Numerical Simulation of Astrophysical Gas Dynamics, and Application to the Gravitational Stability of Protostellar Discs

Christopher Peter Batty

A thesis submitted to
Cardiff University
for the degree of
Doctor of Philosophy

March 2011

Declaration

This work has not previously been accepted in substance for any degree and is not being concurrently submitted in candidature for any degree.

Signed: _____
Candidate

Date: _____

Statement 1

This thesis is the result of my own investigations, except where otherwise stated. Other sources are acknowledged by giving explicit references. A bibliography is appended.

Signed: _____
Candidate

Date: _____

Statement 2

I hereby give consent for my thesis, if accepted, to be available for photocopying and for inter-library loan, and for the title and summary to be available to outside organisations.

Signed: _____
Candidate

Date: _____

Adfyd a ddwg wybodaeth,
a gwybodaeth ddoethineb.

Acknowledgements

I would like to thank:

Anthony Whitworth, who supervised this project and made it all possible;

The Particle Physics and Astronomy Research Council (PPARC), later merging into the Science and Technology Facilities Council (STFC), who funded this project;

David Hubber, with whom I wrote the SEREN SPH code;

Dimitris Stamatellos, who wrote the radiative diffusion approximation subroutines and assisted with my first steps into disc simulations;

Annabel Cartwright, who highlighted several quirks in SPH in general and disc simulations in particular, and assisted with my first steps into writing LaTeX documents;

Daniel Price, who wrote the SPLASH visualisation code, used to make column density plots of the simulations;

Richard Wünsch, who assisted with the development philosophy of SEREN;

Andrew McLeod, who assisted in developing SEREN and wrote its MPI implementation;

Thomas Bisbas, whose HEALPIX ray-casting algorithm made the octal-spatial decomposition tree and the block timestepping scheme look far less complicated by comparison;

Merlin, the Advanced Research Computing at Cardiff (ARCCA) High Performance Computing (HPC) Cluster;

Geoff & Gaby, my parents, for their continued support, along with my family and friends;

Jess Kelly, for being there when it counted most. Go raibh maith agat.

Diolch yn fawr iawn i chi i gyd.

Abstract

In this thesis we investigate the development and use of numerical methods to study astrophysical problems, particularly the formation and evolution of objects via gravitational instability in circumstellar discs.

We begin with a comprehensive overview of the development, validation and optimisation of numerical tools. These formed the basis of SEREN, a Smoothed Particle Hydrodynamics (SPH) code for modelling self-gravitating fluid dynamics. SEREN has been rigorously tested and optimised, and is now being used for high-performance research in various areas of star formation.

We then consider in some depth the problems associated with shocks, instabilities and shear flows in numerical simulations, detailing why such problems arise and what can be done to alleviate them.

Finally we model circumstellar discs, investigating the influence of both physical and computational parameters upon the formation of objects via gravitational instability. We then model the interaction of discs with stars and other disc systems, investigating the influence of the orbital parameters upon the evolution of a marginally stable disc.

Ultimately we find that gravitational instability in massive extended circumstellar discs is a viable mechanism for the formation of brown dwarfs and massive planets, and provides an explanation for the "brown dwarf desert" and free-floating planets.

We also find that while disc-star and disc-disc interactions might produce accretion bursts and exert an influence over the disc evolution, they are not a likely mechanism for triggering fragmentation in marginally stable discs.

Contents

List of Tables

List of Figures

"We are star-stuff."

Carl Sagan

Chapter 1

Introduction

In this chapter we summarise the processes involved in the formation of stars and planets.

Beginning with a historical overview we then examine the collapse of interstellar gas, condensing and heating until it forms a star, and also address the issues of binary systems, brown dwarfs and extra-solar planets.

This allows us to locate our topic of interest (the evolution of circumstellar discs and the formation of objects within them) in the wider astronomical context.

1.1 A Brief History of Star Formation

Stare up into the sky on a clear night and you get the chance to look forwards in space and backwards in time as the light of nearby stars reaches your eyes. Even on an overcast day our world is comfortably illuminated by the nearest star, our Sun. This star lies at the heart of our solar system, providing the energy that fuels our existence, and stars lie at the heart of some of the most fundamental questions in astronomy. Galaxies are composed of hundreds of billions of stars, so to fully understand the stucture and evolution of galaxies we need to understand stars themselves. Planets orbit stars, born from the material that surrounds them, so a full understanding of planets also requires that stars be understood. Even the heavy elements that make up our bodies and the world around us had their origins in the nuclear processes of stars. The fact that stars are important on all scales, from the structure of the universe at large right down to our daily lives in orbit around one, makes them truly profound objects.

Stars have been studied since the dawn of science, from the ancient Babylonians and Egyptians right through to the present day, yet we still do not fully understand them. Their motions and properties have been plotted through the ages, and in recent centuries much has been learned about their evolution along the main sequence and even their often-violent deaths. But their births have long been shrouded in dust and mystery, hidden deep within dense clouds of molecular hydrogen, where our sight cannot penetrate. In recent decades the shroud was pierced when telescopes operating at wavelengths longer than visible light, in the infra-red and sub-millimetre bands, allowed us to peer inside to uncover the secrets within.

When we make astronomical observations we obtain snapshots in time of the

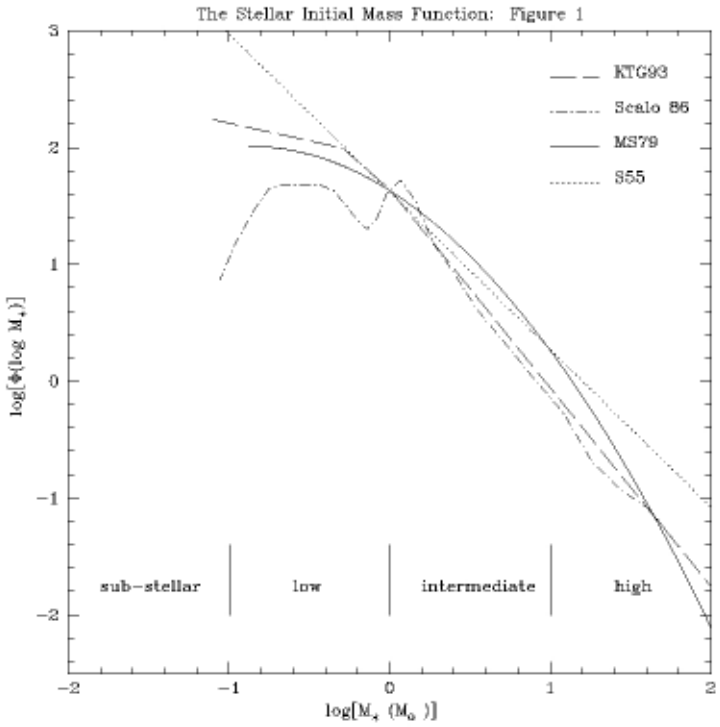


Figure 1.1: The initial mass function (Meyer et al. 2000).

various stages as interstellar gas collapses from low densities of $< 10^{-17}\,\mathrm{kg\,m^{-3}}$ and cold temperatures of $\sim 10\,\mathrm{K}$, through 20 or more orders of magnitude in density to $> 10^{3}\,\mathrm{kg\,m^{-3}}$, and heating to $> 3 \times 10^{6}\,\mathrm{K}$, when hydrogen fusion begins and a star is born.

The conversion process from cold diffuse gas to hot dense stars involves non-linear self-gravitating fluid dynamics. This is the domain of star formation theory, which aims to probe the underlying physical processes involved in forming stars from gas, and to determine the relative importance of factors such as turbulence or magnetic fields, in order to ultimately explain why stars have their observed properties.

Given that the behaviour of a star is predominantly governed by its mass, perhaps the most important observation to explain has been the initial mass function (IMF), the normalised mass spectrum of all stars ever formed, with their mass measured immediately after formation (Salpeter 1955). This appears to follow a power law at higher masses, but the behaviour at lower masses is less clear, since it is more difficult to observe low-mass stars. Figure 1.1 shows several models for the initial mass function.

Recent advances in infra-red and sub-millimetre astronomy have provided observations of prestellar cores and young stellar objects, and these new findings have initiated the refinement of theories regarding star formation.

As with stars, questions have long been asked about how planets are formed, driven by the fact that our solar system contains several planets and that we live upon one of them. This once unique arrangement within the universe has been dramatically

disrupted by the discovery of hundreds of extra-solar planets: planets orbiting stars other than the Sun. Mayor & Queloz (1995) documented the first planet observed around a Sun-like star. New findings once again spurred the refinement of theories regarding planet formation, especially since our own planetary system now seems far from a typical case. More commonly seen, perhaps because they are most easily detected, are "hot Jupiters": massive planets orbiting very close to stars.

In the following sections we will examine the current understanding of the star formation process in some detail, beginning with cold diffuse interstellar gas and following the collapse process until a star is formed. We will also address other issues that arise, such as binary systems, brown dwarfs and the formation of planets.

1.2 The Collapse of a Molecular Cloud

The diffuse interstellar gas within the disc of a galaxy assembles into clouds of molecular hydrogen by unknown processes (Dobbs, Bonnell & Pringle 2006). Concentrating and condensing near the spiral arms, it attains an optical thickness that shields the molecular hydrogen against dissociation from ultraviolet radiation. These molecular clouds range in size from diameters of a fraction of a parsec up to hundreds of parsecs, and contain hundreds or even millions of solar masses of gas at densities of 10^{-19} to $10^{-17}\,\mathrm{kg\,m^{-3}}$ and temperatures of $10-20\,\mathrm{K}$ (Ferrière 2001). Those ~ 50 pc in size, with masses $10^5-10^6\,\mathrm{M_\odot}$, are typically referred to as giant molecular clouds (GMCs); they have a hierarchical structure, consisting of clouds within clouds, and tend towards irregular, filamentary shapes (Williams, Blitz & McKee 2000).

The nearby star-forming regions of Taurus and Orion are well-studied. Taurus, at a distance of ~ 140 pc, with mass $\sim 10^4\,\mathrm{M_\odot}$ and extent ~ 20 pc (Pineda et al. 2010), has low star and gas density, and high binary fraction (McKee & Ostriker 2007). Orion, at a distance of ~ 400 pc, with mass $\sim 10^5\,\mathrm{M_\odot}$ and extent ~ 50 pc, has high star density and several OB associations (Myers 1999). These differing characteristics, with low-mass isolated systems being the norm in Taurus and high-mass clusters in Orion, suggest that star formation depends upon the initial conditions and dominant forces at work.

1.2.1 The Jeans Condition

In order to investigate the collapse of a molecular cloud under its own gravity, we shall consider a spherical cloud for the sake of simplicity. The self-gravitational potential energy of a sphere is

$$-\int_{r=0}^{r=\infty}\frac{GM(r)}{r}dM(r) \tag{1.1}$$

where G is the gravitational constant ($G=6.67\times10^{-11}\,\mathrm{m^3\,kg^{-1}\,s^{-2}}$) and $M(r)$ is the mass contained within radius r. For a sphere of mass M_S and radius R_S, assuming uniform density we can write $M(r)=M_S\left(\frac{r}{R_S}\right)^3$. This can be rearranged as $r(M)=R_S\left(\frac{M}{M_S}\right)^{1/3}$, allowing the integral to be evaluated.

$$-\int_{M=0}^{M=M_S} \frac{GM}{r(M)} dM = -\frac{GM_S^{1/3}}{R_S} \int_0^{M_S} M^{2/3} dM = -\frac{3GM_S^2}{5R_S} \tag{1.2}$$

The internal radial kinetic energy of the sphere is

$$\int_{r=0}^{r=R_S} \frac{1}{2} \frac{dM(r)}{dr} v^2(r) dr \tag{1.3}$$

where $v(r)$ is the radial velocity at radius r. If we assume that radial excursions are homologous, we can write $v(r) = \left(\frac{r}{R_S}\right) \frac{dR_S}{dt}$. Substituting $\frac{dM}{dr} = 4\pi r^2 \rho(r)$, where ρ is the uniform density, the integral can now be evaluated with respect to r.

$$\begin{aligned} \int_{r=0}^{r=R_S} \tfrac{1}{2}\rho 4\pi r^2 \left(\tfrac{r}{R_S}\tfrac{dR_S}{dt}\right)^2 dr &= 2\pi\rho \left(\tfrac{1}{R_S}\tfrac{dR_S}{dt}\right)^2 \int_0^{R_S} r^4 dr \\ &= \tfrac{2}{5}\pi\rho R_S^3 \left(\tfrac{dR_S}{dt}\right)^2 \end{aligned} \tag{1.4}$$

Using $M_S = \rho\frac{4}{3}\pi R_S^3$ allows this kinetic energy to be expressed as $\frac{3}{10} M_S \left(\frac{dR_S}{dt}\right)^2$.

Using the first law of thermodynamics, and assuming isothermal collapse of the sphere, the increase in its internal energy must be accounted for by heating due to the work done by compression. If, for the sake of simplicity, we neglect any external pressure acting upon the system then

$$\frac{d}{dt}\left(\frac{3}{10} M_S \left(\frac{dR_S}{dt}\right)^2 - \frac{3GM_S^2}{5R_S}\right) = P\frac{dV_S}{dt} \tag{1.5}$$

where P is the pressure and V_S is the volume of the sphere. Under isothermal collapse, $P = c_s^2\rho$, where c_s is the isothermal sound speed. The compression rate may be written as $\frac{dV_S}{dt} = \frac{d}{dt}\left(\frac{4}{3}\pi R_S^3\right) = 4\pi R_S^2 \frac{dR_S}{dt}$.

Making these substitutions on the right hand side, and differentiating the left hand side,

$$\frac{3}{5} M_S \frac{dR_S}{dt}\frac{d^2R_S}{dt^2} + \frac{3GM_S^2}{5R_S^2}\frac{dR_S}{dt} = c_s^2 \rho 4\pi R_S^2 \frac{dR_S}{dt} \tag{1.6}$$

The common $\frac{dR_S}{dt}$ term now cancels throughout, and using $M_S = \rho\frac{4}{3}\pi R_S^3$ allows one factor of M_S to be cancelled throughout, resulting in

$$\frac{3}{5}\frac{d^2R_S}{dt^2} + \frac{3GM_S}{5R_S^2} = c_s^2 \frac{3}{R_S} \tag{1.7}$$

Rearranging gives

$$\frac{d^2R_S}{dt^2} = -\frac{GM_S}{R_S^2} + 5\frac{c_s^2}{R_S} \tag{1.8}$$

which is the equation of motion for the radius R_S of the sphere. For the sphere to undergo collapse, the acceleration $\frac{d^2R_S}{dt^2}$ must be negative, and therefore the self-gravity must overcome the internal pressure support.

$$\frac{GM_S}{R_S^2} > 5\frac{c_s^2}{R_S} \tag{1.9}$$

Using $M_S = \rho\frac{4}{3}\pi R_S^3$, and rearranging,

$$R_S^2 > \frac{15c_s^2}{4\pi G\rho} \tag{1.10}$$

This defines a length scale known as the Jeans length: $R_{Jeans} = \left(\frac{15}{4\pi G\rho}\right)^{1/2} c_s$. The Jeans length is essentially the radius at which a cloud is first gravitationally bound, so for a spherical cloud to collapse its radius must be greater than the Jeans length.

Instead of a minimum length, the condition can alternatively be described in terms of a minimum mass, the Jeans mass.

$$M_{Jeans} = \left(\frac{375}{4\pi}\right)^{1/2} \frac{c_s^3}{G^{3/2}\rho^{1/2}} \tag{1.11}$$

Consideration of other forces, such as rotation, turbulent motions and magnetic fields, only increases the support of the cloud, so the Jeans conditions really represent the lowest possible values at which a cloud can undergo collapse. The only plausible way in which a smaller or less massive cloud could undergo collapse would be through the action of external pressure confinement.

For a molecular cloud of temperature $T = 10 - 20\,\mathrm{K}$, the isothermal sound speed is $c_s = \left(\frac{kT}{\overline{m}}\right)^{1/2} \simeq 200 - 300\,\mathrm{m\,s^{-1}}$, where k is the Boltzmann constant ($k = 1.38 \times 10^{-23}\,\mathrm{JK^{-1}}$) and $\overline{m}$ is the mean molecular mass ($3.35 \times 10^{-27}\,\mathrm{kg}$ for molecular hydrogen). Typical densities of $\rho = 10^{-19} - 10^{-17}\,\mathrm{kg\,m^{-3}}$ therefore result in Jeans lengths of $R_{Jeans} \simeq 0.03 - 0.4\,\mathrm{pc}$, or Jeans masses of $M_{Jeans} \simeq 13 - 378\,\mathrm{M_\odot}$. Since even the smallest molecular clouds are of this size and mass range, and giant molecular clouds are many times greater in size ($\sim 100\,\mathrm{pc}$) and mass ($\sim 10^6\,\mathrm{M_\odot}$), most molecular clouds would be expected to undergo collapse.

As the density increases during collapse, the Jeans mass decreases, so separate parts of the cloud might fragment, collapsing independently in a hierarchical fashion. The minimum mass of an object formed by fragmentation might be as low as 3 Jupiter masses (Boyd & Whitworth 2005).

The observed rate of star formation is lower than the Jeans conditions predict. Since thermal pressure alone cannot support such massive clouds, this is an indication that other supporting forces are at work. Observed supersonic velocity dispersions suggest that turbulence plays a role. Clumps within clouds are often observed at scales of ~ 1 parsec radius and mass $\sim 10^2\,\mathrm{M_\odot}$ (Williams, Blitz & McKee 2000), which might be suggestive of the effective Jeans scales.

The gas might also be supported by magnetic fields, regulating its collapse (Shu, Lizano & Adams 1987; Hosking & Whitworth 2004). If the conductivity of the gas is sufficiently high then it is coupled to the magnetic field, and constrained to move along field lines by the Lorentz force. Neutral particles then experience this force indirectly via collisions with ions, described as "ion-neutral friction". Magnetic support might be lost through ambipolar diffusion (Mestel & Spitzer 1956), as neutral particles diffuse through the ions over time. In molecular clouds, the ionisation fraction is low (10^{-8} to 10^{-6}) so this could happen quite rapidly (Caselli et al. 1998).

While magnetic support can delay collapse, it cannot completely prevent it due to ambipolar diffusion, so eventual collapse is inevitable (Mouschovias 1976).

1.2.2 The Freefall Time

In the densest coldest regions, this additional support may fail to overcome the self-gravity, allowing the gas to condense and form a star. If we consider the case where self-gravity completely dominates over thermal pressure support then we can write the equation of motion for the radius R_S of a collapsing sphere of gas as

$$\frac{d^2R_S}{dt^2} = -\frac{GM_S}{R_S^2} \tag{1.12}$$

Using an integrating factor of $2\frac{dR_S}{dt}$, multiplied on each side of the equation, allows us to integrate it.

$$\left(\frac{dR_S}{dt}\right)^2 = \frac{2GM_S}{R_S} + C \tag{1.13}$$

The integration constant C can be determined by considering the sphere to have an initial radius of R_0 (at time $t = 0$), and to collapse from an initial state of rest ($\frac{dR_S}{dt} = 0$), giving $C = -\frac{2GM_S}{R_0}$. Substituting this, and taking the negative square root (which describes the collapse solution),

$$\frac{dR_S}{dt} = -(2GM_S)^{1/2}\left(\frac{1}{R_S} - \frac{1}{R_0}\right)^{1/2} \tag{1.14}$$

The substitution $R_S = R_0\cos^2(\theta)$ can be used to solve this equation, where $\theta = \theta(t)$ is essentially a normalised parameterisation of the radius as a function of time.

$$-2R_0\cos(\theta)\sin(\theta)\frac{d\theta}{dt} = -\left(\frac{2GM_S}{R_0}\right)^{1/2}\left(\sec^2(\theta) - 1\right)^{1/2} \tag{1.15}$$

Using the trigonometric identity $\tan^2(\theta) = \sec^2(\theta) - 1$ and then dividing through by $\tan(\theta)$ and R_0,

$$2\cos^2(\theta)\frac{d\theta}{dt} = \left(\frac{2GM_S}{R_0^3}\right)^{1/2} \tag{1.16}$$

Using the identity $2\cos^2(\theta) = 1 + \cos(2\theta)$, and integrating with respect to t,

$$\theta + \frac{1}{2}\sin(2\theta) = \left(\frac{2GM_S}{R_0^3}\right)^{1/2} t \tag{1.17}$$

Using the identities $\sin(2\theta) = 2\cos(\theta)\sin(\theta)$ and $\sin^2(\theta) = 1 - \cos^2(\theta)$,

$$\theta + \cos(\theta)(1 - \cos^2(\theta))^{1/2} = \left(\frac{2GM_S}{R_0^3}\right)^{1/2} t \tag{1.18}$$

Initially at $t = 0$ (when $R_S = R_0$), $\cos^2(\theta) = 1$, so $\theta = 0$. When the sphere has collapsed to a singularity ($R_S = 0$), $\cos^2(\theta) = 0$, so $\theta = \frac{\pi}{2}$. The time taken to collapse to a singularity through the action of self-gravity is the freefall time t_{ff}.

$$t_{ff} = \frac{\pi}{2}\left(\frac{2GM_S}{R_0^3}\right)^{-1/2} \tag{1.19}$$

Since the mass M_S of the sphere remains constant throughout collapse, the freefall time can be written in terms of the initial density $\rho_0 = \frac{3M_S}{4\pi R_0^3}$, giving

$$t_{ff} = \frac{\pi}{2}\left(\frac{8\pi G\rho_0}{3}\right)^{-1/2} = \left(\frac{3\pi}{32G\rho_0}\right)^{1/2} \tag{1.20}$$

For a density of $\rho = 10^{-17}\,\mathrm{kg\,m^{-3}}$, the freefall time is $\sim 666,000$ years. The collapse of dense clumps within a cloud might therefore occur on timescales of less than a megayear.

Molecular clouds as a whole used to be thought to have long lifetimes ~ 1 gigayear, but now appear to be regarded as shorter-lived transient rather than equilibrium structures (Ballesteros-Paredes, Klessen & Vásquez-Semadeni 2003): assembling, forming stars, and then dispersing, probably destroyed by the luminosity and mechanical feedback of any high-mass stars they produce.

1.2.3 Prestellar Cores

Highly condensed regions have been observed inside molecular clouds and, since they are believed to be the precursors of stars (Ward-Thompson et al. 1994), are referred to as prestellar cores. They are tens of thousands of astronomical units (AU) in diameter, with masses ranging from a fraction of a solar mass to tens of solar masses (André, Ward-Thompson & Barsony 2000). Their flattened rather than spherical geometry (Kirk 2002) hints at rotation or perhaps the influence of a magnetic field. They may be triaxial ellipsoids rather than prolate or oblate spheroids (Gammie et al. 2003). Protostellar cores, however, are observed to be less elliptical than starless cores (Goodwin, Ward-Thompson & Whitworth 2003).

The mass function for cores is similar in shape to the initial mass function for stars, so the IMF could be determined by fragmentation at the prestellar stage (André, Ward-Thompson & Barsony 2000; Bonnell, Larson & Zinnecker 2006), but whether this can produce the smallest masses in the IMF remains an open question.

Figure 1.2 shows a possible prestellar core inside the Horsehead nebula.

1.3 Theoretical Evolution

The initial collapse phase of a cloud, as described in the previous section, is essentially isothermal (Larson 1969): the gas is optically thin and gravitational potential energy can be freely radiated away. At only $\sim 10\,\mathrm{K}$ this collapse phase approaches the freefall timescale, with thermal pressure support overwhelmed by self-gravity.

Figure 1.2: A possible prestellar core inside the horsehead nebula (Ward-Thompson et al. 2006). On the left is an optical image and on the right is a sub-millimetre image (450μm) of the same region, penetrating the dust to reveal a condensation in the "throat" of the horse.

Once the core reaches a density of $10^{-10}\,\mathrm{kg\,m^{-3}}$, however, it becomes opaque to the radiation it emits, and enters a short-lived adiabatic phase (Larson 1969), heating up until the resulting thermal pressure halts collapse at $\sim 200\,\mathrm{K}$.

It then enters a quasi-static phase of Kelvin-Helmholtz contraction, remaining close to hydrostatic equilibrium (Larson 1969), and is considered a pre-main sequence (PMS) star. The timescale for contraction is governed by the luminosity, since that is the rate at which the released gravitational potential energy is able to escape from the protostellar surface; this is much longer than a freefall time. While it contracts, the rest of the cloud continues to collapse around it, causing it to heat up (Masunaga et al. 1998).

At $\sim 2000\,\mathrm{K}$, molecular hydrogen dissociates and the collapse becomes almost isothermal again (Masunaga & Inutsuka 2000) because energy goes into hydrogen dissociation rather than heating. The PMS star then evolves along a Hayashi track (Hayashi 1966), in convective equilibrium, getting denser and so reducing its surface area, causing its luminosity to decrease.

Once all the hydrogen is dissociated the star heats up again, and at $\sim 3000\,\mathrm{K}$ it enters a radiative equilibrium and evolves along a Henyey track (Henyey, Lelevier & Levée 1955), getting denser, hotter and gradually more luminous, and the accretion of any infalling material may increase its mass.

If it reaches $3 \times 10^6\,\mathrm{K}$ then hydrogen burning can occur and the star joins the main sequence. It will spend most of its life here since the energy from nuclear reactions, fusing hydrogen to form helium, provides pressure support against further collapse. This requires it to exceed the Kumar mass (Kumar 1963), which is $\sim 0.08\,\mathrm{M_\odot}$ (assuming protosolar elemental abundances of 70% hydrogen, 28% helium and 2% heavy elements).

A less massive star cannot ignite hydrogen, so it does not join the main sequence. Instead it may burn deuterium and become a brown dwarf, supported against collapse by electron degeneracy pressure. Such objects are faint ($< 10^{-4}\,\mathrm{L_\odot}$) making them hard to observe.

Very massive stars also pose some problems, since they are observed to exist beyond the Jeans mass (when they should have fragmented) or beyond the mass limit where their own radiation pressure should have prevented accretion. Such cases have prompted explanations such as mergers of lower-mass stars, accretion via a circumstellar disc, or runaway accretion processes in the denser regions of a protocluster (Zinnecker & Yorke 2007).

Once the fuel for nuclear fusion has been exhausted, a star will die. Lower-mass stars cool and fade to white dwarfs or neutron stars. Higher-mass stars may explode in a supernova, enriching the galaxy with heavy elements and perhaps triggering a new phase of star formation.

1.4 Observational Evolution

Young stellar objects (YSOs) with different observed characteristics have been interpreted as an evolutionary sequence (André, Ward-Thompson & Barsony 1993). This has only been well-established for low to intermediate mass stars, and is shown in

Figure 1.3.

Ages or timescales for each class have been inferred from statistical considerations and evolutionary models (Lada 1999) and are $10^4, 10^5, 10^6$ and 10^7 years for each class from O to III respectively.

1.4.1 Class 0 Objects

A Class 0 object is the earliest distinguishable stage of a protostar, with a central luminosity source at its core (André, Ward-Thompson & Barsony 2000). Such objects cannot be observed directly because they are highly embedded, but their presence is inferred by compact radio continuum emission, highly energetic bipolar molecular outflows of collimated carbon monoxide, or an internal heating source.

With an emission peak in the sub-millimetre/far-infrared (SMM/FIR) range $150 - 250\,\mu$m, their spectral energy distributions (SED) can be fitted by a relatively cold modified blackbody $\sim 15 - 30\,$K (André, Ward-Thompson & Barsony 2000).

They are rapidly accreting material from their envelope at a rate $> 10^{-5}\,\mathrm{M_\odot yr^{-1}}$ in what may be the main accretion phase (Ward-Thompson et al. 2006), but have not yet accreted half their final mass, and so these early-type protostars are less massive than their envelopes.

1.4.2 Class I Objects

These consist of a dense central object surrounded by a residual envelope of gas and dust (Ward-Thompson & Whitworth 2011), with bipolar outflows.

With an emission peak in the FIR range, their SEDs have an infrared (IR) excess, and so cannot be modelled as a single blackbody or modified blackbody (greybody), only as a composite of them at different temperatures. The protostar emits in the near-infrared (NIR), the envelope in the FIR, and the IR excess is emission from material in a disc that has started to form.

With a lower accretion rate $< 10^{-6}\,\mathrm{M_\odot yr^{-1}}$ taking the protostar almost to its final mass, these late-type protostars are at a more evolved stage (Whitworth & Ward-Thompson 2001).

1.4.3 Class II Objects

These are classical T-Tauri stars in the pre-main sequence phase (Ward-Thompson & Whitworth 2011), so their luminosity is mainly from Kelvin-Helmholtz contraction, with their envelopes dissipated, and surrounded by geometrically thin but optically thick discs with masses from $\sim 0.001\,\mathrm{M_\odot}$ to $\sim 0.5\,\mathrm{M_\odot}$ (Eisner et al. 2008). The high-mass equivalents of T-Tauri stars at $> 2\,\mathrm{M_\odot}$ are Herbig Ae-Be stars (Waters & Waelkens 1998).

The star is revealed rather than embedded, and therefore visible in the optical range, with the disc indicated by IR excess.

These stars are probably a few megayears old, and distinguished by strong H-alpha emission and the veiling of photospheric absorption lines in the UV and

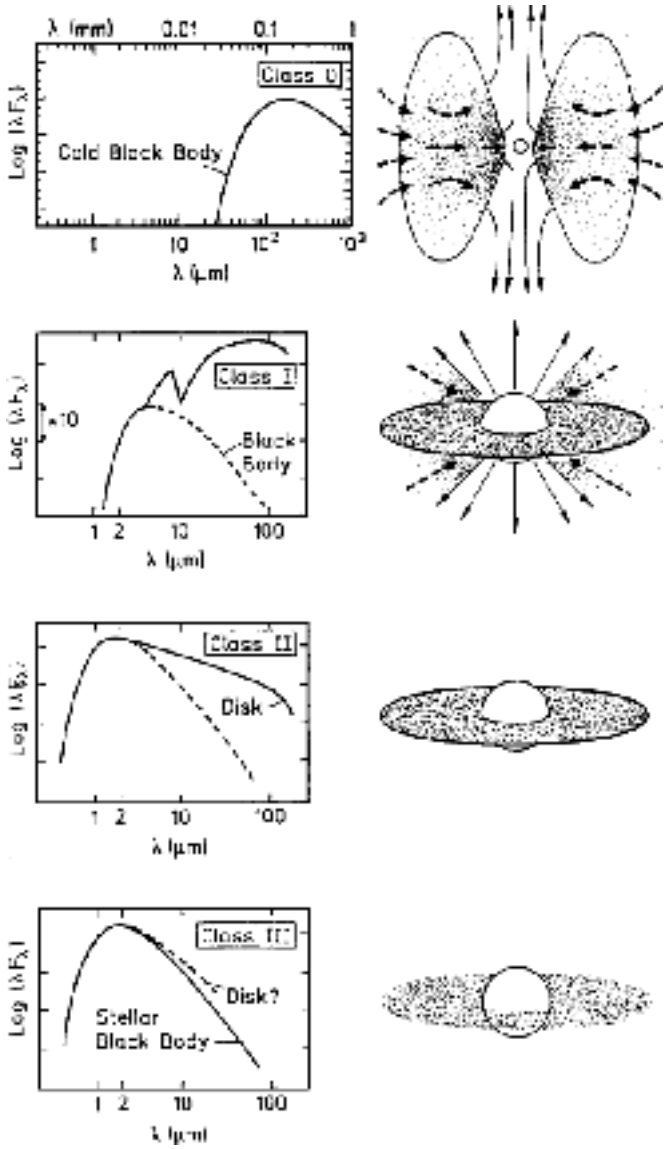


Figure 1.3: Evolutionary sequence of young stellar objects (André 1994) from class O (top) to III (bottom), shown with typical spectral energy distributions (SED).

optical bands due to hot continuum emission. This is indicative of disc material accreting onto the stellar surface, since H-alpha emission occurs when mass flows through magnetospheric accretion columns between the disc and the star (Calvet et al. 2000).

1.4.4 Class III Objects

These are weak-line T-Tauri Stars (Ward-Thompson & Whitworth 2011), distinguished by weak H-alpha emission and little IR or UV excess, indicating negligible accretion. The inner disc region has dissipated, or has been emptied of dust by coagulation into larger rocks or protoplanets.

Blackbody emission peaks in the NIR or optical range. The disc is optically thin and disappears completely after ~ 10 megayears (Hollenbach, Yorke & Johnstone 2000), by which time the star evolves onto the main sequence.

1.4.5 Clusters and Associations

Embedded Clusters

The majority of stars form in embedded clusters, optically obscured by gas and dust but observable in the infrared. These clusters are considered the dominant mode of star formation (Lada & Lada 2003), as opposed to isolated star formation. Most clusters are destroyed when the parent molecular cloud is destroyed, since the loss of gas leaves the cluster gravitationally unbound, and the constituent stars disperse into the field; many young clusters are therefore undergoing violent relaxation (Goodwin & Bastian 2006). In the rare cases where they remain bound after this residual gas loss they become open clusters, and may contain anything from a few dozen stars (such as Ursa Major) to thousands (in the case of the Pleiades).

OB associations

Groups of O stars (with mass $> 20\,\mathrm{M}_\odot$) and B stars (with mass $> 4\,\mathrm{M}_\odot$) are typically found close to molecular clouds. Orion contains 9 O stars and over 300 B stars. With high luminosities and short lifetimes, these stars have little time to move far from their birth site, so they provide good evidence that molecular clouds truly are the sites of star formation.

1.5 Complications

1.5.1 Accretion Discs

Molecular clouds possess angular momentum, either as ordered rotation or within turbulent motion (Burkert & Bodenheimer 2000; Ballesteros-Paredes et al. 2006).

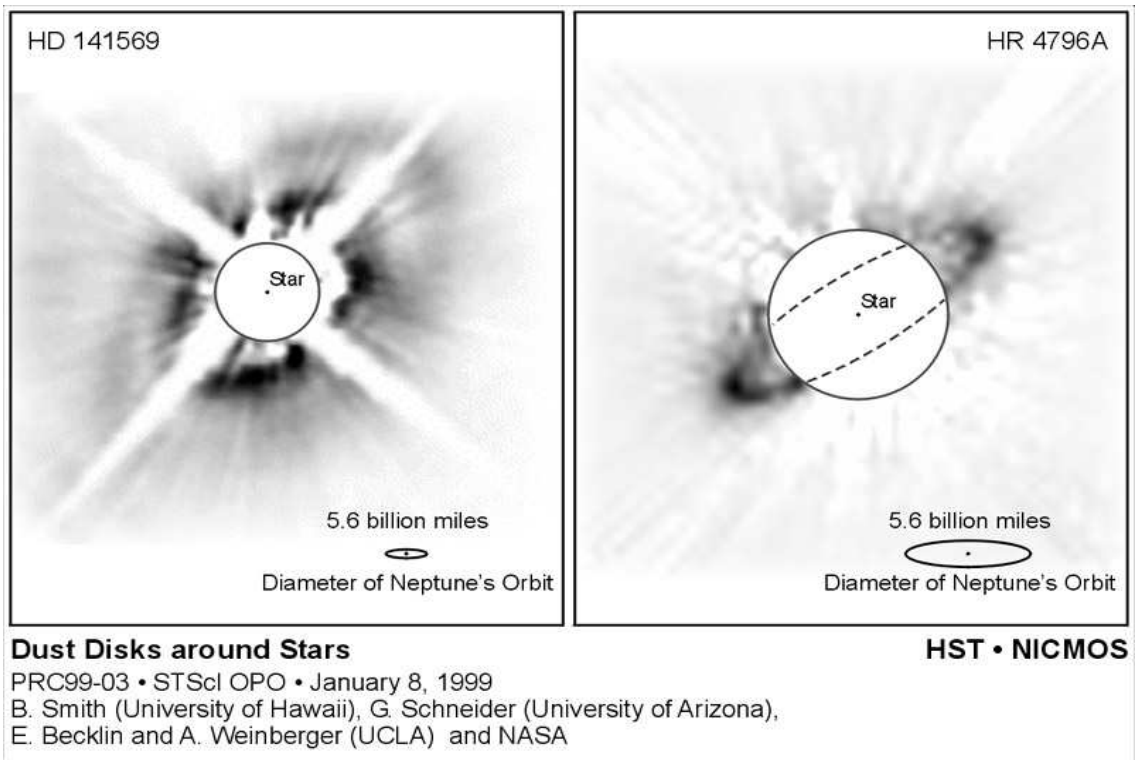


Figure 1.4: Circumstellar discs observed by the Hubble Space Telescope using coronagraphs.

Gas and dust may therefore be prevented from falling onto a protostar by centrifugal forces, and instead collapse down to form a disc around it. In the protostellar stages it is likely that most of the envelope material will fall onto the disc rather than the star.

Protostellar discs are highly embedded, with radiation reprocessed by the surrounding envelope, but have been inferred from IR excesses in SEDs. Some discs have also been observed directly, by the HST in the NIR range and by ground-based adaptive optics (McCaughrean, Stapelfeldt & Close 2000). Figure 1.4 shows a couple of examples. They typically have sizes in the range $10 - 1000$ AU (Watson et al. 2006; McCabe, Stapelfeldt & Pham 2011) and masses $0.001 - 0.5\,\mathrm{M}_\odot$ (Eisner et al. 2008; Jørgensen et al. 2009), and young discs around Class 0 protostars have been observed with masses $0.04 - 1.7\,\mathrm{M}_\odot$ (Enoch et al. 2011). Formed in the early stages of evolution as protostellar discs, they last right through to the final stages, as T-Tauri discs around Class II objects and then protoplanetary discs around Class III objects, having an estimated lifetime of ~ 10 megayears (Hollenbach, Yorke & Johnstone 2000).

T-Tauri and protoplanetary discs have masses 10^{-5} to a few $10^{-1}\,\mathrm{M}_\odot$, according to estimates based on assumed dust opacities, gas-dust ratios and disc density profiles, although these are all poorly determined (Zuckerman 2001). T-Tauri discs could be active, radiating energy produced by viscous dissipation, or passive, simply reprocessing stellar radiation. They are believed to be flared in profile (Dutrey

& Guilloteau 2004), and truncated by the stellar magnetic field in the inner parts, within a few stellar radii of the star (Stone et al. 2000).

Density perturbations in discs can make them gravitationally unstable (Toomre 1964), allowing for the possibility of fragmentation into multiple systems. This would occur at the protostellar stage, when the disc has mass comparable to the protostar. Low-mass binary companions, triples and quadruples could be formed in discs (Stamatellos & Whitworth 2009b).

Discs rotate in almost Keplerian orbit (Jørgensen et al. 2009; Dutrey & Guilloteau 2004), and angular momentum must be transported outwards in order for the disc material to drift inwards and accrete onto the protostar. This redistribution of angular momentum might be achieved via viscosity (Lynden-Bell & Pringle 1974) or gravitational torques. From modelling, the initially formed disc surface density profile scales with radius as r^{-1}, but viscosity and self-gravity steepen this toward $r^{-3/2}$ (Lin & Pringle 1990). The origin of viscosity is uncertain, but might be supplied by turbulence generated by convection (Lin & Papaloizou 1980) or by magnetic fields, through the magneto-rotational instability (Balbus & Hawley 1991) or by magnetic braking (Bodenheimer 1995), where angular momentum is transported by the Alfvén waves generated by the field lines being twisted into helical patterns. Due to these uncertainties, a heuristic treatment has tended to be used, such as the alpha-prescription (Shakura & Sunyaev 1973).

1.5.2 Outflows

A small fraction of accreted material is ejected in bipolar outflows or jets, typically at $\sim 10\%$ of the accretion rate (Richer et al. 2000; Hartigan et al. 1995). These are oriented perpendicular to the disc, are highly collimated, and extend to distances of $0.01 - 1$ pc with velocities of hundreds of $\mathrm{km\,s^{-1}}$ (Bally, Reipurth & Davis 2006).

Outflows only occur during accretion phases, so they are an indicator of stellar youth, and an accretion stage is almost always accompanied by outflows. This suggests that outflows are powered by accretion, and might originate from discs as disc winds (Königl & Pudritz 2000), where material is driven out centrifugally from a magnetised disc, or from the star-disc interface as X-winds (Shu et al. 2000), where infalling material at the inner truncation radius is partially redirected into a helical orbit by magnetic fields.

The T-Tauri stage possesses strong magnetic fields from dynamo action in the central star. This would truncate the disc and might produce jets (Donati et al. 2011). It could also provide an explanation for the disc viscosity.

Accretion discs and jets seem to be ubiquitous in astrophysics. They appear on galactic scales in the form of active galactic nuclei and quasars. At the lower end of the scale, many young brown dwarfs appear to have discs, since they tend to exhibit an IR excess in their SEDs, and outflows have also been observed (Whitworth et al. 2006).

1.5.3 Binary and Multiple Systems

Over 50% of main sequence systems are binaries or multiples (Larson 2001), including perhaps all pre-main sequence solar-mass stars (Padgett, Strom & Ghez 1997). Many stars are found in multiple systems, and Taurus contains a particularly high proportion of young low-mass binary systems (Mathieu et al. 2000).

Separations can range from a few solar radii up to a parsec, but averaging $20 - 50$ AU, giving a typical period of 200 years. The binary frequency varies from region to region, suggesting either an evolutionary effect (Goodwin et al. 2006; Parker et al. 2009) or a variable formation efficiency (Bouvier et al. 2001; Kroupa, Petr & McCaughrean 1999).

Capture between two initially unbound stars requires significant kinetic energy loss, via a three-body interaction or the dissipation of energy through a disc (Watkins et al. 1998). While such capture might be plausible in small, dense clusters of young stars with discs, it cannot realistically explain the high observed binary frequency in the galactic field. Most of these must have formed as binaries, probably during the collapse and fragmentation of prestellar cores, or possibly from gravitational instabilities in discs (Stamatellos & Whitworth 2009a).

Binary or multiple formation appears to be the dominant mode over single star formation (Goodwin et al. 2006). Since single isolated stars are the exception, they might occur when fragmentation is suppressed by feedback processes (Boyd 2003) or magnetic fields (Hosking 2002).

About 15% of binaries are very close, with separation < 1 AU (Mathieu 2007). This requires a large amount of angular momentum to have been extracted from the system, unless it was formed from the accretion of particularly low angular momentum gas. These tight binaries might be explained by dynamic interactions which ejected a third body carrying away the angular momentum, or by dissipation through tidal interactions between the binary system and a circumbinary disc, or by formation in situ through disc fragmentation (Bonnell & Bate 1994).

There is also a wider angular momentum problem (Mestel 1965). Molecular clouds possess specific angular momentum of $\sim 10^{17}\,\mathrm{m^2 s^{-1}}$, so angular momentum conservation on their collapse would result in rapidly rotating stars. These are not observed, however, with T-Tauri stars possessing specific angular momentum of $\sim 5 \times 10^{13}\,\mathrm{m^2 s^{-1}}$ (Bodenheimer 1995).

The missing angular momentum might be extracted via magnetic fields during the intial stages of collapse, when the gas is still coupled to it, or by dissipative processes in the disc, whatever the source of the disc viscosity may be. It might also be explained by the formation of binary or multiple systems, might be carried away with the material ejected in outflows or jets, or might be incorporated into the formation of planets.

1.5.4 Brown Dwarfs

The observed lack of brown dwarf companions around main sequence stars (Marcy & Butler 1994), known as the "brown dwarf desert", suggests that brown dwarfs might have a different formation mechanism than other stars. Some possibilities include

the ejection of low-mass embryos from their pre-stellar cores (Reipurth & Clarke 2001; Goodwin, Whitworth & Ward-Thompson 2003), gravitational instabilities in discs (Stamatellos et al. 2007a), opacity-limited fragmentation in turbulent molecular clouds (Boyd & Whitworth 2005) or the photo-evaporation of massive cores overrun by a HII region (Whitworth & Zinnecker 2004).

1.5.5 Extra-Solar Planets

Our solar system appears to be atypical: a single isolated star with planets on relatively distant orbits. Most stars are in binary systems, and in clusters, and most planets are found on very tight orbits, although admittedly these are the easiest to detect.

Planets have been observed with masses ranging from a few Earth masses to many Jupiter masses (Schneider 2011). This makes the distinction between giant planets and brown dwarfs somewhat blurred. The uncertainty regarding brown dwarf formation mechanisms extends to giant planets. One possibility is that a large gas giant might form via gravitational instability in a disc (Kuiper 1949), in a top-down fashion. The conventional model has long been that planets are formed via core accretion (Pollack et al. 1996), in a bottom-up fashion, through the gradual agglomeration of dust grains (Lissauer 1993). While this must certainly be the case for rocky terrestrial planets like our Earth, such a process might struggle to form a gas giant within the lifetime of a disc. The two mechanisms might overlap, or could even combine into a hybrid mechanism, to fully explain the formation of giant planets (Rice et al. 2006; Boley & Durisen 2010).

Free-floating planets, typically < 13 Jupiter masses, have also been observed within young stellar clusters (Lucas & Roche 2000; Zapatero Osirio et al. 2000), suggesting that planet formation could be a chaotic and violent process. Indeed free-floating Jupiter-mass objects might be twice as common as main sequence stars (Sumi et al. 2011).

The high metallicity of the Sun suggests an origin closer to the galactic centre or to an OB association, while the tilt of the ecliptic plane with respect to the solar equator suggests the possibility of an encounter with another star. Our solar system clearly has an interesting history, and leads one to wonder how much any chance encounter with another star might have influenced its unique pattern of planet formation. In fact many extra-solar planets are found to have very high inclinations (Schneider 2011), which lends significant weight to turbulent formation and the possibility of encounters.

1.6 Numerical Star Formation

Astronomical observations only give us snapshots of the whole story, slices in time. While statistical considerations allow us to infer the timescales involved at these various observed stages, the onus is upon theory to attempt to join them up and fully explain them.

The aim of this thesis is to investigate the objects that might be formed via gravitational instability in discs around young stars. As we have discussed in this

chapter, such objects could be low-mass companion stars, brown dwarfs or giant gas planets. Some principal questions to address are:

- Under what circumstances can such objects form?
- When and where do they form?
- What factors determine their initial properties?
- What factors influence their subsequent orbital evolution?

Since an isolated disc might have the capacity to stabilise itself, the influence of interactions with other stars and star-disc systems will also be considered.

Due to the non-linear and chaotic processes involved, analytic methods are not capable of providing the answers. It therefore becomes necessary to resort to numerical methods, so the investigation is performed through the aid of computer simulations.

Indeed most of star formation theory has been ushered into an era of computer simulation, since the isolated and quiescent modes of star formation studied in the past using analytic methods have been replaced by the modern picture of star formation as a turbulent, dynamic and sometimes violent process.

The nature of our problem requires formidable computational resources, in terms of both hardware and the development of effective and efficient algorithms. While computers have become a vital element in the investigation of theoretical models, their use involves many difficulties that are sometimes not immediately apparent to the casual observer. Due to its critical importance in the investigation, a substantial portion of the thesis is necessarily devoted to this computational aspect.

1.7 Overview of the Thesis

The main focus of the thesis is to investigate object formation via gravitational instability in circumstellar discs. Since the avenue of investigation is computational, much of the early material is concerned with establishing the method used and its fidelity. These foundations form the basis of the astrophysical results, so they must be seen to be safely and solidly constructed.

In the first few chapters, we chart our path through the minefield of numerical physics, developing the tools to deal with our problem in a trustworthy manner. We develop, validate and optimise our tools in a comprehensive fashion, with our method of choice ultimately being Smoothed Particle Hydrodynamics (SPH). Its effectiveness and efficiency will be demonstrated by submitting it to rigorous testing across a range of physical problems, to ensure its fidelity. Only a detailed knowledge of the construction of a numerical tool, together with the often tacit knowledge required to develop it, can truly provide the level of trust required to believe any results it produces.

Chapter 2 (Numerical Methods) explores the powers and pitfalls of computation and examines several numerical integration schemes. Chapter 3 (Gravity) then uses numerical integration to develop an N-body computer program which is used to solve gravitational problems. Chapter 4 (Hydrodynamics) then incorporates computational fluid dynamics, developing the N-body code into a Smoothed Particle Hydrodynamics (SPH) code which is used to solve self-gravitating hydrodynamic problems.

Chapter 5 (Optimisation) demonstrates several key refinements (a tree algorithm, block timestepping, sink particles and parallelisation) that are used to improve the efficiency of the code, allowing higher resolution simulations to be performed within a reasonable timeframe.

Having developed an efficient numerical code for self-gravitating hydrodynamic simulations, in the next couple of chapters we look in depth at one of the trickiest physical problems of numerical hydrodynamics, or indeed hydrodynamics in general, that of viscosity. We expose some of the difficulties that can crop up during the use of a numerical code, and some ways to alleviate them.

Chapter 6 (Shocks and Instabilities) demonstrates how the need to resolve shocks can be addressed using the concept of artificial viscosity, and discusses the issues associated with its application; how to resolve hydrodynamic instabilities using artificial thermal conductivity is also briefly examined. Chapter 7 (Shear Flows) focuses upon the modelling of shear viscosity since this is of crucial importance to the fidelity of disc simulations, where viscous and pseudo-viscous processes must be treated correctly to produce valid results.

Having adapted the numerical code to handle viscous processes, in the final chapters we apply it to the astrophysical problem of gravitational instability in circumstellar discs.

Chapter 8 (Circumstellar Discs) details how to model gravitational instability in a massive extended disc, exploring the influence of the various physical and computational parameters involved, and demonstrates that it is a viable mechanism for the formation of free-floating planets and brown dwarfs. Chapter 9 (Disc Interactions) investigates the effect of disc-star and disc-disc interactions on the evolution of a marginally stable disc using a variety of orbital modes. Chapter 10 (Summary) summarises our conclusions.

"There are more ways to
the wood than one."

English proverb

Chapter 2

Numerical Methods

In this chapter we describe why numerical methods are necessary and how they can be used to solve difficult problems. This provides a foundation for our aim to model the behaviour of an astrophysical fluid.

Particular attention is paid to the difficulties and dangers introduced by the use of computers.

The requirements for a numerical integration scheme are first considered. Then a variety of basic schemes are studied in detail, revealing some of their merits and flaws through their application to the simplest of problems. Finally, some of the effects introduced by more complicated problems are considered.

2.1 The Need for Numerical Methods

Our ultimate aim is to investigate the behaviour of an astrophysical fluid: the compressible, self-gravitating gas found in molecular clouds. In general, the behaviour of this gas is described by a system of partial differential equations that are non-linear, with the behaviour of any fluid element coupled to its neighbouring elements hydrodynamically, and to all other elements gravitationally.

Analytic techniques can only be used to obtain solutions in the simplest cases, typically those involving symmetry (such as spherically symmetric one-dimensional problems), linear equations and few variables. Numerical methods have great power, since they allow us to move beyond these limiting cases and deal with asymmetry, non-linearity and many variables. These more difficult problems are modelled by using a computer program to solve the equations numerically.

Numerical methods have their own fundamental limitation, however: they can only model systems that are finite and discrete.

Computers represent numbers using fixed amounts of binary digits (bits), so concepts like infinity or a continuum simply do not compute. The number of decimal places that can be represented is fundamentally limited, making exact solutions technically impossible to achieve.

In order to model a continuous fluid, it must be approximated as a system of discrete elements. The time dimension must also be divided into discrete elements for numerical methods to be able to operate. Computers cannot differentiate or integrate analytically, because they cannot represent the infinitesimal limit. They must operate

using finite limits, which are determined by the size of the discrete elements into which the system being modelled is divided (discretised) in space and time.

Given the approximate nature of a computer model (simulation), it is vitally important to ensure that physical results are produced (corresponding to the exact solutions of the system equations) which are not corrupted by numerical effects (errors introduced by the method itself).

2.1.1 Discretisation in Space

Our system of interest is an astrophysical fluid: a continuous medium that must be discretised before computation is possible. This is done by constructing a spatial mesh, which may consist of either stationary cells or mobile sampling points.

Stationary cells (grid cells) are basically volume elements that are fixed in space, and the fluid flows through them; thus the entire volume occupied by the continuous fluid is chopped into a finite number of cell-sized pieces.

Mobile sampling points (particles) are basically fluid elements containing a fixed mass of fluid, which move with the fluid flow; thus the entire mass of continuous fluid is chopped into a finite number of particle-sized pieces.

These discrete, finite elements, either grid cells or particles, can be handled numerically. Ultimately we will be using the particle-based approach, modelling the fluid using a finite number of particles; later in this chapter, however, we will adopt the grid-based approach, using a uniform grid with spacing h, since this is arguably the simplest conception of a spatial mesh and therefore makes the stability analysis more accessible.

2.2 Numerically Solving the Initial-Value Problem

Essentially, we are trying to solve an initial-value problem: given a system whose initial state is known, we want to predict its subsequent behaviour.

Stating the initial-value problem in general terms, we can describe a system by its state vector $\mathbf{s}$, which is a function of space and time. If $\mathbf{s}$ is defined at time $t = 0$, we want to find $\mathbf{s}$ for all time t by solving the initial-value equation

$$\frac{d\mathbf{s}}{dt} = \mathbb{D}\mathbf{s} \tag{2.1}$$

where $\mathbb{D}$ is a matrix operator that provides the details of the differential equations governing the system (Potter 1973).

For example, consider a particle with a state described by its position and velocity, and a matrix operator describing its equation of motion as it orbits a single body of mass M located at the origin of the co-ordinate system.

$$\mathbf{s} = (\mathbf{r}(t), \mathbf{v}(t)) \text{ and } \mathbb{D} = \begin{pmatrix} 0 & 1 \\ -\frac{GM}{|r|^3} & 0 \end{pmatrix} \tag{2.2}$$

Substitution into the initial-value equation gives the equations of motion for the particle as

$$\frac{d\mathbf{r}}{dt} = \mathbf{v} \text{ and } \frac{d\mathbf{v}}{dt} = -\frac{GM}{|r|^3}\mathbf{r} \tag{2.3}$$

For a system of N gravitationally interacting particles, $\mathbf{s}$ becomes a vector with $2N$ components and $\mathbb{D}$ becomes a non-linear $2N \times 2N$ matrix, but the essential principles remain the same.

2.2.1 Discretisation in Time

We need to integrate the initial-value equation in time, and achieve this by computation over small steps, dividing time into small finite intervals Δt so that

$$\mathbf{s}(t + \Delta t) = \mathbf{s}(t) + \int_t^{t+\Delta t} \mathbb{D}\mathbf{s} dt \tag{2.4}$$

Since $\mathbf{s}(t)$ is not known for all times between t and $t+\Delta t$, the integral cannot be evaluated exactly. A finite difference approximation is used, which for small timesteps uses a finite Taylor expansion, resulting in (evaluated to second-order accuracy)

$$\mathbf{s}(t + \Delta t) = \mathbf{s}(t) + \mathbb{D}\mathbf{s}(t)\Delta t + \left(\frac{d}{dt}(\mathbb{D}\mathbf{s}(t)\right) \frac{(\Delta t)^2}{2} \tag{2.5}$$

To achieve first-order accuracy in time, we can ignore the final term in the expansion, and only the use of values at the immediate time step (t) is required.

To achieve second-order accuracy in time, we need to define the time derivative in the final term. This will require either the use of values from the previous time step ($t - \Delta t$), the use of values at intermediate time steps ($t + \frac{\Delta t}{2}$), or even the use of values at the subsequent time step ($t + \Delta t$) itself. The flexibility in this choice allows for a range of possible second-order integration schemes.

Integrating over small time steps, the initial-value problem is basically reduced to the problem of obtaining a sequence of solutions at discrete time steps, where the state at a given time step is related to the state at the previous time step by some time integration operator $\mathbb{T}(\Delta t)$.

$$\mathbf{s}(t + \Delta t) = \mathbb{T}(\Delta t)\mathbf{s}(t) \tag{2.6}$$

2.3 Requirements for an Integration Scheme

The integration operator $\mathbb{T}$ is not unique, allowing for a choice of integration schemes. This choice is guided by requiring that an integration scheme be consistent, accurate, efficient and stable. The requirement for consistency is fundamental. Accuracy and efficiency are inversely proportional to each other, and a suitable compromise must be reached. Numerical stability is vitally important, and the most complex requirement to address, especially since it is often dependent upon the form of the specific equations being modelled.

2.3.1 Consistency

For the integration scheme $\mathbb{T}$ to approximate the differential system $\mathbb{D}$, then in the limit of infinitesimal time step they must be identical.

$$\lim_{\Delta t \to 0} \frac{\mathbb{T}(\Delta t) - \mathbb{I}}{\Delta t} = \mathbb{D} \tag{2.7}$$

where $\mathbb{I}$ is an identity matrix. This condition is fundamental, because if it is not satisfied then the initial-value problem of interest is not being modelled. While necessary it is not sufficient, however, because the finite time steps used will produce errors which cause the solution obtained from computation to deviate from the true differential solution.

2.3.2 Accuracy

Accuracy is limited by rounding and truncation errors.

Rounding Error

Rounding error is introduced by the computer representation of numbers, which uses a discrete number of bits (usually 32, or 64 when using double precision). Arithmetic in this floating-point form (where 1 bit is used for representing the sign of the number, 8 bits for the exponent and 23 bits for the mantissa: the fractional values after an assumed leading 1. in the significand) is not exact (Carter 2006). Accuracy is limited by the 23-bit mantissa, which is equivalent to about 7 decimal places ($\log_{10}(2^{23}) = 6.92$), so rounding error is not usually a major issue, but can become important in the limit of very small (or large) numbers, and in particular their summation order. The smallest decimal value that can be represented is $\sim 10^{-35}$ (the largest being $\sim 10^{35}$). A number that is 7 orders of magnitude smaller than another cannot effectively be added to it ($10^{-7}+1$ still equals 1), so numbers should ideally be added from smallest to next smallest and upwards to largest, to minimise this effect.

Using double precision (twice as many bits to represent numbers) can reduce these effects, introducing a trade-off between accuracy and memory usage (a component of efficiency). Double precision uses an 11-bit exponent and a 52-bit mantissa, giving $\log_{10}(2^{52}) = 15.65$ decimal places, and $\sim 10^{\pm 308}$ as the size limit.

Mathematics assumes infinite precision, which is not possible on a computer; while $(a+b)-b = a$ mathematically, this might not be the case on a computer due to rounding error. Floating-point operations possess many intricacies (the decimal value 0.1 is infinitely recurring when represented in binary: $0.00\overline{0110}$) but the conclusion may be stated simply: computer arithmetic is not exact.

Truncation Error

Truncation errors are introduced by the finite difference approximation and by discretisation.

The order to which the Taylor expansion is taken fundamentally limits the accuracy of a numerical method. A scheme of sufficiently high order should be chosen, to minimise the effects of truncation error from cutting off the higher-order terms of the Taylor expansion. With first-order schemes, errors are $\sim \Delta t$, while with second-order schemes, errors are $\sim (\Delta t)^2$.

When continuous variables are discretised in time and space they suffer from incomplete representation. Any behaviour on timescales shorter than the time step, or on size scales smaller than the spatial resolution length, cannot be resolved. These

cut-off points in resolution are also sources of truncation errors. The time step Δt and the spatial resolution length (cell or particle size) should be sufficiently small to minimise truncation errors.

2.3.3 Efficiency

Computers are finite in processing power and memory capacity, so it is not feasible to use an arbitrarily complex integration scheme. Efficiency is defined as the total number of operations required to obtain the solution per unit time of the problem. While efficiency is decreased by using a more complicated higher-order integration scheme, accuracy may be increased, so a suitable compromise between efficiency and accuracy must be settled upon. For simple problems it is appropriate to use sophisticated high-order schemes, but for systems with many variables it is necessary to compromise accuracy for efficiency in order to obtain a solution in a reasonable time span.

2.3.4 Stability

Stability determines whether small errors (from rounding and truncation) can grow without bound. If an integration scheme can produce a solution which is not bounded then the result can be catastrophic, and the scheme is numerically unstable. If an error is amplified with each time step, it will swamp the solution and invalidate the result. For an error ϵ (in general an error vector)

$$\epsilon(t + \Delta t) = A\epsilon(t) \tag{2.8}$$

where A is the amplification factor (in general an amplification matrix, related to the integration operator $\mathbb{T}$). For a numerical method to be stable, any small error at any stage must produce a smaller cumulative error. If it grows larger, it is numerically unstable. Stability therefore requires $|A| \leq 1$.

In the general case of an amplification matrix, error eigenvectors are related between time steps by the eigenvalues A_i, so numerical stability requires that every eigenvalue individually satisfies $|A_i| \leq 1$.

2.4 Integration Schemes and their Properties

We will investigate a variety of basic integration schemes and use them to solve a single first-order ordinary differential equation (ODE) in order to examine their stability (by calculating their error amplification factor A).

Our initial value problem is therefore

$$\frac{ds}{dt} = f(s, t) \tag{2.9}$$

where $s = s(t)$ is now a scalar state variable (rather than a vector), with an initial value $s(t = 0) = s_0$, and f is a function (in place of the matrix operator $\mathbb{D}$).

Integrating this ODE between time steps t_n and $t_{n+1} = t_n + \Delta t$ gives

$$s_{n+1} = s_n + \int_{t_n}^{t_{n+1}} f(s,t)dt \tag{2.10}$$

We must now approximate the integral using some form of integration scheme.

2.4.1 The Euler Scheme

The simplest procedure is to approximate the integrand $f(s,t)$ as $f(s_n,t_n)$ across the whole of the interval $t_n \leq t \leq t_{n+1}$, so that

$$s_{n+1} = s_n + f(s_n,t_n)\Delta t \tag{2.11}$$

This is the Euler scheme (Press et al. 1992). It is accurate to first-order, and its simplicity makes it very efficient, but we must determine whether it is stable.

If we assume that an error ϵ_n exists in the state variable s_n at time t_n then

$$s_{n+1} + \epsilon_{n+1} = s_n + \epsilon_n + f\left((s_n + \epsilon_n), t_n\right)\Delta t \tag{2.12}$$

Assuming that ϵ_n is small, we can approximate $f\left((s_n + \epsilon_n), t_n\right)$ with a Taylor expansion to first order about s_n.

$$f\left((s_n + \epsilon_n), t_n\right) = f(s_n,t_n) + \frac{\partial f}{\partial s}\epsilon_n \tag{2.13}$$

Using the Euler scheme itself to substitute for s_{n+1}, we find that

$$\epsilon_{n+1} = \epsilon_n + \frac{\partial f}{\partial s}\Delta t\epsilon_n \tag{2.14}$$

The error amplification factor A for the Euler scheme is therefore

$$A = 1 + \frac{\partial f}{\partial s}\Delta t \tag{2.15}$$

For the Euler scheme to be numerically stable, $|A| \leq 1$.

If $\frac{\partial f}{\partial s}$ is positive (corresponding to growth equations, where s increases monotonically with time) then it is unstable. The top-left plot in Figure 2.1 demonstrates this, since all numerical solutions diverge from the analytic solution.

If $\frac{\partial f}{\partial s}$ is negative (corresponding to decay equations, where s decreases monotonically with time) then it is stable if $|\frac{\partial f}{\partial s}|\Delta t \leq 2$. This defines an upper limit on the time step as $\Delta t \leq 2/|\frac{\partial f}{\partial s}|$. With a time step larger than this limit the scheme is unstable. For example, consider a typical decay equation $\frac{ds}{dt} = -\frac{s}{\tau}$, which has the analytic solution $s = e^{-t/\tau}$, where τ is the decay timescale. The condition for the Euler scheme to produce a stable numerical solution is $\Delta t \leq 2\tau$. The top-right plot in Figure 2.1 demonstrates this: numerical solutions satisfying the stability condition (dashed and short-dashed lines) converge towards the analytic solution, while failing to satisfy the condition (the dot-dashed line for $\Delta t = 3\tau$) allows errors to grow without bound. Note the neutrally stable solution (the dotted line for $\Delta t = 2\tau$), just

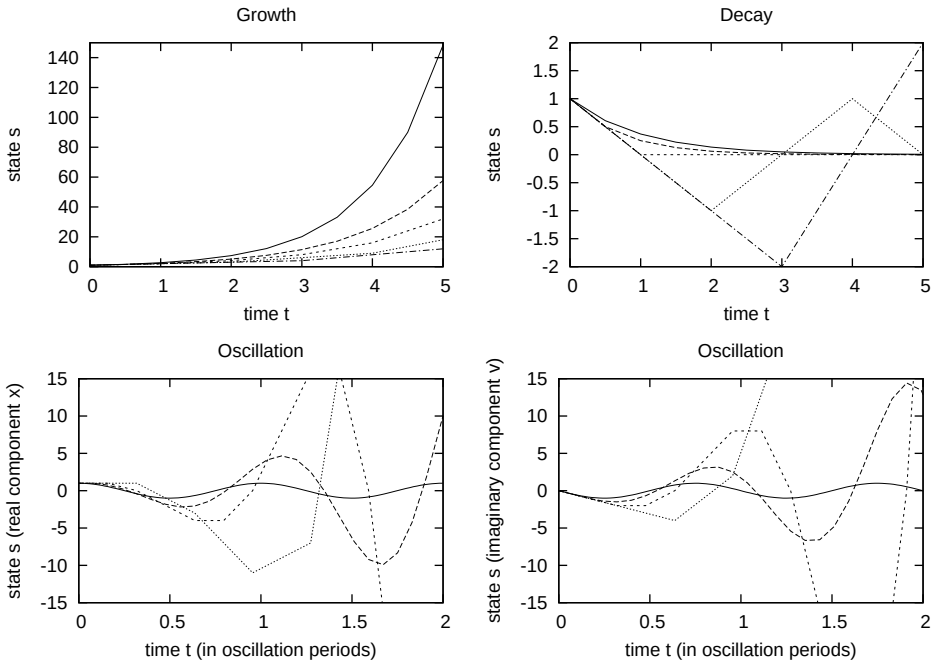


Figure 2.1: Normalised demonstration of the stability properties of the Euler scheme. The solid lines are analytic solutions. For growth and decay equations, the dashed, short-dashed, dotted and dot-dashed lines are numerical solutions with $\Delta t = 0.5, 1, 2$ and 3 respectively, with the growth or decay timescale normalised to $\tau = 1$. For oscillation equations, the dashed, short-dashed and dotted lines are numerical solutions with $\Delta t = \frac{1}{2\omega}, \frac{1}{\omega}$ and $\frac{2}{\omega}$ respectively, with the oscillation frequency normalised to $\omega = 2\pi$ to give oscillation period $T = 1$.

satisfying the condition, where the solution oscillates between -1 and $+1$ (about the asymptotic value of 0) but errors do not grow.

In the case of non-linear equations, where $\frac{\partial f}{\partial s}$ is a function of s, the stability condition can vary with time. It is then advisable to use a time step Δt that can vary with time, in an effort to maintain both numerical stability and optimal efficiency.

If $\frac{\partial f}{\partial s}$ is imaginary then we have an oscillatory equation. When s is complex this corresponds to a pair of coupled equations, where the real and imaginary components of s oscillate with time. For example, consider a simple harmonic oscillator, $\frac{ds}{dt} = -i\omega s$, where $s = x + iv$ and ω is the oscillation frequency. This gives two coupled first-order equations, $\frac{dx}{dt} = \omega v$ and $\frac{dv}{dt} = -\omega x$, which may be combined into the familiar second-order equation $\frac{d^2x}{dt^2} = -\omega x$.

In this oscillatory case, the amplification factor is complex ($A = 1 - i\omega\Delta t$). In general,

$$|A| = \sqrt{AA^*} = \sqrt{1 + \left(\frac{\partial f}{\partial s}\right)^2 (\Delta t)^2} \tag{2.16}$$

Here the function should now be expressed as if, in order to correctly account for the signs involved in multiplying by the complex conjugate. This factor always exceeds unity, so the Euler scheme is always unstable for oscillatory equations. This renders it inappropriate for resolving orbital motion, as we shall see in the next chapter. The bottom plots in Figure 2.1 demonstrate this, since all numerical solutions diverge from the analytic solution.

The Euler scheme is the simplest integration scheme and therefore the prototypical example against which others may be compared.

2.4.2 The Leapfrog Scheme

To improve upon the Euler method we need to move to second-order accuracy. This can be achieved by using an approximation that time-centres the integrand $f(s,t)$ as $f(s_{n+\frac{1}{2}}, t_{n+\frac{1}{2}})$ across the whole of the interval $t_n \leq t \leq t_{n+1}$, using an intermediate time step $t_{n+\frac{1}{2}}$ which introduces a two-level formulation.

$$s_{n+\frac{1}{2}} = s_{n-\frac{1}{2}} + f(s_n, t_n)\Delta t \tag{2.17}$$

$$s_{n+1} = s_n + f(s_{n+\frac{1}{2}}, t_{n+\frac{1}{2}})\Delta t \tag{2.18}$$

This is the Leapfrog scheme (Potter 1973). It is accurate to second-order, but has a problem because the initial conditions only specify $s_0 = s(t = 0)$ while this scheme also requires $s_{-\frac{1}{2}} = s(t = -\frac{\Delta t}{2})$ to be specified. This can be estimated using the Euler scheme ($s_{-\frac{1}{2}} = s_0 - f(s_0, t_0)\frac{\Delta t}{2}$) or ideally some higher-order method (since the overall accuracy of the Leapfrog scheme is strongly affected by the accuracy of this initial estimate).

In determining its stability we need only consider the second equation, since the equations in the two-level formulation are essentially interchangeable.

If an error ϵ_n exists in the state variable s_n at time t_n, then

$$s_{n+1} + \epsilon_{n+1} = s_n + \epsilon_n + f\left((s_{n+\frac{1}{2}} + \epsilon_{n+\frac{1}{2}}), t_{n+\frac{1}{2}}\right)\Delta t \tag{2.19}$$

Assuming that ϵ_n is small, we approximate $f((s_{n+\frac{1}{2}}+\epsilon_{n+\frac{1}{2}}), t_{n+\frac{1}{2}})$ with a Taylor expansion to first order about $s_{n+\frac{1}{2}}$.

$$f\left((s_{n+\frac{1}{2}}+\epsilon_{n+\frac{1}{2}}), t_{n+\frac{1}{2}}\right) = f(s_{n+\frac{1}{2}}, t_{n+\frac{1}{2}}) + \frac{\partial f}{\partial s}\epsilon_{n+\frac{1}{2}} \tag{2.20}$$

Using the Leapfrog scheme itself to substitute for s_{n+1}, we find that

$$\epsilon_{n+1} = \epsilon_n + \frac{\partial f}{\partial s}\Delta t \epsilon_{n+\frac{1}{2}} \tag{2.21}$$

Within the two-level formulation, $\epsilon_{n+1} = A\epsilon_{n+\frac{1}{2}} = A^2\epsilon_n$, where A is the amplification factor, so

$$A^2 = 1 + \frac{\partial f}{\partial s}\Delta t A \tag{2.22}$$

There are two roots for the amplification factor A.

$$A = -\frac{1}{2}\frac{\partial f}{\partial s}\Delta t \pm \sqrt{\left(\frac{1}{2}\frac{\partial f}{\partial s}\Delta t\right)^2 + 1} \tag{2.23}$$

For the Leapfrog scheme to be numerically stable, $|A| \leq 1$ for both roots.

If $\frac{\partial f}{\partial s}$ is real then one of the roots will always be greater than unity, making the scheme unstable. This makes the Leapfrog scheme inappropriate for growth or decay equations. The top plots in Figure 2.2 demonstrate this, since all numerical solutions diverge from the analytic solution.

If $\frac{\partial f}{\partial s}$ is imaginary, as is the case for a simple harmonic oscillator ($\frac{ds}{dt} = -i\omega s$), then the amplification factor A is complex, and (expressing the function as if) has magnitude

$$|A| = \sqrt{AA^*} = \sqrt{\left(\frac{1}{2}\frac{\partial f}{\partial s}\Delta t\right)^2 \mp \left(\frac{1}{2}\frac{\partial f}{\partial s}\Delta t\right)^2 + 1} \tag{2.24}$$

Taking the root with the minus-sign gives $A = 1$ (which is sufficient for stability), while the root with the plus-sign gives $A = \sqrt{2\left(\frac{1}{2}\frac{\partial f}{\partial s}\Delta t\right)^2 - 1}$. This results in the stability condition $\frac{\partial f}{\partial s}\Delta t \leq 2$. For the simple harmonic oscillator, $\Delta t \leq \frac{2}{\omega}$.

The Leapfrog scheme can therefore be stable for oscillatory equations. This renders it appropriate for resolving closed orbits, as we shall see in the next chapter. The bottom plots in Figure 2.2 demonstrate this: numerical solutions satisfying the stability condition (dashed and short-dashed lines) converge towards the analytic solution. Note that the neutrally stable solution (dotted line) here allows errors to grow without bound; this is due to rounding error causing the stability condition to be broken. Because number representation on a computer is not exact, a certain buffer of caution should be used to account for this fact. Strange behaviour can occur as the neutrally stable solution is approached: when $\Delta t = \frac{1.99}{\omega}$, amplitudes can grow but then decay, themselves oscillating with an envelope frequency, as shown in Figure 2.3. The buffer of caution should be chosen to screen out such behaviour; $\Delta t = \frac{1.5}{\omega}$ is probably sufficient.

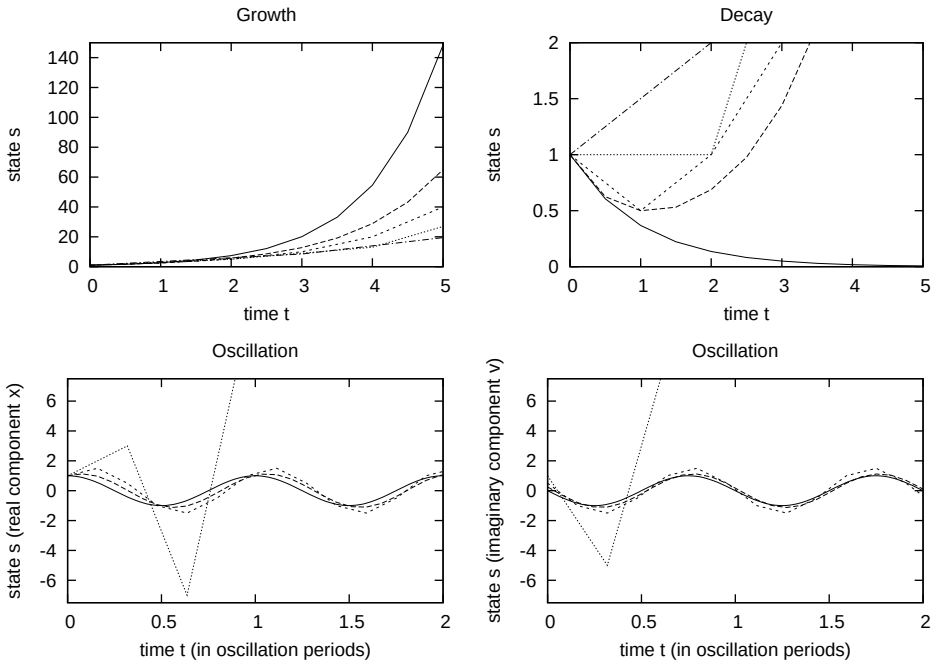


Figure 2.2: Normalised demonstration of the stability properties of the Leapfrog scheme. The solid lines are analytic solutions. For growth and decay equations, the dashed, short-dashed, dotted and dot-dashed lines are numerical solutions with $\Delta t = 0.5, 1, 2$ and 3 respectively, with the growth or decay timescale normalised to $\tau = 1$. For oscillation equations, the dashed, short-dashed and dotted lines are numerical solutions with $\Delta t = \frac{1}{2\omega}, \frac{1}{\omega}$ and $\frac{2}{\omega}$ respectively, with the oscillation frequency normalised to $\omega = 2\pi$ to give oscillation period $T = 1$.

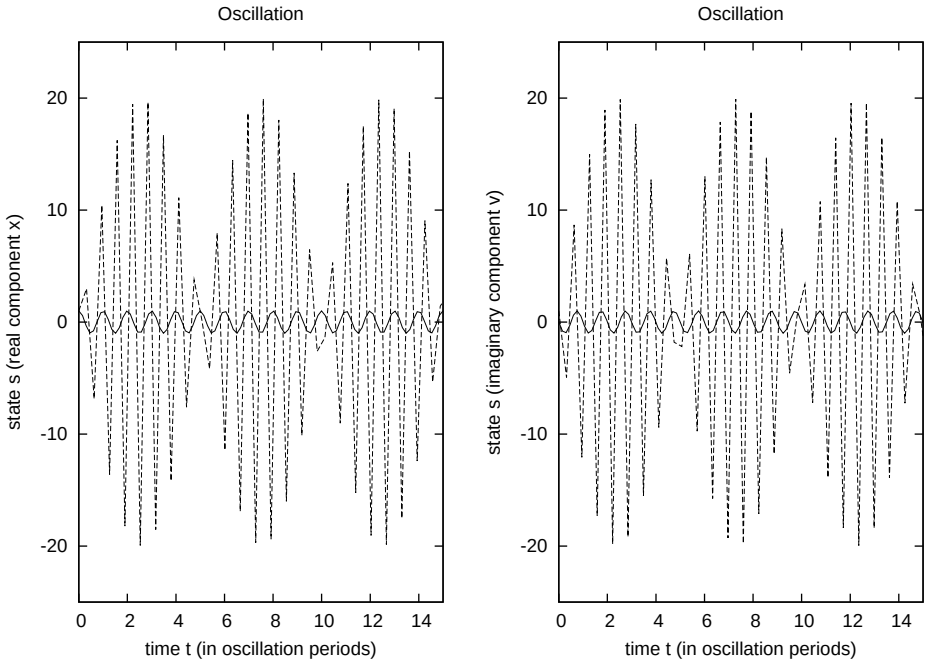


Figure 2.3: Normalised demonstration of the strange behaviour of the Leapfrog scheme as the neutrally stable solution is approached. The solid lines are analytic solutions. The dashed lines are numerical solutions with $\Delta t = \frac{1.99}{\omega}$, with the oscillation frequency normalised to $\omega = 2\pi$ to give oscillation period $T = 1$.

Although conditionally stable against unbounded growth in amplitude, it should be noted that an error in phase does accumulate over time, so a small timestep Δt is needed to minimise this.

It should also be noted that in the case of non-linear equations, where $\frac{\partial f}{\partial s}$ is a function of s, and the time step Δt is varied with time (to maintain stability and efficiency), the Leapfrog scheme ceases to be time-centred due to its two-level formulation. This can become a source of numerical errors.

Nonetheless it is the simplest and most efficient second-order scheme, and will be modified in the next chapter to maintain time-centred integration under varying time steps and to provide stability for open (hyperbolic) orbits.

2.4.3 The Runge-Kutta Scheme

Another approach is to use an approximation that time-centres the integrand with a two-step scheme, again using an intermediate time step $t_{n+\frac{1}{2}}$. Unlike the two-level formulation of the Leapfrog scheme, where the intermediate level forms part of the solution, here it is essentially an auxiliary calculation. The value of s_n is first extrapolated to the intermediate time step using the Euler scheme, and then this intermediate value of $s_{n+\frac{1}{2}}$ is used to inform the approximation of the integrand $f(s,t)$ as $f(s_{n+\frac{1}{2}}, t_{n+\frac{1}{2}})$ across the whole of the interval $t_n \leq t \leq t_{n+1}$, giving

$$s_{n+\frac{1}{2}} = s_n + f(s_n, t_n)\frac{\Delta t}{2} \tag{2.25}$$

$$s_{n+1} = s_n + f(s_{n+\frac{1}{2}}, t_{n+\frac{1}{2}})\Delta t \tag{2.26}$$

This scheme is accurate to second-order and, being essentially identical to the Runge-Kutta method taken to second-order, can be referred to as the Runge-Kutta scheme (Press et al. 1992). While the Euler scheme is a direct application of the Taylor approximation of the time integral taken to first-order, this scheme is essentially a direct application of the Taylor expansion to second-order.

To determine its stability, both steps must be considered.

If an error ϵ_n exists in the state variable s_n at time t_n, then

$$s_{n+\frac{1}{2}} + \epsilon_{n+\frac{1}{2}} = s_n + \epsilon_n + f\left((s_n + \epsilon_n), t_n\right)\frac{\Delta t}{2} \tag{2.27}$$

$$s_{n+1} + \epsilon_{n+1} = s_n + \epsilon_n + f\left((s_{n+\frac{1}{2}} + \epsilon_{n+\frac{1}{2}}), t_{n+\frac{1}{2}}\right)\Delta t \tag{2.28}$$

Assuming that ϵ_n is small, we approximate $f((s_n + \epsilon_n), t_n)$ and $f((s_{n+\frac{1}{2}} + \epsilon_{n+\frac{1}{2}}), t_{n+\frac{1}{2}})$ with Taylor expansions to first order about s_n and $s_{n+\frac{1}{2}}$.

$$f\left((s_n + \epsilon_n), t_n\right) = f(s_n, t_n) + \frac{\partial f}{\partial s}\epsilon_n \tag{2.29}$$

$$f\left((s_{n+\frac{1}{2}} + \epsilon_{n+\frac{1}{2}}), t_{n+\frac{1}{2}}\right) = f(s_{n+\frac{1}{2}}, t_{n+\frac{1}{2}}) + \frac{\partial f}{\partial s}\epsilon_{n+\frac{1}{2}} \tag{2.30}$$

Using the Runge-Kutta scheme itself to substitute for $s_{n+\frac{1}{2}}$ and s_{n+1}, we find that

$$\epsilon_{n+\frac{1}{2}} = \epsilon_n + \frac{\partial f}{\partial s}\frac{\Delta t}{2}\epsilon_n \tag{2.31}$$

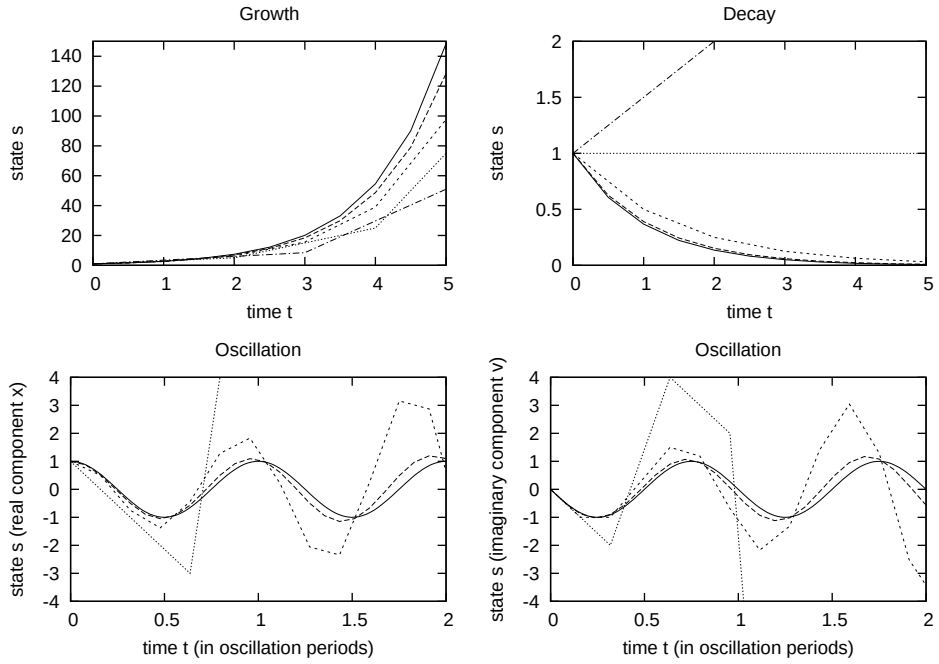


Figure 2.4: Normalised demonstration of the stability properties of the Runge-Kutta scheme. The solid lines are analytic solutions. For growth and decay equations, the dashed, short-dashed, dotted and dot-dashed lines are numerical solutions with $\Delta t = 0.5, 1, 2$ and 3 respectively, with the growth or decay timescale normalised to $\tau = 1$. For oscillation equations, the dashed, short-dashed and dotted lines are numerical solutions with $\Delta t = \frac{1}{2\omega}, \frac{1}{\omega}$ and $\frac{2}{\omega}$ respectively, with the oscillation frequency normalised to $\omega = 2\pi$ to give oscillation period $T = 1$.

$$\epsilon_{n+1} \quad = \quad \epsilon_n + \frac{\partial f}{\partial s}\Delta t \epsilon_{n+\frac{1}{2}} \tag{2.32}$$

Combining these results gives

$$\epsilon_{n+1} = \epsilon_n + \frac{\partial f}{\partial s}\Delta t \left(1 + \frac{\partial f}{\partial s}\frac{\Delta t}{2}\right)\epsilon_n \tag{2.33}$$

Within the two-step formulation, $\epsilon_{n+1} = A\epsilon_n$, where A is the amplification factor, so

$$A = 1 + \frac{\partial f}{\partial s}\Delta t + \left(\frac{\partial f}{\partial s}\right)^2 \frac{(\Delta t)^2}{2} \tag{2.34}$$

For the Runge-Kutta scheme to be numerically stable, $|A| \leq 1$.

If $\frac{\partial f}{\partial s}$ is positive (corresponding to growth equations) then it is unstable. The top-left plot in Figure 2.4 demonstrates this, since all numerical solutions diverge from the analytic solution.

If $\frac{\partial f}{\partial s}$ is negative (corresponding to decay equations) then it is stable if $|\frac{\partial f}{\partial s}|\Delta t \leq 2$. The stability condition constrains the time step $\Delta t \leq 2/|\frac{\partial f}{\partial s}|$. The top-right plot in Figure 2.4 demonstrates this: numerical solutions satisfying the stability condition

(dashed and short-dashed lines) converge towards the analytic solution, while failing to satisfy the condition (the dot-dashed line for $\Delta t = 3\tau$) allows errors to grow without bound. Note the neutrally stable solution (the dotted line for $\Delta t = 2\tau$), just satisfying the condition, where the solution remains at the initial value of 1 but errors do not grow.

If $\frac{\partial f}{\partial s}$ is imaginary, as in the case for a simple harmonic oscillator ($\frac{ds}{dt} = -i\omega s$), then the amplification factor A is complex, and (expressing the function as if) has magnitude

$$|A| = \sqrt{AA^*} = \sqrt{1 + \left(\frac{\partial f}{\partial s}\right)^2 \frac{(\Delta t)^4}{4}} \tag{2.35}$$

Since this is always greater than unity, the Runge-Kutta scheme is numerically unstable for oscillatory equations, but only marginally so because the dependence on the time step is to the fourth power. The bottom plots in Figure 2.4 demonstrate this, since all numerical solutions gradually diverge from the analytic solution.

The Runge-Kutta scheme behaves similarly to the Euler scheme in terms of stability, but gives a higher order of accuracy. While it is a simple and useful method, it is unfortunately marginally unstable when resolving closed orbits, as we shall see in the next chapter.

2.4.4 The Implicit Scheme

A different approach is to use an approximation that uses a time-average for the integrand $f(s,t)$ across the whole of the interval $t_n \leq t \leq t_{n+1}$, giving

$$s_{n+1} = s_n + \frac{1}{2}\left(f(s_n, t_n) + f(s_{n+1}, t_{n+1})\right)\Delta t \tag{2.36}$$

This scheme is accurate to second-order, but the inclusion of $f(s_{n+1}, t_{n+1})$ makes it an implicit method (Potter 1973), and therefore more difficult to implement than the explicit methods shown so far.

Once the form of f is specified, the equation can be rearranged and solved algebraically if possible; otherwise it must be solved iteratively to find a consistent solution. Such iteration comes at considerable computational expense, which hampers efficiency. Even algebraic solutions typically involve division, which is one of the more expensive floating-point operations to perform.

To determine its stability, if an error ϵ_n exists in the state variable s_n at time t_n, then

$$s_{n+1} + \epsilon_{n+1} = s_n + \epsilon_n + \frac{1}{2}\left(f(s_n + \epsilon_n, t_n) + f(s_{n+1} + \epsilon_{n+1}, t_{n+1})\right)\Delta t \tag{2.37}$$

Assuming that ϵ_n is small, we approximate $f((s_n + \epsilon_n), t_n)$ and $f((s_{n+1} + \epsilon_{n+1}), t_{n+1})$ with Taylor expansions to first order about s_n and s_{n+1}.

$$f\left((s_n + \epsilon_n), t_n\right) = f(s_n, t_n) + \frac{\partial f}{\partial s}\epsilon_n \tag{2.38}$$

$$f\left((s_{n+1} + \epsilon_{n+1}), t_{n+1}\right) = f(s_{n+1}, t_{n+1}) + \frac{\partial f}{\partial s}\epsilon_{n+1} \tag{2.39}$$

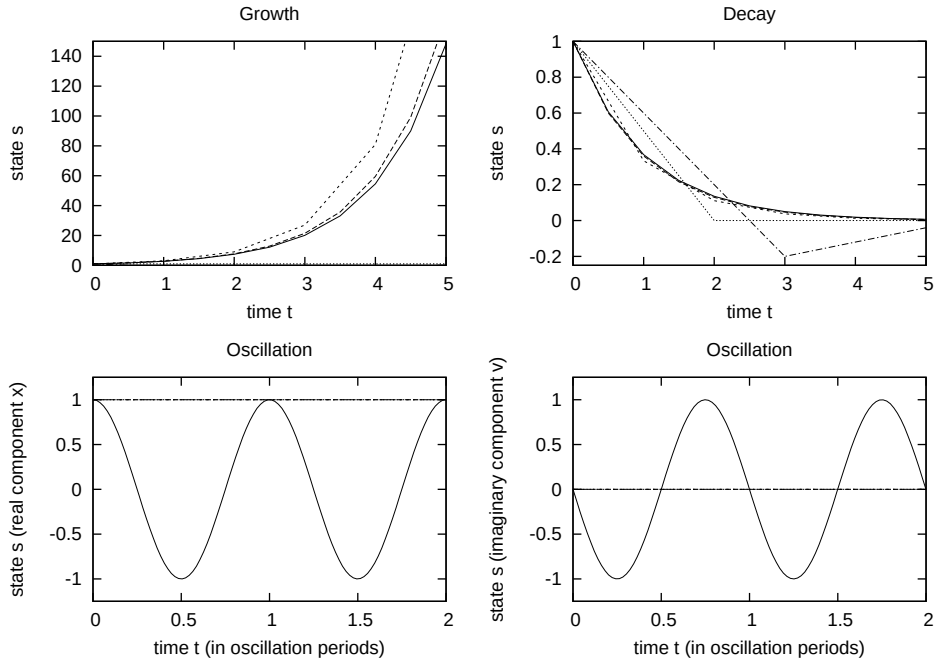


Figure 2.5: Normalised demonstration of the stability properties of the implicit scheme. The solid lines are analytic solutions. For growth and decay equations, the dashed, short-dashed, dotted and dot-dashed lines are numerical solutions with $\Delta t = 0.5, 1, 2$ and 3 respectively, with the growth or decay timescale normalised to $\tau = 1$. For oscillation equations, the dashed, short-dashed and dotted lines are numerical solutions with $\Delta t = \frac{1}{2\omega}, \frac{1}{\omega}$ and $\frac{2}{\omega}$ respectively, with the oscillation frequency normalised to $\omega = 2\pi$ to give oscillation period $T = 1$.

Using the implicit scheme itself to substitute for s_{n+1}, we find that

$$\epsilon_{n+1} = \epsilon_n + \frac{1}{2}\left(\frac{\partial f}{\partial s}\epsilon_n + \frac{\partial f}{\partial s}\epsilon_{n+1}\right)\Delta t \tag{2.40}$$

Rearranging this result into the form $\epsilon_{n+1} = A\epsilon_n$ gives the amplification factor A as

$$A = \frac{1 + \frac{1}{2}\frac{\partial f}{\partial s}\Delta t}{1 - \frac{1}{2}\frac{\partial f}{\partial s}\Delta t} \tag{2.41}$$

For the implicit scheme to be numerically stable, $|A| \leq 1$.

If $\frac{\partial f}{\partial s}$ is positive (corresponding to growth equations) then it is unstable. The top-left plot in Figure 2.5 demonstrates this, since all numerical solutions diverge from the analytic solution.

If $\frac{\partial f}{\partial s}$ is negative (corresponding to decay equations) then it is unconditionally stable. Since there is no stability condition, it is stable for any time step, no matter how large. The top-right plot in Figure 2.5 demonstrates this, since all numerical solutions converge towards the analytic solution.

If $\frac{\partial f}{\partial s}$ is imaginary then the amplification factor A is complex, and its magnitude

is

$$|A| = \sqrt{AA^*} = \sqrt{\frac{1 + \left(\frac{1}{2}\frac{\partial f}{\partial s}\Delta t\right)^2}{1 + \left(\frac{1}{2}\frac{\partial f}{\partial s}\Delta t\right)^2}} \tag{2.42}$$

Since this is always equal to unity, the implicit scheme is unconditionally stable for oscillatory equations. The bottom plots in Figure 2.5 demonstrate that, in its simplest implementation, while errors in the numerical solutions do not grow, they remain at their initial value. A more sophisticated (and therefore less efficient) inplementation is required for true convergence.

The unconditional stability of the implicit scheme makes it very desirable, but this comes with a high computational cost. Fortunately a more efficient explicit scheme can be constructed which shares similar properties.

2.4.5 The Predictor-Corrector Scheme

The implicit scheme can be made explicit by simply estimating the state at t_{n+1} (with a prediction step using the Euler scheme or some higher-order method) as $\tilde{s}_{n+1}$, and then using that to calculate a $f(\tilde{s}_{n+1}, t_{n+1})$ term which can be used to time-average the solution s_{n+1} (in a correction step).

$$\tilde{s}_{n+1} = s_n + f(s_n, t_n)\Delta t \tag{2.43}$$

$$s_{n+1} = s_n + \frac{1}{2}\left(f(s_n, t_n) + f(\tilde{s}_{n+1}, t_{n+1})\right)\Delta t \tag{2.44}$$

This is the Predictor-Corrector scheme (Press et al. 1992). It is accurate to second-order, and its explicit form makes it more efficient.

To determine its stability, both steps must be considered.

If an error ϵ_n exists in the state variable s_n at time t_n, then

$$\tilde{s}_{n+1} + \tilde{\epsilon}_{n+1} = s_n + \epsilon_n + f((s_n + \epsilon_n), t_n)\Delta t \tag{2.45}$$

$$s_{n+1} + \epsilon_{n+1} = s_n + \epsilon_n + \frac{1}{2}\left(f((s_n + \epsilon_n), t_n) + f((\tilde{s}_{n+1} + \tilde{\epsilon}_{n+1}), t_{n+1})\right)\Delta t \tag{2.46}$$

Assuming that ϵ_n is small, we approximate $f((s_n + \epsilon_n), t_n)$ and $f((\tilde{s}_{n+1} + \tilde{\epsilon}_{n+1}), t_{n+1})$ with Taylor expansions to first order about s_n and $\tilde{s}_{n+1}$.

$$f\left((s_n + \epsilon_n), t_n\right) = f(s_n, t_n) + \frac{\partial f}{\partial s}\epsilon_n \tag{2.47}$$

$$f\left((\tilde{s}_{n+1} + \tilde{\epsilon}_{n+1}), t_{n+1}\right) = f(\tilde{s}_{n+1}, t_{n+1}) + \frac{\partial f}{\partial s}\tilde{\epsilon}_{n+1} \tag{2.48}$$

Using the Predictor-Corrector scheme itself to substitute for $\tilde{s}_{n+1}$ and s_{n+1}, we find that

$$\tilde{\epsilon}_{n+1} = \epsilon_n + \frac{\partial f}{\partial s}\Delta t\epsilon_n \tag{2.49}$$

$$\epsilon_{n+1} = \epsilon_n + \frac{1}{2}\left(\frac{\partial f}{\partial s}\epsilon_n + \frac{\partial f}{\partial s}\tilde{\epsilon}_{n+1}\right)\Delta t \tag{2.50}$$

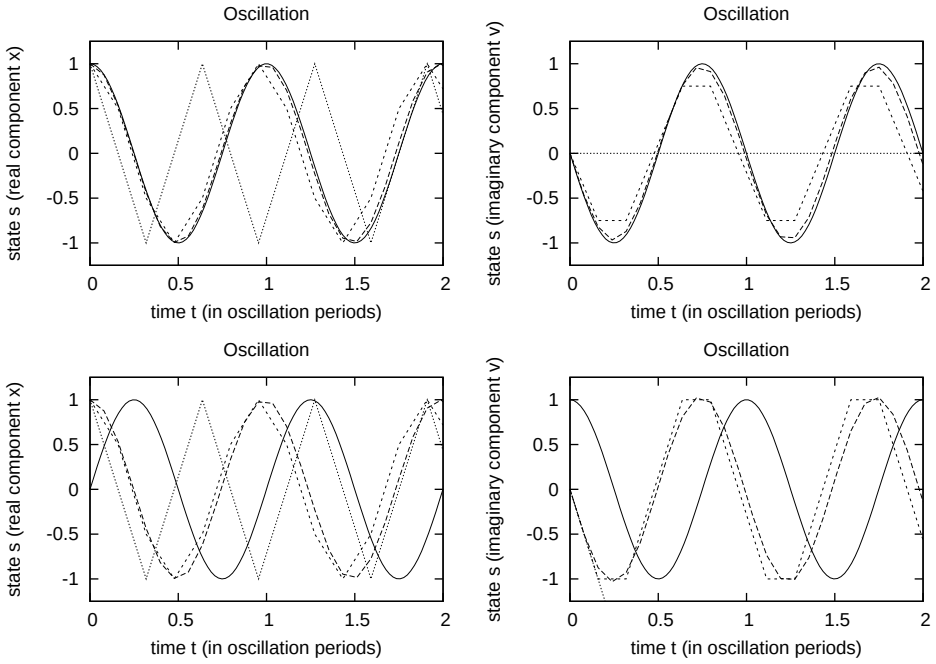


Figure 2.6: Normalised demonstration of the stability properties of the Predictor-Corrector scheme for oscillation equations. The solid lines are analytic solutions. The dashed, short-dashed and dotted lines are numerical solutions with $\Delta t = \frac{1}{2\omega}, \frac{1}{\omega}$ and $\frac{2}{\omega}$ respectively, with the oscillation frequency normalised to $\omega = 2\pi$ to give oscillation period $T = 1$.

Combining these results gives

$$\epsilon_{n+1} = \epsilon_n + \frac{1}{2}\left[\frac{\partial f}{\partial s}\epsilon_n + \frac{\partial f}{\partial s}\left(\epsilon_n + \frac{\partial f}{\partial s}\Delta t \epsilon_n\right)\right]\Delta t \tag{2.51}$$

$$= \epsilon_n\left[1 + \frac{1}{2}\frac{\partial f}{\partial s}\left(2 + \frac{\partial f}{\partial s}\Delta t\right)\Delta t\right] \tag{2.52}$$

Since $\epsilon_{n+1} = A\epsilon_n$, the amplification factor A is

$$A = 1 + \frac{\partial f}{\partial s}\Delta t + \frac{1}{2}\left(\frac{\partial f}{\partial s}\right)^2 (\Delta t)^2 \tag{2.53}$$

This is identical to the Runge-Kutta scheme. In fact, for real functions f, the schemes are algebraically equivalent and therefore possess the same stability properties (unstable for growth equations, but stable for decay equations with the condition $\Delta t \leq 2/|\frac{\partial f}{\partial s}|$). These properties have already been seen in the top plots of Figure 2.4.

It would be expected that the same equivalence would hold for oscillatory equations, and the scheme would be unconditionally unstable like the Runge-Kutta. This is indeed true for the standard implementation, where the correction step uses the immediate value and that of the prediction step for its time-averaging. Coupled equations, however, allow a subtle alteration: one component can use the value from

the correction step of the other component for its time-averaging, instead of the prediction step value. Such a trick is not possible for the Runge-Kutta; it uses an intermediate value rather than time-averaging.

For example, a simple harmonic oscillator has two coupled equations, $\frac{dx}{dt} = \omega v$ and $\frac{dv}{dt} = -\omega x$, which are implemented as

$$\tilde{x}_{n+1} = x_n + \omega v_n \Delta t \tag{2.54}$$
$$\tilde{v}_{n+1} = v_n - \omega x_n \Delta t \tag{2.55}$$

for the prediction step. If the correction step is simply implemented as

$$x_{n+1} = x_n + \frac{1}{2}\omega(v_n + \tilde{v}_{n+1})\Delta t \tag{2.56}$$
$$v_{n+1} = v_n - \frac{1}{2}\omega(x_n + \tilde{x}_{n+1})\Delta t \tag{2.57}$$

then the scheme is unconditionally unstable, in exactly the same manner as the Runge-Kutta scheme (as seen in the bottom plots of Figure 2.4).

Modifying the correction step to

$$x_{n+1} = x_n + \frac{1}{2}\omega(v_n + \tilde{v}_{n+1})\Delta t \tag{2.58}$$
$$v_{n+1} = v_n - \frac{1}{2}\omega(x_n + x_{n+1})\Delta t \tag{2.59}$$

makes the scheme unconditionally stable, in a similar manner to the implicit scheme. Alternatively modifying the correction step to

$$v_{n+1} = v_n - \frac{1}{2}\omega(x_n + \tilde{x}_{n+1})\Delta t \tag{2.60}$$
$$x_{n+1} = x_n + \frac{1}{2}\omega(v_n + v_{n+1})\Delta t \tag{2.61}$$

makes the scheme unconditionally stable in x but only conditionally stable in v, for $\Delta t \leq 2/\frac{\partial f}{\partial s}$. At least this is the case when at $t = 0$, $x(0) = 1$ and $v(0) = 0$. If instead $x(0) = 0$ and $v(0) = 1$ then the first formulation becomes only conditionally stable in x (but unconditionally stable in v), while the second formulation is now unconditionally stable. The implicitly treated state component must initially be zero for unconditionally stability, in order to maintain its average value of zero for all time steps.

The plots in Figure 2.6 demonstrate this: numerical solutions converge towards the analytic solution. In the top-right plot, note how using a large time step stabilises the numerical solution to its average (and initial) value of zero; if instead its initial value was 1, this time step would be conditionally unstable.

Although stable against unbounded growth in amplitude, it should be noted that an error in phase accumulates over time, and a small timestep Δt is needed to minimise this.

The Predictor-Corrector scheme is therefore a very useful method, capable of being unconditionally stable when resolving closed orbits, as we shall see in the next chapter. Although achieving this unconditional stability is dependent upon matching the implementation to the initial conditions, conditional stability is guaranteed for any implementation.

2.4.6 Higher-Order Schemes

More sophisticated schemes exist, such as higher-order Runge-Kutta methods or the fourth-order Hermite scheme. While they can be useful in performing highly accurate gravitational N-body calculations, they are not readily applicable to partial differential equations or fluid dynamics problems. We therefore limit our discussion to second-order methods.

2.5 Complications

Growing Solutions

Although growth equations have been shown to be unconditionally unstable, it is technically possible to relax the stability condition ($|A| \leq 1$) to allow an error to grow without bound provided that it always remains smaller than the growing solution.

Higher-Order Equations

Extending these methods to higher-order ODEs is fairly straightforward, since any equation of order n can be reduced to n first-order ODEs. The stability analysis is made more complicated, since the error amplification factor A is replaced by an error amplification matrix $\mathbb{A}$ of size $n \times n$, but the results are in line with those discussed previously, as we shall see in the next chapter.

N-Body Systems

N-body problems complicate the stability analysis further, since they increase the size of the error amplification matrix by a factor of N. Since gravitation is governed by a second-order equation, it involves a $2N \times 2N$ matrix. The equations are strongly coupled, so the the matrix is not sparse, and it is difficult to obtain the eigenvalues A_i. It seems reasonable, however, to extend the results of lower-N systems to higher-N systems, since stability is most strongly affected by the closest interactions.

Non-Linearity

Non-linear equations affect the stability conditions, making them time-varying. This means that a variable time step is usually necessary to maintain stability (by being smaller than the condition requires) and efficiency (by being as large as possible given the accuracy requirements).

2.6 Partial Differential Equations

Hydrodynamics introduces partial differential equations (PDEs), which have spatial derivatives and therefore couple points in time and space. The solution s of a PDE

can therefore be considered as a composition of waves, or Fourier modes in time t and space x.

$$s(x,t) = \hat{s}e^{i(\omega t - kx)} \tag{2.62}$$

The amplitude of the mode is $\hat{s}$. The angular frequency of the mode is $\omega = \frac{2\pi}{T}$, describing a characteristic period or timescale T. The wavenumber of the mode is $k = \frac{2\pi}{\lambda}$, describing a characteristic wavelength or length scale λ.

The frequency ω can be related to the wavenumber k by a dispersion relation $\omega = \omega(k)$. A real value of ω indicates an oscillation or wave mode, while an imaginary value of ω indicates the growth or decay of a mode.

2.6.1 Wave Propagation

Since we are considering the solution of a PDE to be a superposition of waves, it makes sense to examine the wave equation as the prototypical example. In one dimension,

$$\frac{\partial^2 s}{\partial t^2} = v^2 \frac{\partial^2 s}{\partial x^2} \tag{2.63}$$

The characteristic timescale is the period of the wave, $T = \frac{2\pi}{\omega}$. The characteristic length scale is the wavelength, $\lambda = \frac{2\pi}{k}$.

Introducing a Fourier mode $s(x,t) = \hat{s}e^{i(\omega t - kx)}$ as a solution to the wave equation gives the dispersion relation $\omega = vk$, where v is the phase velocity of the wave; all wave modes propagate at this same speed. Using this result allows the timescale to be expressed as $T = \frac{2\pi}{\omega} = \frac{2\pi}{vk} = \lambda v$.

Stability analysis can be performed upon any numerical approximation to a PDE by introducing a Fourier mode $s_n = \hat{s}_n e^{ikx}$. Considering modes k separately essentially allows the analysis to be decoupled from the spatial discretisation involved in the numerical method.

The mode amplitudes are related by the integration operator $\mathbb{T}$.

$$\hat{s}_{n+1} = e^{-ikx}\mathbb{T}(\Delta t, h)e^{ikx}\hat{s}_n \tag{2.64}$$

Here h is the scale of spatial discretisation, just as Δt is the scale of discretisation in time. This allows an error amplification matrix $\mathbb{A}$ to be determined.

$$\hat{s}_{n+1} = \mathbb{A}(\Delta t, h, k)\hat{s}_n \tag{2.65}$$

Diagonalising this matrix $\mathbb{A}$ relates the amplitudes of Fourier modes between time steps. To be stable, a Fourier mode must be bounded. The condition for stability is therefore that if the amplitude of a Fourier mode is initially finite (at $t = 0$) then it must remain finite for all t. The condition on each eigenvalue then becomes $|A_i|^n \leq K$, where K is some (positive) finite number and n is the number of time steps (amplification occurring between each time step), in order to maintain bounded values. It follows that $|A_i| \leq K^{1/n}$, and over a large time duration ($n \to \infty$) this becomes $|A_i| \leq 1$.

With non-linear PDEs, stability conditions can vary across space, so that they then become local conditions for stability.

2.6.2 Advection

The PDE that describes the convective transport of fluid properties is the advection equation. In its simplest form (linear and without source terms) in one dimension,

$$\frac{\partial s}{\partial t} = -v\frac{\partial s}{\partial x} \tag{2.66}$$

Here v is now the centre-of-mass velocity of a fluid element, not a wave phase velocity. Introducing a Fourier mode again gives the dispersion relation $\omega = vk$. This allows the characteristic timescale to be expressed as $T = \frac{2\pi}{\omega} = \frac{2\pi}{vk} = \frac{\lambda}{v}$, which simply describes the time taken for a fluid element moving with velocity v to traverse a distance λ.

The advection equation can be solved numerically using an explicit first-order scheme.

$$s_{n+1}(x) = s_n(x) - \frac{v}{2h}\left(s_n(x+h) - s_n(x-h)\right)\Delta t \tag{2.67}$$

where we have introduced a spatial discretisation with resolution length h to allow us to approximate the spatial derivative.

To determine its stability, we introduce a Fourier mode $s = \hat{s}e^{ikx}$,

$$\hat{s}_{n+1}e^{ikx} = \hat{s}_n e^{ikx} - \frac{v}{2h}\hat{s}_n\left(e^{ik(x+h)} - e^{ik(x-h)}\right)\Delta t \tag{2.68}$$

so

$$\hat{s}_{n+1} = \hat{s}_n\left(1 - \frac{iv}{h}\sin(kh)\Delta t\right) \tag{2.69}$$

and this gives the error amplification factor A as

$$A(\Delta t, h, k) = 1 - \frac{iv}{h}\sin(kh)\Delta t \tag{2.70}$$

To satisfy the condition for stability, $|A| = |AA^*| \leq 1$, for all wavenumbers k we must take the maximum of the sine function, so

$$1 + \frac{v^2}{h^2}(\Delta t)^2 \leq 1 \tag{2.71}$$

indicating that this scheme is numerically unstable. While consistent and efficient, it is nonetheless not appropriate to solve the advection equation.

An alternative explicit first-order method, the Lax scheme (Lax 1954), uses a spatial average for $s_n(x)$, giving

$$s_{n+1}(x) = \frac{1}{2}\left(s_n(x+h) + s_n(x-h)\right) - \frac{v}{2h}\left(s_n(x+h) - s_n(x-h)\right)\Delta t \tag{2.72}$$

Introducing a Fourier mode $s = \hat{s}e^{ikx}$ gives

$$\hat{s}_{n+1} = \hat{s}_n\left(\cos(kh) - \frac{iv}{h}\sin(kh)\Delta t\right) \tag{2.73}$$

and this gives the error amplification factor A as

$$A(\Delta t, h, k) = \cos(kh) - \frac{iv}{h}\sin(kh)\Delta t \tag{2.74}$$

which has magnitude

$$\begin{aligned} AA^* &= \cos^2(kh) + \frac{v^2}{h^2}\sin^2(kh)(\Delta t)^2 &(2.75)\\ &= 1 - \sin^2(kh)\left(1 - \frac{v^2}{h^2}(\Delta t)^2\right) &(2.76)\end{aligned}$$

To satisfy the condition for stability for all wavenumbers k,

$$\frac{v^2}{h^2}(\Delta t)^2 \leq 1 \tag{2.77}$$

which constrains the time step $\Delta t \leq \frac{h}{|v|}$. This is to be expected, since in our method information can only travel through space at a speed of $\frac{h}{\Delta t}$. Obviously the time step Δt must be chosen so that this speed is sufficient to transmit information across the resolution length h, as demonstrated by our constraint, which requires that this speed be greater than the physical velocity v in order to resolve the convective motion. This is essentially the Courant-Friedrichs-Lewy condition (Courant et al. 1928).

2.6.3 Diffusion

The PDE that describes the conductive transport of fluid properties is the diffusion equation. In its simplest form (linear and without source terms) in one dimension,

$$\frac{\partial s}{\partial t} = \frac{\partial}{\partial x}\left(\kappa\frac{\partial s}{\partial x}\right) \tag{2.78}$$

Here κ is the conductivity (coefficient of diffusion). Introducing a Fourier mode here results in $i\omega = -\kappa k^2$, which gives the dispersion relation $\omega = i\kappa k^2$. The imaginary value of ω indicates that modes decay in time. This allows the characteristic timescale of decay to be expressed as $T = \frac{2\pi}{\omega} = \frac{2\pi}{\kappa k^2} = \frac{\lambda^2}{2\pi\kappa}$, which describes the time taken to diffuse over a distance λ.

The diffusion equation can be solved numerically using an explicit first-order scheme.

$$s_{n+1}(x) = s_n(x) + \frac{\kappa}{h^2}\left(s_n(x+h) - 2s_n(x) + s_n(x-h)\right)\Delta t \tag{2.79}$$

Introducing a Fourier mode $s = \hat{s}e^{ikx}$,

$$\hat{s}_{n+1}e^{ikx} = \hat{s}_n e^{ikx} + \frac{\kappa}{h^2}\hat{s}_n\left(e^{ik(x+h)} - 2e^{ikx} + e^{ik(x-h)}\right)\Delta t \tag{2.80}$$

so

$$\hat{s}_{n+1} = \hat{s}_n\left[1 + \frac{2\kappa}{h^2}\left(\frac{1}{2}e^{ikh} + \frac{1}{2}e^{-ikh} - 1\right)\Delta t\right] \tag{2.81}$$

and this gives the error amplification factor A as

$$A(\Delta t, h, k) = 1 + \frac{2\kappa}{h^2}\left(\cos(kh) - 1\right)\Delta t \tag{2.82}$$

$$= 1 - \frac{4\kappa}{h^2}\sin^2\left(\frac{kh}{2}\right)\Delta t \tag{2.83}$$

To satisfy the condition for stability, $|A| \leq 1$, for all wavenumbers k,

$$\frac{4\kappa}{h^2}\Delta t \leq 2 \tag{2.84}$$

which constrains the time step $\Delta t \leq \frac{h^2}{2\kappa}$. This is to be expected, since in our method information can only travel through space at a speed of $\frac{h}{\Delta t}$ because only values at $x+h$ and $x-h$ are used to integrate in time by Δt at position x. Obviously the time step Δt must be chosen so that this speed is sufficient to transmit information across the resolution length h, as demonstrated by our constraint.

The consistency of the method is demonstrated by introducing a Fourier mode $s = \hat{s}e^{i(\omega t - kx)}$, giving

$$\hat{s}_{n+1}e^{i\omega\Delta t} = \hat{s}_n\left(1 + \frac{2\kappa}{h^2}\left(\cos(kh) - 1\right)\Delta t\right) \tag{2.85}$$

In the limit of small time steps ($\Delta t \to 0$), where $\hat{s}_{n+1} = \hat{s}_n$ and to first-order expansion $e^{i\omega\Delta t} = 1 + i\omega\Delta t$, it follows that

$$1 + i\omega\Delta t = 1 + \frac{2\kappa}{h^2}\left(\cos(kh) - 1\right)\Delta t \tag{2.86}$$

In the limit of large wavelengths ($k \to 0$), then to first-order expansion $\cos(kh) = 1 - \frac{k^2h^2}{2}$, giving

$$i\omega = \frac{2\kappa}{h^2}\left(1 - \frac{k^2h^2}{2} - 1\right) = -\kappa k^2 \tag{2.87}$$

which gives the expected dispersion relation $\omega = i\kappa k^2$, so the method is confirmed as an approximation to the diffusion equation: it is consistent.

Its accuracy is of order Δt in time and h^2 in space.

2.6.4 Numerical Dispersion and Diffusion

The Lax scheme is a stable approximation to the advection equation, while the simpler first-order scheme is unstable. Instability in first-order integration stems from the equations not being time-centred.

We can write the simpler scheme

$$s_{n+1}(x) - s_n(x) = -\frac{v}{2h}\left(s_n(x+h) - s_n(x-h)\right)\Delta t \tag{2.88}$$

in a second-order accurate form as

$$\begin{array}{c} \frac{1}{2}\left(s_{n+1}(x) - s_{n-1}(x)\right) \\ +\frac{1}{2}\left(s_{n+1}(x) - 2s_n(x) + s_{n-1}(x)\right) \end{array} = -\frac{v}{2h}\left(s_n(x+h) - s_n(x-h)\right)\Delta t \tag{2.89}$$

which reveals that it is not quite equivalent to the advective equation but instead models

$$\frac{\partial s}{\partial t} + \frac{\Delta t}{2}\frac{\partial^2 s}{\partial t^2} = -v\frac{\partial s}{\partial x} \tag{2.90}$$

The lack of time-centred integration gives rise to the second term, which is where instability arises since it allows errors to grow without bound.

Writing the Lax scheme in a second-order accurate form gives

$$\begin{array}{c} \frac{1}{2}\left(s_{n+1}(x) - s_{n-1}(x)\right) \\ +\frac{1}{2}\left(s_{n+1}(x) - 2s_n(x) + s_{n-1}(x)\right) \end{array} = \begin{array}{c} -\frac{v}{2h}\left(s_n(x+h) - s_n(x-h)\right)\Delta t \\ +\frac{1}{2}\left(s_n(x+h) - 2s_n(x) + s_n(x-h)\right) \end{array} \tag{2.91}$$

which, through the inclusion of spatial averaging, models

$$\frac{\partial s}{\partial t} + \frac{\Delta t}{2}\frac{\partial^2 s}{\partial t^2} = -v\frac{\partial s}{\partial x} + \frac{h^2}{2\Delta t}\frac{\partial^2 s}{\partial x^2} \tag{2.92}$$

as an approximation to the advective equation. This now has a diffusion term which can counteract the growth term, stabilising the scheme when it is equal to or larger than the growth term. For a Fourier mode $s = \hat{s}e^{i(\omega t - kx)}$, this condition for stability is therefore

$$\frac{\Delta t}{2}\omega \leq \frac{h^2}{2\Delta t}k^2 \tag{2.93}$$

which constrains the time step $\Delta t \leq \frac{hk}{\omega}$. Given the dispersion relation $\omega = vk$, this is $\Delta t \leq \frac{h}{|v|}$ as expected.

When $\Delta t = \frac{h}{|v|}$ then the terms cancel each other and the approximation converges to the advective equation. Unfortunately when $\Delta t < \frac{h}{|v|}$ then, despite being stable, numerical diffusion occurs. This results in modes decaying when they physically should not. In situations where the stability constraint is a local condition, such numerical diffusion is difficult to avoid.

The details of numerical effects can be seen by deriving the dispersion relation for the scheme, by using the Fourier mode $s = \hat{s}e^{i(\omega t - kx)}$ to get

$$e^{i\omega\Delta t} = \cos(kh) - \frac{iv}{h}\sin(kh)\Delta t \tag{2.94}$$

In general, ω is complex, so by considering $\omega = a + ib$ and equating real and imaginary parts,

$$e^{ia\Delta t} = \cos(kh) - \frac{iv}{h}\sin(kh)\Delta t \tag{2.95}$$

$$\Rightarrow \cos(a\Delta t) + i\sin(a\Delta t) = \cos(kh) - i\frac{v\Delta t}{h}\sin(kh) \tag{2.96}$$

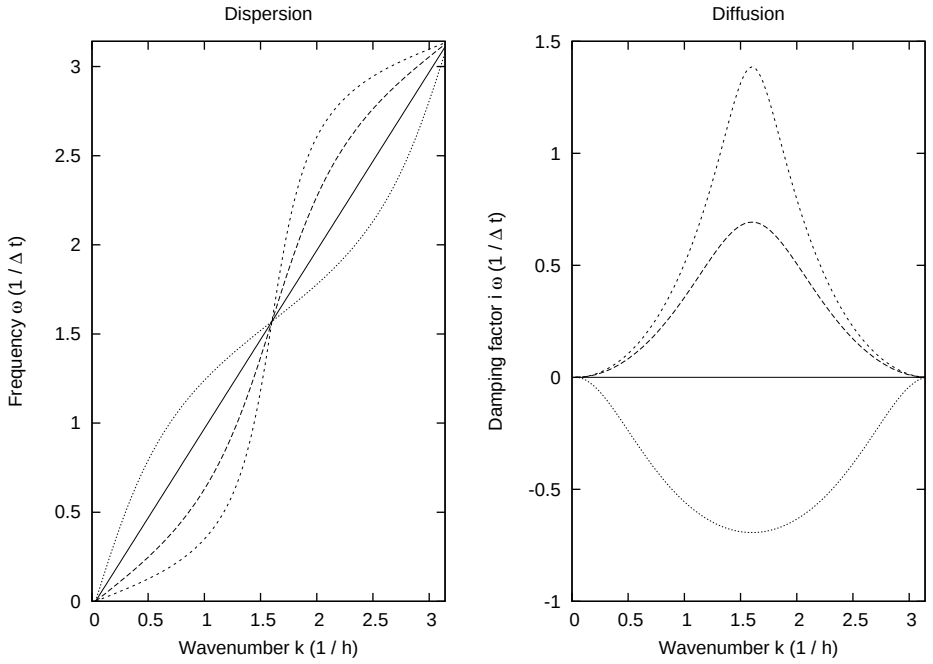


Figure 2.7: Normalised demonstration of numerical dispersion and diffusion for advection equations. The solid lines are the numerical solutions with $\frac{v\Delta t}{h} = 1$, which correspond to the analytic solutions. The dashed, short-dashed and dotted lines are numerical solutions with $\frac{v\Delta t}{h} = \frac{1}{2}, \frac{1}{4}$ and 2 respectively.

and

$$e^{-b\Delta t} = \cos(kh) - \frac{iv}{h}\sin(kh)\Delta t \tag{2.97}$$

$$\Rightarrow e^{-2b\Delta t} = \cos^2(kh) + \left(\frac{v\Delta t}{h}\right)^2 \sin^2(kh) \tag{2.98}$$

When $\Delta t = \frac{h}{v}$ then $a\Delta t = kh$ and $e^{-2b\Delta t} = 1$, which gives $a = vk$ and $b = 0$. This gives the correct dispersion relation $\omega = vk$, with no imaginary component. For any other time step, however, it can be seen that the real component of the dispersion relation becomes non-linear and an imaginary component appears. The non-linear real component gives rise to numerical dispersion, where modes with different wavenumbers propagate with different velocities. The existence of an imaginary component gives rise to numerical diffusion, where modes with different wavenumbers are damped by various amounts.

The plots in Figure 2.7 show these effects, with dispersion on the left and diffusion on the right. Note the negative sign for the damping factor in the numerical solution when the stability condition is broken (dotted line); this indicates terms that are growing rather than decaying.

These numerical effects tend to become worse for larger wavenumbers k (short

wavelength modes), while long wavelength modes ($\lambda \gg h$, so $kh \to 0$) are in reasonably good agreement with the true dispersion relation. From the imaginary component, numerical diffusion is maximised at $kh = \frac{\pi}{2}$, so $\lambda = 4h$ is the most strongly damped mode. The scale of spatial discretisation will therefore affect the results.

Numerical dispersion and diffusion are caused by difference approximations to differential equations. They are most severe in these first-order methods, and higher-order methods reduce their effects but do not eliminate them. They are essentially truncation errors from the Taylor expansion; only by taking an infinite number of terms would exact convergence with the differential equation be achieved.

2.6.5 Conservative Formulation

Non-linear equations allow for a variety of ways in constructing a difference approximation to them. Since many PDEs, such as the advective hydrodynamic equations, are conservative (being derived from principles of conservation), it is sensible to ensure that their difference approximations are also conservative.

$$\frac{\partial s}{\partial t} = -\mathbf{v}\frac{\partial s}{\partial x} \tag{2.99}$$

This conservative equation may be integrated over finite volume elements of space-time $\Delta V \Delta t$ so

$$\int_t^{t+\Delta t} dt \int_V \frac{\partial s}{\partial t} dV = -\int_t^{t+\Delta t} dt \int_V \mathbf{v}\frac{\partial s}{\partial x} dV \tag{2.100}$$

Integrating the left-hand side over time, and applying the divergence theorem to the right-hand side to make it a surface area integral,

$$\int_V s_{n+1} dV - \int_V s_n dV = -\int_t^{t+\Delta t} dt \int_A \mathbf{v} s \cdot dA \tag{2.101}$$

The total quantity s within a volume element can be defined as

$$s_n(x)\Delta V = \int_V s_n dV \tag{2.102}$$

and the flux $s \cdot dA$ (where A is area) can be defined by summing over fluxes F through the adjoining surfaces of N neighbouring volume elements

$$\sum_N F_N(x) = \int_A s \cdot dA \tag{2.103}$$

giving

$$s_{n+1}(x) = s_n(x) - \int_t^{t+\Delta t} dt \frac{1}{\Delta V} \sum_N F_N(x) \tag{2.104}$$

This defines a conservative difference scheme. All that remains is to determine a method to evaluate the fluxes such that they can be integrated in time.

2.6.6 Hyperbolic Equations

The advection equation is hyperbolic. A conservative, explicit first-order accurate method to approximate it is

$$s_{n+1}(x) = s_n(x) - \frac{1}{2h}\left(F_n(x+h) - F_n(x-h)\right)\Delta t \tag{2.105}$$

where $F_n = vs_n$. We have seen that this scheme is unstable, and previously replaced it with the Lax scheme.

A time-centred, second-order accurate method can be constructed using the Leapfrog scheme. Fluxes are defined on the intermediate time step level.

$$F_{n+1/2}(x) = F\left(s_{n+1/2}(x)\right) \tag{2.106}$$

$$s_{n+1}(x) = s_n - \frac{1}{h}\left(F_{n+1/2}(x+h) - F_{n+1/2}(x-h)\right)\Delta t \tag{2.107}$$

$$s_{n+1/2}(x) = s_{n-1/2} - \frac{1}{h}\left(F_n(x+h) - F_n(x-h)\right)\Delta t \tag{2.108}$$

The solutions on the two levels are uncoupled and so may freely drift out of phase. It is therefore advisable to consider only one of the solutions and discard the other, rather than combining them.

Introducing a Fourier mode $s = \hat{s}e^{ikx}$ for the advection equation, where $F = vs$, gives

$$s_{n+1}(x) = s_n(x)\left(1 - \frac{v}{h}\left(s_{n+1/2}e^{ikh} - s_{n+1/2}e^{-ikh}\right)\Delta t\right) \tag{2.109}$$

The error amplification factor A, using $\epsilon_{n+1} = A\epsilon_{n+1/2} = A^2\epsilon_n$, gives

$$A^2 = 1 - i\frac{2v}{h}\sin(kh)\Delta t A \tag{2.110}$$

which has roots

$$A = -i\frac{v}{h}\sin(kh)\Delta t \pm \sqrt{-\left(\frac{v}{h}\sin(kh)\Delta\right)^2 + 1} \tag{2.111}$$

For stability, $AA^* \leq 1$, and

$$AA^* = \left(\frac{v}{h}\sin(kh)\Delta t\right)^2 - \left(\frac{v}{h}\sin(kh)\Delta t\right)^2 + 1 \tag{2.112}$$

so $AA^* = 1$ provided that the term inside the square root is positive (giving a real result not an imaginary one), which gives the condition

$$\frac{v}{h}\sin(kh)\Delta t \leq 1 \tag{2.113}$$

Considering all wavenumbers k, we take the maximum of the sine function, and the time step is contrained to be $\Delta t \leq \frac{h}{|v|}$ for stability. This is the Courant condition.

2.6.7 Parabolic Equations

The diffusion equation is parabolic. A conservative, explicit first-order accurate method to approximate it is

$$s_{n+1}(x) = s_n(x) + \frac{\kappa}{h^2}\left(s_n(x+h) - 2s_n(x) + s_n(x-h)\right)\Delta t \tag{2.114}$$

We have seen that this scheme is stable with the condition that $\Delta t \leq \frac{h^2}{2\kappa}$. Note that the time step is limited here by the square of the spatial resolution length; doubling the spatial resolution quarters the time step, so parabolic equations can place serious limitations upon computational efficiency.

A time-centred, second-order accurate method can again be constructed using the Leapfrog scheme.

$$s_{n+1}(x) = s_n + \frac{2\kappa}{h^2}\left(s_{n+1/2}(x+h) - 2s_{n+1/2}(x) + s_{n+1/2}(x-h)\right)\Delta t \tag{2.115}$$

Introducing a Fourier mode $s = \hat{s}e^{ikx}$ for the diffusion equation gives

$$s_{n+1}(x) = s_n\left(1 + \frac{2\kappa}{h^2}\left(s_{n+1/2}e^{ikh} + s_{n+1/2}e^{-ikh}\right)\Delta t\right) \tag{2.116}$$

The error amplification factor A, using $\epsilon_{n+1} = A\epsilon_{n+1/2} = A^2\epsilon_n$, gives

$$A^2 = 1 + \frac{4\kappa}{h^2}\cos(kh)\Delta t A \tag{2.117}$$

which has roots

$$A = -\frac{2\kappa}{h^2}\cos(kh)\Delta t \pm \sqrt{\left(\frac{2\kappa}{h^2}\cos(kh)\Delta t\right)^2 + 1} \tag{2.118}$$

For stability, $|A| \leq 1$, but the term inside the square root is always positive, so one of the roots is always greater than unity and the scheme, while consistent, is unconditionally unstable.

Therefore schemes that are stable for parabolic equations may be unstable for hyperbolic equations (like the explicit first-order scheme) or vice-versa (like the Leapfrog scheme). Given that complicated problems may exhibit parabolic, hyperbolic and wave-like features, care must be taken in the application of a numerical approximation.

2.6.8 Elliptic Equations

The PDE that describes boundary-value problems is elliptic. A familiar example is Poisson's equation.

$$\nabla^2\phi = -\rho \tag{2.119}$$

Here the form of the source function ρ (mass density) governs the form of the gravitational potential ϕ.

Using the analogy with wave propagation, gravitational interaction assumes that information is transported instantaneously, so the timescale for information to propagate over any length scale is effectively zero ($T \to 0$), making the frequency of a Fourier mode effectively infinite ($\omega \to \infty$).

Solving an elliptic equation such as Poisson's equation using a numerical approximation results in a matrix equation. This can be solved exactly, using an arithmetic approach (such as the Gauss elimination method), but is very inefficient for large matrices, scaling as N^3 (Fox 1964). Alternatively an inexact but consistent solution can be found by iteration to convergence (using Jacobi or Gauss-Seidel iteration, successive over-relaxation, cyclic Chebyshev method, or alternating-direction implicit method (Potter 1973)), but this can require many iteration steps, which also limits efficiency.

We shall see in subsequent chapters that gravitation is the most computationally demanding aspect of our astrophysical problem.

2.7 Further Complications

Growing Modes

To account for the possibility of a growing local term in a system of PDEs, the stability condition $|A_i| \leq 1$ may be relaxed to $|A_i| \leq 1 + O(\Delta t)$ for all wavenumbers k and all eigenvalues i (Richtmyer & Morton 1967).

Multi-Dimensional Explicit Methods

Explicit conservative methods can be extended to more than one spatial dimension, but the Courant-Friedrichs-Lewy condition for stability may need to be modified to $\Delta t \leq \frac{h}{|\mathbf{v}|\sqrt{D}}$, where D is the number of spatial dimensions (Potter 1973).

Non-Linearity

Non-linear PDEs affect the stability conditions, making them space-varying as well as time-varying. This means that a local condition is usually necessary to constrain the time step. It seems reasonable to take the stability conditions obtained from simple linear systems (such as the Courant condition) and apply these as local conditions in more complex non-linear systems.

2.8 Summary

Computer arithmetic is not exact. Whenever a numerical approach is chosen, it is vital to remember that it is inherently an approximation, and to be aware of the limitations and shortcomings involved.

Numerical methods must be consistent with the problem they are attempting to solve, must reach a suitable compromise between accuracy and efficiency, and must be stable against the unbounded growth of numerical errors. Stability is typically

conditional, and stability in one regime does not guarantee stability in others. Care should be taken to ensure that numerical effects such as anomolous dispersion or diffusion do not contaminate the results.

We have investigated the properties of various integration schemes in simple situations, and this has served to highlight some of the difficulties involved. We can now begin to apply them to more complicated problems, with an understanding of the hazards allowing us to proceed with suitable caution.

In the next chapter we will apply numerical integration to gravitational problems, starting with the two-body problem (which can be solved analytically) before modelling a stable three-body system and an N-body sphere in freefall collapse.

"The proof of the pudding
is in the eating."

British proverb

Chapter 3

Gravity

In this chapter we use numerical methods to solve gravitational problems. Since our aim is to model self-gravitating gas, this forms a vital component of the investigation.

By considering the simple two-body problem, which has an analytic solution, integration schemes may be studied comparatively and in detail to reveal their merits and flaws.

An N-body computer program is developed in stages to reveal the issues involved in writing a numerical code, and the concepts of adaptive timestepping and gravitational softening are introduced.

3.1 The Theory of Gravity

Although gravity is the weakest of the fundamental forces, it is the most important in astrophysics because it dominates at large scales. The weak and strong nuclear forces are intrinsically short range, and only affect the sub-atomic domain. The electromagnetic force, while a long range force like gravity, and considerably stronger, can be both attractive and repulsive, so it ultimately cancels out in neutral matter and is only of concern when there is a significant degree of ionisation. Gravity, for all its weakness, is always attractive, and so the more matter involved, the more important it becomes. At astronomical scales its importance becomes overwhelming.

3.1.1 Kepler's Laws of Planetary Motion

The theory of gravity was itself born out of astronomical observations. Using the observational data of planetary motions accumulated by Tycho Brahe, Johannes Kepler showed that the planets did not orbit in circles, and formulated his three laws.

1. The planets have elliptical orbits, with the Sun at one focus.

An ellipse has two foci, and is described when the sum of the distances from each focus is constant. Its shape is described by the eccentricity e, defined as the ratio of the focus-to-centre distance with the semi-major axis a.

The orbit equation, describing the locus of an ellipse, can be obtained from its geometry by applying the cosine rule to the triangle formed by a point on the ellipse

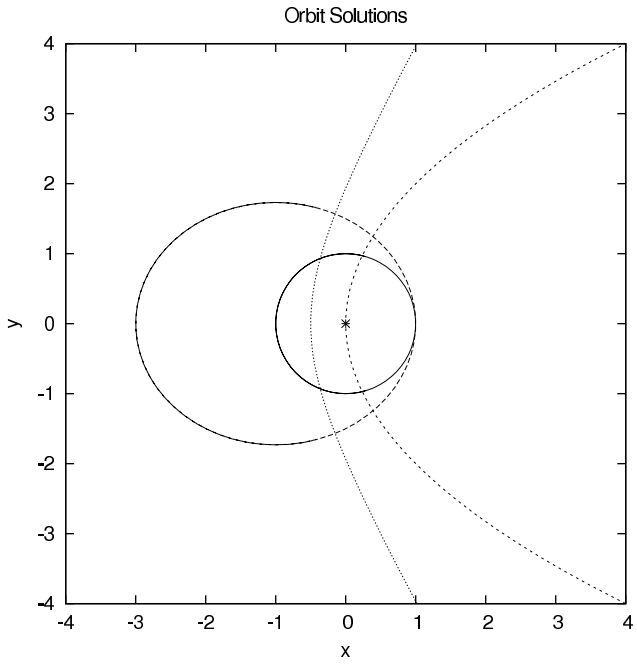


Figure 3.1: Solutions of the orbit equation. The smallest orbit is a circular orbit solution for $e = 0$. The larger closed orbit is an elliptical orbit solution for $e = \frac{1}{2}$. The open orbit passing through the origin is a parabolic orbit solution for $e = 1$. The remaining open orbit is a hyperbolic orbit solution for $e = 2$. The star point at the origin indicates the occupied focus.

and the two foci. Polar coordinates are used, with r being the distance from the focus to the point on the ellipse and θ being the true anomaly (angle from perihelion).

$$r = \frac{a(1 - e^2)}{1 + e\cos\theta} \tag{3.1}$$

This equation describes an elliptical orbit when $0 < e < 1$, with the occupied focus somewhere between the centre and the orbital path. When $e = 0$, it describes a circular orbit, with the focus at the centre. When $e = 1$, it describes a parabolic orbit, with the focus on the orbital path. When $e > 1$, it describes a hyperbolic orbit, with the focus outside the orbital path.

Circular and elliptical orbits have finite perimeters, forming closed orbits. Parabolic and hyperbolic orbits are infinite, and therefore open orbits. Figure 3.1 shows all four types of solutions to the orbit equation. These analytic solutions will be used later to examine the stability of numerical solutions to the gravitational two-body problem of a planet orbiting a star.

2. The position vector from the Sun to a planet sweeps out equal areas in equal times.

It follows that the orbital velocity $v = r\frac{d\theta}{dt} \sim r^{-1}$ (indicating that a planet moves fastest at perihelion) and that the specific angular momentum of the planet $h =$

$\mathbf{r} \times \mathbf{v} = r^2 \frac{d\theta}{dt}$ is constant, so Kepler's second law is essentially a statement of the conservation of angular momentum.

3. The cube of the semi-major axis of the orbit is proportional to the square of the sidereal period.

$a^3 = kT^2$, where k is Kepler's constant.

3.1.2 Newton's Law of Gravitation

Isaac Newton, using his laws of motion, considered how Kepler's laws worked. To do this, the radial component of the acceleration of an orbiting planet must be considered.

$$a_r = \frac{d^2r}{dt^2} - r\left(\frac{d\theta}{dt}\right)^2 = \frac{d^2r}{dt^2} - \frac{h^2}{r^3} \tag{3.2}$$

The second derivative can be calculated by implicit differentiation of the reciprocal of the orbit equation, and ultimately leads to

$$a_r = \frac{h^2}{r^2}\left(\frac{e\cos\theta}{a(1-e^2)} - \frac{1}{r}\right) = -\frac{h^2}{a(1-e^2)}\frac{1}{r^2} \tag{3.3}$$

The radial acceleration is negative (directed towards the occupied focus along r), and proportional to r^{-2}. This indicates that the planet is under the influence of a force acting towards the occupied focus which obeys an inverse square law. Expressed in terms of Kepler's constant:

$$a_r = -\frac{4\pi^2 k}{r^2} \tag{3.4}$$

Introducing ideas from Newton's equations of motion (of force being proportional to inertial mass, and of action and reaction being equal and opposite) gives rise to the concept of mutually interacting pairs of masses (m_1 and m_2) and the inverse square law of gravitation.

$$F_r = -\frac{Gm_1m_2}{r^2} \tag{3.5}$$

where G is the gravitational constant ($6.67 \times 10^{-11}\,\mathrm{m^3\,kg^{-1}\,s^{-2}}$).

In vector form, Newton's law of gravitation is

$$F_1 = m_1\frac{d^2\mathbf{r}_1}{dt^2} = -\frac{Gm_1m_2}{|\mathbf{r}_1 - \mathbf{r}_2|^3}(\mathbf{r}_1 - \mathbf{r}_2) \tag{3.6}$$

Since the law of gravitation is essentially derived from Kepler's laws, it follows that the orbit equation is the solution to the equation of motion for a system of two gravitationally interacting bodies, where the reference frame is chosen such that one body occupies a focus while the other body orbits about it.

It is also interesting to reflect that Newton's theory (published in 1687) was based upon Kepler's work (published from around 1605 to 1619) building upon the earlier observations of Brahe. One of the most fundamental theories from the dawn of modern physics in the 17th century ultimately developed from observations by a 16th century astronomer who had worked with a geocentric conception of the solar system.

3.2 The Two-Body Problem

Any self-gravitating system can ultimately be considered as an ensemble of mass elements whose masses m, positions $\mathbf{r}$ and velocities $\mathbf{v}$ are all known at some initial point in time $t = 0$. To determine how the system evolves, for each mass element we make the simplifying assumption that its mass does not vary, and need to find $\mathbf{r}$ and $\mathbf{v}$ as functions of t, which entails solving its equation of motion.

For a system of N mass elements we have a system of N second-order ordinary differential equations, all coupled to each other via the law of gravitation; such a system cannot be solved in general by analytic means, so only a numerical method can provide us with the answer. When fluid dynamics is incorporated in the next chapter, partial differential equations appear, making the situation even more complicated.

For the simplest case of a binary star system, where $N = 2$ and each mass element represents a star, we have a pair of coupled equations. By describing the state of this system in terms of relative positions and velocities we can reduce this to a single equation which has an analytic solution. This two-body problem can then be used as a basis for comparison with numerical solutions, to ensure convergence with the analytic solution.

Given a system of two stars with masses m_1 and m_2, initial positions $\mathbf{r}_1$ and $\mathbf{r}_2$, and initial velocities $\mathbf{v}_1$ and $\mathbf{v}_2$, we set out to determine their subsequent motions.

Each star experiences a gravitational force towards the other star, so their equations of motion are

$$\frac{d^2\mathbf{r}_1}{dt^2} = -\frac{Gm_2}{(\mathbf{r}_1 - \mathbf{r}_2)^2}\frac{\mathbf{r}_1 - \mathbf{r}_2}{|\mathbf{r}_1 - \mathbf{r}_2|} \tag{3.7}$$

$$\frac{d^2\mathbf{r}_2}{dt^2} = -\frac{Gm_1}{(\mathbf{r}_2 - \mathbf{r}_1)^2}\frac{\mathbf{r}_2 - \mathbf{r}_1}{|\mathbf{r}_2 - \mathbf{r}_1|} \tag{3.8}$$

Using the relative position (i.e. separation) $\mathbf{r} = \mathbf{r}_2 - \mathbf{r}_1$ gives

$$\frac{d^2\mathbf{r}_1}{dt^2} = \frac{Gm_2}{r^3}\mathbf{r} \tag{3.9}$$

$$\frac{d^2\mathbf{r}_2}{dt^2} = -\frac{Gm_1}{r^3}\mathbf{r} \tag{3.10}$$

Consideration of the gravitational forces experienced by each body reveals that they are equal and opposite, as expected:

$$m_1\frac{d^2\mathbf{r}_1}{dt^2} + m_2\frac{d^2\mathbf{r}_2}{dt^2} = 0 \tag{3.11}$$

This can be written as

$$\frac{d^2}{dt^2}\left(m_1\mathbf{r}_1 + m_2\mathbf{r}_2\right) = 0 \tag{3.12}$$

Comparing this with the position of the centre-of-mass of the system

$$\mathbf{r}_{CoM} = \frac{m_1\mathbf{r}_1 + m_2\mathbf{r}_2}{m_1 + m_2} \tag{3.13}$$

reveals that $\frac{d^2\mathbf{r}_{CoM}}{dt^2} = 0$, so the centre of mass experiences no acceleration, remaining at rest or at uniform velocity, and can be used as an inertial reference frame. Within this frame, the equation of motion for the separation is

$$\frac{d^2\mathbf{r}}{dt^2} = \frac{d^2\mathbf{r}_2}{dt^2} - \frac{d^2\mathbf{r}_1}{dt^2} = -\frac{Gm_1}{r^3}\mathbf{r} - \frac{Gm_2}{r^3}\mathbf{r} = -\frac{G(m_1+m_2)}{r^3}\mathbf{r} \tag{3.14}$$

This is essentially the same as the law of gravitation for a body of negligible mass in orbit around a body of mass $(m_1 + m_2)$, so the analytic solution for this system must be the orbit equation.

3.2.1 Numerical Stability

The binary system can be described by a state vector $\mathbf{s}(t)$, where $\mathbf{s} = (\mathbf{r}(t), \mathbf{v}(t))$ describes the separation and effective orbital velocity (the orbital velocity of one body in the rest frame of the other body). If at time $t = 0$ the initial separation and effective orbital velocity are supplied, the subsequent orbital evolution at all times t can be evaluated.

Essentially we are solving

$$\frac{d\mathbf{s}}{dt} = \mathbb{D}\mathbf{s} \text{ where } \mathbb{D} = \begin{pmatrix} 0 & 1 \\ -\frac{GM}{|r|^3} & 0 \end{pmatrix} \tag{3.15}$$

with $M = m_1 + m_2$ being the total mass of the system. This gives the equations of motion $\frac{d\mathbf{r}}{dt} = \mathbf{v}$ and $\frac{d\mathbf{v}}{dt} = -\frac{GM}{r^3}\mathbf{r}$, which can be solved numerically.

The binary system as described by a second-order ODE, equivalent to two coupled first-order ODEs, has an error amplification matrix $\mathbb{A}$ of size 2×2.

In general, for an integration scheme described by the time integration operator $\mathbb{T}(\mathbf{s}, \delta t)$, errors are related by $\mathbf{s}_{n+1} + \epsilon_{n+1} = \mathbb{T}(\mathbf{s}_n + \epsilon_n, \Delta t)$. Assuming that ϵ_n is small, we approximate the right hand side with a Taylor expansion to first order about $\mathbf{s}_n$.

$$\mathbb{T}(\mathbf{s}_n + \epsilon_n, \Delta t) = \mathbb{T}(\mathbf{s}_n, \Delta t) + \left(\frac{\partial}{\partial \mathbf{s}_n}(\mathbb{T}\mathbf{s}_n)\right)\epsilon_n \tag{3.16}$$

Since $\mathbf{s}_{n+1} = \mathbb{T}\mathbf{s}_n$,

$$\epsilon_{n+1} = \left(\frac{\partial \mathbf{s}_{n+1}}{\partial \mathbf{s}_n}\right)\epsilon_n \tag{3.17}$$

The amplification matrix for the binary system is therefore

$$\mathbb{A} = \begin{pmatrix} \frac{\partial \mathbf{r}_{n+1}}{\partial \mathbf{r}_n} & \frac{\partial \mathbf{r}_{n+1}}{\partial \mathbf{v}_n} \\ \frac{\partial \mathbf{v}_{n+1}}{\partial \mathbf{r}_n} & \frac{\partial \mathbf{v}_{n+1}}{\partial \mathbf{v}_n} \end{pmatrix} \tag{3.18}$$

Numerical stability is determined by the 2 eigenvalues A_i of this matrix, with $|A_i| \leq 1$ required for stability.

$$\left(\frac{\partial \mathbf{r}_{n+1}}{\partial \mathbf{r}_n} - A\right)\left(\frac{\partial \mathbf{v}_{n+1}}{\partial \mathbf{v}_n} - A\right) - \left(\frac{\partial \mathbf{r}_{n+1}}{\partial \mathbf{v}_n}\right)\left(\frac{\partial \mathbf{v}_{n+1}}{\partial \mathbf{r}_n}\right) = 0 \tag{3.19}$$

gives the quadratic equation

$$A^2 - \left(\frac{\partial \mathbf{r}_{n+1}}{\partial \mathbf{r}_n} + \frac{\partial \mathbf{v}_{n+1}}{\partial \mathbf{v}_n}\right)A - \left(\frac{\partial \mathbf{r}_{n+1}}{\partial \mathbf{v}_n}\right)\left(\frac{\partial \mathbf{v}_{n+1}}{\partial \mathbf{r}_n}\right) = 0 \tag{3.20}$$

which can be solved to give the two eigenvalues A.

3.2.2 Initial Conditions

The orbit equation (Equation 3.1) may describe a circle ($e = 0$), an ellipse ($0 < e < 1$), a parabola ($e = 1$) or a hyperbola ($e > 1$). The eccentricity e and the semi-major axis a will determine the initial values of position r and velocity v.

The gravitational potential energy is a maximum at the greatest separation (apastron), when $r = a(1+e)$. At this point, the specific kinetic energy is $\frac{1}{2}\frac{h^2}{r^2}$. Using Kepler's laws, and noting that Kepler's constant $k = \frac{GM}{4\pi^2}$, and the area of an ellipse $A^2 = \pi^2 a^4(1-e^2)$, the specific kinetic energy is $\frac{1}{2}\frac{GMa(1-e^2)}{r^2}$.

The total specific energy at apastron is therefore

$$-\frac{GM}{a(1+e)} + \frac{1}{2}\frac{GM(1-e^2)}{a(1+e)^2} = -\frac{GM}{2a}\left(\frac{2}{1+e} - \frac{1-e^2}{(1+e)^2}\right) = -\frac{GM}{2a} \tag{3.21}$$

Applying conservation of energy, at any point in the orbit

$$\frac{1}{2}v_\theta^2 - \frac{GM}{r} = -\frac{GM}{2a} \tag{3.22}$$

Specifying the initial conditions at periastron, when the true anomaly $\theta = 0$ and the distance from the focus is $r = a(1-e)$,

$$v_\theta^2 = GM\left(\frac{2}{a(1-e)} - \frac{1}{a}\right) = GM\left(\frac{2-(1-e)}{a(1-e)}\right) = GM\frac{1+e}{r} \tag{3.23}$$

which allows the initial tangential velocity to be calculated.

3.2.3 The Euler Scheme

We will begin by considering the simplest integration scheme: the first-order Euler method.

$$\mathbf{r}_{n+1} = \mathbf{r}_n + \mathbf{v}_n\Delta t \tag{3.24}$$
$$\mathbf{v}_{n+1} = \mathbf{v}_n + \mathbf{a}_n\Delta t \tag{3.25}$$

where the acceleration $\mathbf{a}_n = -\frac{GM}{|\mathbf{r}_n|^3}\mathbf{r}_n$ is a function of position.

Evaluating the components of the error amplification matrix $\mathbb{A}$ gives the eigenvalue equation as

$$A^2 - (1+1)A - (\Delta t)\left(\frac{\partial \mathbf{a}_n}{\partial \mathbf{r}_n}\Delta t\right) = 0 \tag{3.26}$$

which gives the eigenvalues

$$A = 1 \pm \sqrt{1 + \frac{\partial \mathbf{a}_n}{\partial \mathbf{r}_n}(\Delta t)^2} \tag{3.27}$$

Since one of these values is always greater than unity, the Euler scheme is numerically unstable. The plots in Figure 3.2 demonstrate this, since the numerical solutions strongly diverge from the analytic solution even with a small timestep.

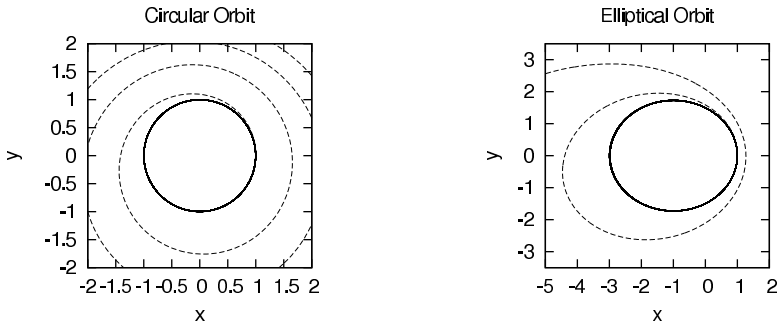


Figure 3.2: Normalised demonstration of the orbital stability properties of the Euler scheme. The solid lines are analytic solutions and the dashed lines are numerical solutions with $\Delta t = \frac{2\pi}{100}$ over 1000 integration steps. The circle has a radius of 1, and the ellipse has eccentricity $e = \frac{1}{2}$ giving a semi-major axis of 2.

3.2.4 The Leapfrog Scheme

Moving to second-order, we consider the Leapfrog scheme.

$$\mathbf{r}_{n+1} = \mathbf{r}_n + \mathbf{v}_{n+\frac{1}{2}}\Delta t \tag{3.28}$$

$$\mathbf{v}_{n+\frac{1}{2}} = \mathbf{v}_{n-\frac{1}{2}} + \mathbf{a}_n\Delta t \tag{3.29}$$

This requires an initial extrapolation for the value of $\mathbf{v}_{-\frac{1}{2}}$, which can be made with an Euler step $\mathbf{v}_{-\frac{1}{2}} = \mathbf{v}_0 - \mathbf{a}_n\frac{\Delta t}{2}$.

The two-level formulation makes evaluating the components of the error amplification matrix $\mathbb{A}$ complicated, but the stability properties can be shown graphically. The top plots in Figure 3.3 demonstrate stability with a small timestep. The bottom plots use a large timestep to reveal deviations from the true orbital path, notably significant variation in the orbital parameters (eccentricity and semi-major axis) and the precession of the apastron position.

3.2.5 The Runge-Kutta Scheme

The Runge-Kutta scheme can be implemented to second-order.

$$\mathbf{r}_{n+1} = \mathbf{r}_n + \mathbf{v}_{n+\frac{1}{2}}\Delta t \tag{3.30}$$

$$\mathbf{v}_{n+1} = \mathbf{v}_n + \mathbf{a}_{n+\frac{1}{2}}\Delta t \tag{3.31}$$

This introduces a "halfstep" midway between timesteps t and $t+\Delta t$, at which the acceleration must be evaluated to provide the value of $\mathbf{a}_{n+\frac{1}{2}}$.

$$\mathbf{r}_{n+\frac{1}{2}} = \mathbf{r}_n + \mathbf{v}_n\frac{\Delta t}{2} \tag{3.32}$$

$$\mathbf{v}_{n+\frac{1}{2}} = \mathbf{v}_n + \mathbf{a}_n\frac{\Delta t}{2} \tag{3.33}$$

The halfstep makes evaluating the components of the error amplification matrix $\mathbb{A}$ complicated, so the stability properties are shown graphically. The top plots in

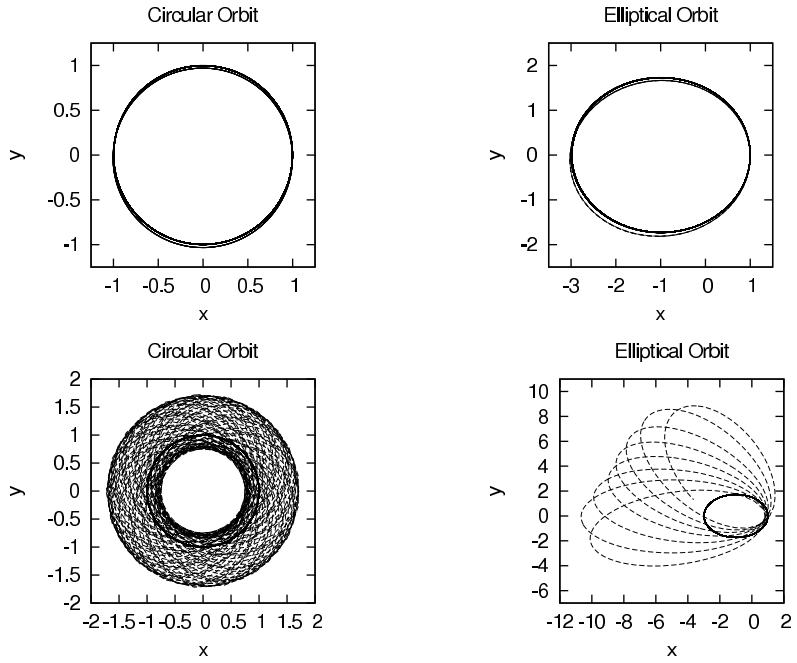


Figure 3.3: Normalised demonstration of the orbital stability properties of the Leapfrog scheme. The solid lines are analytic solutions and the dashed lines are numerical solutions over 1000 integration steps with $\Delta t = \frac{2\pi}{100}$ for the top plots and $\Delta t = \frac{2\pi}{10}$ for the bottom plots. The circle has a radius of 1, and the ellipse has eccentricity $e = \frac{1}{2}$ giving a semi-major axis of 2.

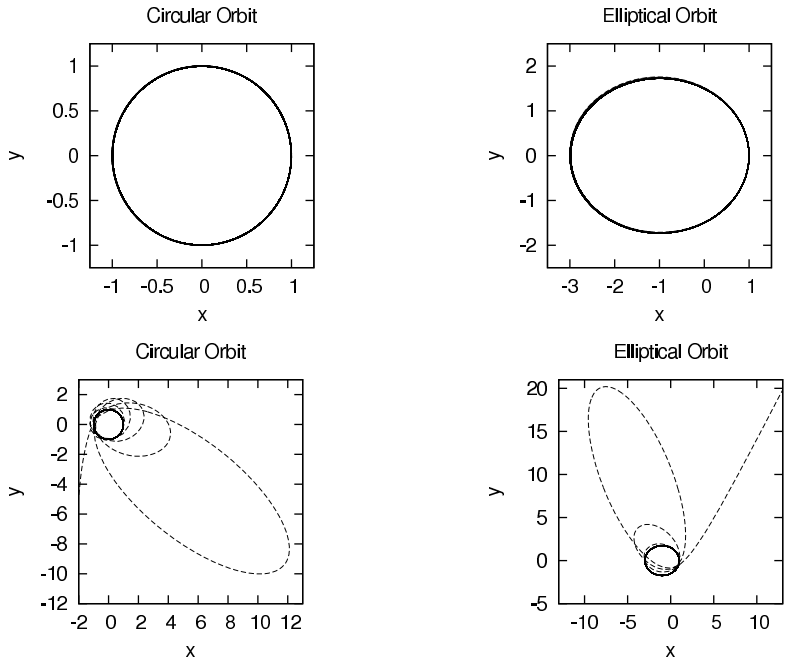


Figure 3.4: Normalised demonstration of the orbital stability properties of the Runge-Kutta scheme. The solid lines are analytic solutions and the dashed lines are numerical solutions over 1000 integration steps with $\Delta t = \frac{2\pi}{100}$ for the top plots and $\Delta t = \frac{2\pi}{10}$ for the bottom plots. The circle has a radius of 1, and the ellipse has eccentricity $e = \frac{1}{2}$ giving a semi-major axis of 2.

Figure 3.4 demonstrate stability with a small timestep. The bottom plots use a large timestep to reveal deviations from the true orbital path, and only the inner parts of the numerical solutions are shown since they are found to be unstable.

3.2.6 The Predictor-Corrector Scheme

The Predictor-Corrector scheme can be implemented by using a prediction step to extrapolate the position to second-order accuracy using Taylor expansion, and then calculating the acceleration at this extrapolated position in order to time-average the velocity solution using a correction step.

$$\mathbf{r}_{n+1} = \mathbf{r}_n + \mathbf{v}_n \Delta t + \mathbf{a}_n \frac{(\Delta t)^2}{2} \tag{3.34}$$

$$\mathbf{v}_{n+1} = \mathbf{v}_n + \frac{\mathbf{a}_n + \mathbf{a}_{n+1}}{2} \Delta t \tag{3.35}$$

Although there is no halfstep, two acceleration values are used in the formulation so evaluating the components of the error amplification matrix $\mathbb{A}$ is complicated and the stability properties are shown graphically. The top plots in Figure 3.5 demonstrate stability with a small timestep. The bottom plots use a large timestep to reveal deviations from the true orbital path, notably some variation in the orbital parameters (eccentricity and semi-major axis), but much less than was observed with the Leapfrog scheme, and again the precession of the apastron position.

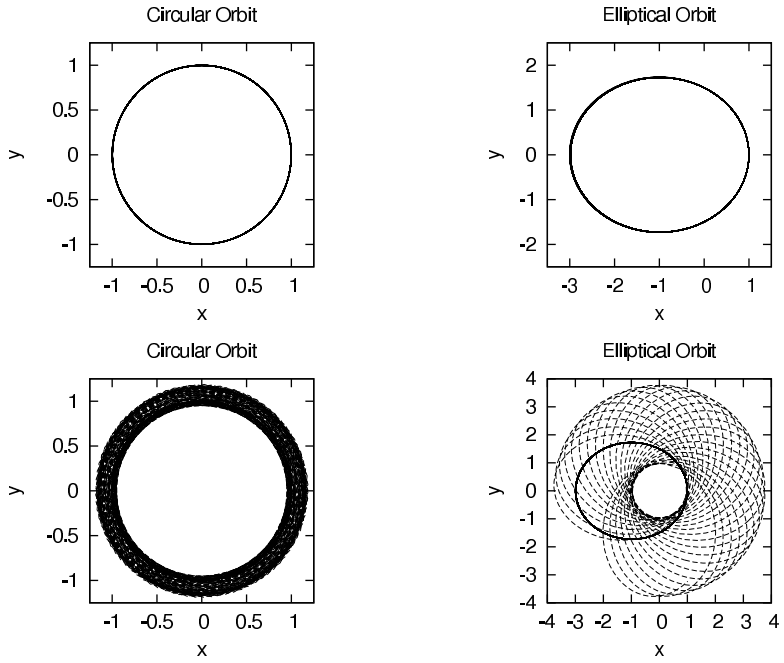


Figure 3.5: Normalised demonstration of the orbital stability properties of the Predictor-Corrector scheme. The solid lines are analytic solutions and the dashed lines are numerical solutions over 1000 integration steps with $\Delta t = \frac{2\pi}{100}$ for the top plots and $\Delta t = \frac{2\pi}{10}$ for the bottom plots. The circle has a radius of 1, and the ellipse has eccentricity $e = \frac{1}{2}$ giving a semi-major axis of 2.

3.2.7 The Modified Leapfrog Scheme

A modification can be made to the Leapfrog scheme. Instead of using a two-level formulation with position and velocity on different levels (and acceleration evaluated on the same level as position), we can choose to evaluate position and velocity on the same level. This provides an improved solution of the state of the system since the full state is now known at each timestep, but it requires the acceleration to be evaluated on a different level for second-order accuracy. The "halfstep" midway between timesteps t and $t + \Delta t$ can be used to provide this acceleration $\mathbf{a}_{n+\frac{1}{2}}$.

$$\mathbf{r}_{n+\frac{1}{2}} = \mathbf{r}_n + \mathbf{v}_n \frac{\Delta t}{2} \tag{3.36}$$

$$\mathbf{v}_{n+\frac{1}{2}} = \mathbf{v}_n + \mathbf{a}_{n-\frac{1}{2}} \Delta t \tag{3.37}$$

This requires a modification on the first timestep, since $\mathbf{a}_{-\frac{1}{2}}$ is not known. Instead the initial acceleration $\mathbf{a}_0$ is evaluated, and $\mathbf{v}_{\frac{1}{2}} = \mathbf{v}_0 + \mathbf{a}_0 \frac{\Delta t}{2}$.

The halfstep acceleration can then be used to evaluate the subsequent velocity, and this can be used to time-average the position solution in a similar manner to the Predictor-Corrector scheme.

$$\mathbf{v}_{n+1} = \mathbf{v}_n + \mathbf{a}_{n+\frac{1}{2}} \Delta t \tag{3.38}$$

$$\mathbf{r}_{n+1} = \mathbf{r}_n + \frac{\mathbf{v}_n + \mathbf{v}_{n+1}}{2} \Delta t \tag{3.39}$$

The halfstep makes evaluating the components of the error amplification matrix $\mathbb{A}$ complicated, so the stability properties are shown graphically. The top plots in Figure 3.6 demonstrate stability with a small timestep. The bottom plots use a large timestep to reveal deviations from the true orbital path, notably some small variation in the orbital parameters (eccentricity and semi-major axis), but less than was observed with the Predictor-Corrector scheme, and again the precession of the apastron position. The ubiquitous apastron precession is due to the accumulation of phase errors; while it might be minimised with a small timestep it will never be eliminated.

The traditional Leapfrog scheme can be described as a "kick-drift-kick" implementation (KDK), since the acceleration is evaluated on the fullsteps (giving the "kick") but not the halfsteps (where it simply "drifts"). The modified Leapfrog scheme can therefore be described as a "drift-kick-drift" implementation (DKD) (Springel 2005). Since this modified Leapfrog scheme can be expressed as a form of Predictor-Corrector (predicting the velocity and correcting the position), the Predictor-Corrector scheme in Section 3.2.6 (predicting the position and correcting the velocity) could also be expressed as a modified Leapfrog with a KDK implementation.

3.2.8 Conservation of Energy

While a reasonable idea of stability can be gained from these observations of the position vector as a function of time, it is perhaps more easily seen via examination

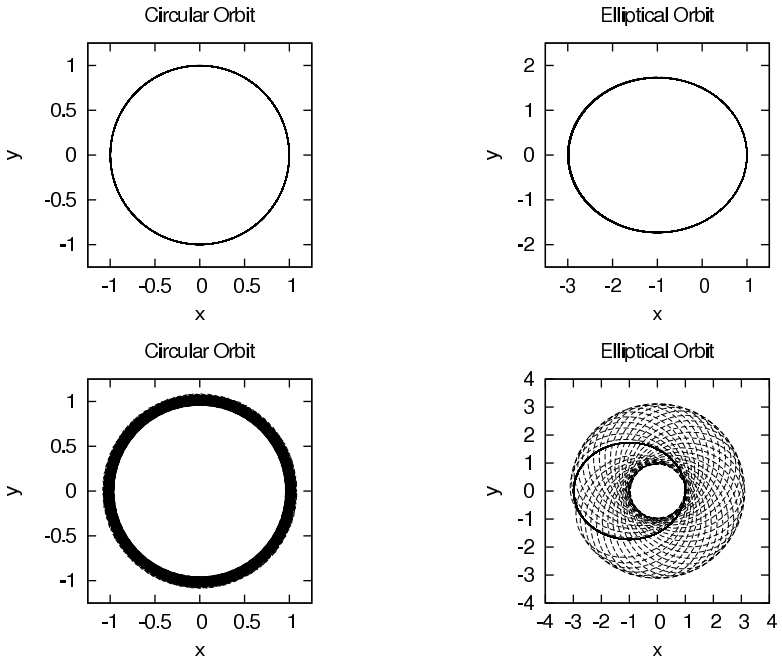


Figure 3.6: Normalised demonstration of the orbital stability properties of the Modified Leapfrog scheme. The solid lines are analytic solutions and the dashed lines are numerical solutions over 1000 integration steps with $\Delta t = \frac{2\pi}{100}$ for the top plots and $\Delta t = \frac{2\pi}{10}$ for the bottom plots. The circle has a radius of 1, and the ellipse has eccentricity $e = \frac{1}{2}$ giving a semi-major axis of 2.

of a scalar quantity. Energy is the obvious choice since it is an orbital constant. Stability can therefore be gauged by how well energy is conserved.

Since the conservation of energy is a consequence of a system being symmetric with respect to time through the application of Noether's theorem, it follows that time symmetry should be a good indicator of stability for an integration scheme. If an integration scheme is time-symmetric then stepping backwards in time, using a timestep of $-\Delta t$, will return the system to its initial state.

For the Euler scheme,

$$\begin{array}{rcccl} \mathbf{r}_{n+1} - \mathbf{v}_{n+1}\Delta t & = & (\mathbf{r}_n + \mathbf{v}_n\Delta t) - \mathbf{v}_{n+1}\Delta t & \neq & \mathbf{r}_n \\ \mathbf{v}_{n+1} - \mathbf{a}_{n+1}\Delta t & = & (\mathbf{v}_n + \mathbf{a}_n\Delta t) - \mathbf{a}_{n+1}\Delta t & \neq & \mathbf{v}_n \end{array} \tag{3.40}$$

so it is not time-symmetric, and its numerical instability comes as no surprise.

For the Leapfrog scheme,

$$\begin{array}{rcccl} \mathbf{r}_{n+1} - \mathbf{v}_{n+\frac{1}{2}}\Delta t & = & (\mathbf{r}_n + \mathbf{v}_{n+\frac{1}{2}}\Delta t) - \mathbf{v}_{n+\frac{1}{2}}\Delta t & = & \mathbf{r}_n \\ \mathbf{v}_{n+\frac{1}{2}} - \mathbf{a}_n\Delta t & = & (\mathbf{v}_{n-\frac{1}{2}} + \mathbf{a}_n\Delta t) - \mathbf{a}_n\Delta t & = & \mathbf{v}_{n-\frac{1}{2}} \end{array} \tag{3.41}$$

so it is time-symmetric between states $(\mathbf{r}_n, \mathbf{v}_{n-\frac{1}{2}})$ and $(\mathbf{r}_{n+1}, \mathbf{v}_{n+\frac{1}{2}})$.

For the Runge-Kutta scheme,

$$\begin{array}{rcccl} \mathbf{r}_{n+1} - \mathbf{v}_{n+\frac{1}{2}}\Delta t & = & (\mathbf{r}_n + \mathbf{v}_{n+\frac{1}{2}}\Delta t) - (\mathbf{v}_{n+1} + \frac{1}{2})\Delta t & = & \mathbf{r}_n \\ \mathbf{v}_{n+1} - \mathbf{a}_{n+\frac{1}{2}}\Delta t & = & (\mathbf{v}_n + \mathbf{a}_{n+\frac{1}{2}}\Delta t) - \mathbf{a}_{n+\frac{1}{2}}\Delta t & = & \mathbf{v}_n \end{array} \tag{3.42}$$

so it is time-symmetric between states $(\mathbf{r}_n, \mathbf{v}_n)$ and $(\mathbf{r}_{n+1}, \mathbf{v}_{n+1})$.

For the Predictor-Corrector scheme,

$$\begin{array}{rcccl} \mathbf{r}_{n+1} - \mathbf{v}_{n+1}\Delta t + \mathbf{a}_{n+1}\frac{(\Delta t)^2}{2} & = & (\mathbf{r}_n + \mathbf{v}_n\Delta t + \mathbf{a}_n\frac{(\Delta t)^2}{2}) - & & \\ & & (\mathbf{v}_n + \frac{\mathbf{a}_n+\mathbf{a}_{n+1}}{2}\Delta t)\Delta t + \mathbf{a}_{n+1}\frac{(\Delta t)^2}{2} & = & \mathbf{r}_n \\ \mathbf{v}_{n+1} - \frac{\mathbf{a}_{n+1}+\mathbf{a}_n}{2}\Delta t & = & (\mathbf{v}_n + \frac{\mathbf{a}_n+\mathbf{a}_{n+1}}{2}\Delta t) - & & \\ & & \frac{\mathbf{a}_{n+1}+\mathbf{a}_n}{2}\Delta t & = & \mathbf{v}_n \end{array} \tag{3.43}$$

so it is time-symmetric between states $(\mathbf{r}_n, \mathbf{v}_n)$ and $(\mathbf{r}_{n+1}, \mathbf{v}_{n+1})$.

For the modified Leapfrog scheme,

$$\begin{array}{rcccl} \mathbf{v}_{n+1} - \mathbf{a}_{n+\frac{1}{2}}\Delta t & = & (\mathbf{v}_n + \mathbf{a}_{n+\frac{1}{2}}\Delta t) - \mathbf{a}_{n+\frac{1}{2}}\Delta t & = & \mathbf{v}_n \\ \mathbf{r}_{n+1} - \frac{\mathbf{v}_{n+1}+\mathbf{v}_n}{2}\Delta t & = & (\mathbf{r}_n + \frac{\mathbf{v}_n+\mathbf{v}_{n+1}}{2}\Delta t) - \frac{\mathbf{v}_{n+1}+\mathbf{v}_n}{2}\Delta t & = & \mathbf{r}_n \end{array} \tag{3.44}$$

so it is time-symmetric between states $(\mathbf{r}_n, \mathbf{v}_n)$ and $(\mathbf{r}_{n+1}, \mathbf{v}_{n+1})$.

The Euler scheme is clearly unsuitable for resolving orbital motion, but the time-symmetric second-order schemes all seem capable of doing so, given a small enough timestep. As seen previously in Figures 3.3 to 3.6, with an arbitrarily chosen time resolution of 100 timesteps per orbit, all the second-order schemes satisfactorily resolved the circular and elliptical orbits.

The specific energy of the orbiting body is $\frac{1}{2}v^2 - \frac{1}{r}$, giving a value of $-\frac{1}{2}$ for the circular orbit and $-\frac{1}{4}$ for the elliptical orbit with eccentricity $\frac{1}{2}$. Inspecting this energy, as shown in Figure 3.7, reveals details about each scheme. Note that for the

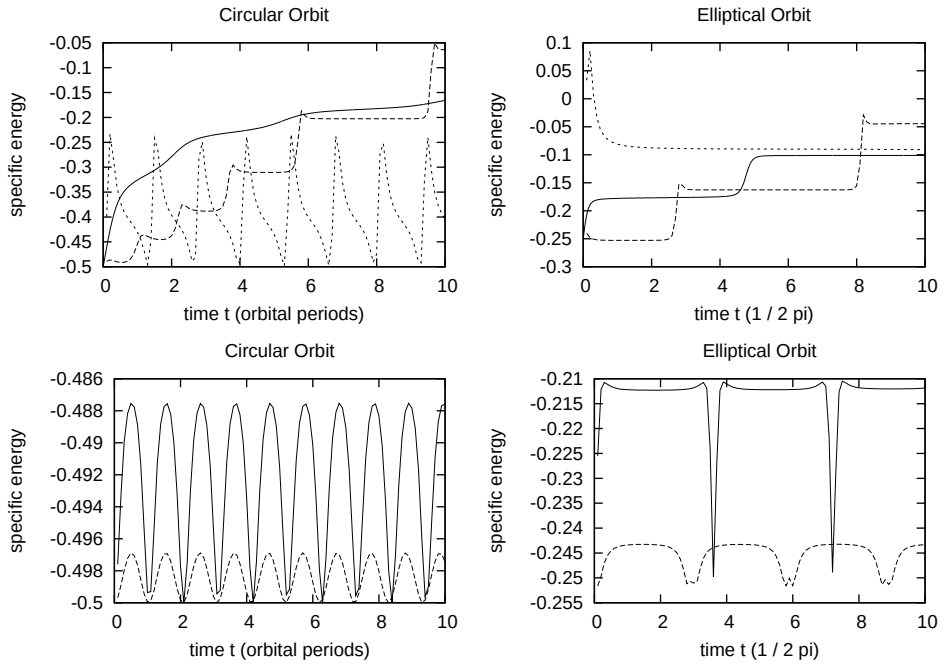


Figure 3.7: Energy conservation of integration schemes. In the top plots the solid lines are the Euler scheme, the dashed lines are the Runge-Kutta scheme and the short-dashed lines are the Leapfrog scheme. In the bottom plots the solid lines are the Predictor-Corrector scheme and the dashed lines are the modified Leapfrog scheme. Numerical solutions using $\Delta t = \frac{2\pi}{10}$ are shown for all schemes, except the Euler scheme which is shown for $\Delta t = \frac{2\pi}{100}$. The circular orbit has specific energy $-\frac{1}{2}$ and the elliptical orbit has specific energy $-\frac{1}{4}$.

Leapfrog scheme the velocity is interpolated between timesteps in order to evaluate the kinetic energy at the same time as the gravitational potential energy.

The Euler scheme is clearly unstable, demonstrating catastrophic growth of energy; this growth is most pronounced at each periastron in the elliptical orbit, since this is the point of maximum velocity.

The Runge-Kutta scheme also demonstrates instability, with a similar energy boost at each periastron leading to the catastrophic results observed in Figure 3.4.

The Leapfrog scheme demonstrates stability, but with significant errors corresponding to the orbital paths observed in Figure 3.3. In the circular orbit the eccentricity appears to oscillates between ~ 0 at perisatron and ~ 0.2 at apastron, with the apastron itself precessing. Such oscillation between two distinct solution types is an unfortunate side-effect of the two-level formulation of the scheme. The elliptical orbit fares even worse, oscillating between different elliptical solutions, with a systematic error probably introduced by the initial extrapolation step, but it is still ultimately stable.

The Predictor-Corrector scheme demonstrates stability, with only small errors. The circular orbit periodically develops slight eccentricity with a precessing apastron. The elliptical orbit develops increased eccentricity at apastron but returns to the correct value at periastron, again with a precessing apastron.

The modified Leapfrog scheme behaves similarly to the Predictor-Corrector scheme, but with errors smaller by a factor of ~ 4.

Although these stable schemes are all time-symmetric, note that they only conserve energy exactly (within the limits of rounding error) at the point of completion of each orbit. During each orbit the energy may vary greatly, as is particularly apparent in the behaviour of the Leapfrog scheme.

Figure 3.8 reveals the behaviour with a small timestep, when all the second-order schemes appeared to resolve the orbital paths correctly.

The marginal instability of the Runge-Kutta scheme is demonstrated, with unbounded growth of energy. This quickly exceeds the small systematic error of the stable Leapfrog scheme for the circular orbit, but for the elliptical orbit the error growth is slower. Such comparative behaviour is to be expected given the analysis of these two schemes in the previous chapter: the Runge-Kutta scheme was shown to be marginally unstable for oscillatory equations, while the Leapfrog scheme was conditionally stable ($\Delta t \leq \frac{2}{\omega}$). A circular orbit is essentially simple harmonic oscillation, and so the worst case scenario for the Runge-Kutta scheme and the best case scenario for the Leapfrog scheme. The elliptical orbit plot indicates that eccentricity effectively damps the error growth for the Runge-Kutta scheme and amplifies the energy deviations for the Leapfrog scheme.

The bottom plots show the Predictor-Corrector and modified Leapfrog schemes approaching the accuracy limit of rounding error for the circular orbit, as suggested by the very small values and the quantised nature of fluctuations. The elliptical orbit plot indicates that eccentricity amplifies the energy deviations for these schemes too.

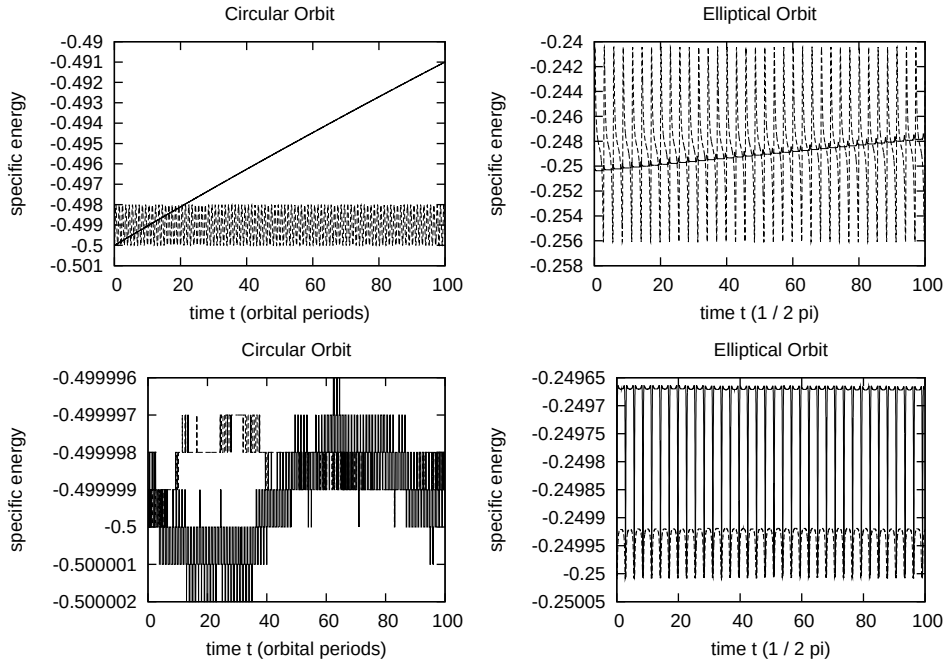


Figure 3.8: Energy conservation of second-order integration schemes using a small timestep. In the top plots the solid lines are the Runge-Kutta scheme and the dashed lines are the Leapfrog scheme. In the bottom plots the solid lines are the Predictor-Corrector scheme and the dashed lines are the modified Leapfrog scheme. Numerical solutions using $\Delta t = \frac{2\pi}{100}$ are shown for all schemes. The circular orbit has specific energy $-\frac{1}{2}$ and the elliptical orbit has specific energy $-\frac{1}{4}$.

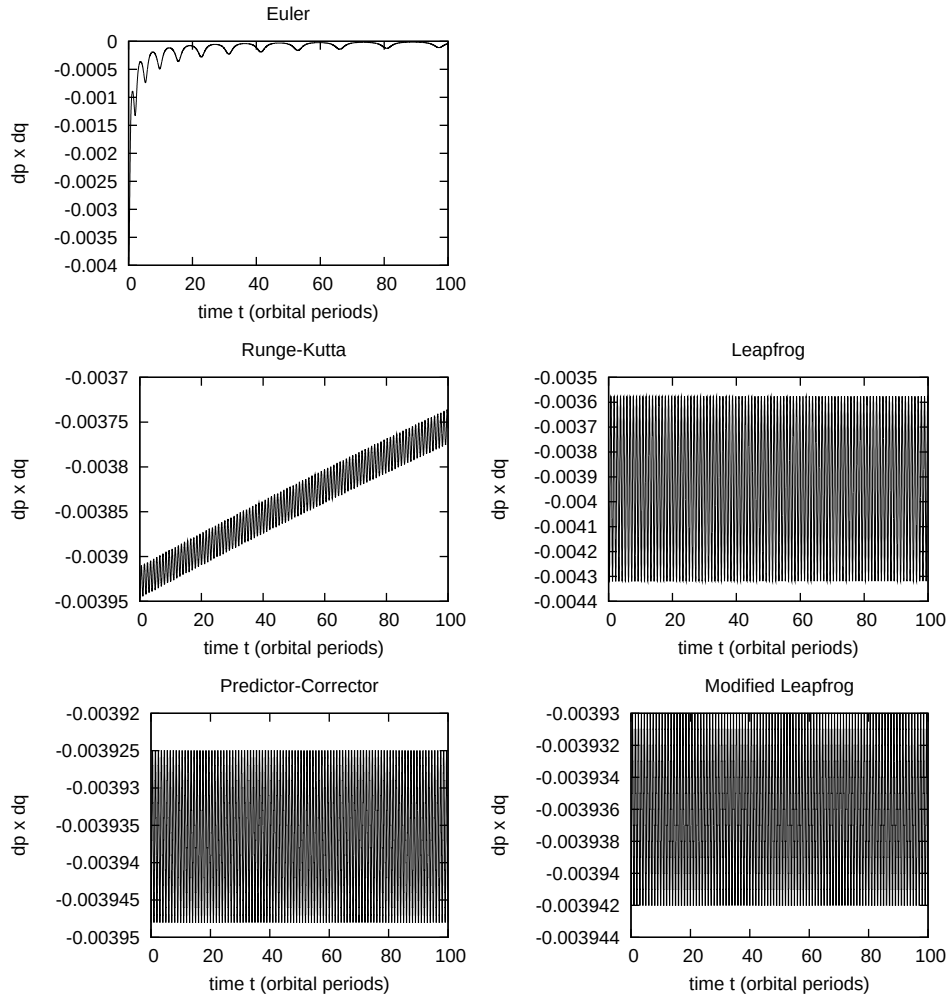


Figure 3.9: Symplectic behaviour of integration schemes. Numerical solutions of the circular orbit using $\Delta t = \frac{2\pi}{100}$ are shown for all schemes, with the z component of $dp \times dq$ plotted.

3.2.9 Symplectic Integration

The Leapfrog schemes and the Predictor-Corrector scheme are found to be symplectic integration schemes, since they conserve a Hamiltonian ($H = T + V$) only slightly perturbed from the original one. This effectively means that they preserve the correct phase space in terms of position q and momentum p as defined by the solution of Hamilton's equations $\frac{dq}{dt} = \frac{\partial H}{\partial p}$ and $\frac{dp}{dt} = -\frac{\partial H}{\partial q}$.

The strict definition of a symplectic integration scheme is one that conserves $dp \times dq$, since that is the conserved quantity in the time evolution of Hamilton's equations due to its symplectomorphism.

Figure 3.9 reveals the behaviour with a small timestep, plotting the time evolution of $dp \times dq$. Since the circular orbit is restricted to the (x, y) plane, only the z component of $dp \times dq$ is non-zero. The Euler and Runge-Kutta schemes reveal

growth instability, showing that they are not symplectic. The Leapfrog, Predictor-Corrector and modified Leapfrog schemes demonstrate deviations of $\sim 10\%$, $\sim 0.4\%$ and $\sim 0.15\%$ respectively, and are symplectic. This makes them well-suited to determining the long-term evolution of chaotic systems.

3.2.10 Assessment

The Euler scheme is catastrophically unstable (it is unconditionally unstable, and rapidly deviates from the correct orbital path even when a small timestep is used, as seen in Figure 3.2), and therefore clearly not suitable for resolving orbital motion.

The Runge-Kutta scheme is relatively inefficient, requiring two acceleration evaluations every timestep (on both the halfstep and fullstep), and also marginally unstable and not symplectic, making it a poor choice.

The Leapfrog scheme is more efficient, requiring only one acceleration evaluation every timestep, relatively stable and symplectic, but its two-level formulation makes it a dubious choice, since it oscillates between two distinct solution types.

The Predictor-Corrector scheme is also efficient, requiring only one acceleration evaluation each timestep, and a reasonable choice.

The modified Leapfrog scheme is equally efficient, requiring only one acceleration evaluation each timestep (on the halfstep but not the fullstep). This midpoint evaluation, consequently allowing the correction step to be applied to position rather than velocity, appears to improve the performance by a factor of ~ 4 with respect to the Predictor-Corrector scheme in terms of energy conservation. This lower deviation in energy and phase space makes it our method of choice. It also has an established pedigree, having been used in GADGET (Springel, Yoshida & White 2001).

3.2.11 Open Orbits

A circular orbit is essentially a simple harmonic oscillation, and an elliptical orbit is closed and therefore also oscillatory. A parabolic or hyperbolic orbit is open, however, and therefore not oscillatory; tending towards infinity, it essentially corresponds to a growing solution. From the analysis in the previous chapter, we recall that none of the integration schemes are capable of producing stable numerical solutions for growth equations.

Figure 3.10 shows how the numerical solutions diverge from the analytic solutions. For the parabolic orbit the Euler scheme performs particularly badly, and the Runge-Kutta scheme slightly better than the other second-order schemes, although all diverge strongly from the true solution, tending towards hyperbolic paths. For the hyperbolic orbit all schemes diverge from the true solution, tending towards excess orbital energy, although the Runge-Kutta scheme performs slightly better than the others.

The better performance of the Runge-Kutta scheme relative to the others implies that the eigenvalues of its error amplification matrix $\mathbb{A}$ have a weaker dependence on the timestep Δt in the asymptotic case (where position $r \to \infty$, velocity $v \to$ constant and acceleration $a \to 0$), through the influence of the partial derivatives of the acceleration with respect to position and velocity as they appear in the

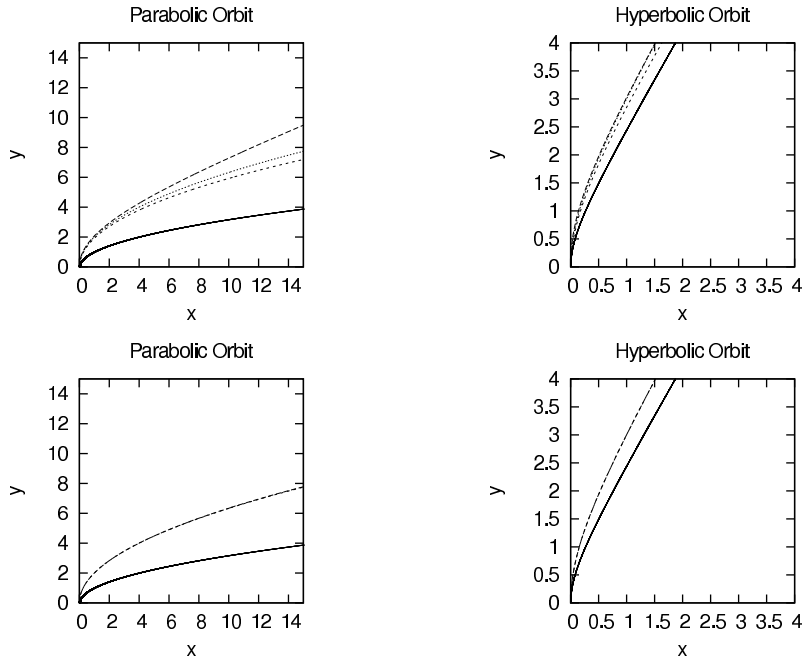


Figure 3.10: Normalised demonstration of the stability properties of integration schemes for open orbits. The solid lines are analytic solutions and the dashed lines are numerical solutions with $\Delta t = \frac{2\pi}{100}$. The hyperbola has an effective eccentricity of 2. In the top plots the dashed lines are the Euler scheme, the short-dashed lines are the Leapfrog scheme and the dotted lines are the Runge-Kutta scheme. In the bottom plots the dashed lines are the Predictor-Corrector scheme and the short-dashed lines are the modified Leapfrog scheme.

components of $\mathbb{A}$.

Regardless of this, no numerical method handles growth solutions well. Open orbits and other problems involving expansion out towards infinity should therefore be avoided if possible. Fortunately most astrophysical scenarios involve closed orbits and collapse towards singularity, which correspond to oscillatory and decay solutions, and these are within the capabilities of numerical methods.

3.2.12 Stability Condition

The Leapfrog scheme is conditionally stable if $\Delta t \leq \frac{2}{\omega}$. Recall, however, that as the neutrally stable value is approached the error amplitudes periodically fluctuate, which in turn affects the stability condition. Accounting for this, and aiming to be conservative enough to resolve ~ 1000 orbits, a condition $\Delta t \lesssim \frac{0.68}{\omega}$ will suffice for a circular orbit. Since $\omega = \frac{2\pi}{T}$, this is equivalent to requiring that the orbital period T is resolved by at least 10 timesteps: $\Delta t \leq \frac{T}{10}$. This forms a useful basic stability condition for any scheme attempting to resolve orbital motion, and with the modified Leapfrog scheme will tend to keep orbital migration within $\sim 10\%$. Better is $\Delta t \leq \frac{T}{100}$ which, due to the second-order accuracy of the modified Leapfrog scheme, will tend to keep orbital migration within $\sim 0.1\%$, an improvement of two orders of magnitude.

3.2.13 The N-Body Problem

Gravitation can be solved analytically for 2 bodies using the orbit equation, because it can be reduced to a single mutual interaction. With 3, 4, 5 or more bodies this pure analytic approach is no longer possible, since each body interacts with every other body and the system of equations becomes non-linear, with 3, 6, 10 or more mutual interactions, scaling as $\frac{N^2-N}{2}$ mutual interactions for N bodies.

A numerical method is necessary to solve systems of N bodies when $N > 2$, and such N-body methods are commonly used to model the dynamics of stellar clusters, where gravity is the only significant force. It is this force calculation, scaling as $\frac{N^2-N}{2}$, that provides the main computational difficulty. It is interesting to note that some early N-body calculations were performed using lightbulbs to represent stellar masses (Holmberg 1941), with light intensity from all lightbulbs in the system measured using a photocell as a proxy for the gravitational force, since the intensity falls off geometrically with the same $\frac{1}{r^2}$ scaling.

The advent of computers has made force calculation much easier, but it still remains computationally expensive. The N^2 scaling means that doubling the number of particles quadruples the time taken to calculate forces. Even with the dramatic advancement of processor speeds, two orders of magnitude in speed are required to match each order of magnitude increase in particle number.

3.3 Developing an N-Body Code

N-body methods (Aarseth 2003) are in many respects the forerunners of all particle-based numerical methods, including Smoothed Particle Hydrodynamics. An N-body

computer code can therefore form the foundation of any other particle-based code, and is a useful preliminary stage in developing a more sophisticated code.

All the numerical solutions shown so far can be obtained with relatively trivial computer programs, but developing a computer program to model a self-gravitating fluid at high resolution is far from trivial.

A fully developed numerical code can be a large and complex entity, tens of thousands or even hundreds of thousands of lines in length. Fortunately it is possible for development to be made incrementally, in a modular fashion, using a top-down approach.

Given the likelihood that any non-trivial program contains at least one bug, it is worth remembering Kernighan's Law: debugging is twice as hard as programming, so it follows that if you write the cleverest possible program then you are, by definition, not clever enough to debug it.

When developing a complex computer program, it is therefore sensible to proceed incrementally, allowing for frequent testing and debugging. The first development stage of our particle-based code will be to ensure that it can solve an N-body gravitational problem correctly.

Before we begin to consider the structure of the program itself, we will briefly consider some fundamental issues: the choice of programming language, the structures used to store particle data, and the use of compiler flags. We will then begin development of the main program, considering some key elements (model parameters, input/output format, memory allocation and dimensionless code units) before implementing the integration scheme, the gravitational calculation and adaptive timesteps.

3.3.1 Programming Languages

Using a computer to apply a numerical method requires it to be supplied with suitable instructions. These must be written in an appropriate programming language.

There are many programming languages, but they fall into two broad categories: interpreted languages and compiled languages.

Programs written in interpreted languages are run on-the-fly, line by line, by an interpreter program. BASIC, Perl, Python and scripting languages such as BASH are all interpreted languages. Their main advantages are ease of use (being able to run immediately after being written) and interactivity potential (with the user able to manipulate the program at run-time). The cost of this is that processing the instructions line by line is a much slower approach than compiling instructions.

Programs written in compiled languages are converted into machine-code instructions before use. Fortran, C and C++ are all compiled languages. The additional step of compiling source code into machine code before being able to use it adds a level of complication and removes a level of interactivity, but allows for considerable gain in performance by transforming the human-readable source-code instructions into a very efficient computer-processable machine-code form.

For this performance reason, using a compiled language is the method of choice. Which one to use is ultimately a matter of familiarity and taste.

Fortran has a long tradition of use within the scientific community, and a good track record when it comes to floating-point operations, compiling to efficient

machine code. Its Fortran 90 incarnation allows dynamic memory allocation, and memory management is high-level (easy to use, being controlled by the compiler more than the programmer). It is somewhat old-fashioned and protective in nature, typically trading programming flexibility for stability and safety, making it clunky but robust. Its extensive library of intrinsic mathematical functions and conformable array operations help to compensate for its otherwise conservative nature.

C is less protective in its use, giving a finer degree of control over memory management and more flexibility (allowing the use of pointers), albeit with the risk of more dangerous programming errors. Its lower-level, more general-purpose approach means it can be more time-consuming to program than Fortran, since more aspects are ultimately left under the control of the programmer.

C++ is a superset of C, stricter in its checking for errors (and thus more strongly enforcing good programming practice), and is designed to facilitate an object-oriented approach. While initially taking longer to learn than C, its new features typically allow programs to be developed more quickly through the application of modern programming concepts. Variables can be declared on-the-fly rather than at the start of a program. Care must be taken in using object-oriented programming, since although it can make things easier from a conceptual (and developmental) standpoint, it may have a performance cost at run-time, specifically the overhead costs of the virtual function tables when using polymorphism (Eckel 2000). Fortunately it follows a strict design philosophy in this regard: you only pay for what you use.

Procedural programming focuses on methods rather than objects, and is based upon lists of instructions, but has a much greater risk of developing tangled and messy code. Such "spaghetti" code is difficult to read and debug; it is worth remembering that code will be read more often than it is written.

Structured programming involves dividing the instructions into subroutines: methods or functions which are called by the main program or by other routines. Each such procedure called will typically receive input in the form of arguments and deliver output in the form of return values. In the ideal case, each subroutine should do one thing, and do it well. This structured approach becomes much more manageable than simple sequential programming, especially as a task becomes larger. Modular programs are much easier to maintain and develop than monolithic ones.

Further advice on programming style can be found in "Writing Scientific Software: A Guide to Good Style" (Oliveira & Stewart 2006).

3.3.2 Data Structures

Particle data may be stored in individual arrays (for masses, positions and velocities) or grouped together. Fortran types, C structures or C++ classes are an option for grouping together certain particle properties. A similar outcome can be achieved by using a two-dimensional array to store several particle properties together: one index denotes the particle number while the other denotes the type of property (mass, position or velocity). Since a loop over properties will typically be nested inside a loop over particles, the optimal index order should be chosen: property then particle number for Fortran (column-major order), or particle number then property for C (row-major order).

Data should be stored in an optimal order in memory because gravitational force calculation is an expensive area of computation. Grouping the properties involved in such calculations contiguously in memory (positions, masses and also gravitational softening lengths, described later) is considered to be the most efficient arrangement.

Global variables, considered dangerous from a programming perspective (since they cannot be protected from accidental manipulation), are nonetheless a reasonable choice for particle data, since the majority of subroutines will require access to this information, but appropriate care should be taken.

3.3.3 Compiler Flags

Since subroutines are favoured over monolithic code it is sensible to write each such subroutine in its own file. This makes the compilation command very long-winded, but the process can be simplified by using a Makefile to allow the GNU `make` utility to compile the program. This Makefile is actually written in an interpreted programming language, similar to a shell script, and allows for conditional compilation through the use of variables referred to as "compiler flags". By setting different compiler flags, different sections of code may be compiled, allowing the implementation to be generalised at the compilation-time level without sacrificing performance at run-time level.

Consider generalising the number of spatial dimensions, for example, so that it is not "hard-wired" to one value but can be set by the user to a value of 1, 2 or 3 as befits the particular problem. This could be achieved by using an integer variable of the programming language which is referenced whenever the number of dimensions needs to be known, such as when referencing array elements for positions or velocities. Unfortunately this referencing operation incurs a small performance cost; while admittedly very small, it is at great risk of being compounded, particularly within loops of instructions. Fortunately this can be avoided by instead using a substitution variable from the Makefile at the compilation level. By setting a compiler flag for the number of dimensions in the Makefile, it can then be used as a "macro", with its value being substituted into the source code wherever the number of dimensions needs to be known *before* compilation occurs. This provides the flexibility of a generalised approach without any run-time performance cost. The cost is instead paid at compilation-time: if the value needs to be changed, the entire program must be recompiled with the new value before being run, whereas a variable of the programming language could be manipulated at run-time. Given that run-times are much greater than compilation-times (except for the most trivial problems), a macro should clearly be chosen over a variable.

The issues to consider when deciding between a macro and a variable are that a macro cannot be changed at run-time (so only constants need be considered) and that a macro is never type-checked (which might incur a greater risk of errors). If a value is constant and occurs frequently in the program, such as inside a loop of instructions, then a macro should be considered. If a value is not constant then a macro cannot be used, and if it does not occur frequently then any small performance gain is probably outweighed by the benefits of clarity and type-checking that would

come with the use of a standard variable.

3.3.4 The Main Program

Using a top-down approach, our program essentially consists of three main parts: input, integration and output.

Pseudocode (main):

```
input (supply initial state of system and prepare for integration)
REPEAT
      integration (numerically integrate system in time)
      output (current state of system)
UNTIL some end condition is satisfied
```

In the initial input part, we supply any model parameters and the initial conditions (initial state of the system at time $t = 0$), and perform other preparations such as allocating memory for data stuctures, converting to code units and initialising variables (such as setting time $t = 0$).

In the integration part, which is inside the main program loop and forms the core processing element, we perform the calculation itself through the application of our numerical method in algorithmic form.

In the output part, also inside the main program loop, we deliver the results of the calculation by writing the state of the system to an output file. This could be done at every integration step, but since an excessive amount of data might be produced it is best done conditionally, typically at some periodic interval.

The main program loop continues until some end condition is satisfied, typically when the integration time t exceeds some pre-defined end point t_{end}.

3.3.5 Parameters

The model parameters could be read in from a file, and might include:

- A run identifier, used to generate output filenames;
- A filename for the file containing the initial conditions;
- An end time t_{end} for the integration (providing the condition to end the main program loop);
- An output time, used to determine the periodic interval at which output is made;
- A backup time, used to determine how often output is made for backup purposes, in the event of program failure (typically specified as an interval in terms of integration steps rather than a physical timescale);
- A timestep multiplier, used to modify any timestep condition;

- Scaling unit preferences, used to specify the physical units used for scales of mass, position, velocity, acceleration, time, energy, momentum and so forth, and also scaling factors for position and mass.

3.3.6 Input/Output Format

Initial conditions and output files should specify the state of the system. This requires a record of the mass, position and velocity of each particle, in appropriate physical units as specified by the model parameters, together with the corresponding time t of the state.

Additional quantities, which are not required for input as initial conditions (since they are calculated as functions of the state variables) but might be useful to include in output for diagnostic purposes, include acceleration, energy and momentum.

For large N scenarios, file sizes can become very large: the state of the system requires $N \times (2D + 1)$ variables in D dimensions (1 for mass, D for position and D for velocity). Significant compression might be possible, particularly if particles have the same mass (effectively reducing the state to $N \times 2D$ variables); favouring single over double precision halves the data size, and storage in binary form (machine representation) rather than as formatted output (human readable numbers which, being strings of characters, can easily increase the data size by a factor of $\sim 2 - 3$) also minimises the data size.

Including acceleration, energy and momentum in the output would significantly increase the data size. For this reason, their inclusion is best considered as a compiler-flagged option. Since energy and momentum could be treated not simply as particle properties but instead summed to provide system-wide totals, it is recommended that they be incorporated in this form, providing useful information at negligible storage cost.

Also note that while binary data saves storage space, a conversion program to translate them into human readable numbers (formatted in rows and columns) is advisable, since this allows the data to be plotted quickly and easily with third-party software such as GNUPLOT. The ability to switch the output between unformatted binary data (to save space) and formatted data (for ease of use) is recommended as a compiler-flagged option.

Since memory allocation requires knowledge of the number of particles in the system, this should be stored at the head of the input file.

In summary, a data file should include a header detailing:

- The number of particles in the system;
- The time t corresponding to the system state;
- Optionally the corresponding integration step n;
- Optional diagnostic information on the energy and momentum of the system.

This is followed by a record of the state of the system:

- The mass of each particle;
- The position of each particle;
- The velocity of each particle;
- Optionally the acceleration of each particle;
- Optional information on the energy and momentum of each particle.

3.3.7 Memory Allocation

Memory for the data structures to store particle properties (including mass, position, velocity and acceleration) can only be allocated once the number of particles is known. Any memory that has been allocated should always be deallocated when it is no longer needed, to prevent "memory leaks". In the case of particle data stored in global variables and in use until the very end of the program, it should automatically be deallocated when the program ends but it is good practice to explicitly deallocate it using a final "clean up" subroutine. Memory leaks are one of the most insidious programming bugs, and the C language is particularly susceptible to them, so it is recommended that memory management is enforced rigidly.

3.3.8 Code Units

Input and output should be made in physical units, as specified by the scaling unit preferences in the model parameters. Unfortunately these units often result in very large or small values being used in computations, which enhances numerical errors and risks the overflow or underflow of the machine representation of numbers. Numerical errors can be minimised by keeping computational values close to unity, and this is achieved by normalising them into dimensionless units referred to as code units.

Conversion between physical units (used in input and output) and code units (used within the program) must therefore be performed at the input/output interfaces.

This first requires a specification of the physical unit for each quantity X used: mass in solar masses, position in astronomical units, velocity in $\mathrm{km\,s^{-1}}$ and time in years, for example. Units for acceleration, energy and momentum may also be needed for output purposes.

Each such physical unit can be concisely described using the numeric value X_{SI} corresponding to its equivalent in SI units: a solar mass is $m_{SI} = 1.98892 \times 10^{30}\,\mathrm{kg}$, an astronomical unit is $r_{SI} = 1.49598 \times 10^{11}\,\mathrm{m}$, $1\,\mathrm{km\,s^{-1}}$ is $v_{SI} = 10^{3}\,\mathrm{m\,s^{-1}}$ and a year is $t_{SI} = 3.1556952 \times 10^{7}\,\mathrm{s}$, for example.

To normalise values we require a scaling factor for each unit. One code unit will then correspond to X_{scale} physical units, so physical units may be normalised into code units by dividing by X_{scale} upon input, and code units may be converted into physical units by multiplying by X_{scale} upon output.

In order to determine X_{scale} for each unit, we first specify the scaling factors for position and mass as model parameters, supplying them as r_{scale} and m_{scale}. This

allows us to derive all the other scaling factors X_{scale} for each physical quantity X using dimensional analysis.

In order to relate units of time t to length r and mass m, consider the law of gravitation in dimensional terms: $rt^{-2} = \frac{Gm}{r^2}$, where $G = 6.673 \times 10^{-11}\,\mathrm{m^3\,kg^{-1}\,s^{-2}}$ is the gravitational constant in SI units. Rearranging this provides a scaling factor for time:

$$t_{scale} \times t_{SI} = \sqrt{\frac{(r_{scale} \times r_{SI})^3}{G \times m_{scale} \times m_{SI}}} \tag{3.45}$$

The inclusion of G in this relationship effectively normalises the units of time in the code such that $G = 1$. This is a very desirable arrangement because a value of unity for the gravitational constant simplifies the gravitational calculation (eliminating one multiplication operation), which greatly improves the efficiency of the code.

Scaling factors for velocity v, acceleration a, linear momentum p, angular momentum L and energy E can be derived very simply:

$$v_{scale} \times v_{SI} = \frac{r_{scale} \times r_{SI}}{t_{scale} \times t_{SI}} \tag{3.46}$$

$$a_{scale} \times a_{SI} = \frac{v_{scale} \times v_{SI}}{t_{scale} \times t_{SI}} \tag{3.47}$$

$$p_{scale} \times p_{SI} = m_{scale} \times m_{SI} \times v_{scale} \times v_{SI} \tag{3.48}$$

$$L_{scale} \times L_{SI} = m_{scale} \times m_{SI} \times v_{scale} \times v_{SI} \times r_{scale} \times r_{SI} \tag{3.49}$$

$$E_{scale} \times E_{SI} = m_{scale} \times m_{SI} \times (v_{scale} \times v_{SI})^2 \tag{3.50}$$

All physical quantities must be normalised upon input (divided by X_{scale}), so the initial conditions will require mass, position and velocity normalisation, and model parameters specifying the end time and output interval will require time normalisation. Conversely all such physical quantities must be converted back to physical units upon output (multiplied by X_{scale}).

3.3.9 Integration

The integration routine within the main program loop contains the core processing elements.

Pseudocode (integration):

```
calculate total energy and momentum of system [optional diagnostic]
IF adaptive timestep is used THEN
      calculate new timestep Δt
ENDIF
FOR each particle
      apply first part of integration scheme
      (predict position)
ENDFOR
calculate gravitational accelerations (using predicted positions)
FOR each particle
```

```
      apply second part of integration scheme
      (advance velocity and correct position)
ENDFOR
advance time:  t = t + Δt
```

This description applies to the modified Leapfrog scheme, and requires additional data storage for the predicted positions. Similarly the Runge-Kutta scheme requires additional storage for the halfstep positions and velocities, and the Predictor-Corrector scheme for the previous accelerations. The Leapfrog scheme requires no additional storage, nor does the Euler scheme.

Using different integration schemes therefore requires some modifications to the implementation in each case. In particular, the Euler and Leapfrog schemes can be implemented in one part rather than two, and the Runge-Kutta scheme requires an additional calculation of gravitational accelerations.

3.3.10 Gravitational Calculations

The full gravitational calculation involves a loop over all particles. For each particle the contribution to its gravitational acceleration from every other particle must be evaluated, requiring another loop over all particles nested inside the first. This makes the calculation an N^2 operation. Since a particle has no gravitational influence on itself, the self-contribution in the loop is skipped, resulting in an $N^2 - N$ operation. Since the gravitational interaction is mutual, with each particle exerting an equal and opposite force upon the other, the operation can be reduced to $\frac{N^2-N}{2}$ by accounting for this and applying both contributions at the same time.

Pseudocode (calculate gravitational accelerations):

```
FOR each particle i = 1 to N
      initialise acceleration to zero:  a_i = 0
ENDFOR i
FOR each particle i = 1 to N
      FOR each subsequent particle j = i + 1 to N
            calculate gravitational acceleration a_ij between i and j
            apply contribution to particle i:  a_i = a_i + a_ij
            apply contribution to particle j:  a_j = a_j − (m_i/m_j) a_ij
      ENDFOR j
ENDFOR i
```

The gravitational acceleration experienced by particle i due to particle j, with G normalised to unity, is

$$\mathbf{a}_{ij} = -m_j \frac{\mathbf{r}_{ij}}{|\mathbf{r}_{ij}|^3} \tag{3.51}$$

where m_j is the mass of particle j and $\mathbf{r}_{ij} = \mathbf{r}_i - \mathbf{r}_j$ is the relative position vector of particle i with respect to particle j. For particle j,

$$\mathbf{a}_{ji} = m_i \frac{\mathbf{r}_{ij}}{|\mathbf{r}_{ij}|^3} \tag{3.52}$$

3.3.11 Adaptive Timesteps

In choosing the timestep, the classic computational trade-off between accuracy and speed comes into play. The timestep must be small enough to provide a stable and accurate solution, but preferably no smaller since that would slow the computation unnecessarily.

By using an adaptive timestep it can be adjusted at each integration step and set to the value most appropriate to the state of the system at that point in time.

The downside is that this breaks the time symmetry of the integration scheme, which leads to energy no longer being strictly conserved. This unfortunate side effect becomes acceptable for collapse problems, where the choice must be made between accurate fixed-timestep simulations and slightly inaccurate adaptive timestep simulations which have run-times that are orders of magnitude faster; the dramatic gain in speed makes the price worth paying.

To achieve convergence with the analytic solution for orbital motion, we previously found that the condition $\Delta t \leq \frac{T}{100}$ maintained accuracy within $\sim 0.1\%$. For the circular orbit a fixed timestep is feasible; for the general case of potentially chaotic N-body motion, however, a fixed timestep is inappropriate since it will at times be too large and at other times too small.

In formulating a condition for an adaptive timestep it seems sensible to begin with circular motion, which obeys the relationship $a = \frac{v^2}{r}$. The orbital velocity may be expressed as $v = \frac{2\pi r}{T}$, and substituting this and rearranging gives $T = 2\pi\sqrt{\frac{r}{a}}$. This provides a general timestep condition $\Delta t = \frac{2\pi}{100}\sqrt{\frac{r}{a}}$. Applying this condition to each mutual gravitational interaction between particles i and j gives

$$\Delta t = \frac{2\pi}{100}\sqrt{\frac{|\mathbf{r}_{ij}|}{|\mathbf{a}_{ij}|}} \tag{3.53}$$

This conservative condition assumes that each particle might be in circular orbit around any other particle, and gives a timestep sufficiently small to resolve such an orbit. For more complicated trajectories it essentially scales as $|\mathbf{r}_{ij}|^{3/2}$, which provides the desired asymptotic behaviour: strongly decreasing the timestep for close encounters between bodies as $|\mathbf{r}_{ij}| \to 0$, and likewise increasing it for weak interaction as $|\mathbf{r}_{ij}| \to \infty$.

Taking a similar approach using the freefall time $t_{ff} = \frac{\pi}{2}\left(\frac{2GM}{r^3}\right)^{-1/2}$, and substituting $|a| = \frac{GM}{r^2}$, leads to the condition $\Delta t = \frac{\pi}{2\sqrt{2}}\sqrt{\frac{|r|}{|a|}}$. Therefore the previous "orbital" condition also provides sufficient time resolution in the case where two bodies are approaching one another in gravitational freefall, due to the conservative application of the factor of 100 in the denominator.

This freefall scenario reveals another timestep condition, provided by the "collision" time $\Delta t = \frac{|\mathbf{r}_{ij}|}{|\mathbf{v}_{ij}|}$, which is the time taken for any two freely drifting bodies to collide with each other. This expression essentially corresponds to the Courant-Friedrichs-Lewy condition. It is usually applied more conservatively as $\Delta t = C\frac{|\mathbf{r}_{ij}|}{|\mathbf{v}_{ij}|}$, with $C = 0.25$ or 0.3 being typical.

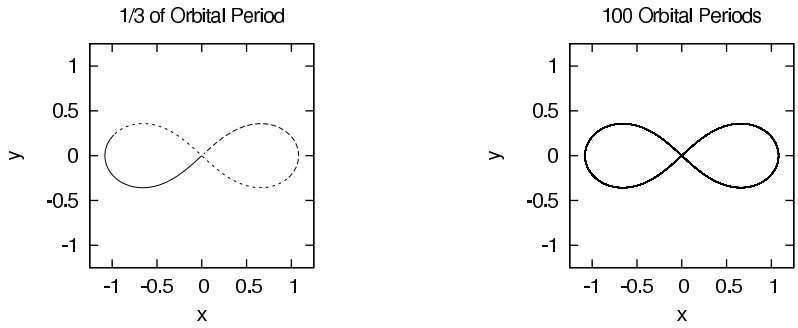


Figure 3.11: The "figure of eight" configuration for three bodies. The left plot shows the tracks of each particle after one third of an orbital period, when they have changed places. The right plot shows the tracks over 100 orbits, demonstrating the stability of the adaptive timestep.

3.4 The Three-Body Problem

The first test to perform after writing an N-body code is to verify that it solves a two-body problem correctly, such as a circular orbit. Once this has been achieved, more stringent tests can be applied.

The dynamical motion of N-body systems with $N > 2$ is generally non-linear, but stable configurations exist such as the three-body "figure of eight" configuration (Chenciner & Montgomery 2000). In dimensionless units each particle has a mass of 1 and the initial positions and velocities are given in Table 3.1.

Particle	1	2	3
x	0.0	0.9700436	-0.9700436
y	0.0	-0.24308753	0.24308753
v_x	-0.93240737	0.466203685	0.466203685
v_y	-0.86473146	0.43236573	0.43236573

Table 3.1: Initial conditions for the three-body "figure of eight" configuration.

The particles move along a cyclic "figure of eight" orbit with period $T \sim 6.3264$. The orbit is stable to perturbations within a fraction of a percent, so this provides a good test for the adaptive timestep. Each particle is displaced along the orbit by a third of a period, so after $t \sim 2.1088$ the particles will have changed places.

Figure 3.11 shows the numerical solutions, verifying that the adaptive timestep is capable of resolving complicated orbital motion. Due to the breach of time symmetry caused by the adaptive timestep, and the gradual accumulation of phase errors in the modified Leapfrog scheme, the system loses energy at a rate of $\sim 10^{-3}\%$ per orbit. This loss will therefore start to become appreciable after ~ 1000 orbits, making the validity of further orbital evolution questionable. Conversely the marginally unstable Runge-Kutta scheme gains energy, and at a rate more than an order of magnitude faster.

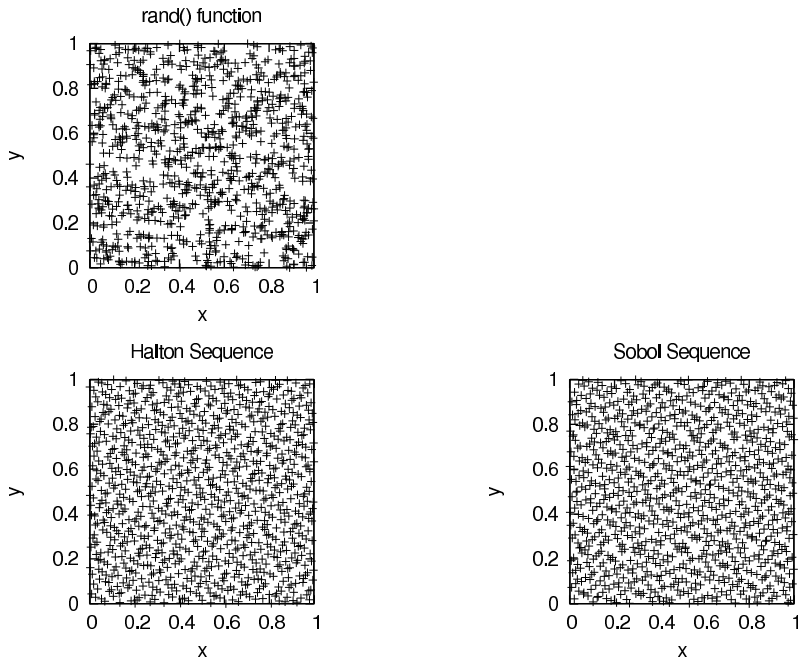


Figure 3.12: Demonstration of noise and space-filling behaviour in random number generation. The upper-left plot shows 1000 points placed using the pseudo-random `rand()` function from the C math library. The lower-left plot shows 1000 points placed using the quasi-random Halton sequence at base 2 in x and base 3 in y. The right plot shows 1000 points placed using the quasi-random Sobol sequence.

3.5 Freefall Collapse

In Chapter 1 we derived the analytic solution for the collapse of a uniform density sphere of mass M and initial radius R under gravitational freefall,

$$\theta + \cos(\theta)(1 - \cos^2(\theta))^{1/2} = \left(\frac{2GM}{R^3}\right)^{1/2} t \tag{3.54}$$

where $\cos^2(\theta) = \frac{r(t)}{R}$ gives the evolution of the radius as a function of time. The time taken to collapse to a singularity through the action of self-gravity is the freefall time $t_{ff} = \frac{\pi}{2}\left(\frac{2GM}{R^3}\right)^{-1/2}$, so for a sphere of unit mass and unit initial radius with G normalised to 1, $t_{ff} = \frac{\pi}{2\sqrt{2}} \simeq 1.111$.

This uniform density sphere can be modelled by placing N particles, each of mass $\frac{1}{N}$ and with no initial velocity, inside a spherical volume of unit radius. Placing the particles at random will approximate uniform density; there is, however, the risk of two particles having a very small initial separation, which consequently enforces a very small timestep, slowing the computation. Due to the noisy behaviour of a typical pseudo-random number generator on a computer, as shown in the top-left plot of Figure 3.12, this effectively cripples the computation for large N. A space-filling number generator reduces this problem, as shown in the bottom plots. This essentially produces an irregular lattice; such self-avoiding sequences are not actually random, but are often referred to as quasi-random or sub-random number generators.

For the Halton sequence, each dimension is a Van der Corput sequence with a different prime number used as its base; Table 3.2 shows the sequence with base 2. The Sobol sequence has slightly better behaviour, but is more complicated in its implementation (Press et al. 1992).

Term in Sequence	In Base 2 (Binary)	Reversed and written as a fraction	In Base 10 (Decimal)
1	1	0.1	0.5
2	10	0.01	0.25
3	11	0.11	0.75
4	100	0.001	0.125
5	101	0.101	0.625
6	110	0.011	0.375
7	111	0.111	0.875
8	1000	0.0001	0.0625
9	1001	0.1001	0.5625
10	1010	0.0101	0.3125

Table 3.2: The Van der Corput sequence with base 2.

Even with particles spaced apart in the initial conditions, however, a collapse problem inevitably results in the timestep decreasing as all particles approach each other. Besides this timestep problem, there is a more general problem in computing gravitational accelerations, because as $\mathbf{r}_{ij} \to 0$ when approaching singularity, $\mathbf{a}_{ij} \to \infty$. A computer can never represent this asymptotic behaviour correctly due to its discrete number representation, so any close encounters always risk large numerical errors.

3.5.1 Gravitational Softening

One way to address this problem is to soften the gravitational interactions through the introduction of a softening length ϵ. The softened gravitational acceleration experienced by particle i due to particle j, with G normalised to unity, is given by

$$\mathbf{a}_{ij} = -m_j \frac{\mathbf{r}_{ij}}{(|\mathbf{r}_{ij}|^2 + \epsilon^2)^{3/2}} \tag{3.55}$$

When $\mathbf{r}_{ij} \gg \epsilon$ the gravitational acceleration behaves as before, but as $\mathbf{r}_{ij} \to 0$ the denominator can never be smaller than ϵ^3 and so $\mathbf{a}_{ij} \to 0$ not ∞, preventing the singularity problem. Although this circumvents the numerical issues, it does clearly come at the expense of deviation from the analytic form of gravity at small separations $\sim \epsilon$, as shown in Figure 3.13. The minimum occurs at $r = \frac{\epsilon}{\sqrt{2}}$, where $a = -\frac{2}{3\sqrt{3}}\frac{1}{\epsilon^3}$. A physical interpretation of this softened gravitational interaction can be made by considering that a particle is no longer a point-mass but effectively has a radial extent of order a few ϵ, with the mass distributed within this according to some density profile that gives the resulting softened gravitational form shown.

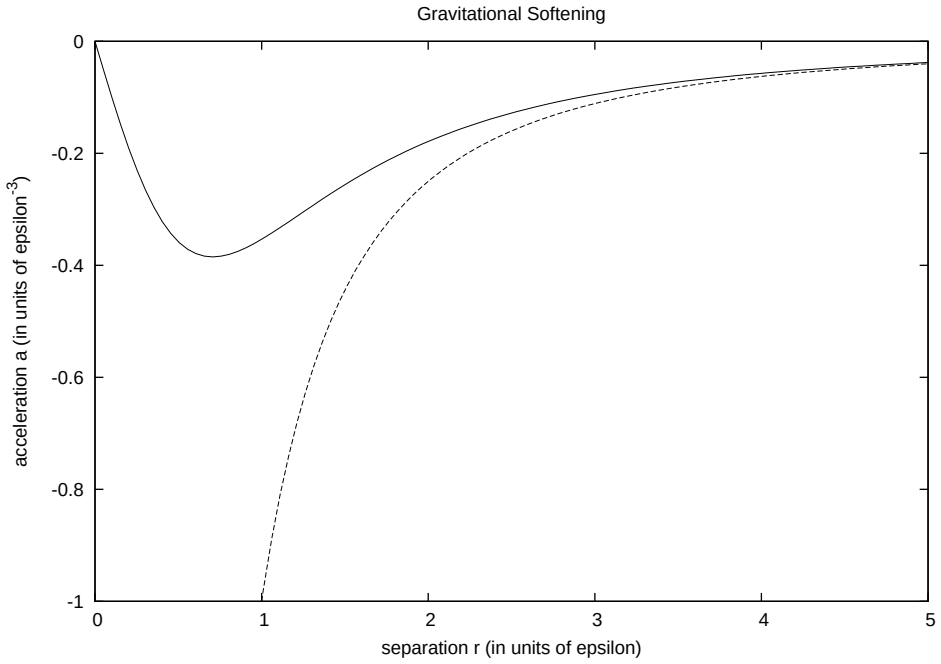


Figure 3.13: Normalised demonstration of gravitational softening, with the analytic form (inverse square law) shown by the dashed line.

The adaptive timestep Δt no longer scales as $|\mathbf{r}_{ij}^{3/2}|$ but as $(|\mathbf{r}_{ij}|^2 + \epsilon^2)^{3/4}$, which effectively limits the minimum timestep to $\sim \frac{2\pi}{100}\epsilon^{3/2}$. This makes large N computations more feasible.

An alternative to gravitational softening is regularisation. The equation of motion for any two-body interaction can be regularised by performing a co-ordinate transformation such that this regularised equation of motion does not become singular at any point, allowing it to be accurately integrated and then transformed back to the original co-ordinate frame (Steifel & Scheifele 1971). Close interactions involving multiple bodies require a higher-order regularisation scheme to be implemented, such as chain regularisation (Mikkola & Aarseth 1993).

Since gravitational softening provides a conceptual foundation for hydrodynamical smoothing, we will neglect the more sophisticated methods of regularisation in favour of this simpler approach.

Figure 3.14 shows the computer modelling of the freefall collapse of a uniform density sphere represented by 1000 gravitationally softened particles placed using the Halton sequence. It appears to be a reasonable approximation to the analytic solution, given the relatively low resolution of 1000 particles, but softening ultimately gives rise to deviations. The delay of collapse towards singularity in the centre, once particles approach within a few ϵ of each other, has a cumulative effect. It is most pronounced for those particles at the surface of the sphere, since the descent of each particle is essentially governed by all those particles interior to it.

Energy is conserved within $\sim 0.01\%$ provided that softening is also accounted for within the calculation of gravitational potential energy.

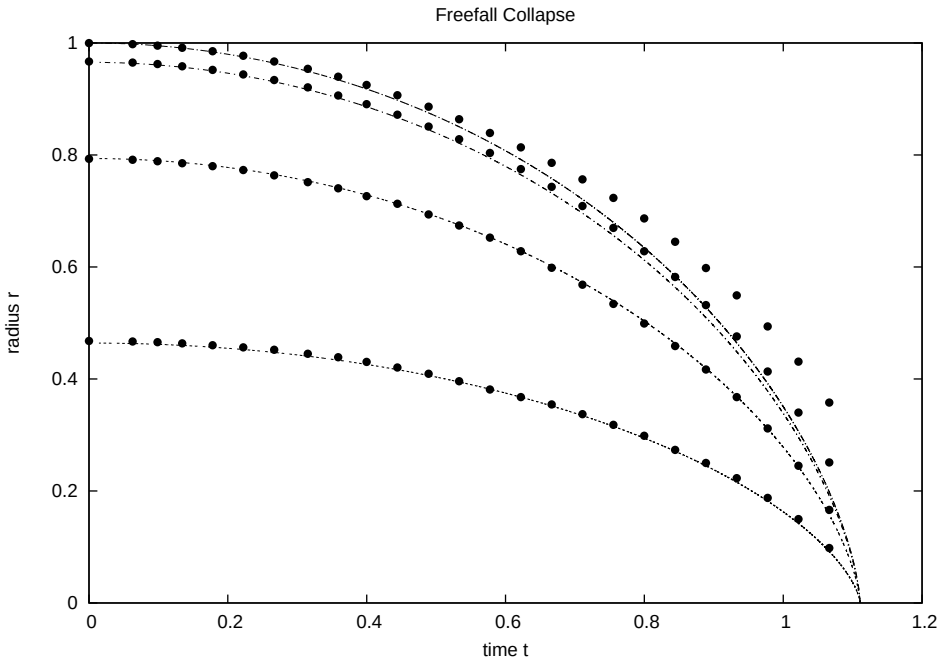


Figure 3.14: Freefall collapse for $N = 1000$ particles with gravitational softening $\epsilon = 10^{-3}$. The lines show the analytic solutions for the sphere radius and the 90%, 50% and 10% mass radii inside it. The points are the numerical solution values at periodic intervals.

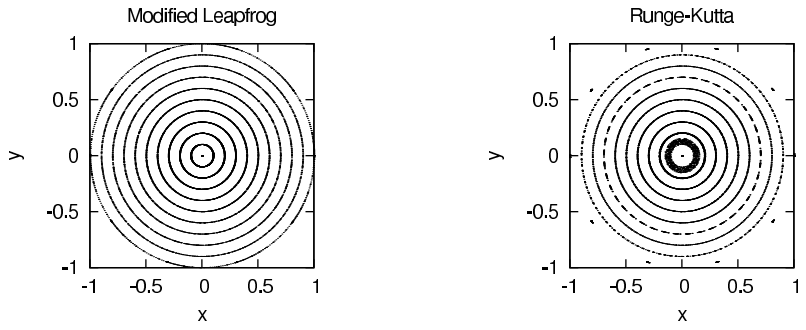


Figure 3.15: System of 10 bodies of zero mass in Keplerian orbit around a stationary body of unit mass. Tracks are plotted over 100 outer orbital periods at periodic intervals (10 times per outer orbital period).

3.6 A Planetary System

Setting 10 bodies with zero mass on circular orbits with tangential velocity $v = \frac{1}{\sqrt{r}}$ around a body of unit mass placed at the origin provides a crude approximation of a planetary system. After evolving the system for 100 orbital periods of the outermost body, the results are shown in Figure 3.15.

The modified Leapfrog scheme demonstrates good orbital stability with the adaptive timestep. The Runge-Kutta scheme, while appearing to accumulate less phase error on the outermost orbit, demonstrates the effects of its marginal instability on the innermost orbit, which has migrated outwards by $\sim 50\%$. The second orbit has suffered migration of $\sim 2.5\%$, while the third orbit has remained reasonably stable. Even after only 10 outer rotation periods, the innermost orbit has already migrated outward by $> 5\%$.

While this trivial system is physically similar to a planetary system around a star, the results have obvious implications for the evolution of circumstellar disks. For a large disk being evolved over many outer rotation periods, the innermost material risks being incorrectly integrated by a Runge-Kutta scheme. Since much of the mass resides in the innermost regions this is clearly a serious issue. Any minor benefits of using a Runge-Kutta scheme to minimise phase errors and to better approximate any open (parabolic or hyperbolic) orbits look certain to be outweighed by the major drawbacks of marginal instability for closed orbits and relative inefficiency in terms of both speed and memory usage (requiring twice as many acceleration calculations and twice as much data storage).

3.7 A Stellar Cluster

When most of the mass in a cloud or core is still molecular gas, hydrodynamics plays an important role and must therefore be modelled (using SPH for example) in order to resolve the critical processes of fragmentation leading to star and planet formation; this remains the focus of our investigation.

Once most of the gas is accreted onto stars (or evaporated away by O stars), however, gravity alone dominates the stellar cluster and its dynamics can be modelled

by an N-body code: stars are treated as point masses and their ballistic evolution is followed.

Due to the highly chaotic nature of most N-body problems, it is vital that an accurate integration scheme is used to correctly evolve a system and achieve convergence. A higher-order method such as the fourth-order Hermite scheme is therefore recommended. Hut & Makino (2007) provide a useful primer on the development of an N-body code to model stellar dynamics; it assisted greatly in guiding the code development within this chapter.

3.8 Summary

The modified Leapfrog scheme has proved to be the method of choice: it is efficient, time-symmetric and symplectic. While it resolves closed orbital motion with the greatest capability, it should be noted that no numerical method copes well with open (parabolic or hyperbolic) orbits.

We have developed an N-body computer program, which has served to highlight some important general issues involved in code development. Adaptive timestepping and gravitational softening have been incorporated into this code.

Now that gravitational dynamics have been satisfactorily modelled we can safely extend the application of the code into the realm of particle-based hydrodynamics.

In the next chapter we will apply particle-based numerical modelling to self-gravitating hydrodynamic problems such as an N-body sphere in isothermal collapse and an N-body sphere of polytropic gas in hydrostatic equilibrium.

"However far the stream flows,
it never forgets its source."

Nigerian proverb

Chapter 4

Hydrodynamics

In this chapter we use numerical methods to solve hydrodynamic problems, which then allows us to complete our aim of modelling self-gravitating gas.

The N-body computer program of the previous chapter is extended to reveal the issues involved in developing a particle-based code for computational fluid dynamics, specifically Smoothed Particle Hydrodynamics (SPH).

By considering simple problems with analytic solutions, such as the isothermal collapse of a uniform-density sphere and a sphere of polytropic gas in hydrostatic equilibrium, the SPH code can be tested to ensure its fidelity.

4.1 Fluid Dynamics

Most astrophysical problems essentially involve gravitation and fluid dynamics. While our simple N-body code is sufficient for modelling systems such as stellar clusters as a collection of point masses, the clouds of gas involved in star and planet formation necessitate the incorporation of computational fluid dynamics.

The foundations of fluid dynamics are the conservation laws:

1. the conservation of mass (the continuity equation);
2. the conservation of linear momentum (Newton's second law of motion);
3. the conservation of energy (first law of thermodynamics).

These laws may be formulated using the Reynolds transport theorem, which essentially states that the rate of change of a physical property X defined over a volume V must equal whatever X is lost (or gained) through the surface S enclosing the volume, plus whatever X is consumed by sinks (or created by sources) within the volume.

$$\frac{d}{dt}\int_V X dV + \int_V X\mathbf{v}\cdot\hat{\mathbf{n}}dS + \int_V \dot{X}_{sink}dV = 0 \tag{4.1}$$

where $\mathbf{v}$ is the fluid velocity (with respect to the stationary surface S), $\hat{\mathbf{n}}$ is the normal unit vector out of the surface S, and $\dot{X}_{sink}$ represents any net sink of X (or source if negative in value).

Applying Leibniz's rule to the first term (bringing the differential operator inside the integral) and using Gauss's divergence theorem to convert the second term into a volume integral allows all the terms to be combined into one volume integral. For this combined integral to be zero for any volume, the integrand itself must be zero.

$$\frac{\partial X}{\partial t} + \nabla \cdot (X\mathbf{v}) + \dot{X}_{sink} = 0 \tag{4.2}$$

4.1.1 The Continuity Equation

To formulate the mass continuity equation we use the mass density ρ of the fluid as our property X and assume no sinks or sources of mass.

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho\mathbf{v}) = 0 \tag{4.3}$$

For the special case of an incompressible fluid, ρ is constant and this equation reduces to $\nabla \cdot \mathbf{v} = 0$ (conservation of volume).

4.1.2 The Equation of Motion

To formulate the momentum equation we use $\rho\mathbf{v}$ as our property X and body forces $\mathbf{F}$ (which act throughout the volume of a body rather than only on the surface) as our sinks or sources of momentum per unit volume, since a force is simply a rate of change of momentum (from Newton's second law of motion).

$$\frac{\partial \rho\mathbf{v}}{\partial t} + \nabla \cdot (\rho\mathbf{v}\mathbf{v}) + \mathbf{F} = 0 \tag{4.4}$$

Expanding the derivatives, then noting that $\mathbf{v} \cdot \nabla\rho + \rho\nabla \cdot \mathbf{v} = \nabla \cdot (\rho\mathbf{v})$ and collecting terms,

$$\mathbf{v}\left(\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho\mathbf{v})\right) + \rho\left(\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla\mathbf{v}\right) + \mathbf{F} = 0 \tag{4.5}$$

The first term is zero due to the continuity equation, and the body forces $\mathbf{F}$ can be split into two terms: forces resulting from stresses, and other forces $\mathbf{f}$, giving the Cauchy momentum equation

$$\rho\left(\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla\mathbf{v}\right) - \nabla \cdot \sigma_{ij} + \mathbf{f} = 0 \tag{4.6}$$

where σ_{ij} is the Cauchy stress tensor,

$$\begin{aligned} \sigma_{ij} &= \begin{pmatrix} \sigma_{xx} & \tau_{xy} & \tau_{xz} \\ \tau_{yx} & \sigma_{yy} & \tau_{yz} \\ \tau_{zx} & \tau_{zy} & \sigma_{zz} \end{pmatrix} \\ &= -\begin{pmatrix} P & 0 & 0 \\ 0 & P & 0 \\ 0 & 0 & P \end{pmatrix} + \begin{pmatrix} \sigma_{xx} + P & \tau_{xy} & \tau_{xz} \\ \tau_{yx} & \sigma_{yy} + P & \tau_{yz} \\ \tau_{zx} & \tau_{zy} & \sigma_{zz} + P \end{pmatrix} \\ &= -P\mathbb{I} + \mathbb{T} \end{aligned} \tag{4.7}$$

where σ are normal stresses and τ are shear stresses, and pressure $P = -\frac{1}{3}(\sigma_{xx} + \sigma_{yy} + \sigma_{zz})$ is, by definition, the mean normal stress. $\mathbb{I}$ is a 3×3 identity matrix, which when multiplied by P gives the mean hydrostatic (or volumetric) stress tensor, describing changes in the volume of the body (normal stress), and $\mathbb{T}$ is a traceless stress deviator tensor, describing distortions of the body (shear stress).

Using this, the momentum equation may now be written as

$$\rho\left(\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v}\cdot\nabla\mathbf{v}\right) + \nabla P - \nabla\cdot\mathbb{T} + \mathbf{f} = 0 \tag{4.8}$$

which is the general form of the Navier-Stokes equation. The stress tensor $\mathbb{T}$ contains too many unknowns for this equation to be of any use yet, so more information must be supplied by considering the viscous behaviour of the fluid.

In the simplest case of a inviscid flow, we arrive at the Euler equation of momentum.

$$\rho\left(\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v}\cdot\nabla\mathbf{v}\right) + \nabla P + \mathbf{f} = 0 \tag{4.9}$$

Including gravity (via the gravitational potential ϕ) as our only body force,

$$\rho\left(\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v}\cdot\nabla\mathbf{v}\right) + \nabla P + \rho\nabla\phi = 0 \tag{4.10}$$

The treatment of viscous flow begins with the assumptions that the fluid is isotropic and that $\nabla\cdot\mathbb{T} = 0$ when the fluid is at rest, so that hydrostatic pressure is experienced. Consideration of a Newtonian fluid (such as water), which has a stress tensor that is a linear function of the strain rate, gives

$$\mathbb{T}_{ij} = \mu\left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i}\right) + \delta_{ij}\lambda\nabla\cdot\mathbf{v} \tag{4.11}$$

where δ_{ij} is the Kronecker delta function, and μ is the (first coefficient of) viscosity (dynamic viscosity). λ is the second coefficient of viscosity, the bulk viscosity associated with changes in volume. For incompressible flows $\lambda = 0$, and for compressible flows the most common approximation is $\lambda \approx -\frac{2}{3}\mu$ (Batchelor 1967). This gives the Navier-Stokes equation for a Newtonian fluid as

$$\rho\left(\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v}\cdot\nabla\mathbf{v}\right) + \nabla P - \nabla\cdot\left(\left[\mu\cdot\left(\nabla\mathbf{v} + (\nabla\mathbf{v})^T\right)\right] + \nabla\left(\lambda\nabla\cdot\mathbf{v}\right)\right) + \mathbf{f} = 0 \tag{4.12}$$

where $()^T$ indicates the transpose.

4.1.3 The Energy Equation

To formulate the energy equation we use ρu as our property X, where u is the specific internal energy, and the rate of heat transfer out of the system Q_{loss} and work W done by the system as our sinks of energy per unit volume, in accordance with the first law of thermodynamics.

$$\frac{\partial \rho u}{\partial t} + \nabla\cdot(\rho u\mathbf{v}) + \frac{\partial}{\partial t}\left(Q_{loss} + W\right) = 0 \tag{4.13}$$

Expanding the derivatives in the same manner as for the equation of motion, we arrive at

$$\rho\left(\frac{\partial u}{\partial t}+\mathbf{v}\cdot\nabla u\right)+\frac{\partial}{\partial t}\left(Q_{loss}+W\right)=0 \tag{4.14}$$

The fluid element does mechanical work W by expanding, where the product of its force and displacement is given by the product of its pressure P and change in volume δV. Since the rate of change of volume $\frac{\partial V}{\partial t}=\nabla\cdot\mathbf{v}$ by Gauss's divergence theorem, we have as our energy equation

$$\rho\left(\frac{\partial u}{\partial t}+\mathbf{v}\cdot\nabla u\right)+P\nabla\cdot\mathbf{v}+\frac{\partial Q_{loss}}{\partial t}=0 \tag{4.15}$$

4.1.4 The Eulerian Equations of Fluid Dynamics

By considering a fixed volume in space, we have the Euler equations of fluid dynamics, which express the properties of the fluid as functions of position and time.

$$\frac{\partial\rho}{\partial t}+\nabla\cdot(\rho\mathbf{v})=0 \tag{4.16}$$

$$\rho\left(\frac{\partial\mathbf{v}}{\partial t}+\mathbf{v}\cdot\nabla\mathbf{v}\right)+\nabla P+\rho\nabla\phi=0 \tag{4.17}$$

$$\rho\left(\frac{\partial u}{\partial t}+\mathbf{v}\cdot\nabla u\right)+P\nabla\cdot\mathbf{v}+\frac{\partial Q_{loss}}{\partial t}=0 \tag{4.18}$$

These equations give rates of change of quantities at a fixed point in space (the volume considered) through which the fluid flows. This Eulerian formulation is used by grid-based codes such as finite difference methods, which use a grid of cells fixed in space (the computational volumes) as their reference frame.

4.1.5 The Lagrangian Equations of Fluid Dynamics

An alternative is the Lagrangian formulation of the equations, which does not use a fixed frame of reference but instead one that moves with a fluid element, following the flow. This Lagrangian formulation is used by particle-based methods such as SPH, which use a system of sampling points moving within the flow (the computational fluid elements) as their reference frame.

The Lagrangian derivative is defined as

$$\frac{DX}{Dt}=\frac{\partial X}{\partial t}+(\mathbf{v}\cdot\nabla)X \tag{4.19}$$

It is an advective derivative, representing $\frac{\partial X}{\partial t}$ in the reference frame that comoves with the fluid element along the flow velocity field $\mathbf{v}$ (which is the *flow* velocity, *not* the fluid element velocity).

Expanding the derivative in the continuity equation

$$\frac{\partial\rho}{\partial t}+\rho(\nabla\cdot\mathbf{v})+(\mathbf{v}\cdot\nabla)\rho=0 \tag{4.20}$$

allows us to write the Lagrangian continuity equation

$$\frac{D\rho}{Dt} + \rho(\nabla \cdot \mathbf{v}) = 0 \tag{4.21}$$

which shows the relation of rate of change of density to the rate of change of the volume occupied by the fluid element (from Gauss's divergence theorem).

For the equation of motion,

$$\rho\frac{D\mathbf{v}}{Dt} + \nabla P + \rho\nabla\phi = 0 \tag{4.22}$$

and for the energy equation,

$$\rho\frac{Du}{Dt} + P\nabla \cdot \mathbf{v} + \frac{\partial Q_{loss}}{\partial t} = 0 \tag{4.23}$$

which expresses the first law of thermodynamics ($\delta u = \delta Q_{gain} - P\delta V$).

4.1.6 The Equation of State

Knowing the mass density ρ of the fluid and its flow velocity $\mathbf{v}$ at all spatial positions $\mathbf{r}$ is sufficient to specify the gravitational potential ϕ in the momentum equation. The pressure P, however, which appears in the momentum and energy equations, is not yet specified.

In addition to the three conservation equations (fluid dynamics equations), a fourth equation is therefore required to close the system of equations.

For the special case of an incompressible fluid, $\nabla \cdot \mathbf{v} = 0$ (conservation of volume) is itself sufficient to close the system of equations.

In the more general case, an equation of state is used. This specifies the pressure P of the fluid as a function of other thermodynamic variables, typically density ρ and specific internal energy u. The isothermal equation of state, for example, is $P = c_s^2\rho$, where c_s is the isothermal sound speed of the fluid.

4.2 Computational Fluid Dynamics

The fundamental equations of hydrodynamics are continuum equations that describe a continuous fluid. Computations, however, cannot operate in this manner and so the equations must first be discretised through the introduction of a finite number of interpolation points. There are two main approaches that each achieve this requirement in a fundamentally different manner:
a) Grid-based (Eulerian) approach - use the vertices of a grid (typically fixed in space) as the interpolation points.
b) Particle-based (Lagrangian) approach - use particles that move with the flow as interpolation points.

4.2.1 Grid-Based Methods

By constructing a grid of points, cells can be defined in the spaces between them. In the class of "finite difference" schemes, fluxes (of mass, momentum and energy) between adjoining cells are then calculated by taking finite differences of the fluid quantities in space.

Using a grid, the spatial scale is fixed and the mass scale is free to vary. The spatial resolution is thus limited by the cell size, making it difficult to follow large changes in density once an object becomes smaller than the cell size. This is a major issue for simulations of star and planet formation, where objects condense through many orders of magnitude in density.

Multiple grid methods (finer grids placed within regions of interest) have been used to improve this, culminating in "adaptive mesh refinement" (Truelove et al. 1998), where the grid is effectively dynamically rescaled to ensure sufficient resolution. This is computationally expensive, however, slowing the method down considerably.

For disc simulations the grid is typically chosen to be cylindrical, or spherical with cell divisions concentrated about the midplane. Such an arrangement is optimal since it aligns with the fluid motion, thus minimising the shearing effects that would be experienced by cells on a Cartesian grid. The drawback to this arrangement, however, is that it essentially imposes a preferred geometry, and typically assumes a fixed star at the origin of the coordinate system, preventing any modes involving relative motion between the star and the disc from occurring. If a disc is of comparable mass to the star then the prevention of such behaviour becomes physically unrealistic.

4.2.2 Particle-Based Methods

Gridless particle-based methods such as Smoothed Particle Hydrodynamics (SPH) use freely moving particles to represent the fluid. Each particle is essentially a point at which the underlying fluid quantities (of mass, momentum and energy) are sampled.

In order to calculate hydrodynamic quantities such as density and pressure, contributions from other particles must be considered. Each particle is therefore given a sphere of influence, and those with overlapping spheres interact with each other. The size of this sphere might be fixed or allowed to vary, and is typically chosen such that it always contains a suitable number of neighbouring particles. The total hydrodynamic quantity for a particle can then be determined by a weighted average over all its neighbours.

Using particles, the mass scale is fixed but the spatial scale is free to vary. This makes it ideally suited to situations in which densities evolve through many orders of magnitude to high values, such as star and planet formation. Conversely it makes it difficult to resolve low density regimes correctly.

4.2.3 Complications

The Euler equations of fluid dynamics are nonlinear hyperbolic equations. As shown in Chapter 2 this leads to various computational issues, including the risk of numerical dispersion and diffusion.

The general solutions to the equations are waves, and it is possible for these to "break" and form shock waves where the solution becomes discontinous and effectively multi-valued. Physically this essentially breaches the assumptions used to formulate the differential equations. This problem could be circumvented by returning to the integral equations and including discontinuities in the fluid quantities in order to derive appropriate solutions. Physical quantities are rarely discontinuous, however, since in reality any discontinuity is smoothed out through the action of viscosity. Since this is not possible when the inviscid Euler momentum equation is used, and since the Navier-Stokes equation can be difficult to implement, artificial viscosity is often introduced to alleviate this problem, as will be shown in Chapter 6.

4.3 Smoothed Particle Hydrodynamics

Smoothed particle hydrodynamics (Lucy 1977; Gingold & Monaghan 1977) is a gridless method using freely moving particles to represent the fluid.

The foundation of SPH begins with the simple identity

$$A(\mathbf{r}_a) = \int A(\mathbf{r}_b)\delta(|\mathbf{r}_a - \mathbf{r}_b|)d\mathbf{r}_b \tag{4.24}$$

Here A is any fluid quantity defined at all spatial positions $\mathbf{r}$ and δ is the Dirac delta function.

We now replace the Dirac delta function with a smoothing function W, which has a characteristic width h where $\lim_{h\to 0} W(|\mathbf{r}_a - \mathbf{r}_b|, h) = \delta(|\mathbf{r}_a - \mathbf{r}_b|)$ and is normalised such that $\int W(|\mathbf{r}_a - \mathbf{r}_b|, h)d\mathbf{r}_b = 1$.

$$A(\mathbf{r}_a) = \int A(\mathbf{r}_b)W(|\mathbf{r}_a - \mathbf{r}_b|, h)d\mathbf{r}_b + O(h^2) \tag{4.25}$$

Incorporating the mass density ρ provides a mass element $m = \rho d\mathbf{r}$.

$$A(\mathbf{r}_a) = \int \frac{A(\mathbf{r}_b)}{\rho(\mathbf{r}_b)}W(|\mathbf{r}_a - \mathbf{r}_b|, h)\rho(\mathbf{r}_b)d\mathbf{r}_b + O(h^2) \tag{4.26}$$

This integral is discretised onto a finite set of interpolation points, replacing the integral with a summation.

$$A(\mathbf{r}_a) \approx \sum_b \frac{A_b}{\rho_b}W(|\mathbf{r}_a - \mathbf{r}_b|, h)m_b + O(h^2) \tag{4.27}$$

The subscripts b refer to quantities evaluated at the positions of the interpolation points. These points have an associated mass m_b given by the mass element $\rho(\mathbf{r}_b)d\mathbf{r}_b$.

This interpolated sum forms the basis of SPH.

Gradient terms can be calculated by analytic derivation of the integral.

$$\nabla A(\mathbf{r}_a) = \frac{\partial}{\partial \mathbf{r}_a}\int \frac{A(\mathbf{r}_b)}{\rho(\mathbf{r}_b)}W(|\mathbf{r}_a - \mathbf{r}_b|, h)\rho(\mathbf{r}_b)d\mathbf{r}_b + O(h^2) \tag{4.28}$$

Discretising this results in the following expression.

$$\nabla A(\mathbf{r}_a) \approx \sum_b \frac{A_b}{\rho_b}\frac{\partial}{\partial \mathbf{r}_a}W(|\mathbf{r}_a - \mathbf{r}_b|, h)m_b + O(h^2) \tag{4.29}$$

4.3.1 The SPH Continuity Equation

Taking density ρ as our fluid quantity A, we have

$$\rho(\mathbf{r}_a) = \sum_b m_b W(|\mathbf{r}_a - \mathbf{r}_b|, h) \tag{4.30}$$

The time derivative of this

$$\begin{aligned}
\frac{d\rho(\mathbf{r}_a)}{dt} &= \frac{d}{dt}\left(\sum_b m_b W(|\mathbf{r}_a - \mathbf{r}_b|, h)\right) && (4.31)\\
&= \sum_b m_b \frac{d}{dt} W(|\mathbf{r}_a - \mathbf{r}_b|, h) && (4.32)\\
&= \sum_b m_b \frac{d(\mathbf{r}_a - \mathbf{r}_b)}{dt} \frac{W(|\mathbf{r}_a - \mathbf{r}_b|, h)}{d(\mathbf{r}_a - \mathbf{r}_b)} && (4.33)\\
&= \sum_b m_b (\mathbf{v}_a - \mathbf{v}_b) \cdot \frac{W(|\mathbf{r}_a - \mathbf{r}_b|, h)}{d(\mathbf{r}_a - \mathbf{r}_b)} && (4.34)\\
&= \sum_b m_b (\mathbf{v}_a - \mathbf{v}_b) \cdot \nabla_a W_{ab} && (4.35)
\end{aligned}$$

leads to the SPH continuity equation

$$\frac{d\rho(\mathbf{r}_a)}{dt} = \sum_b m_b (\mathbf{v}_a - \mathbf{v}_b) \cdot \nabla W(|\mathbf{r}_a - \mathbf{r}_b|, h) = -\rho(\mathbf{r}_a)(\nabla \cdot \mathbf{v})_a \tag{4.36}$$

4.3.2 The SPH Equation of Motion

The SPH momentum equation can be derived by consideration of the Lagrangian

$$L = T - V = \int \left(\frac{1}{2}\rho v^2 - \rho u - \rho\phi\right) dV \tag{4.37}$$

which (noting that $m = \rho dV$) has the SPH form

$$L_{SPH} = \sum_b m_b \left(\frac{1}{2} v_b^2 - u_b(\rho_b, s_b) - \phi\right) \tag{4.38}$$

The equation of motion will be provided by the Euler-Lagrange equation.

$$\frac{d}{dt}\left(\frac{\partial L}{\partial v}\right) - \frac{\partial L}{\partial r} = 0 \tag{4.39}$$

The first term is simply

$$\frac{d}{dt}\left(\frac{\partial L_{SPH}}{\partial \mathbf{v}_a}\right) = \frac{d}{dt}(m_a \mathbf{v}_a) = m_a \frac{d\mathbf{v}_a}{dt} \tag{4.40}$$

The second term is

$$\frac{\partial L_{SPH}}{\partial \mathbf{r}_a} = \sum_b m_b \left(-\frac{\partial u_b}{\partial \rho_b}\Big|_s \frac{\partial \rho_b}{\partial \mathbf{r}_a} - \frac{\partial \phi_a}{\partial \mathbf{r}_a} \right) \tag{4.41}$$

where $\Big|_S$ indicates constant entropy (an adiabat).

From the first law of thermodynamics, $du = Tds - Pdv$, it follows that $\frac{\partial u}{\partial \rho}\Big|_s = -P\frac{dv}{d\rho} = -P\frac{d}{d\rho}\left(\frac{1}{\rho}\right) = \frac{P}{\rho^2}$.

Noting that $\rho_b = \sum_c m_c W_{bc} = \sum_c m_c W\left(\frac{|\mathbf{r}_b - \mathbf{r}_c|}{h}\right)$, it follows that

$$\begin{aligned}
\frac{\partial \rho_b}{\partial \mathbf{r}_a} &= \sum_c m_c \frac{\partial W(|\mathbf{r}_b - \mathbf{r}_c|, h)}{\partial \mathbf{r}_a} && (4.42)\\
&= \delta_{ab} \sum_c m_c \frac{\partial W(|\mathbf{r}_a - \mathbf{r}_c|, h)}{\partial \mathbf{r}_a} + \delta_{ac} \sum_c m_c \frac{\partial W(|\mathbf{r}_b - \mathbf{r}_a|, h)}{\partial \mathbf{r}_a} && (4.43)\\
&= \delta_{ab} \sum_c m_c \frac{\partial W_{ac}}{\partial \mathbf{r}_a} + \delta_{ac} m_a \frac{\partial W_{ba}}{\partial \mathbf{r}_a} && (4.44)
\end{aligned}$$

Substituting these expressions into the second term, and noting that the smoothing function is symmetric ($\frac{\partial W_{ba}}{\partial \mathbf{r}_a} = \frac{\partial W_{ab}}{\partial \mathbf{r}_a}$), gives

$$\begin{aligned}
\frac{\partial L_{SPH}}{\partial \mathbf{r}_a} &= \sum_b m_b \left(-\frac{P_b}{\rho_b^2} \right) \frac{\partial \rho_b}{\partial \mathbf{r}_a} && (4.45)\\
&= -\left(\sum_b m_b \frac{P_b}{\rho_b^2} \delta_{ab} \sum_c m_c \frac{\partial W_{ac}}{\partial \mathbf{r}_a} + \sum_b m_b \frac{P_b}{\rho_b^2} \delta_{ac} m_a \frac{\partial W_{ba}}{\partial \mathbf{r}_a} \right) && (4.46)\\
&= -\left(m_a \frac{P_a}{\rho_a^2} \sum_b m_b \frac{\partial W_{ab}}{\partial \mathbf{r}_a} + m_a \sum_b m_b \frac{P_b}{\rho_b^2} \frac{\partial W_{ba}}{\partial \mathbf{r}_a} \right) && (4.47)\\
&= -m_a \sum_b m_b \left(\frac{P_a}{\rho_a^2} + \frac{P_b}{\rho_b^2} \right) \frac{\partial W_{ab}}{\partial \mathbf{r}_a} && (4.48)
\end{aligned}$$

Finally substituting into the Euler-Lagrange equation gives the SPH equation of motion.

$$\frac{d\mathbf{v}_a}{dt} = -\sum_b m_b \left(\frac{P_a}{\rho_a^2} + \frac{P_b}{\rho_b^2} \right) \frac{\partial W_{ab}}{\partial \mathbf{r}_a} \tag{4.49}$$

It can be seen that the pressure force between two particles a and b is symmetric (equal and opposite), conserving linear momentum.

The SPH equation of motion also conserves angular momentum, which can be seen by considering the time derivative of the angular momentum of a particle.

$$\begin{aligned}
\frac{d}{dt} \sum_a \mathbf{r}_a \times m_a \mathbf{v}_a &= \sum_a m_a \left(\mathbf{r}_a \times \frac{d\mathbf{v}_a}{dt} \right) && (4.50)\\
&= \sum_a \sum_b m_a m_b \left(\frac{P_a}{\rho_a^2} + \frac{P_b}{\rho_b^2} \right) \left(\mathbf{r}_a \times \frac{\partial W_{ab}}{\partial \mathbf{r}_a} \right) && (4.51)
\end{aligned}$$

Since $\frac{\partial W_{ab}}{\partial \mathbf{r}_a}$ is ultimately a function of $\mathbf{r}_{ab} = \mathbf{r}_a - \mathbf{r}_b$, the final term essentially becomes a $-(\mathbf{r}_a \times \mathbf{r}_b)$ term. Over the double summation, each $-(\mathbf{r}_a \times \mathbf{r}_b)$ contribution is cancelled by a $+(\mathbf{r}_b \times \mathbf{r}_a)$ contribution, giving a net result of zero, which indicates the conservation of angular momentum.

4.3.3 The SPH Energy Equation

From the first law of thermodynamics, $du = Tds - Pdv$, we have seen that $\left.\frac{\partial u}{\partial \rho}\right|_s = \frac{P}{\rho^2}$. Using the chain rule, $\frac{du}{dt} = \frac{\partial u}{\partial \rho}\frac{d\rho}{dt}$, so the specific internal energy is

$$\frac{du_a}{dt} = \frac{P_a}{\rho_a^2} \sum_b m_b \left(\mathbf{v}_a - \mathbf{v}_b\right) \cdot \nabla_a W(|\mathbf{r}_a - \mathbf{r}_b|, h) \tag{4.52}$$

Since this result has been obtained at constant entropy s, it applies only in the absence of dissipation or heat transfer dQ/dt.

Alternatively, the energy equation could be formulated in terms of specific entropy s, which might be considered a stricter and more desirable arrangement (Springel & Hernquist 2002).

The total energy of the system, which is a conserved quantity, can be calculated using the Hamiltonian $H = T + V$, which has the SPH form

$$H_{SPH} = \sum_a m_a (\frac{1}{2} v_a^2 + u_a(\rho_a, s_a) + \phi) \tag{4.53}$$

Taking the time derivative,

$$\frac{dH}{dt} = \sum_a m_a \left(\mathbf{v}_a \cdot \frac{d\mathbf{v}_a}{dt} + \frac{du_a}{dt} + \frac{d\phi}{dt}\right) \tag{4.54}$$

and then considering that $\frac{dH}{dt} = \sum_a m_a \frac{de_a}{dt}$, where e is the specific energy, and substituting using the momentum and energy equations, gives the equation

$$\begin{array}{rcl} \frac{de_a}{dt} & = & \mathbf{v}_a \cdot \left[-\sum_b m_b \left(\frac{P_a}{\rho_a^2} + \frac{P_b}{\rho_b^2}\right) \frac{\partial W_{ab}}{\partial \mathbf{r}_a} \right] + \\ & & \frac{P_a}{\rho_a^2} \sum_b m_b \left(\mathbf{v}_a - \mathbf{v}_b\right) \cdot \nabla_a W(|\mathbf{r}_a - \mathbf{r}_b|, h) \end{array} \tag{4.55}$$

which simplifies to

$$\frac{de_a}{dt} = \sum_b m_b \left(-\left(\frac{P_a}{\rho_a^2} + \frac{P_b}{\rho_b^2}\right) \mathbf{v}_a \cdot \nabla_a W_{ab} + \frac{P_a}{\rho_a^2} \left(\mathbf{v}_a - \mathbf{v}_b\right) \cdot \nabla_a W_{ab} \right) \tag{4.56}$$

$$= \sum_b m_b \left(\left(-\frac{P_a}{\rho_a^2} \mathbf{v}_a - \frac{P_b}{\rho_b^2} \mathbf{v}_a + \frac{P_a}{\rho_a^2} \mathbf{v}_a - \frac{P_a}{\rho_a^2} \mathbf{v}_b \right) \cdot \nabla_a W_{ab} \right) \tag{4.57}$$

$$= -\sum_b m_b \left(\frac{P_b}{\rho_b^2} \mathbf{v}_a + \frac{P_a}{\rho_a^2} \mathbf{v}_b \right) \cdot \nabla_a W_{ab} \tag{4.58}$$

4.3.4 The Smoothing Function

The smoothing function $W(|\mathbf{r}_a - \mathbf{r}_b|, h)$ appears throughout the SPH equations, and is referred to as the smoothing kernel or kernel function. It must satisfy two requirements:
a) In the limit of zero width h it must tend towards the Dirac delta function: $\lim_{h\to 0} W(|\mathbf{r}_a - \mathbf{r}_b|, h) = \delta(|\mathbf{r}_a - \mathbf{r}_b|)$.
b) It must be normalised such that $\int W(|\mathbf{r}_a - \mathbf{r}_b|, h) d\mathbf{r}_b = 1$.

There remains a great deal of flexibility in the choice of function, and its chosen form will strongly affect the accuracy of the SPH implementation. It is sensible to choose an even function (symmetric), and one function meeting all these requirements is the Gaussian kernel $W(|\mathbf{r}_a - \mathbf{r}_b|, h) = \frac{n(D)}{h^D} e^{-\left(\frac{|\mathbf{r}_a - \mathbf{r}_b|}{h}\right)^2}$, where D is the number of spatial dimensions and $n(D)$ is the normalisation factor required for that number of dimensions ($\pi^{-1/2}$, π^{-1} or $\pi^{-3/2}$ for 1, 2 or 3 dimensions respectively). The Gaussian is infinitely smooth (all its spatial derivatives are continous functions), which provides it with good stability properties (Price 2004). Its infinite extent, however, gives it a high computational cost: by spanning the entire computational domain, the hydrodynamic calculations scale in the same manner as gravity (as $O(N^2)$). Given that the individual hydrodynamic calculations tend to involve more operations than the gravitational calculations, this could effectively cripple a computer simulation.

Since the contribution from neighbouring particles decreases with increasing distance, many of the particles in the system will have negligible effect upon each other, so an N^2 calculation is unnecessary. It is therefore more efficient to choose a kernel function that has compact support (finite extent), limiting the hydrodynamic calculations to those involving the nearest neighbours, resulting in an $O(kN)$ calculation where k is the number of neighbours involved. This does incur the additional cost of finding the k nearest neighbours for each particle, which remains an $O(N^2)$ operation, but it can be reduced to an $O(N \log N)$ operation through the use of a tree algorithm (see Chapter 5).

Kernel functions that are similar in form to the Gaussian tend to give the best performance (Fulk & Quinn 1996), and one such function is the M4 cubic spline (Monaghan & Lattanzio 1985):

$$W(q,h) = \frac{n(D)}{h^D} \begin{cases} 1 - \frac{3}{2}q^2 + \frac{3}{4}q^3 & 0 \le q \le 1 \\ \frac{1}{4}(2-q)^3 & 1 \le q \le 2 \\ 0 & q > 2 \end{cases} \tag{4.59}$$

where $q = \frac{|\mathbf{r}_a - \mathbf{r}_b|}{h}$. It is a piecewise continuous function with compact support of size $2h$, and $n(D)$ is $\frac{2}{3}$, $\frac{10}{7\pi}$ or $\frac{1}{\pi}$ for 1, 2 or 3 dimensions respectively.

The first spatial derivative of the M4 kernel function is

$$\frac{dW}{dr}(q,h) = \frac{n(D)}{h^{D+1}} \begin{cases} -3q + \frac{9}{4}q^2 & 0 \le q \le 1 \\ -\frac{3}{4}(2-q)^2 & 1 \le q \le 2 \\ 0 & q > 2 \end{cases} \tag{4.60}$$

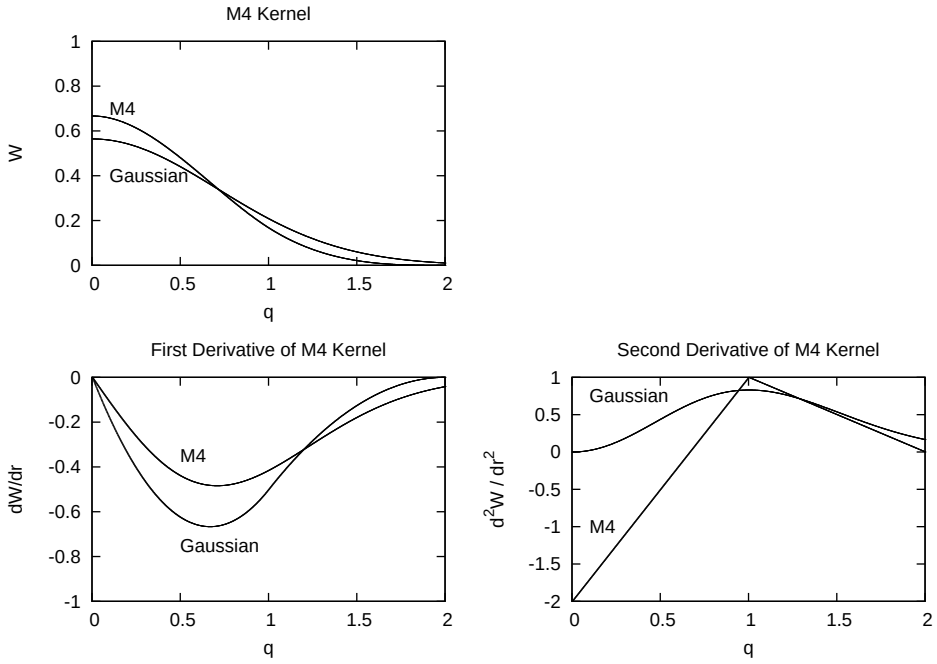


Figure 4.1: The M4 cubic spline kernel and its derivatives, with the Gaussian kernel shown for comparison.

and the second spatial derivative is

$$\frac{d^2W}{dr^2}(q,h) = \frac{n(D)}{h^{D+2}} \begin{cases} -3+\frac{9}{2}q & 0 \le q \le 1 \\ \frac{3}{2}(2-q) & 1 \le q \le 2 \\ 0 & q > 2 \end{cases} \tag{4.61}$$

Figure 4.1 shows the M4 kernel function and its derivatives. It can be seen that the first and second derivatives are also continuous, which aids in limiting the effects of noise from the particle distribution on the calculation of the fluid properties. A discontinuity appears in the third derivative, so use of the M4 kernel function suggests that second derivatives of SPH quantities should be avoided, since they will be subject to more pronounced effects from noise.

Smoother kernel functions can be implemented, by increasing the size of the compact support region (at the expense of increasing the computational cost factor k) and/or by using higher order functions, such as quartic or quintic splines, but the gain in accuracy over the cubic spline is minor (Price 2004).

4.3.5 Stability Properties

The one-dimensional dispersion relation for SPH may be derived by perturbing the SPH equations by a factor $e^{i(kx_a-\omega t)}$ (Price 2004), giving

$$\begin{aligned} \omega_a^2 &= \frac{2mP_0}{\rho_0^2}\sum_b \left(1-\cos\left[k(x_a-x_b)\right]\right)\frac{\partial^2 W}{\partial x^2}(x_a-x_b,h) \\ &+ \frac{m^2}{\rho_0^2}\left(c_s^2-\frac{2P_0}{\rho_0}\right)\left(\sum_b \sin\left[k(x_a-x_b)\right]\frac{\partial W}{\partial x}(x_a-x_b,h)\right)^2 \end{aligned} \tag{4.62}$$

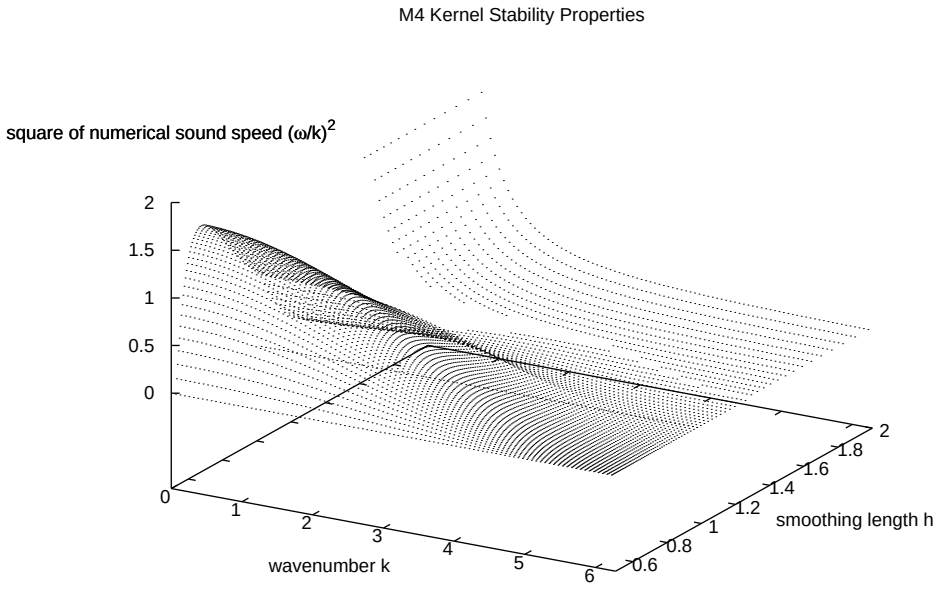


Figure 4.2: Stability phase-space diagram for the M4 cubic spline kernel, with the square of the numerical sound speed shown as a function of both wavenumber k and smoothing length h, both plotted in units of particle spacing Δx.

where particles of equal mass m have been assumed, the subscript 0 refers to the unperturbed quantities and $c_s = \frac{\partial P}{\partial \rho}$ is the sound speed, with the result being applicable to any equation of state.

With an isothermal equation of state $c_s^2 = \frac{P_0}{\rho_0}$, and assuming this to be unity and the particle spacing $\Delta x = (x_a - x_b)$ to be unity, the normalised square of the numerical sound speed $c_n^2 = \frac{\omega^2}{k^2}$ can be plotted as a function of wavenumber k and kernel width h, both in dimensionless units of particle spacing Δx, as shown in Figure 4.2.

$kx \to 0$ represents the limit of an infinite number of particles per wavelength, and $h \to \infty$ represents an infinite number of neighbours.

Numerical dispersion only occurs when $0.5 \lesssim \frac{h}{\Delta x} \lesssim 1.5$, and is maximum when $\frac{h}{\Delta x} = \frac{2}{3}$, and then decreases as $h \to \infty$.

Numerical diffusion begins when $\frac{h}{\Delta x} \gtrsim 1.3$, is maximum when $\frac{h}{\Delta x} = 2$, and then decreases as $h \to \infty$, becoming essentially negligible except in the very small k regime. For $1.3 \lesssim \frac{h}{\Delta x} \lesssim 1.5$, numerical diffusion only occurs at extremes of k, but at all values of k when $\frac{h}{\Delta x} \gtrsim 1.5$, although only the very small k regime contributes significantly.

The numerical sound speed $\frac{\omega}{k} \to -\infty$ as $k \to 0$ when $\frac{h}{\Delta x} \gtrsim 1.4$, due to numerical diffusion, but remains bounded for $0.5 \lesssim \frac{h}{\Delta x} \lesssim 1.4$, having a maximum at $\frac{h}{\Delta x} = \frac{2}{3}$ (the minimum of the kernel gradient).

Numerical effects only occur when $\frac{h}{\Delta x} \gtrsim 0.5$ because a neighbour must be within the compact support region of the kernel in order to contribute. They only

become catastrophic (negative numerical sound speed) when $\frac{h}{\Delta x} \gtrsim 1.4$, in the innermost $\sim \frac{2}{3}$ of the kernel. The Thomas & Couchman modification to the kernel helps to alleviate this issue (see Section 4.4.12).

Numerical effects become significant in the small k regime, and as $k\Delta x \to 0$ the wavelength $\lambda \gg \Delta x$, resulting in many particles per wavelength. The longest wavelength modes are therefore most susceptible to numerical effects.

Ideally the numerical sound speed should be equal to 1 for correct propagation, but the phase space plot indicates that this is not always possible. Nonetheless, so long as catastrophic numerical diffusion is avoided the behaviour should be acceptable for most purposes. Catastrophic numerical diffusion is typically indicated by the "clumping instability" of SPH, when a particle experiences an attractive rather than repulsive inter-particle force in the innermost $\sim \frac{2}{3}$ of the kernel; such behaviour demonstrates the necessity of the Thomas & Couchman modification, as will be seen in Section 4.4.12. While the catastrophic behaviour of numerical diffusion can be averted through such a modification, the milder effects of numerical dispersion are unfortunately practically inevitable, and must therefore simply be accepted and endured.

Morris (1996) provides a more detailed stability analysis of SPH.

4.3.6 The Smoothing Length

The kernel function possesses a characteristic width h, referred to as the smoothing length, which essentially determines the radius of interaction for each particle (out to $2h$ using the M4 kernel function). This in turn determines the number of nearest neighbours k which contribute to the hydrodynamic calculations. The total hydrodynamic quantity for a particle is then effectively a weighted average over these k neighbours, with the smoothing kernel W acting as the weighting function.

The simplest approach is to use a fixed smoothing length for all particles. The sphere of hydrodynamic interaction (with diameter $4h$ using the M4 kernel function), however, determines the effective size of a particle, which corresponds to a minimum spatial resolution length. Allowing the smoothing length to vary according to local conditions can therefore substantially improve the spatial resolution, and also improves stability by acting to normalise the kernel gradient, providing a major improvement in accuracy (Price 2004).

In order to maintain resolution during a collapse process, the smoothing length h must decrease as the density ρ increases. One way to achieve this is to determine h by ensuring the sphere of interaction of a particle always contains the same number of neighbours k. As density increases, particles become closer together and so h decreases. Conversely in diffuse regions where particles spread out, h increases to maintain sufficient contributions to the hydrodynamic calculations, preventing any particle from effectively "decoupling" from the others. In keeping the number of neighbours constant, an appropriate value for this factor must be chosen. More neighbours provide greater accuracy, but at the cost of more computational processing. A value of ~ 50 neighbours seems to be the typical choice in the trade-off between accuracy and speed. The number of neighbours k should be fixed, without any tolerance Δk, in order to minimise numerical dispersion and diffusion (Attwood 2008).

Another way to determine h is to take the inverse scaling relation with density as $h \sim \rho^{-\frac{1}{D}}$, where D is the number of spatial dimensions, and apply it directly. This approach is more complicated to implement, since for each particle a the SPH density ρ_a is a function of h_a; a self-consistent solution therefore requires an iterative process.

The time evolution of h can be calculated using the chain rule, $\frac{dh_a}{dt} = \frac{dh_a}{d\rho_a}\frac{d\rho_a}{dt} = -\frac{h_a}{D\rho_a}\frac{d\rho_a}{dt}$. Using this relation to update h will maintain the inverse scaling relation well enough for most practical purposes, particularly if ρ is updated using the SPH continuity equation. To be fully self-consistent, however, terms involving the derivative of h should be incorporated into the momentum and energy equations (Springel & Hernquist 2002).

"Grad-h" Formulation

Taking the SPH density summation

$$\rho_a = \rho(\mathbf{r}_a) = \sum_b m_b W(|\mathbf{r}_a - \mathbf{r}_b|, h_a) \tag{4.63}$$

and noting that $h_a = h(\rho_a)$ is now variable and so a function of time, the time derivative is then

$$\frac{d\rho_a}{dt} = \sum_b m_b \left(\frac{\partial |\mathbf{r}_a - \mathbf{r}_b|}{\partial t}\frac{dW_{ab}}{d\mathbf{r}} + \frac{\partial h_a}{\partial t}\frac{dW_{ab}}{dh} \right) \tag{4.64}$$

which, using the chain rule $\frac{\partial h_a}{\partial t} = \frac{\partial \rho_a}{\partial t}\frac{\partial h_a}{\partial \rho_a}$, gives

$$\frac{d\rho_a}{dt}\left(1 - \frac{\partial h_a}{\partial \rho_a}\sum_b m_b \frac{dW_{ab}}{dh}\right) = \sum_b m_b(\mathbf{v}_a - \mathbf{v}_b)\cdot\nabla_a W_{ab}(h_a) \tag{4.65}$$

which can be arranged in the usual form

$$\frac{d\rho_a}{dt} = \frac{1}{\Omega_a}\sum_b m_b(\mathbf{v}_a - \mathbf{v}_b)\cdot\nabla_a W_{ab}(h_a) \tag{4.66}$$

where

$$\Omega_a = 1 - \frac{\partial h_a}{\partial \rho_a}\sum_c m_c \frac{\partial W_{ab}(h_a)}{\partial h_a} \tag{4.67}$$

is then the correction term that accounts for the variation in h; this can be computed at the same time as the SPH density summation.

For "grad-h" SPH, the Ω correction term kernel function in D dimensions is given by

$$\frac{\partial W}{\partial h}(q,h) = \frac{n(D)}{h^{D+1}}\begin{cases} -D + \frac{3}{2}(D+2)q^2 - \frac{3}{4}(D+3)q^3 & 0 \le q \le 1 \\ -2D + 3(D+1)q - \frac{3}{2}(D+2)q^2 + \frac{1}{4}(D+3)q^3 & 1 \le q \le 2 \\ 0 & q > 2 \end{cases} \tag{4.68}$$

or equivalently $\frac{\partial W}{\partial h} = -DW - q\frac{dW}{dr}$ (see Figure 4.3).

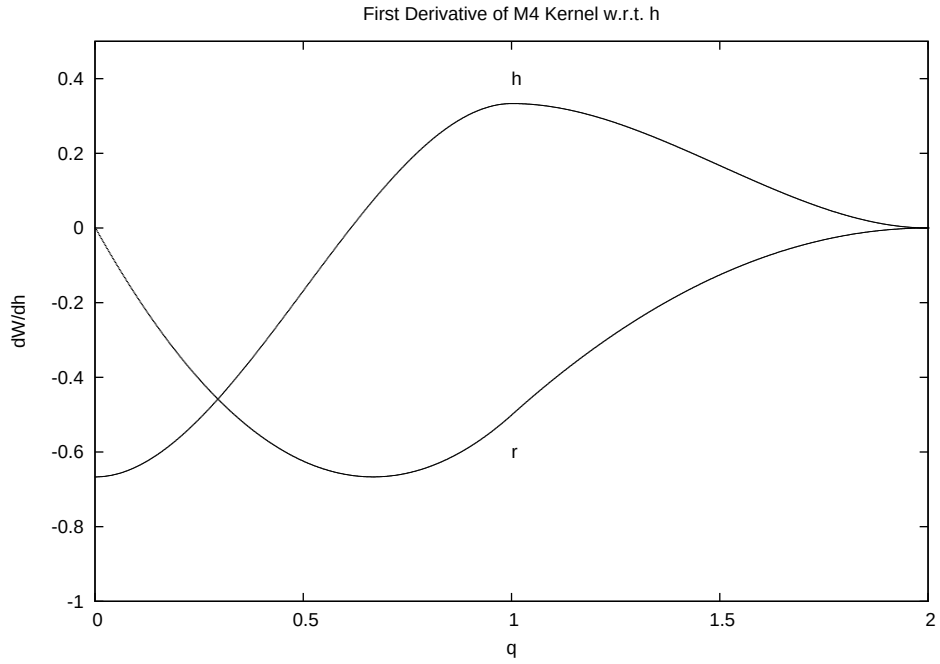


Figure 4.3: The derivative of the M4 kernel with respect to h. The spatial derivative with respect to r is also shown for comparison.

Since $\frac{\partial h_a}{\partial \rho_a}$ is negative, close neighbours within $\sim \frac{2}{3}$ will contribute to reducing Ω below 1 (thus increasing their contribution to density), while more distant neighbours, particularly those at a distance $\sim h$, will contribute to increasing Ω (thus reducing their contribution to density).

The equation of motion can again be found from the Euler-Lagrange equation, thus conserving both linear and angular momentum, as

$$\frac{d\mathbf{v}_a}{dt} = -\sum_b m_b \left(\frac{P_a}{\Omega_a \rho_a^2} \frac{\partial W_{ab}(h_a)}{\partial \mathbf{r}_a} + \frac{P_b}{\Omega_b \rho_b^2} \frac{\partial W_{ab}(h_b)}{\partial \mathbf{r}_a} \right) \tag{4.69}$$

and the energy equation as

$$\frac{du_a}{dt} = \frac{P_a}{\Omega_a \rho_a^2} \sum_b m_b \left(\mathbf{v}_a - \mathbf{v}_b \right) \cdot \nabla_a W(|\mathbf{r}_a - \mathbf{r}_b|, h_a) \tag{4.70}$$

Self-Consistent Calculation

To self-consistently calculate the SPH density summation ρ_a and the smoothing length h_a, the time evolution of h can be used to determine its expected value, which is then used to calculate ρ in the SPH summation. Then a new value of the smoothing length h_{new} can be calculated from this density using the inverse scaling relation $h_{new} = \eta \rho^{-\frac{1}{D}}$, where η is a scaling factor (effectively determining the number of neighbours k). Convergence can be determined using a condition such as $\frac{h_{new} - h}{h} < 0.01$ to maintain consistency within 1%. Any particles not satisfying convergence can

then set their smoothing length to h_{new} and recalculate ρ using this value. Since ρ_a is independent of h_b, only particles that have not converged require this iterative treatment, which can continue until the condition is met.

The iteration process can be improved by implementing the Newton-Raphson method, using the calculated Ω gradient terms, since this converges faster than simple fixed point iteration. It is necessary, however, to maintain a simple brute force approach, such as a bisection method, as a fall-back option to deal with any cases where the Newton-Raphson method becomes unstable and gets trapped in an infinite oscillation between values.

Since $h = \eta\,(m/\rho)^{1/D}$, rearranging gives $\rho = m\,(\eta/h)^D$. In order to find a relationship between η and the equivalent number of neighbours k, consider that $\rho = \frac{(k+1)m}{V}$, where the $+1$ accounts for self-density and V is the volume enveloped by the kernel, which for the M4 kernel (extending to $2h$) is $2(2h)$, $\pi(2h)^2$ or $\frac{4}{3}\pi(2h)^3$ for 1, 2 or 3 spatial dimensions D.

$$\eta = \left(\frac{h^D}{V}(k+1)\right)^{1/D} \tag{4.71}$$

For the fiducial value of $k = 50$ neighbours in 3 dimensions, $\eta \simeq 1.15$. Conversely, the popular value of $\eta = 1.3$ is equivalent to ~ 72 neighbours in 3 dimensions.

The prevailing orthodoxy of $k = 50$ neighbours has arisen from the necessary compromise between high values, which increases the "smoothness" of SPH (reducing the noise and bringing the SPH summations closer to the integrals that they approximate through the increased number of contributions), and low values, which not only provides computational efficiency (less calculations) but also increases the spatial resolution (due to the reduction in the values of the smoothing lengths h). It is therefore desirable to choose the highest value of k or η that still provides reasonable efficiency and sufficient resolution.

4.3.7 Time-Stepping Issues

Since SPH can be derived from a Hamiltonian variational principle (in the absence of dissipation terms), it benefits from symplectic integration in time. Time symmetry, however, becomes a complicated issue when considering equations with rates of change involving the particle velocity (such as the energy equation or the continuity equation), or when using individual particle timesteps.

The Courant-Friedrichs-Lewy condition must be modified to incorporate $\Delta t < \frac{h}{v_{sig}}$, where the signal velocity $v_{sig} = c_s + h|\nabla \cdot \mathbf{v}|$ must account for both the local sound speed and the relative motion of the particles.

The stability condition based upon acceleration, $\Delta t < (h/|\mathbf{a}|)^{1/2}$, must also be maintained.

If the SPH energy equation is explicitly solved then it is sensible to limit the fractional change in internal energy u per timestep by using $\Delta t < u/\left(|\frac{du}{dt}| + \epsilon\right)$, where ϵ is a small number that prevents division by zero.

4.3.8 Including Gravity

From the previous chapter, the gravitationally softened acceleration experienced by particle i due to particle j can be written as

$$\mathbf{a}_{ij} = -m_j \frac{\mathbf{r}_{ij}}{(|\mathbf{r}_{ij}|^2 + \epsilon^2)^{3/2}} \tag{4.72}$$

where G is normalised to unity and ϵ is the softening length.

Since our formulation of hydrodynamics incorporates a smoothing length h and a kernel function W, it is sensible to implement gravitational softening based on these parameters, so that gravity and hydrodynamics are both resolved at the same resolution scale. If they are not then if hydrodynamics is resolved in regions where gravity is unresolved, collapse processes will be artificially suppressed, while conversely if gravity is resolved in regions where hydrodynamics is not resolved then collapse processes will be artificially enhanced (Nelson 2006). Using the smoothing kernel W,

$$\mathbf{a}_{ij} = -m_j \frac{\mathbf{r}_{ij}}{|\mathbf{r}_{ij}|^3} \int_0^r W(r,h) 4\pi r^2 dr \tag{4.73}$$

is the gravitational acceleration corresponding to the mass density distribution described by the SPH equation $\rho(\mathbf{r}_i) = \sum_j m_j W(\mathbf{r}_{ij}, h)$; this can be seen by applying Poisson's equation $\nabla^2\Phi = 4\pi G\rho$, where Φ is the specific gravitational potential, so $\mathbf{a} = -\nabla\Phi$. Beyond the compact support region of the kernel function ($2h$ for the M4 kernel) this acceleration takes the form of an inverse square law, but within the sphere of hydrodynamic interaction gravity is softened in a manner consistent with the smoothed nature of the mass density distribution.

For conservation of momentum, the accelerations must be symmetrised such that particle j experiences an equal and opposite contribution from particle i. With a constant smoothing length h this is not a problem, but when h is variable it requires symmetrisation, such as the formulation $h = \frac{h_i + h_j}{2}$. Unfortunately this spatially varying value of the softening length results in the total energy of the system no longer being conserved. It is possible, however, to use a Lagrangian formulation of gravitational softening that conserves energy and momentum exactly, within the constraints of errors introduced by the time integration scheme (Price & Monaghan 2007).

The Softening Kernel

We can write the specific gravitational potential as

$$\Phi(\mathbf{r}) = -G \sum_b m_b \phi(|\mathbf{r} - \mathbf{r}_b|, h) \tag{4.74}$$

where the softening kernel ϕ is a function of particle separation and softening length h. This softening length h is set to be the same as the hydrodynamic smoothing length, for reasons detailed above. The softening kernel then determines the form of the softened gravitational force law.

$$\mathbf{F}(\mathbf{r}) = -\nabla\Phi = G \sum_b m_b \frac{\partial\phi}{\partial|\mathbf{r} - \mathbf{r}_b|}(|\mathbf{r} - \mathbf{r}_b|, h) \frac{\mathbf{r} - \mathbf{r}_b}{|\mathbf{r} - \mathbf{r}_b|} \tag{4.75}$$

Since Poisson's equation $\nabla^2\Phi = 4\pi G\rho$ must result in

$$\rho(\mathbf{r}_i) = \sum_b m_b W(|\mathbf{r} - \mathbf{r}_b, h) \tag{4.76}$$

this allows the smoothing kernel W to be related to the softening kernel ϕ using Equation 4.73.

$$-\frac{\partial\phi}{\partial r} = \frac{1}{r^2}\int_0^r W(r)4\pi r^2 dr \tag{4.77}$$

gives

$$W(r) = -\frac{1}{4\pi r^2}\frac{\partial}{\partial r}\left(r^2\frac{\partial\phi}{\partial r}\right) \tag{4.78}$$

or equivalently

$$\frac{\partial\phi}{\partial r} = \frac{4\pi}{r^2}\int_0^r W r^2 dr \tag{4.79}$$

and, integrating by parts,

$$\phi = \int\frac{\partial\phi}{\partial r}dr = 4\pi\left(-\frac{1}{r}\int_0^r W r^2 dr + \int_0^r W r dr - \int_0^{2h} W r dr\right) \tag{4.80}$$

where the final term is the integration constant determined by the condition that $\phi \to 0$ as $r \to \infty$, with assumed compact support of size $2h$.

Using the M4 kernel function as the smoothing kernel W therefore gives the softening kernel ϕ, normalised in 3 dimensions, as

$$\phi(q,h) = \frac{1}{h}\begin{cases} \frac{7}{5} - \frac{2}{3}q^2 + \frac{3}{10}q^4 - \frac{1}{10}q^5 & 0 \le q \le 1 \\ \frac{8}{5} - \frac{4}{3}q^2 + q^3 - \frac{3}{10}q^4 + \frac{1}{30}q^5 - \frac{1}{15}\frac{1}{q} & 1 \le q \le 2 \\ 1/q & q > 2 \end{cases} \tag{4.81}$$

with $q \equiv \frac{r}{h}$, and its first derivative as

$$-\frac{\partial\phi}{\partial r}(q) = \frac{1}{h^2}\begin{cases} \frac{4}{3}q - \frac{6}{5}q^3 + \frac{1}{2}q^4 & 0 \le q \le 1 \\ \frac{8}{3}q - 3q^2 + \frac{6}{5}q^3 - \frac{1}{6}q^4 - \frac{1}{15}\frac{1}{q^2} & 1 \le q \le 2 \\ 1/q^2 & q > 2 \end{cases} \tag{4.82}$$

Figure 4.4 shows the softening kernel with its first derivative, and the effects they have on the gravitational potential and force laws.

Energy-Conserving Formulation

We can write the gravitational potential term in the SPH Lagrangian as

$$L_{grav} = -\sum_b m_b\Phi_b = -\frac{G}{2}\sum_b\sum_c m_b m_c\phi(|\mathbf{r}_b - \mathbf{r}_c|, h_b) \tag{4.83}$$

where the factor of 2 in the denominator accounts for the double counting in the double summation. Considering the swapping of indices in the double summation gives the symmetrised form

$$L_{grav} = -\frac{G}{2}\sum_b\sum_c m_b m_c\frac{\phi_{bc}(h_b) + \phi_{bc}(h_c)}{2} \tag{4.84}$$

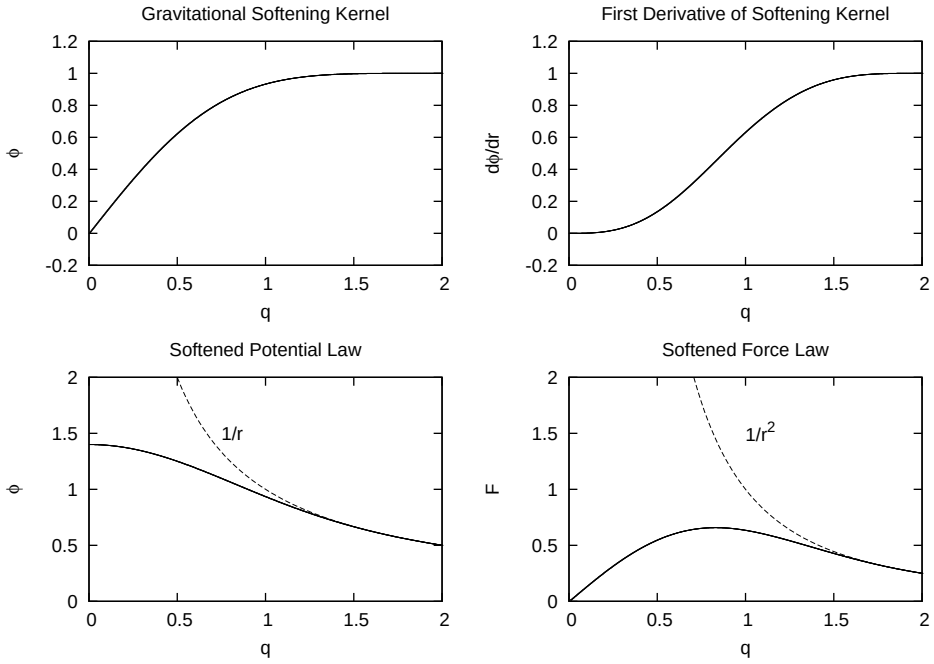


Figure 4.4: The softening kernel and its first derivative (upper plots). The lower plots show the analytic forms of the gravitational potential and force laws (inverse linear and square laws respectively) together with the kernel-softened forms.

which incorporates an average of the softening kernels. Differentiating the unsymmetrised equation gives

$$\frac{\partial L_{grav}}{\partial \mathbf{r}_a} = -\frac{G}{2}\sum_b \sum_c m_b m_c \left(\frac{\partial \phi_{bc}(h_b)}{\partial |r_{bc}|}\bigg|_h \frac{\partial |r_{bc}|}{\partial \mathbf{r}_a} + \frac{\partial \phi_{bc}(h_b)}{\partial h_b}\bigg|_r \frac{\partial h_b}{\partial \mathbf{r}_a} \right) \tag{4.85}$$

where $\frac{\partial |r_{bc}|}{\partial \mathbf{r}_a} = \frac{\mathbf{r}_b - \mathbf{r}_c}{|\mathbf{r}_b - \mathbf{r}_c|}(\delta_{ba} - \delta_{ca})$ and using the chain rule $\frac{\partial h_b}{\partial \mathbf{r}_a} = \frac{\partial h_b}{\partial \rho_b}\frac{\partial \rho_b}{\partial \mathbf{r}_a}$ since $h_b = h(\rho_b)$. Differentiating the SPH density summation gives

$$\frac{\partial \rho_b}{\partial \mathbf{r}_a} = \frac{1}{\Omega_b}\sum_d m_d \frac{\partial W_{bd}(h_b)}{\partial \mathbf{r}_a}(\delta_{ba} - \delta_{da}) \tag{4.86}$$

and making these substitutions gives

$$\begin{aligned} \frac{\partial L_{grav}}{\partial \mathbf{r}_a} &= -\frac{G}{2}\textstyle\sum_b \sum_c m_b m_c \left(\frac{\partial \phi_{bc}(h_b)}{\partial |r_{bc}|}\Big|_h \frac{\mathbf{r}_b - \mathbf{r}_c}{|\mathbf{r}_b - \mathbf{r}_c|}(\delta_{ba} - \delta_{ca}) \right. \\ &\quad \left. + \frac{\partial \phi_{bc}(h_b)}{\partial h_b}\Big|_r \frac{\partial h_b}{\partial \rho_b}\frac{1}{\Omega_b}\sum_d m_d \frac{\partial W_{bd}(h_b)}{\partial \mathbf{r}_a}(\delta_{ba} - \delta_{da}) \right) \end{aligned} \tag{4.87}$$

Using the Euler-Lagrange equation $m_a \frac{dv_a}{dt}\Big|_{grav} = \frac{\partial L_{grav}}{\partial \mathbf{r}_a}$ results in, after simplifying,

$$\begin{aligned} \frac{dv_a}{dt}\Big|_{grav} &= -\frac{G}{2}\textstyle\sum_b m_b \left(\frac{\partial \phi_{ab}(h_a)}{\partial |r_{ab}|} + \frac{\partial \phi_{ab}(h_b)}{\partial |r_{ab}|} \right) \frac{\mathbf{r}_a - \mathbf{r}_b}{|\mathbf{r}_a - \mathbf{r}_b|} \\ &\quad -\frac{G}{2}\textstyle\sum_b m_b \left(\frac{\zeta_a}{\Omega_a}\frac{\partial W_{ab}(h_a)}{\partial \mathbf{r}_a} + \frac{\zeta_b}{\Omega_b}\frac{\partial W_{ab}(h_b)}{\partial \mathbf{r}_a} \right) \end{aligned} \tag{4.88}$$

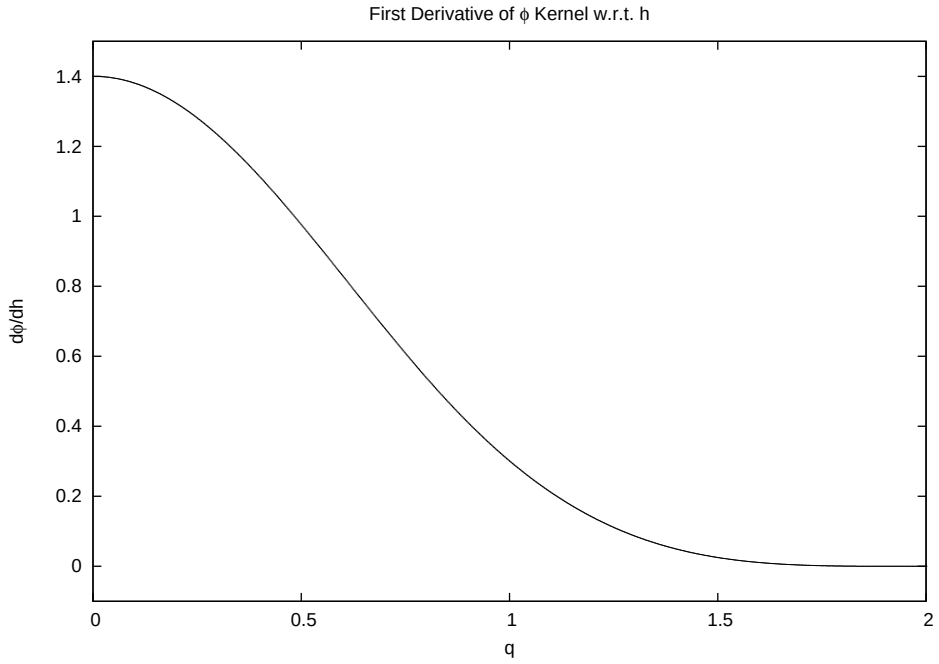


Figure 4.5: The derivative of the softening kernel with respect to h.

where

$$\zeta_a = \frac{\partial h_a}{\partial \rho_a} \sum_b m_b \frac{\partial \phi_{ab}(h_a)}{\partial h_a} \tag{4.89}$$

is essentially a correction term.

For the M4 smoothing kernel the $\frac{\partial \phi}{\partial h}$ function is given by

$$\frac{\partial \phi}{\partial h}(q,h) = \frac{1}{\pi h^3} \begin{cases} \frac{7}{5} - 2q^2 + \frac{3}{2}q^4 - \frac{3}{5}q^5 & 0 \le q \le 1 \\ \frac{8}{5} - 4q^2 + 4q^3 - \frac{3}{2}q^4 + \frac{1}{5}q^5 & 1 \le q \le 2 \\ 0 & q > 2 \end{cases} \tag{4.90}$$

and is shown in Figure 4.5.

4.4 Developing an SPH Code

Using the N-body code from the previous chapter as a foundation, we can continue the incremental development of the code to incorporate the SPH algorithms. For testing and debugging purposes our aim will be to ensure that the code can solve self-gravitating hydrodynamic problems correctly.

As a preliminary step the hydrodynamic settling of a periodic box will be attempted. In the absence of gravity, such a box (filled at random with particles of equal mass, and using an isothermal equation of state) should settle into a state of uniform density through the action of pressure forces between neighbouring particles. Since each particle will exert pressure on its neighbours this should ultimately result in the formation of a glass-like structure. This is recommended as a first test because,

although it requires the implementation of periodic boundary conditions, it is purely hydrodynamic and so does not require the inclusion of gravity.

In keeping with incremental development, it is recommended that the correction terms (Ω and ζ) initially be ignored (they can be set to unity in the implementation), and that a constant smoothing length h is first used, in order to ease the debugging process.

4.4.1 Data Structures

Particle data must now also incorporate smoothing lengths h and densities ρ, plus the correction terms Ω and ζ. Depending on the equation of state used, pressures, sound speeds, temperatures and/or internal energies may also be required.

In order to maintain the optimal order in memory for the gravitational force calculation, the properties involved in such calculations should be grouped contiguously in memory: positions, masses and smoothing lengths (which act as the gravitational softening lengths).

Global variables have been recommended for the particle data, but those quantities that depend upon the equation of state may benefit from being encapsulated within a Fortran module or a C++ class.

4.4.2 Compiler Flags

Besides setting the number of spatial dimensions, compiler flags might be considered for the purpose of switching the gravitational and hydrodynamic calculations on and off, allowing for purely gravitational, purely hydrodynamic or self-gravitating hydrodynamic simulations, with a check to ensure that at least one of these flags is switched on.

Flags to determine the equation of state would also be useful, as would flags to switch on periodic boundary conditions. Periodic boundary conditions need only be applied for the pure hydrodynamic case, but periodic gravity could be implemented using the Ewald method (Hernquist et al. 1991). It would be useful to implement periodic boundary conditions independently in each dimension since this will be required for later hydrodynamic tests in Chapter 6.

Certain parameters could be implemented as macros, such as η (the scaling factor in the $h(\rho)$ equation).

4.4.3 The Main Program

Using a top-down approach the program still consists of three main parts as before: input, integration and output.

In the initial input part we again input any model parameters and the initial conditions (initial state of the system being modelled at time $t = 0$), and perform other preparations such as allocating memory for data stuctures (including the new particle data), converting to code units (including new quantities) and initialising variables (such as setting time $t = 0$, and also initialising the kernel function look-up tables and smoothing lengths).

In the integration part, where we perform the calculation itself through the application of our numerical method in algorithmic form, we will need to make major additions in order to incorporate SPH.

4.4.4 Parameters

New model parameters might include:

- Additional scaling unit preferences, used to specify the physical units used for scales of density, pressure, temperature and so forth;
- Periodic boundary positions such as x_{min}, x_{max}, y_{min}, y_{max}, z_{min} and z_{max};
- The number of elements to use in the kernel function look-up tables (probably better suited to being a macro);
- The scaling factor η in the $h(\rho)$ equation, or the number of neighbours k required (probably better suited to being a macro);
- The mean molecular mass $\bar{\mu}$, used to determine the sound constant.

4.4.5 Input/Output Format

Additional quantities, not required for input as initial conditions since they are calculated as functions of the state variables, that might be useful to include in output for diagnostic purposes, include smoothing length, density, pressure, sound speed, temperature and so forth. Density and temperature are probably the most useful to incorporate here.

Any diagnostic information on acceleration could be split into two components: the gravitational contribution and the hydrodynamic contribution. Similarly any diagnostic information on energy could be split into three components: kinetic energy, gravitational potential energy and internal thermal energy.

4.4.6 Memory Allocation

Memory for the data structures to store new particle properties (such as smoothing length, density, and so forth) will need to be allocated. Memory will also be required for the kernel function look-up tables; the number of elements to use in each table could either be supplied as a parameter (requiring dynamic memory allocation) or as a macro (using static arrays).

4.4.7 Code Units

Input and output should be made in physical units, as specified by the scaling unit preferences in the model parameters.

Conversion between physical units (used in input and output) and code units (used within the program) must be performed at the input/output interfaces.

Additional units for density, pressure, temperature and so forth may need to be incorporated for output purposes.

For gravitational calculations, t_{scale} was determined in a manner that normalised the gravitational constant G to unity. For the case of purely hydrodynamic simulations, where there are no gravitational calculations, t_{scale} itself can simply be set to unity instead.

The scaling factors for density ρ and pressure P can be derived very simply, since density is mass per unit volume and pressure is force per unit area.

$$\rho_{scale} \times \rho_{SI} = \frac{m_{scale} \times m_{SI}}{(r_{scale} \times r_{SI})^3} \tag{4.91}$$

$$P_{scale} \times P_{SI} = \frac{m_{scale} \times m_{SI}}{r_{scale} \times r_{SI} \times (t_{scale} \times t_{SI})^2} \tag{4.92}$$

$$\tag{4.93}$$

The sound constant $\hat{C}_s$ can also be calculated here. This is the constant of proportionality between sound speed c_s and temperature T, such that $c_s = \hat{C}_s\sqrt{T}$.

$$c_s^2 = \frac{k_B T}{m_H \bar{\mu}} \quad \Rightarrow \hat{C}_s = \sqrt{\frac{k_B}{m_H \bar{\mu}}} \tag{4.94}$$

where k_B is the Boltzmann constant ($1.38 \times 10^{-23}\text{JK}^{-1}$), m_H is the mass of hydrogen ($1.67 \times 10^{-27}\text{kg}$), and $\bar{\mu}$ is the mean molecular mass (in units of m_H, set as a model parameter, with 2.29 being a typical value that is representative of the solar nebula).

The sound constant $\hat{C}_s$ is then normalised (divided by $v_{scale} \times v_{SI}$) in order to allow temperatures to be converted directly into sound speeds with dimensionless code units when using the formula $c_s = \hat{C}_s\sqrt{T}$.

All physical quantities must be normalised upon input (divided by X_{scale}), and conversely must be converted back to physical units upon output (multiplied by X_{scale}).

Smoothing lengths should be multiplied by r_{scale} on output, and sound speeds by v_{scale}, in order to make them consistent with the physical units specified for positions and velocities.

Temperatures T may be left in units of K throughout, eliminating the need for T_{scale}.

4.4.8 Kernel Function Look-Up Tables

Since the kernel function W (and its derivative) appears throughout the SPH equations it will need to be evaluated many times during a simulation. Rather than explicitly calculating W each time, it may be more efficient to tabulate it as a function of $q = \frac{|\mathbf{r}_a - \mathbf{r}_b|}{h}$ in the form of an array, and to simply look-up the corresponding element each time it is needed.

Including more elements in the look-up table improves the accuracy of the evaluations, but at the expense of greater memory requirements. Since it is sensible to try to keep the table small enough to fit into one of the fast-access memory caches of

the processor, $10^3 - 10^4$ elements is probably a reasonable compromise. CPU speeds have been improving at a faster rate than memory access speeds, however, so an explicit calculation of W may prove to be faster than referencing a table in memory; a speed comparison should be made in order to determine the optimal approach.

If each table is an array from $i = 0$ to $i = n$ then each element i will correspond to a value of $q = \frac{2i}{n}$, providing tabulation for $0 \leq q \leq 2$.

Each table should be normalised appropriately for the number of spatial dimensions, but since the table is only a function of q and not h, the denominator involving h has not been accounted for in the table and must therefore be explicitly incorporated into the implementation of the SPH equations instead.

Tables should be made for the kernel function W and its first derivative $\frac{dW}{dr}$. The Thomas-Couchman modification should be made to the first derivative (see Section 4.4.12), and tables made for the softening kernel ϕ and its first derivative $\frac{d\phi}{dr}$. For the correction terms, tables should be made for $\frac{dW}{dh}$ for the Ω term, and $\frac{d\phi}{dh}$ for the ζ term.

The initialisation of the kernel tables is a good place to also initialise the smoothing lengths to some reasonable value. A basic starting point is to assume uniform density within the computational volume V and set the smoothing length as $h = \left(\frac{V}{N}\right)^{1/D}$ for N particles in D spatial dimensions.

4.4.9 Integration

The integration routine within the main program loop contains the same core processing elements as before, but with some new SPH elements now incorporated; these additions have been marked ($\star$) and *highlighted* to stand out from the previous pseudocode.

Pseudocode (integration):

```
calculate total energy and momentum of system [optional diagnostic]
IF adaptive timestep is used THEN
      calculate new timestep Δt
ENDIF
FOR each particle
      apply first part of integration scheme
      (predict position)
ENDFOR
⋆ FOR each particle
⋆       calculate smoothing length and density [and any correction terms]
⋆ ENDFOR
⋆ FOR each particle
⋆       calculate pressure using the equation of state
⋆ ENDFOR
⋆ calculate gravitational accelerations [and potentials]
⋆ calculate hydrodynamic accelerations
FOR each particle
```

```
      apply second part of integration scheme
      (advance velocity and correct position)
ENDFOR
advance time:  t = t + Δt
```

This description applies to the modified Leapfrog scheme. For the Runge-Kutta scheme, or any other scheme, each and every calculation of gravitational accelerations should be preceded by the new SPH elements, thus allowing hydrodynamic accelerations to also be calculated.

Note that if a non-barotropic equation of state is used then the internal energy u will need to be integrated for each particle using the SPH energy equation in each part of the integration scheme. Likewise any energy diagnostics may need to account for the internal thermal energy.

The calculation of an adaptive timestep should also implement the appropriate Courant-Friedrichs-Lewy condition for hydrodynamic simulations.

4.4.10 Smoothing Lengths and Densities

Using a fixed smoothing length, writing a subroutine to calculate the SPH density for each particle by summation of the contributions from other particles is fairly straightforward. This can then be tested on a randomly populated box (Figure 4.6, top-left) to check that it works correctly.

Since the kernel function has been tabulated as a function of q, the $\frac{1}{h^D}$ denominator must be incorporated in order to correctly normalise the density. Also note that the self-density (the contibution of a particle to its own density) should be included to avoid under-estimating the density when the particles have settled (Whitworth et al. 1995).

On average each particle should have the same density, except for those particles close to edge of the box because they will have a deficit of neighbours in the outside direction. This edge effect is a peril of SPH, where particles at the edge of the computational domain experience a drop in density (Figure 4.6, top-right). For 1000 particles in 2 spatial dimensions the average smoothing length is $h = \left(\frac{V}{N}\right)^{1/D} = \left(\frac{1}{1000}\right)^{1/2} \simeq 0.0316$. This length scale indicates how far the edge effects can be expected to penetrate the computational domain.

Periodic boundary conditions will eliminate this effect, so they should be implemented and tested next (Figure 4.6, bottom-left). These are fairly straightforward to implement through the use of a dedicated subroutine for calculating relative position vectors ($\mathbf{r}_{ij} = \mathbf{r}_i - \mathbf{r}_j$) between two particles i and j whenever such information is required. If the separation x_{ij} in any spatial dimension x exceeds half the size of the computational domain L_x in that dimension, then the displacement can be adjusted accordingly in order to enforce periodic boundary conditions.
If $x_{ij} > \frac{L_x}{2}$ then x_{ij} maps to $x_{ij} - L_x$.
If $x_{ij} < -\frac{L_x}{2}$ then x_{ij} maps to $x_{ij} + L_x$.

Since the correction term Ω can be calculated at the same time as the density ρ, it may now be implemented and tested. On average each particle should have a

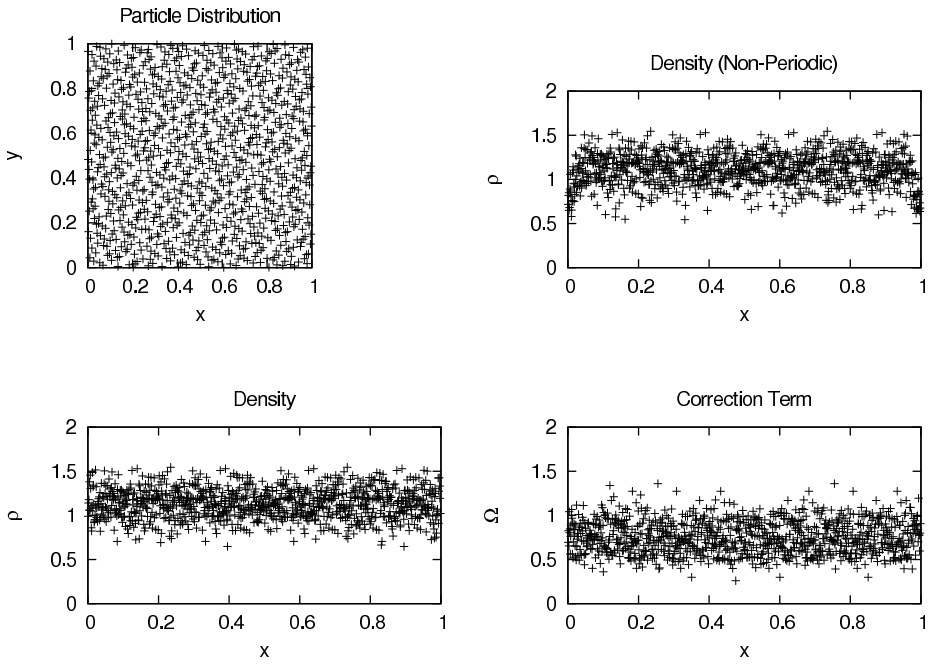


Figure 4.6: A randomly populated box containing 1000 particles (top-left) and its SPH density evaluation (top-right), showing a drop at the edges, which is eliminated by employing periodic boundary conditions (bottom-left). The correction term Ω is also shown (bottom-right).

value of Ω close to unity (Figure 4.6, bottom-right).

To implement the variable smoothing length, the derivative of density with respect to time $\frac{d\rho}{dt}$ should be calculated at the same time as density.

Pseudocode (calculate smoothing length and density):

```
FOR each particle i
      store value of h_i as h_old
      calculate dh_i/dt = -(h_i/(D ρ_i)) dρ_i/dt
      estimate new value h_new = h_old + (dh_i/dt) Δt
      REPEAT
            set h_i = h_new
            calculate ρ_i using SPH summation
            calculate h_new = η (m_i/ρ_i)^(1/D)
            calculate convergence = |h_new − h_old| / h_old
            set h_old = h_new
      UNTIL convergence < 0.01
ENDFOR i
```

When estimating the new value h_{new} it is prudent to check that it is not negative. Ideally the adaptive timestep conditions should prevent this from happening, but since that is not guaranteed then the possibility of an exception should be

addressed.

4.4.11 Equation of State

The equation of state is used to calculate the pressure P for each particle. For barotropic equations of state, where P is solely a function of density ρ and independent of internal energy u, the SPH energy equation will not be required.

The simplest equation of state is the isothermal equation $P = c_s^2\rho$, where c_s is the isothermal sound speed, which is found from the temperature T by $c_s = \hat{C}_s\sqrt{T}$, where $\hat{C}_s = \sqrt{k_B/(m_H\bar{\mu})}$ is the sound constant.

For general barotropic equations of state, the isothermal sound speed $c_s = \sqrt{P/\rho}$ no longer applies, and the local sound speed is given by $c_s = \sqrt{\frac{\partial P}{\partial \rho}}$.

In such cases it might be easier to use a pressure constant $\hat{P} = \frac{k_B}{m_H\bar{\mu}}$ rather than the sound constant $\hat{C}_s$. Once normalised (multiplied by $\rho_{scale} \times \rho_{SI}$ and divided by $P_{scale} \times P_{SI}$), this allows temperatures to be converted directly into pressures using the equation $P = \hat{P}\rho T$.

The sound speed can then be calculated as $c_s = \sqrt{\hat{P}\frac{\partial(\rho T(\rho))}{\partial \rho}}$.

For the case of a polytropic equation of state, where $P = K\rho^\eta$, the temperature can then be calculated as $T = \frac{P}{\hat{P}\rho}$ and the sound speed as $c_s = \sqrt{\frac{\eta P}{\rho}}$.

In general, using any applicable barotropic equation of state, the temperature T, sound speed c_s and pressure P can all be evaluated.

For the non-barotropic case, the use of the SPH energy equation is required and the internal energy variables u and $\frac{du}{dt}$ must be included. The internal energy u is then integrated directly together with r and v in the integration scheme, and $\frac{du}{dt}$ can be calculated alongside the hydrodynamic component of acceleration, using the SPH energy equation. An additional timestep condition, $\Delta t < u/\left(|\frac{du}{dt}| + \epsilon\right)$ is also required to maintain stability in the integration of the energy equation, where ϵ is a small number to prevent division by zero. The pressure may be calculated using the ideal gas equation of state $P = (\gamma - 1)\rho u$, and then the temperature as $T = \frac{P}{\hat{P}\rho}$ and the sound speed as $c_s = \sqrt{\frac{\gamma P}{\rho}}$.

4.4.12 Hydrodynamic Calculations

With the pressure calculated, the hydrodynamic component of acceleration can be calculated for each particle using the SPH equation of motion.

$$\frac{d\mathbf{v}_a}{dt} = -\sum_b m_b \left(\frac{P_a}{\Omega_a \rho_a^2} \frac{\partial W_{ab}(h_a)}{\partial \mathbf{r}_a} + \frac{P_b}{\Omega_b \rho_b^2} \frac{\partial W_{ab}(h_b)}{\partial \mathbf{r}_a} \right) \tag{4.95}$$

The direction is given by $\frac{\partial W_{ab}}{\partial \mathbf{r}_a}$, which is equivalent to $\frac{\partial W_{ab}}{\partial r}\frac{\mathbf{r}_a - \mathbf{r}_b}{|\mathbf{r}_a - \mathbf{r}_b|}$.

Settling a Periodic Box

When a periodic box is allowed to evolve hydrodynamically (using an isothermal equation of state) the particles will spread out through the action of pressure. Due

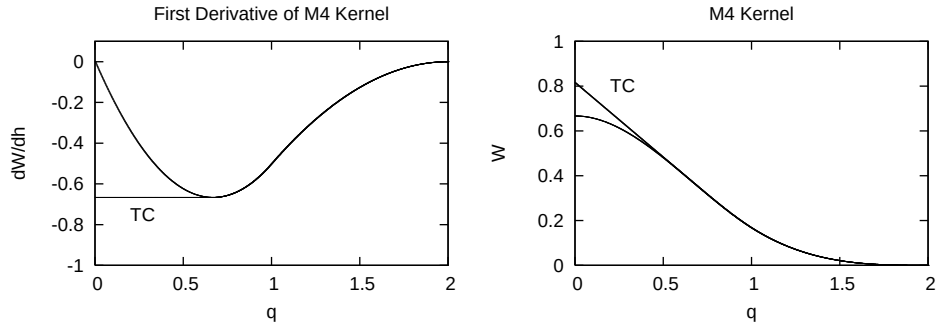


Figure 4.7: The Thomas & Couchman (1992) modification to the M4 kernel derivative (labelled TC) with the unmodified form also shown. The right-hand plot shows the effect that this modification would have on the original M4 kernel.

to the functional form of the kernel gradient $\frac{dW}{dr}$ it is possible for particles to clump together if they approach within $\frac{2}{3}h$ of each other. This can be remedied by modifying the first derivative of the kernel (Thomas & Couchman 1992).

$$\frac{dW}{dr}(q,h) = \frac{n(D)}{h^{D+1}} \begin{cases} -1 & 0 \leq q \leq \frac{2}{3} \\ -3q + \frac{9}{4}q^2 & \frac{2}{3} \leq q \leq 1 \\ -\frac{3}{4}(2-q)^2 & 1 \leq q \leq 2 \\ 0 & q > 2 \end{cases} \tag{4.96}$$

This modification, shown in Figure 4.7, will slightly steepen and raise the central portion of the original kernel W when integrated, but the difference is marginal and affects very few particles (the purpose of the modification is to discourage them from occupying this region of the kernel, after all) so the other kernel functions are typically left in unmodified form.

The particles confined within a periodic box will spread themselves out as much as possible through the action of pressure, essentially jostling each other for space. This thermalised gas will remain in such a jostling state, and to fully settle the particles into their relaxed configuration requires their motion to be damped. This can be rapidly achieved through the application of artificial viscosity (see Chapter 6) or more gradually by applying a damping factor directly to the accelerations, either reducing them by a fixed percentage or in a velocity-dependent manner ($\frac{dv}{dt} = \mathbf{a} - 0.05v$ for example). This damping process allows the gas particles to settle into a relaxed crystalline configuration that resembles a glass (Figure 4.8).

Figure 4.9 shows the result of not using the Thomas & Couchman modification. Although the particles have settled to a similar density range, close inspection reveals that the structure contains a propensity of particle pairs that have clumped together due to the close-range attractive behaviour of the kernel derivative.

A similar kind of clumping behaviour can occur if an adjustment algorithm is used to settle particles towards a desired density profile (Whitworth et al. 1995). Suitable caution is therefore advised whenever particles are settled, to avoid any undesirable behaviour.

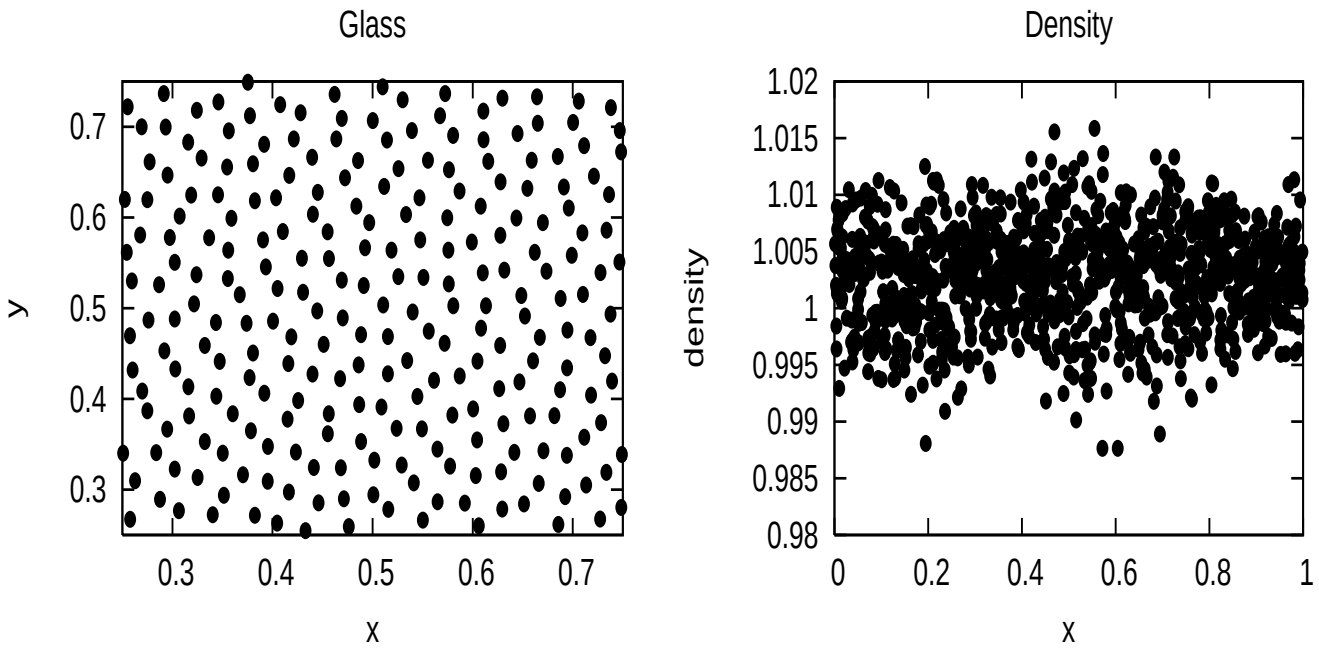


Figure 4.8: 1000 particles in a 2D periodic box settled into a "glass" configuration, to within $\sim 1 - 2\%$ of uniform unit density.

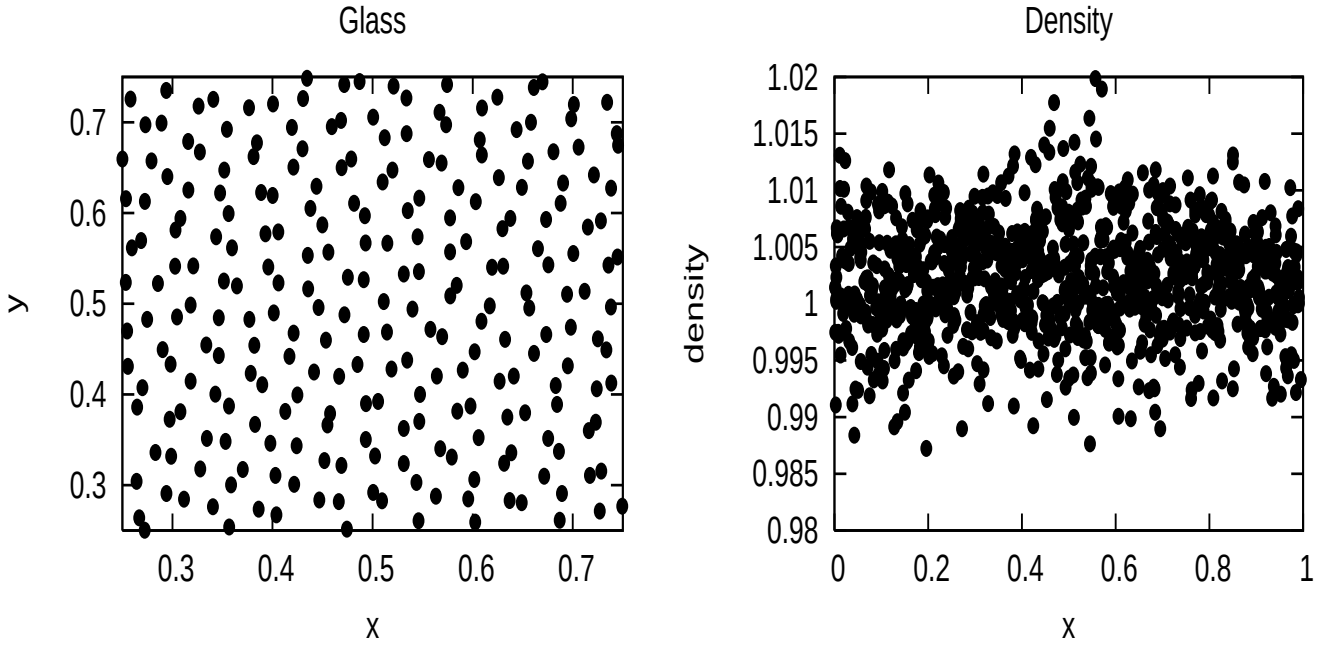


Figure 4.9: 1000 particles in a 2D periodic box settled into a "glass" configuration, without the Thomas & Couchman modification.

4.4.13 Gravitational Calculations

To calculate the gravitational acceleration experienced by particle i due to particle j, with G normalised to unity,

$$\mathbf{a}_{ij} = -m_j \frac{\mathbf{r}_{ij}}{|\mathbf{r}_{ij}|^3} \tag{4.97}$$

where m_j is the mass of particle j and $\mathbf{r}_{ij} = \mathbf{r}_i - \mathbf{r}_j$ is the relative position vector of particle i with respect to particle j. For particle j,

$$\mathbf{a}_{ji} = m_i \frac{\mathbf{r}_{ij}}{|\mathbf{r}_{ij}|^3} \tag{4.98}$$

We can write the specific gravitational potential as

$$\Phi(\mathbf{r}) = -G \sum_b m_b \phi(|\mathbf{r} - \mathbf{r}_b|, h) \tag{4.99}$$

where the softening kernel ϕ is a function of particle separation and softening length h. This softening length h is set to be the same as the hydrodynamic smoothing length for reasons detailed in Section 4.3.8. The softening kernel then determines the form of the softened gravitational force law.

$$\mathbf{a}(\mathbf{r}) = -\nabla\Phi = G \sum_b m_b \frac{\partial \phi}{\partial |\mathbf{r} - \mathbf{r}_b|}(|\mathbf{r} - \mathbf{r}_b|, h) \frac{\mathbf{r} - \mathbf{r}_b}{|\mathbf{r} - \mathbf{r}_b|} \tag{4.100}$$

which in symmetrised form gives

$$\mathbf{a}_{ij} = -m_j \frac{1}{2} \left(\frac{\partial \phi_{ij}(h_a)}{\partial r} + \frac{\partial \phi_{ij}(h_b)}{\partial r} \right) \frac{\mathbf{r}_{ij}}{|\mathbf{r}_{ij}|} \tag{4.101}$$

or, if the energy-conserving formulation is applied,

$$\begin{aligned} \mathbf{a}_{ij} \quad = \quad & -m_j \tfrac{1}{2} \left[\left(\tfrac{\partial \phi_{ij}(h_a)}{\partial r} + \tfrac{\partial \phi_{ij}(h_b)}{\partial r} \right) \tfrac{\mathbf{r}_{ij}}{|\mathbf{r}_{ij}|} \right. \\ & \left. - \left(\tfrac{\zeta_a}{\Omega_a} \tfrac{\partial W_{ab}(h_a)}{\partial r} + \tfrac{\zeta_b}{\Omega_b} \tfrac{\partial W_{ab}(h_b)}{\partial r} \right) \right] \end{aligned} \tag{4.102}$$

where the ζ correction term can be calculated at the same time as Ω.

Once kernel-softened gravity is implemented it can be tested using the freefall collapse of a uniform-density sphere of cold gas ($T = 0$ so no hydrodynamic pressure is experienced).

Note that in practice it is helpful to renormalise the softening kernel functions. By multiplying ϕ by a factor of r (and $\frac{\partial \phi}{\partial r}$ by r^2) the functions can then be tabulated solely as functions of q (where $q = \frac{r}{h}$), removing the presence of h (or h^2) terms in the denominator. All equations involving these renormalised softening kernels will then require multiplication by $\frac{1}{r}$ (or $\frac{1}{r^2}$). Besides tending to be more computationally efficient to express the calculations in this manner, it also preserves the physical behaviour (inverse linear and square laws) with respect to r in an explicit manner, making the equations (and the ensuing source code) easier for humans to read and interpret.

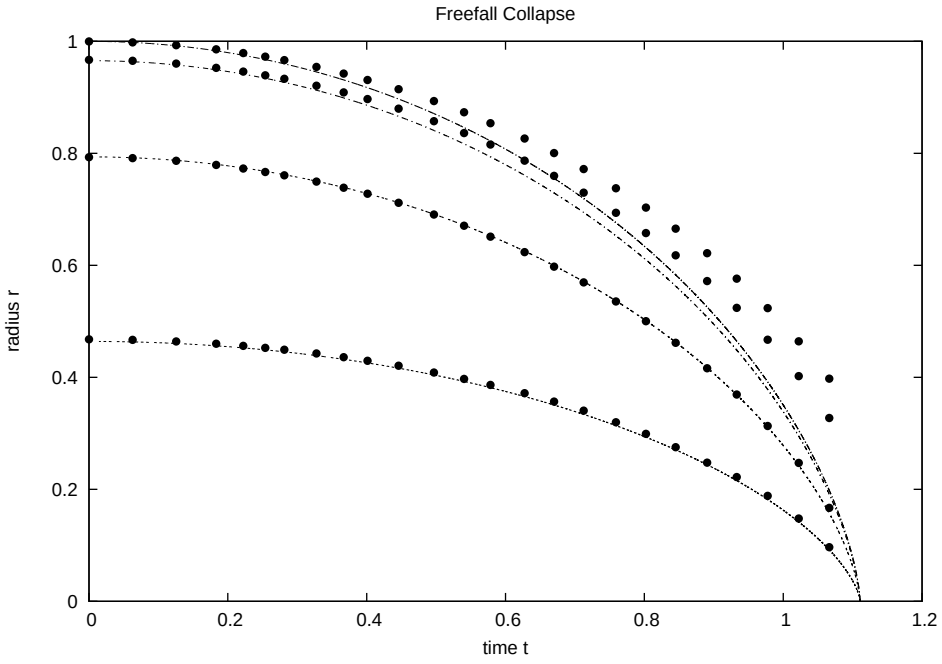


Figure 4.10: Freefall collapse for $N = 1000$ particles with kernel-softened gravity. The lines show the analytic solutions for the sphere radius and the 90%, 50% and 10% mass radii inside it. The points are the numerical solution values at periodic intervals.

Figure 4.10 shows the results of a freefall collapse simulation. As the singularity is approached the simulation deviates slightly from the analytic solution due to gravitational softening, particularly in the outermost parts. By this stage the density is more than 7 orders of magnitude greater than its initial value.

Although the general form of the analytic solution is obeyed the combination of adaptive timesteps and variable softening lengths greatly damages the conservation of energy, particularly in the later stages of collapse. For strict energy conservation, fixed timesteps and softening lengths are required; nonetheless the early stages of collapse provide a useful preliminary test for the code at this developmental stage.

4.5 Isothermal Collapse

A static uniform density gas sphere with no internal (or external) pressure collapses homologously to a singularity on a freefall timescale t_{ff}.

$$t_{ff} = \frac{\pi}{2}\left(\frac{2GM}{R^3}\right)^{-1/2} \tag{4.103}$$

where R and M are the radius and mass of the sphere respectively.

In the first chapter we derived the analytic solution for the evolution of the radius of the sphere $r(t)$ as a function of time under gravitational freefall,

$$\theta + \cos(\theta)(1 - \cos^2(\theta))^{1/2} = \left(\frac{2GM}{R^3}\right)^{1/2} t \tag{4.104}$$

where $\cos^2(\theta) = \frac{r(t)}{R}$. This can be written in terms of the freefall time as

$$\frac{t}{t_{ff}} = \frac{2}{\pi}\left(\arccos\left(\frac{r}{R}\right)^{1/2} + \left(\frac{r}{R}\right)^{1/2}\left(1-\frac{r}{R}\right)^{1/2}\right) \tag{4.105}$$

If internal pressure is included (with an isothermal equation of state) then a rarefaction wave propagates inwards from the surface of the sphere. Ahead of this wavefront the gas collapses in accordance with Equation 4.105, but behind it the gas deviates from the analytic solution. The position of the rarefaction wavefront r_{rf} moves inwards according to the sound speed c_s and the local flow speed $v(r_{rf}, \rho)$, and its progress can be evaluated (Truelove et al. 1998) according to

$$\frac{dr_{rf}}{dt} = -c_s + v(r_{rf}, \rho) \tag{4.106}$$

Introducing dimensionless variables $\xi = \frac{r_{rf}}{R_0}$ for the position of the rarefaction wavefront and $\tau = \frac{t}{t_{ff}}$ for time,

$$\frac{R_0}{t_{ff}}\frac{d\xi}{d\tau} = -c_s + v(r_{rf}, \rho) \tag{4.107}$$

From conservation of energy the freefall velocity v_{ff} at the surface R_0 is

$$v_{ff} = \left(\frac{2GM_0}{R_0}\right)^{1/2} = \frac{\pi}{2}\frac{R_0}{t_{ff}} \tag{4.108}$$

so

$$\frac{2v_{ff}}{\pi}\frac{d\xi}{d\tau} = -c_s + v(r_{rf}, \rho) \tag{4.109}$$

Introducing a dimensionless variable $\delta = (\rho/\rho_0)^{1/3} \equiv \frac{R}{r}$, the freefall equation can be written as

$$\begin{aligned} \tau &= \tfrac{2}{\pi}\left(\arccos\left(\delta^{-1/2}\right) + \delta^{-1/2}\left(1-\delta^{-1}\right)^{1/2}\right) \\ &= \tfrac{2}{\pi}\left(\arccos\left(\delta^{-1/2}\right) + \delta^{-1}\left(\delta-1\right)^{1/2}\right) \end{aligned} \tag{4.110}$$

From conservation of energy,

$$\frac{1}{2}v^2(r) = GM\left(\frac{1}{r} - \frac{1}{R}\right) \tag{4.111}$$

which can be rearranged to give the flow velocity in terms of the freefall velocity v_{ff},

$$\begin{aligned} v(r) &= \left(2GM\left(\tfrac{1}{r} - \tfrac{1}{R}\right)\right)^{1/2} \\ &= v_{ff}\left(\tfrac{R}{r} - 1\right)^{1/2} \\ &= v_{ff}\left(\delta - 1\right)^{1/2} \end{aligned} \tag{4.112}$$

Since the flow velocity is homologous it can be rescaled (using the fact that the mass M is constant throughout the collapse, so $\rho_0 R^3 = \rho r_{rf}^3$, therefore $v(r_{rf}, \rho) \sim (\rho/\rho_0)^{1/3}\,\frac{r_{rf}}{R}v(r)$) to give the value at the position of the rarefaction wavefront,

$$\begin{aligned} v(r_{rf}, \delta(\rho)) &= -\tfrac{r_{rf}}{R}\delta v_{ff}\left(\delta-1\right)^{1/2} \\ &= -\xi v_{ff}\left(\delta-1\right)^{1/2} \end{aligned} \tag{4.113}$$

where the negative sign is used to indicate the converging flow solution.

Substituting this result gives

$$\frac{2v_{ff}}{\pi}\frac{d\xi}{d\tau} = -c_s - \xi v_{ff}\delta\left(\delta-1\right)^{1/2} \tag{4.114}$$

so

$$\frac{d\xi}{d\tau} = -\frac{\pi}{2}\left(\frac{c_s}{v_{ff}} + \xi\delta\left(\delta-1\right)^{1/2}\right) \tag{4.115}$$

Considering the mass inside the rarefaction wavefront $M_{rf} = \frac{4}{3}\pi r_{rf}^3\rho$, and introducing a dimensionless variable $\mu = \left(M_{rf}/M\right)^{1/3}$ for mass (so $\mu = \xi\delta$ and equivalently $\xi = \frac{\mu}{\delta}$) we can consider the rate of change of M_{rf} with respect to the density ρ, using the product rule, as

$$\frac{d\mu}{d\delta} = \frac{d\xi}{d\delta}\delta + \xi \tag{4.116}$$

and then applying the chain rule,

$$\frac{d\mu}{d\delta} = \frac{d\xi}{d\tau}\frac{d\tau}{d\delta}\delta + \xi \tag{4.117}$$

Using the previous results

$$\begin{aligned}
\frac{d\xi}{d\tau} &= -\frac{\pi}{2}\left(\frac{c_s}{v_{ff}} + \xi\delta\left(\delta-1\right)^{1/2}\right) && (4.118)\\
\tau &= \frac{2}{\pi}\left(\arccos\left(\delta^{-1/2}\right) + \delta^{-1}\left(\delta-1\right)^{1/2}\right) && (4.119)
\end{aligned}$$

and differentiating the second, using the identity $\frac{d}{dx}\arccos x = \frac{-1}{(1-x^2)^{1/2}}$ with the chain rule to determine $\frac{d}{d\delta}\arccos(\delta^{-1/2}) = \frac{1}{2}\delta^{-1}(\delta-1)^{-1/2}$, to get

$$\begin{aligned}
\tfrac{d\tau}{d\delta} &= \tfrac{2}{\pi}\left(\tfrac{1}{2}\delta^{-1}(\delta-1)^{-1/2} + \left[-\delta^{-2}(\delta-1)^{1/2}\right] + \left[\delta^{-1}\tfrac{1}{2}(\delta-1)^{-1/2}\right]\right)\\
&= \tfrac{2}{\pi}\left[\delta^{-1}(\delta-1)^{-1/2} - \delta^{-2}(\delta-1)^{1/2}\right]\\
&= \tfrac{2}{\pi}\left(\tfrac{1}{\delta(\delta-1)^{1/2}} - \tfrac{(\delta-1)^{1/2})}{\delta^2}\right)\\
&= \tfrac{2}{\pi}\left(\tfrac{\delta^2-\delta(\delta-1)}{\delta^3(\delta-1)^{1/2}}\right)\\
&= \tfrac{2}{\pi}\tfrac{1}{\delta^2(\delta-1)^{1/2}}
\end{aligned} \tag{4.120}$$

gives

$$\begin{aligned}
\tfrac{d\mu}{d\delta} &= \tfrac{d\xi}{d\tau}\tfrac{d\tau}{d\delta}\delta + \xi\\
&= \left(-\tfrac{\pi}{2}\left[\tfrac{c_s}{v_{ff}} + \xi\delta(\delta-1)^{1/2}\right]\right)\left(\tfrac{2}{\pi}\tfrac{1}{\delta^2(\delta-1)^{1/2}}\right)\delta + \xi\\
&= -\left(\tfrac{c_s}{v_{ff}}\tfrac{1}{\delta(\delta-1)^{1/2}} + \xi\right) + \xi\\
&= -\tfrac{c_s}{v_{ff}}\tfrac{1}{\delta(\delta-1)^{1/2}}
\end{aligned} \tag{4.121}$$

Using the substitution $x^2 = \delta - 1$, so $\frac{dx}{d\delta} = \frac{1}{2x}$,

$$\frac{d\mu}{d\delta} = \frac{d\mu}{dx}\frac{1}{2x} = -\frac{c_s}{v_{ff}}\frac{1}{(1+x^2)x} \tag{4.122}$$

so

$$\frac{d\mu}{dx} = -\frac{c_s}{v_{ff}} \frac{2}{1+x^2} \tag{4.123}$$

which can be integrated using the identity $\frac{d}{dx} \arctan x = \frac{1}{1+x^2}$ to get

$$\mu = \mu_C - 2\frac{c_s}{v_{ff}} \arctan (\delta - 1)^{1/2} \tag{4.124}$$

where μ_C is the integration constant, which is unity since $\mu = 1$ when $\delta = 1$ at time $t = 0$.

Substituting back into physical variables,

$$\left(\frac{M_{rf}}{M}\right)^{1/3} = 1 - 2\frac{c_s}{v_{ff}} \arctan \left[\left(\frac{\rho}{\rho_0}\right)^{1/3} - 1 \right]^{1/2} \tag{4.125}$$

The position of the rarefaction wavefront r_{rf} can therefore be determined as a function of time, using $M_{rf} = \frac{4}{3}\pi r_{rf}^3 \rho$, and density ρ as a proxy for time t via the freefall equation.

Note that in the limit $\frac{\rho}{\rho_0} \to \infty$, $(M_{rf}/M)^{1/3} \to \left(1 - \pi\frac{c_s}{v_{ff}}\right)$, so the condition for the sphere to collapse to a singularity before the rarefaction wavefront reaches the centre and prevents this is $\left(1 - \pi\frac{c_s}{v_{ff}}\right) > 0$, giving $c_s < \frac{v_{ff}}{\pi}$.

This analytic solution allows comparison with numerical solutions, providing a useful basic test that involves both gravitational and hydrodynamic processes.

In order to minimise the effects of noise in the hydrodynamic calculations it is best to use initial conditions that have been settled into a relaxed glass-like state using a periodic box. The required uniform density sphere can then be trimmed from within this uniform density cube.

Figure 4.11 shows the 90%, 50% and 10% mass radii as a function of time. The simulation is dimensionless, with $G = 1$ and $c_s = 1$; this means that the rarefaction wave reaches the centre of the sphere in less than a freefall time, preventing collapse to a singularity. As the rarefaction wavefront passes through each mass radius, collapse is prevented and divergence from the pressureless analytic solution occurs.

The premature deviations in the simulation, occurring before the rarefaction wave should have arrived, are due to hydrodynamic smoothing at low resolution; with only 1000 particles in the sphere its radius is represented by just a few particles in any given direction. Optimisation of the code to improve its performance will allow this test to be repeated at higher resolution in the next chapter (see Section 5.5); nonetheless it remains a useful preliminary test at this developmental stage, even at such poor resolution, since the basic underlying physical behaviour is still exhibited.

4.6 Stable Polytropes

When a polytropic equation of state ($P = K\rho^\eta$) is applied, a uniform density gas sphere will gradually relax to a polytrope.

The radial density profile of the polytrope is given by a Lane-Emden function, which is obtained by numerically solving the Lane-Emden equation.

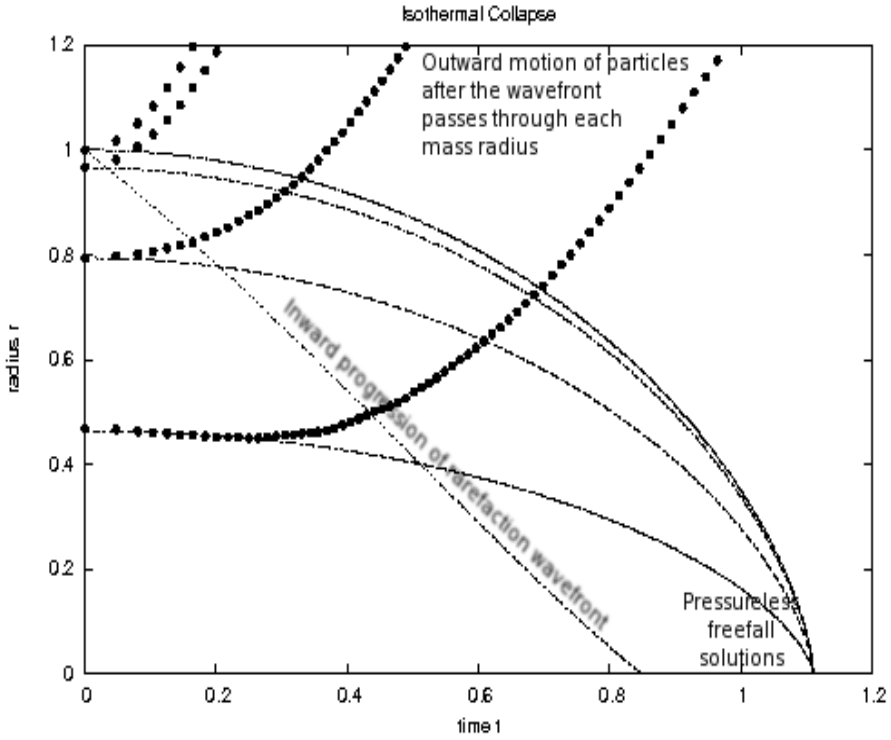


Figure 4.11: Isothermal collapse for $N = 1000$ particles with SPH and kernel-softened gravity. The lines show the freefall (pressureless) analytic solutions for the sphere radius and the 90%, 50% and 10% mass radii inside it. The points are the numerical solution values at periodic intervals. The sloping near-diagonal line is the analytic solution for the inward propagation of the rarefaction wave.

The particles will oscillate about this solution; in order to completely settle to it their motion must be damped. Artificial viscosity (see Chapter 6) can be used for this purpose.

4.6.1 The Polytropic Equation of State

$$P = K\rho^{\eta} \tag{4.126}$$

The pressure P is related to the density ρ by a polytropic constant K and the polytropic exponent η, which may also be expressed in terms of a polytropic index n where $\eta = 1 + \frac{1}{n}$. Polytropes with a higher polytropic index are therefore more centrally condensed, with the smaller value of η providing a softer equation of state; conversely larger values of η give a harder equation of state.

4.6.2 The Lane-Emden Equation

For a spherically symmetric polytrope to be in hydrostatic balance,

$$\frac{1}{\rho(r)}\frac{dP(r)}{dr} = -\frac{GM(r)}{r^2} \tag{4.127}$$

where r is the radial position, $\rho(r)$ is the density at r and $M(r)$ is the mass interior to r, which rearranges to

$$M(r) = -\frac{r^2}{G\rho(r)}\frac{dP(r)}{dr} \tag{4.128}$$

and this differentiates to give

$$\frac{dM(r)}{dr} = -\frac{1}{G}\frac{d}{dr}\left(\frac{r^2}{\rho(r)}\frac{dP(r)}{dr}\right) \tag{4.129}$$

From conservation of mass,

$$M(r) = \int_0^r \rho(r) 4\pi r^2 dr \tag{4.130}$$

so it follows that

$$\frac{d}{dr}\left(\frac{r^2}{\rho(r)}\frac{dP(r)}{dr}\right) + G\rho(r)4\pi r^2 = 0 \tag{4.131}$$

Substituting the polytropic equation of state $P(r) = K\rho^{(1+\frac{1}{n})}(r)$ into this equation gives

$$\frac{d}{dr}\left(r^2\rho^{(\frac{1}{n}-1)}(r)\frac{d\rho(r)}{dr}\right) + \frac{nG\rho(r)4\pi r^2}{K(n+1)} = 0 \tag{4.132}$$

Introducing dimensionless variables $\xi = \frac{r}{R}$ for position and $\theta_n(\xi) = (\rho(r = \xi R)/\rho_c)^{1/n}$ as a proxy for density, where ρ_c is the central density ($\rho(r = 0)$) and R is a scale length defined as $R = \left(\frac{K(n+1)}{4\pi G}\rho_c^{(\frac{1}{n}-1)}\right)^{1/2}$, this equation may then be expressed as

$$\frac{1}{\xi^2}\frac{d}{d\xi}\left(\xi^2\frac{d\theta_n(\xi)}{d\xi}\right) + (\theta_n(\xi))^n = 0 \tag{4.133}$$

which is the Lane-Emden equation.

4.6.3 The Lane-Emden Functions

The Lane-Emden functions are the values of $\theta_n(\xi)$ that are solutions of the Lane-Emden equation for different values of the polytropic index n. Finding these solutions requires the application of boundary conditions.

The first boundary condition is provided by considering the definition of θ_n at $r = 0$, which gives $\rho(r = 0) = \rho_c \left(\theta_n(\xi = 0)\right)^n$. Since $\rho(r = 0)$ must equal ρ_c, $\theta_n(\xi = 0)$ must be unity, giving the first boundary condition as $\theta_n(0) = 1$.

The second boundary condition is provided by considering the pressure gradient $\frac{dP(r)}{dr}$ at $r = 0$. From hydrostatic balance this must be zero since $M(r = 0) = 0$ (there is no point mass at the centre to provide any gravitational acceleration), so

$$\begin{aligned} \tfrac{dP(r=0)}{dr} &= K\left(1+\tfrac{1}{n}\right)\rho^{(\frac{1}{n})}(r)\tfrac{d\rho(r)}{dr} \\ &= \tfrac{K(n+1)}{R}\rho_c^{(1+\frac{1}{n})}\left(\theta_n(\xi=0)\right)^n \tfrac{d\theta_n(\xi=0)}{d\xi} \\ &= 0 \end{aligned} \tag{4.134}$$

giving the second boundary condition as $\frac{d\theta_n(0)}{d\xi} = 0$.

Only three analytic Lane-Emden functions can be determined, corresponding to $n = 0$, 1 and 5; other solutions must be determined by numerical integration.

For $0 \leq n < 5$ (equivalent to $\eta > \frac{6}{5}$) the Lane-Emden functions $\theta_n(\xi)$ fall to zero at some finite value of $\xi = \xi_n$ (so $\theta_n(\xi_n) = 0$), which corresponds to the surface of the polytrope. Having a finite extent these are termed to be hard polytropes.

For $n \geq 5$ and $n < -1$ (equivalent to $0 < \eta \leq \frac{6}{5}$) the Lane-Emden functions $\theta_n(\xi)$ do not fall to zero at finite ξ (effectively $\xi_n = \infty$). Having an infinite extent they are termed to be soft polytropes. This includes the $n = \infty$ solution (equivalent to $\eta = 1$) which corresponds to the isothermal equation of state.

For $-1 \leq n < 0$ (equivalent to $\eta \leq 0$) pressure decreases as density increases, so there can be no hydrostatic balance and therefore no Lane-Emden functions exist. Any such objects could be termed very soft polytropes.

4.6.4 Polytrope Properties

Considering a complete polytrope as a sphere of gas with some constant value of the polytropic constant K and the polytropic index n, its surface (where the density $\rho(r)$ falls to zero) will be located at $r = R_P = \xi_n R$.

$$R_P(\rho_c, K, n) = \xi_n \left(\frac{K(n+1)}{4\pi G}\rho_c^{(\frac{1}{n}-1)}\right)^{1/2} \tag{4.135}$$

The radius R_P of a polytrope therefore scales with the central density as $R_P \sim \rho_c^{\left(\frac{1-n}{2n}\right)}$. This indicates that the $n = 1$ solution is a special case, since the radius R_P is independent of the central density ρ_c and is simply a function of the polytropic constant K. An $n = 1$ polytrope is a plausible model for a neutron star.

The mass M_P of a polytrope is

$$\begin{aligned} M_P(\rho_c, K, n) &= \int_0^{R_P} \rho(r) 4\pi r^2 dr \\ &= \int_0^{\xi_n} \rho_c \left(\theta_n(\xi)\right)^n 4\pi R^3 \xi^2 d\xi \\ &= 4\pi R^3 \rho_c \int_0^{\xi_n} \left(\theta_n(\xi)\right)^n \xi^2 d\xi \end{aligned} \tag{4.136}$$

Rearranging the Lane-Emden equation gives

$$(\theta_n(\xi))^n \xi^2 = -\frac{d}{d\xi}\left(\xi^2 \frac{d\theta_n(\xi)}{d\xi}\right) \tag{4.137}$$

which can be substituted to give

$$\begin{aligned} M_P(\rho_c, K, n) &= 4\pi R^3 \rho_c \left[-\xi^2 \tfrac{d\theta_n(\xi)}{d\xi}\right]_0^{\xi_n} \\ &= 4\pi \left(\tfrac{K(n+1)}{4\pi G}\right)^{3/2} \rho_c^{\left(\frac{3-n}{2n}\right)} \left(-\xi_n^2 \tfrac{d\theta_n(\xi_n)}{d\xi}\right) \end{aligned} \tag{4.138}$$

The mass M_P of a polytrope therefore scales with the central density as $M_P \sim \rho_c^{\left(\frac{3-n}{2n}\right)}$. This indicates that the $n = 3$ solution is a special case, since the mass M_P is independent of the central density ρ_c and is simply a function of the polytropic constant K. An $n = 3$ polytrope is a plausible model for a main sequence star, based upon the Eddington model of stellar structure.

The $n = 5$ solution can also be noted as special, because although it has infinite extent it has finite mass.

Eliminating the central density ρ_c gives the scaling relation $R_P \sim M_P^{\left(\frac{1-n}{3-n}\right)}$, which indicates that for $1 < n < 3$ the radius of a polytrope decreases as its mass increases. Within this range an $n = \frac{3}{2}$ polytrope is a plausible model for an object supported by non-relativistic electron degeneracy pressure, such as the core of a red giant star, a white dwarf star, a brown dwarf, a gas giant planet or even a rocky planet.

4.6.5 Numerically Solving the Lane-Emden Equation

For a given value of the polytropic index n the Lane-Emden function $\theta_n(\xi)$ and the boundary radius ξ_n are uniquely determined, so a family of physical solutions will be described by relations between the polytropic exponent K, the central density ρ_c and the radius R_P and mass M_P of the polytrope. When any two of these properties are specified the remaining two properties are then determined from them, and the dimensionless Lane-Emden function $\theta_n(\xi)$ can be converted into a physical radial density profile $\rho(r)$.

The Lane-Emden equation

$$\frac{1}{\xi^2}\frac{d}{d\xi}\left(\xi^2 \frac{d\theta_n(\xi)}{d\xi}\right) + (\theta_n(\xi))^n = 0 \tag{4.139}$$

can be rearranged to

$$\frac{d}{d\xi}\left(\xi^2 \frac{d\theta_n(\xi)}{d\xi}\right) = -\xi^2 (\theta_n(\xi))^n \tag{4.140}$$

which, differentiating the left hand side, gives

$$\xi^2 \frac{d^2\theta_n(\xi)}{d\xi^2} + 2\xi \frac{d\theta_n(\xi)}{d\xi} = -\xi^2 (\theta_n(\xi))^n \tag{4.141}$$

which can be rearranged to

$$\frac{d^2\theta_n(\xi)}{d\xi^2} = -\left(\frac{2}{\xi}\frac{d\theta_n(\xi)}{d\xi} + (\theta_n(\xi))^n\right) \tag{4.142}$$

A numerical integration scheme can be used to solve this, which is implemented for each discrete integration step i as

$$\begin{aligned} \tfrac{d\theta_n}{d\xi}(i+1) &= \tfrac{d\theta_n}{d\xi}(i) + \tfrac{d^2\theta_n}{d\xi^2}(i)\Delta\xi \\ &= \tfrac{d\theta_n}{d\xi}(i) - \left(\tfrac{2}{\xi(i)}\tfrac{d\theta_n}{d\xi}(i) + (\theta_n(\xi(i)))^n\right)\Delta\xi \end{aligned} \tag{4.143}$$

$$\theta_n(i+1) = \theta_n(i) + \frac{d\theta_n}{d\xi}(i+1)\Delta\xi \tag{4.144}$$

$$\xi(i+1) = \xi(i) + \Delta\xi \tag{4.145}$$

The boundary conditions $\theta_n(0) = 1$ and $\frac{d\theta_n(0)}{d\xi} = 0$ are applied at the start of the integration ($i = 0$). Note that although $\xi(0) = 0$ it should be approximated by a small value $\epsilon \sim \frac{\Delta\xi}{10}$ in order to prevent division by zero when evaluating the $\frac{2}{\xi}$ term in the second derivative.

Stepping forward in dimensionless radius by $\Delta\xi$ with each integration step, the solution θ_n is thus determined as a function of ξ. For a hard polytrope, when $\theta_n \leq 0$ the solution is complete and the integration can cease, noting that the value of ξ at this point gives the boundary surface ξ_n.

Given a value of n (or $\eta = 1 + \frac{1}{n}$) the Lane-Emden function $\theta_n(\xi)$ can therefore be numerically determined for $\epsilon(\approx 0) \leq \xi \leq \xi_n$, which provides a value for ξ_n.

The relations between the polytropic exponent K, the central density ρ_c and the radius R_P and mass M_P of the polytrope can then be used to determine the two unspecified properties from the two specified ones.

$$R_P(K, \rho_c) = \xi_n\left(\frac{K(n+1)}{4\pi G}\rho_c^{(\frac{1}{n}-1)}\right)^{1/2} \tag{4.146}$$

$$M_P(K, \rho_c) = 4\pi\left(\frac{K(n+1)}{4\pi G}\right)^{3/2}\rho_c^{(\frac{3-n}{2n})}\left(-\xi_n^2\frac{d\theta_n(\xi_n)}{d\xi}\right) \tag{4.147}$$

For an SPH simulation it is typical for R_P and M_P to be specified via the initial conditions. These values can then be used to determine the polytropic constant K required for the equation of state,

$$K = \left(\frac{4\pi G}{n+1}\right)\left(\frac{R_P}{\xi_n}\right)^{(\frac{3-n}{n})}\left(-\frac{4\pi\xi_n^2\frac{d\theta_n(\xi_n)}{d\xi}}{M_P}\right)^{(\frac{1-n}{n})} \tag{4.148}$$

which then determines the central density,

$$\rho_c = \left(\frac{K(n+1)}{4\pi G}\right)^{(\frac{-n}{1-n})}\left(\frac{R_P}{\xi_n}\right)^{(\frac{2n}{1-n})} \tag{4.149}$$

Note that when applying this expression using normalised code units it is important to remember that G is normalised to unity.

4.6.6 SPH Simulation of a Polytrope

Now that the Lane-Emden equation can be numerically solved and the resulting radial density profile of a polytrope can be determined, a polytropic equation of state ($P = K\rho^{\eta}$) can be applied to a gas sphere composed of SPH particles which are gradually relaxed via damping. The resultant settled profile can then be compared to the analytic solution (Price & Monaghan 2007).

It is possible to settle an initially uniform density sphere, but since this is far from the solution a relatively long period of time is required for the large-scale oscillations to damp down. In order to speed up the process the initially uniform density sphere can be transformed to the analytic density profile, so that only the damping of oscillations due to particle noise is required.

The numerical solution of the Lane-Emden equation gives θ as a function of dimensionless radius ξ at each integration step i. These values can be used to calculate the mass M_{poly} contained within the radius ξ for the polytrope as

$$M_{poly}(i+1) \;=\; M_{poly}(i) + \theta(i)\left(4\pi\xi^2(i)(\xi(i+1)-\xi(i))\right) \qquad (4.150)$$

beginning at the centre where $M_{poly}(0) = 0$ and integrating outwards, reaching $M_{poly} = M_P$ at the point when $\xi = \xi_n$ (which is equivalent to $r = R_P$).

For a uniform density sphere of mass M_S and radius R_S, the mass contained within a radius r is simply $M_{UDS} = M_S\left(\frac{r}{R_S}\right)^3$. To transform a uniform density sphere to a polytropic density profile, the position r of each particle must be adjusted to a new position $r_{new} = Fr$ such that the mass interior to it remains the same. This requires finding the value of ξ at which $\frac{M_{poly}(\xi)}{M_P} = \frac{M_{UDS}(r)}{M_S}$ for the radial position r of each particle (or in practice determining the closest possible value due to the finite intervals of numerical integration). This value of ξ for each value of r then provides the scaling factor $F = \frac{\xi R_S}{r R_P}$ for each particle, and the particles can be repositioned accordingly; their smoothing lengths h can also be rescaled by the same factor.

Using periodic boundary conditions, particles can be settled into a uniform density "glass" configuration. A uniform density sphere can then be cut from this glass box by keeping particles within a radius R_S and removing those outside. This uniform density sphere can then be transformed to the polytropic density profile and allowed to settle further under the action of self-gravitating SPH with the inclusion of artificial viscosity or some other damping mechanism.

Figure 4.12 shows the resulting radial density profile. The polytropic exponent is 5/3, the mass of the sphere is fixed as 1 and the radius is chosen to be 1, which gives a polytropic constant of $K = 0.424$ and a central density of $\rho_c = 1.43$. The slight gaps at the centre and edge of the sphere are due to hydrodynamic smoothing, and their size is comparable to the smoothing length at each point.

Although there are deviations from the analytic solution, with the central density being too low and the edge density too high, this can be explained by hydrodynamic smoothing at low resolution; with only 1062 particles in the sphere its radius is represented by only a few particles in any given direction.

Optimisation of the code to improve its performance will allow this test to be repeated at higher resolution in the next chapter (see Section 5.5); nonetheless it

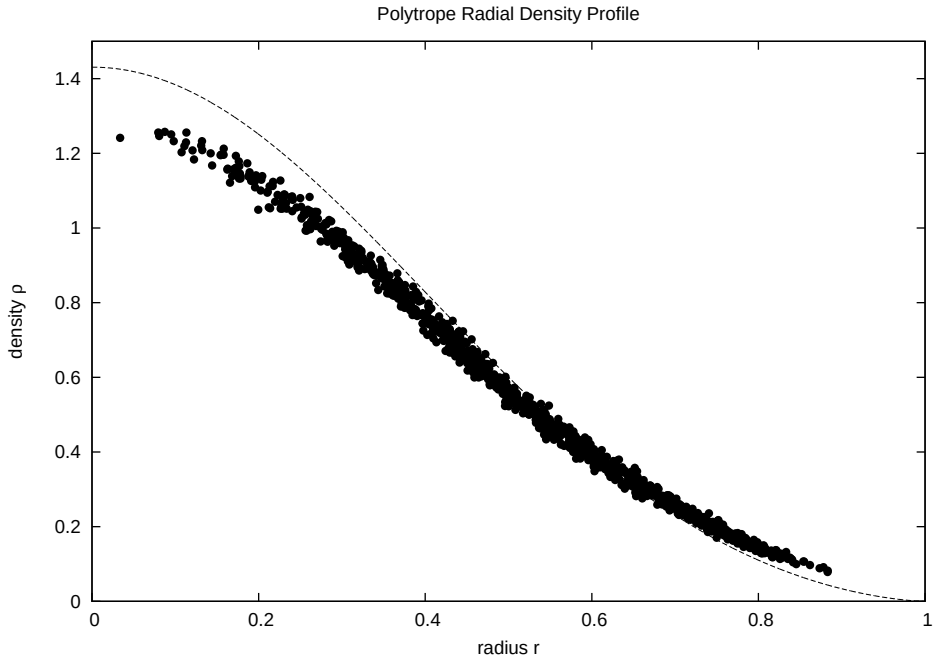


Figure 4.12: Relaxation of a transformed uniform density sphere (composed of 1062 particles) to a polytrope with exponent 5/3. The figure shows the resulting radial density profile. The dashed line indicates the analytic solution.

remains a useful preliminary test at this developmental stage, even at low resolution, since the basic underlying physical behaviour is still exhibited.

Polytrope Oscillations

It is possible to use the radial modes of oscillation of a polytrope in order to test the ζ correction terms of the energy-conserving formulation of SPH (Price & Monaghan 2007). These modes can be imposed via a radial velocity perturbation (such as $v_r = 0.2r$) but the remnant noise after the settling process is sufficient for our purpose. To this end the settled sphere shown in Figure 4.12 can be used, with any artificial viscosity or damping of the oscillations now removed.

It is also beneficial to use the standard M4 kernel function without the Thomas & Couchman modification (since the modification impinges slightly upon energy conservation) and to use a low Courant number (such as a factor of 0.05 in the timestepping condition) in order to minimise the errors due to the integration scheme itself.

Figure 4.13 shows the resulting variation in total energy, and it can be seen that including the ζ correction terms greatly improves the conservation of energy. The same essential behaviour is observed with a higher Courant number or the use of the Thomas & Couchman modification but the amplitude of the noise is increased.

Further similar tests that might be performed at this stage include models of 'Toy Stars' (Price 2004) or acoustic oscillations of an isentropic monatomic gas sphere (Attwood 2008).

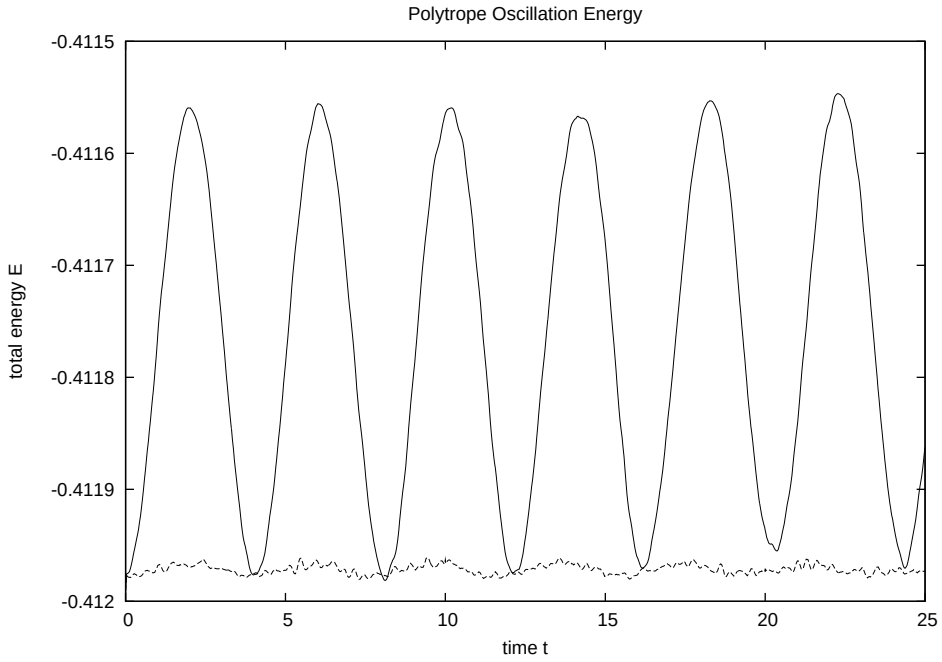


Figure 4.13: Oscillation of a polytrope (composed of 1062 particles) with exponent 5/3. The upper line shows the variation in total energy without the ζ correction terms and the lower line with the ζ correction terms included.

4.7 Summary

Smoothed particle hydrodynamics (SPH) has been introduced as a Lagrangian particle-based method for discretising the fluid dynamics equations and, due to its spatially-varying resolution length being inversely proportional to the cube root of the density (in 3 dimensions), is well-suited to problems involving gravitational collapse.

The M4 kernel function is deemed to be suitable for evaluating first-order derivatives, but noise from the particle distribution makes second-order derivatives unreliable.

We have extended the N-body computer program to incorporate SPH, which has served to highlight further issues involved in code development. Correction terms to account for variable smoothing and softening lengths have been included, and the Thomas & Couchman kernel modification has been introduced.

Now that hydrodynamics has been satisfactorily modelled in combination with gravitational dynamics, demonstrating the accuracy of the numerical method (within the limits of low spatial resolution), we can turn our attention towards improving its speed and efficiency

In the next chapter we will optimise our SPH code in order to allow high resolution simulations to be performed within a reasonable duration, including the Boss & Bodenheimer test.

"As fast as laws are devised,
their evasion is contrived."

German proverb

Chapter 5

Optimisation

In this chapter we describe the method used to optimise the performance of a self-gravitating hydrodynamics code. This allows us to further our aim of modelling self-gravitating gas by achieving higher resolution without considerable cost to efficiency.

In particular, three key developments are introduced and developed: the tree algorithm, block timestepping and sink particles. Their effect upon both efficiency and accuracy is demonstrated using simple performance tests, and higher resolution simulations of tests from the previous chapter are also performed. The Boss & Bodenheimer test is used to demonstrate the sink routines.

Parallelisation is also discussed, allowing the code to run on multiple processors.

5.1 Optimisation

Optimisation is an important process for any complex computational procedure, and various criteria are used to ensure efficiency since an acceptable level of accuracy must be achieved in a reasonable amount of time. When simulation run times extend from hours to days or even weeks or more, code optimisation rapidly becomes a significant issue.

The classic computational trade-offs to consider are accuracy-speed (sacrificing small amounts of accuracy for large gains in speed) and accuracy-memory (sacrificing accuracy or precision for less memory usage, which also improves speed by reducing cache misses).

Depending on the nature of the code, memory usage may or may not be a critical factor; to be most efficient, all relevant variables must simultaneously fit into the memory cache of the computer.

The accuracy of a numerical solution can be measured by its proximity to the analytic solution. The speed can be gauged by the time taken to run the simulation, which is determined by the number of computational operations required. The memory usage can be gauged roughly by inspection (by counting the number of variables used and multiplying by 4 bytes each (or 8 for double precision), or by using the `top` command) or in more detail through the use of profiling tools.

Speed and memory usage are obviously factors of N (the number of particles in the simulation). For high-resolution simulations the total memory (RAM) available

on a computer system will provide an absolute limit to N. Beyond this limit, virtual memory (hard disk space) might be used in place of RAM, but the access speed (reading and writing) is orders of magnitude slower and will cripple the speed of a simulation, so this is to be avoided at all costs.

Between the main memory (RAM) of a computer and the registers of the processor (where data is actually manipulated) there is a layer of low-latency memory known as the cache. If the data to be manipulated can be retrieved from the cache then access times are fast. If the data required is not present in the cache then a "cache miss" occurs and the data must be retrieved from main memory. This typically involves replacing a line of data in the cache with a line from the main memory.

This explains why our data structures were designed to store the data required for gravitational calculations (r, m and h) contiguously in memory, to minimise cache misses in the most frequently called subroutine.

Some optimisation issues have already been addressed, such as the use of compiler flags and macros, and memory management issues such as optimal data structures. In the same vein the kernel tables should be kept sufficiently small that they can be accommodated in the cache.

It is worth noting that modern processors often have multi-level caches structured in a hierarchy with level 1 (L1) being the smallest fastest cache, the next largest being L2 and so on until main memory is reached. Cache misses then result in a search upwards through the hierarchy until the required data is found. As a rule of thumb, typical sizes are kilobytes for the L1 cache (64KB being standard) and megabytes for the L2 upwards (dependent on the chip architecture), with gigabytes usually available in main memory.

5.2 Optimisation Methodology

There are four main avenues of optimisation, which tend to be best approached in the order listed below.

5.2.1 Compiler Optimisation

The first (and simplest) approach is to use the optimisation flags associated with the compiler, typically `-OX` (where `X` is a level of optimisation from 1 to 3 or more, depending upon the compiler). These extend the compilation time but can dramatically speed up the run time by making sure that an efficient machine code is compiled.

The typical convention is for `O1` to optimise for binary size (compiling to the least amount of machine code), `O2` to optimise for speed (usually at the cost of more machine code) and `O3` to aggressively optimise for speed (by unrolling loops, replicating code to eliminate branches, pre-fetching data and so on).

The `O3` option is of particular benefit to loop-intensive codes using many floating-point operations and processing large data sets (Intel 2006), so it should be the best choice for SPH or N-body simulations.

Some other flags such as `-fast` might be appropriate, typically sacrificing mathematical precision for greater speed.

5.2.2 Profiling and Tuning

The next level of improvement is achieved by generating a profile during a run (first compiling with the appropriate `-pg` flag and then running) and then viewing it using a program such as `gprof`. A profile gives a breakdown of how much time is spent in each subroutine, and even on each line of code. This allows the most expensive areas of code to be fine-tuned on a line-by-line basis to enhance speed by ensuring that the most efficient program structures and routines are used.

5.2.3 Algorithmic Replacement

Choosing the best algorithm can be very important. In the numerical integration of functions ($\int f(x)dx$) the rectangle rule has errors that scale as order h (where h is the width of the elements being summed), the trapezium rule errors scale as order h^2 and the Simpson rule errors scale as order h^4. This means faster convergence can be achieved with the latter methods, giving the same result for less iterations, which improves the efficiency. Such considerations guided our choice of time integration scheme in Chapter 3. By accepting small losses in accuracy for large gains in speed this principle can be taken further: an N-body gravitational calculation scales as N^2 but the use of a tree code can reduce this to an $N \log N$ process.

5.2.4 Parallel Computation

Parallelisation is usually a major undertaking and thus the final step in improving performance. It achieves greater speed by spreading the computational work over multiple processors. There is ultimately an overhead in balancing the cross-communication required between processors with the gains from spreading the workload, so more processors are not inevitably better: eventually the communication costs may eat into the computation process too heavily.

5.2.5 Notes on Application

Compiler optimisation, while almost trivial, is greatly assisted by having a well-written and stream-lined code in the first place, since the compiler can only do so much optimisation within its finite compilation time.

The profiling and tuning process will reveal the specific bottlenecks of any given code, but common to all self-gravitating hydrodynamics codes is the intense cost of the gravitational calculations, particularly for high N simulations where it dwarfs every other process. As a constituent of this, the subroutine dedicated to calculating relative position vectors between two particles has the highest profile, and is therefore the prime candidate for optimisation; it should be as stream-lined as possible, since even the slightest deviation from optimal construction will be multiplied dramatically at run-time. This is precisely why dimensionality should be handled with a macro and periodic boundaries using compiler flags, to ensure that they never add unnecessary costs here. It is recommended that the displacement vector and the square of the

separation be returned by this subroutine to maximise efficiency. This prevents the terrible cost of using an expensive square-root operation to calculate the magnitude of separation within the subroutine, only to square it again immediately afterwards in the calling routine. In high-usage routines such as these, extreme care is required in their construction; keeping code as short and simple as possible, the general rule for any subroutine, should be followed most rigorously here.

Algorithmic replacements that can alleviate costs are the tree code (reducing the calculation to $N \log N$), block timestepping (replacing the global timestep with local timesteps prevents the central dense region of a collapse process from slowing down the entire domain) and sink particles (replacing a high density region with a sink particle prevents it from slowing beyond a certain threshold). These three techniques will be introduced and developed in the remainder of this chapter, followed by a brief discussion of parallelisation.

5.3 Developing a Tree Code

A tree code decomposes the computational domain by dividing it into cells, and recursively subdividing these into smaller cells, repeating until each cell contains less than some specified number of particles within it (such a cell is termed a leaf cell).

There are several ways in which the domain can be decomposed and one approach is octal-spatial decomposition, as typified by the Barnes-Hut tree (Barnes & Hut 1986). This divides the 3-dimensional computational domain into eight cubic sub-cells of equal volume and recursively divides these sub-cells down to the leaf cells. A computational domain of uniform density will result in a highly symmetric tree structure but in general it may be highly asymmetric, with leaf cells rapidly reached in low density regions but deeply nested in high density regions. Rather than considering one particle at a time it is more efficient to decompose them as an ensemble (Pfalzner & Gibbon 1996) since this approach can easily be parallelised later.

The tree structure allows particle-particle gravitational interactions to be replaced at long distances by particle-cell interactions, where the cell is treated as if it was a single particle with the combined mass of all the particles contained within it located at its centre of mass. This approach reduces the N^2 gravitational calculation to a more efficient $N \log N$ process.

Determining whether a particle is distant enough to interact with a cell, or whether accuracy demands that it interact with the sub-cells or, in the case of a leaf cell, constituent particles, requires the tree structure to be walked with a suitable criterion applied at each stage (see Section 5.3.10). When this criterion is satisfied, such as a cell subtending a small enough angle with respect to the particle, the tree structure can be walked horizontally, moving from cell to cell on the same level. When not satisfied a cell must be opened, and the tree structure walked vertically down into the next level which contains the child cells, returning to the parent level and the next cell only after all of these subcells have been walked (see Section 5.3.9).

The tree structure can be duplicated and a similar walk performed in order to rapidly obtain neighbour lists for SPH interactions. The reason for using two

separate tree structures, one for gravity and one for hydrodynamics, is that different cell information is required in each case: separating this information out makes the structure more efficient to process for each case by minimising cache misses.

Since the timestep stability conditions limit how far particles can move in a single timestep, the tree structure will not change significantly from timestep to timestep, so there is no need to build the tree structure every single timestep. Instead it can be built every ~ 10 timesteps, although the properties of each cell still need to be calculated every timestep.

5.3.1 Data Structures

Tree data must now be incorporated in addition to the particle data. The tree structure is basically a linked list of cells. Useful global variables for managing the tree structure include the total number of cells, the total number of levels and the identifiers of the first and last cells on each level.

Each cell will possess two links: one to the next cell on its level and one to its first child cell. These will be assigned when the tree is built and then used when the tree is walked. Any cell that is a leaf cell will also need to store the number of particles that it contains, together with a list of their identifiers.

The cells in the gravity tree will need to be stocked with their centre-of-mass positions, the total mass they contain, a measure of their size (used to determine the angle they subtend, for the opening criterion) and possibly any multipole moment terms and multipole acceptance criterion. The size is best stored as a length squared, in order to avoid expensive square root operations.

The cells in the hydrodynamic tree, used for SPH neighbour finding, will need to be stocked with their centre-of-volume positions, a measure of their size (used to determine the bounding box) and the maximum smoothing length contained (used to extend the bounding box for scatter operations).

5.3.2 Compiler Flags

The use of the tree is best implemented with a compiler flag. This allows simulations to be performed without the tree code using direct summation, which is useful for comparison purposes (to determine the effect of the tree upon accuracy) or for low-N simulations where its use is not warranted. It also allows different types of tree to be incorporated if desired, such as a binary tree.

The inclusion of multipole moments (quadrupole or octupole) is best done with compiler flags. This way they can be included when their contribution to accuracy proves useful, or excluded to conserve memory usage in high-N simulations. The possibility of using different types of multipole acceptance criteria is also best handled with compiler flags.

It is often best to obtain lists of SPH neighbours on the fly, since they are only needed to calculate smoothing lengths or hydrodynamic accelerations. If memory usage is not at a premium, however, it can sometimes be worthwhile storing them to save any repeated tree walks. Such an option is best included using a compiler flag.

A macro is recommended for the maximum number of particles allowed within a leaf cell. It may also be useful to have a macro for the maximum number of cells in the tree, or the maximum number of levels in the tree (which in turn constrains the maximum number of cells, since an octal tree can contain at most $1 + \sum_{i=1}^{L} 8^i$ cells, where L is the number of levels beyond the root cell, which can itself be considered to be the zeroth level). Such a macro can be used to curtail excessively large tree structures if memory usage is at a premium.

5.3.3 The Main Program

Using a top-down approach the program still consists of three main parts as before: input, integration and output.

In the initial input part we again input any model parameters and the initial conditions, and perform other preparations such as allocating memory for data structures (including the new tree data), converting to code units and initialising variables (including building and stocking the tree based upon the initial conditions). Note that since the hydrodynamic tree contains information that is dependent upon the smoothing lengths h, it is useful to calculate the smoothing lengths here and then stock the tree again to ensure the inclusion of this information.

In the integration part, where the calculations are performed through the application of a numerical method in algorithmic form, we will need to make modifications in order to incorporate the tree structure. This essentially consists of three components: building the tree, stocking the tree and walking the tree.

5.3.4 Parameters

The frequency with which the tree structure is built can be handled as a parameter, although a macro would be reasonable. Its value, typically a few steps ($\sim$ 5 full timesteps), is unlikely to need changing, but a lower number of steps might be more suitable for violent turbulent simulations and a higher number of steps might be suitable for quiet well-ordered simulations. Ultimately the frequency of tree-building should account for the rate at which the geometry of the particle distribution might change.

The criterion determining whether a cell is opened when the gravity tree is walked is best handled as a parameter. For the geometric criterion this is described by an opening angle. Increasing its size will benefit speed at the cost of accuracy by allowing a greater number of cell-cell interactions. The value should be chosen to satisfy the speed-accuracy requirements of the simulation. Some multipole acceptance criteria may need a different parameter, typically the absolute size of an error term.

5.3.5 Memory Allocation

Memory for the data structures to store the tree information will need to be allocated. If the tree structure is asymmetric then this is more difficult than particle data, where the number of particles N precisely determines the size of the arrays required, because

the number of cells required is dependent not just on the number of particles but also the geometry of their distribution. In general, allocating memory for $2N$ cells is sufficient. Since the majority of cells will be leaf cells, the macro determining how many particles a leaf cell can contain has significant influence on the size of the tree and thus the cell memory requirements. The conservative choice of $2N$ can usually be reduced when leaf cells are able to contain large numbers of particles (typically > 8).

5.3.6 Integration

The integration routine within the main program loop contains the same core processing elements as before, with some new tree code elements now incorporated; these additions have been marked ($\star$) and *highlighted* to stand out from the previous pseudocode.

Pseudocode (integration):

```
calculate total energy and momentum of system [optional diagnostic]
IF adaptive timestep is used THEN
      calculate new timestep Δt
ENDIF
FOR each particle
      apply first part of integration scheme
      (predict position)
ENDFOR
⋆ IF tree needs building [every few steps as specified by parameter]
⋆       build tree (using predicted positions)
⋆ ENDIF
⋆ stock tree (using predicted positions)
FOR each particle
      calculate smoothing length and density [and any correction terms]
⋆       by walking the tree
ENDFOR
FOR each particle
      calculate pressure using the equation of state
ENDFOR
calculate gravitational accelerations [and potentials]
calculate hydrodynamic accelerations
⋆ by walking the tree
FOR each particle
      apply second part of integration scheme
      (advance velocity and correct position)
ENDFOR
advance time:  t = t + Δt
```

This description applies to the modified Leapfrog scheme. For the Runge-Kutta scheme the tree should be stocked a second time for the additional acceleration

calculation.

5.3.7 Building the Tree

A new tree structure need not be built every integration step, but every few instead, as determined by a parameter.

If all particles are self-gravitating gas particles then one tree structure can be built for the gravity tree and simply duplicated for the hydrodynamic tree. If special particle types are later incorporated into the code, such as non-gravitational gas particles (to represent the intercloud medium for example), then the two trees will have to be built separately, with the gravity tree including only gravitationally interacting particles and the hydrodynamics tree including only hydrodynamically interacting particles.

For particle data to be optimally arranged in memory in a contiguous manner (to minimise cache misses) the ordering should follow that of the tree walking process for the gravity tree. Although this requires additional code dedicated to the rearrangement of particle data it is a worthwhile optimisation.

The routine to build a tree structure itself will be quite complicated, involving a sizeable amount of auxiliary variables (local to the routine itself), but the overall process is summarised below.

Psuedocode (build tree):

```
begin at level 0 (root level of tree)
determine cubic bounding box enclosing all particles in computational domain
set cell size to be length of side of this cubic cell
set position of this root cell to be at its centre
REPEAT
      advance to next level of tree
      halve cell size (to get subcell size for this level of tree)
      FOR all cells on parent level (from C_first to C_last)
            IF cell is a leaf cell THEN
                  skip to next cell
            ENDIF
            FOR each particle (P_list) contained in cell
                  determine which subcell it belongs to
            ENDFOR P
      ENDFOR C
      create child cells for each occupied subcell
      assign positions of these child cells (at their centres)
      set up all linked lists between parent and child cells
      store list of particles (P_list) contained in each cell
      record first and last cell identifiers (C_first and C_last) of level
UNTIL all particles are in leaf cells
      (i.e.  all cells on final level are leaf cells)
reorder cells so that tree walks are contiguous in memory
```

At the end of the building process each cell will contain a link to the next cell on that level (for when it is not opened) and a link to its first child cell (for when it is opened), together with its position (the centre of volume). Any leaf cells will also contain the list of particle identifiers (indicating which particles they contain) together with the size of this list (the total number of particles contained).

This information is essentially common to both the gravity and hydrodynamic trees, but will be modified and added to during the stocking process.

5.3.8 Stocking the Tree

The tree structure should be stocked on every acceleration step to ensure that correct values are being used.

Stocking the tree must be performed from the bottom upwards, one level at a time, beginning with the leaf cells of the final level. The properties of a leaf cell are determined from the particles that it contains, and the properties of all other cells are determined by those of their child cells, all the way up to the root cell which contains the entire computational domain.

For the gravity tree each cell must be stocked with the total mass it contains and the centre of mass assigned as its position. Additionally any multipole information (such as the elements of the quadrupole moment tensor) together with any multipole acceptance criterion (such as the opening angle) will be stocked here.

For the hydrodynamic tree each cell must be stocked with the position and size of its bounding box for the purpose of gathering neighbours. The bounding box is the cuboid volume element that is just big enough to contain all the particles within the cell. Its position is given by the centre of this volume element and its size is given by the diagonal length from the centre to a corner. Additionally once particles have had their smoothing lengths calculated, the maximum smoothing length in each cell will also be stocked here; while not necessary for calculating the smoothing lengths using a gather formulation, this information is necessary to extend the range of the bounding box for calculations based upon a scatter formulation, such as hydrodynamic accelerations.

5.3.9 Walking the Tree

Walking the tree must be performed from the top, beginning with the root cell and using the linked list to move from one cell to the next until the end of the list is reached, indicating that the entire tree has been walked.

This is the stage at which the original code must interface with the tree code in order to reap its benefits.

Walking the Hydrodynamic Tree

For the hydrodynamic tree each cell is checked to determine if its bounding box overlaps the smoothing volume of the particle that is walking the tree.

This bounding box may be determined by the constituent particle positions alone (when using a gather formulation to determine smoothing length) or extended

to include their own smoothing volumes (when using a scatter formulation, such as when calculating hydrodynamic accelerations).

If there is no overlap then the cell cannot contain any potential SPH neighbours and so deeper searching can be ignored; the link to the next cell on that level is then followed in the walk through the linked list.

If there is an overlap then the cell is opened by following the link to the first child cell instead. In the case of a leaf cell any SPH neighbours contained within it can then be found directly on a particle-particle basis for each member of that cell, before the walk continues to the next cell.

The last child of each parent cell always links to the next cell on the level of their parent (the same as the link that would have been followed if the parent had not been opened), and the last child of the root cell points to a marker indicating the end of the list. The linked list therefore ensures that the entire tree structure is walked, searching to the necessary depth for each cell to find all SPH neighbours. An N^2 particle-particle search is reduced to a faster $N \log N$ search through this particle-cell approach.

When calculating smoothing lengths, the gathering of neighbours in the SPH density summation can be performed by walking the hydrodynamic tree.

A useful optimisation is to sort the list of SPH neighbours obtained from a hydrodynamic tree walk into a reverse order (starting with the furthest and ending with the closest). This optimises the accuracy of the summation by ensuring that smaller contributions are added before bigger ones, minimising rounding errors. Since a neighbour list typically consists of < 100 integer particle identifiers, a simple injection-sort algorithm is usually sufficient. For more expensive sorting requirements a heap-sort algorithm is recommended, which is best suited to large and highly jumbled arrangements; this is actually a form of tree code itself and enjoys the same $N \log N$ performance.

When calculating hydrodynamic accelerations, the interacting particle pairs (mutual neighbours) for the SPH momentum equation can be found by walking the hydrodynamic tree using the extended bounding box.

It is important that the neighbour-finding method is symmetrised, ensuring that if j is a neighbour of i then i is also a neighbour of j (so that all forces are symmetric in accordance with Newton's third law of motion). This can be achieved via two different gather/scatter formulations: mean smoothing length ($\bar{h}$) or mean kernel ($\bar{W}$). When walking the tree to find neighbours of particle i, a particle j is its neighbour if their separation $r_{ij} < h_i + h_j$ for the $\bar{h}$ method, or if either $r_{ij} < 2h_i$ or $r_{ij} < 2h_j$ for the $\bar{W}$ method. The method chosen should be consistent with the symmetrisation of the SPH momentum and energy equations, particularly with respect to dissipative terms (see Chapter 6) and gravitational terms (see Section 4.4.13 in the previous chapter). In our case the $\bar{W}$ method is the appropriate choice.

Neighbours j of particle i can be found either by gather (where $|\mathbf{r}_{ij}|^2 \leq 4h_i^2$), scatter (where $|\mathbf{r}_{ij}|^2 \leq 4h_j^2$) or both (where $|\mathbf{r}_{ij}|^2 \leq 4h_i^2$ or $|\mathbf{r}_{ij}|^2 \leq 4h_j^2$). Note that comparing the square of the separation to the square of the smoothing radius avoids the need to perform expensive square root operations; cell sizes should therefore be calculated and stored in length-squared terms when the tree is stocked.

Although a list of all potential neighbours is returned from the tree walk, there

is no need to prune down to the true SPH neighbours. This is because those potential neighbours that are not true neighbours will simply fail to contribute anything to the SPH calculations because the kernel functions are zero beyond the interaction radius $2h$.

Walking The Gravity Tree

For the gravity tree each cell is checked with an opening criterion.

If the cell is not opened then the gravitational acceleration is determined by a particle-cell interaction, which assumes that all the mass of the cell is located at its centre of mass, although this can then be modified by multipole correction terms; the link to the next cell on that level is then followed in the walk through the linked list.

If the cell is opened then the link to the first child cell of that cell is followed instead. In the case of a leaf cell the gravitational accelerations are determined directly on a particle-particle basis for each member of that cell, before the walk continues to the next cell.

The linked list ensures that the entire tree structure is walked, progressing down to the necessary depth for each cell based upon the opening criterion. Adding up all the gravitational acceleration contributions along the way gives the net gravitational acceleration. Since the tree replaces the N^2 particle-particle summation with particle-cell interactions whenever the opening criterion allows it, the calculation is reduced to an $N \log N$ process.

When calculating gravitational accelerations, the entire process can be completed simply by walking the gravity tree for each particle in turn.

Note that, depending on the nature of the opening criterion, it might be necessary to force a particle to always open its own cell; any possibility of a particle-cell interaction with its own cell must obviously be eliminated entirely.

Since gravitational softening is typically only applied to particle-particle interactions, the cell opening criterion should be chosen such that particle-cell interactions do not occur within the softening length of the particle. Only long-range interactions should involve cells, and the opening criterion must be conservative enough to ensure this.

The tree cannot use the mutual acceleration optimisation that was available under direct summation, where the N^2 calculation was reduced to $\frac{N^2}{2}$ by applying the result of the interaction between i and j to both i and j at the same time. This results in the tree being less efficient than direct summation for low N simulations, although the $N \log N$ performance wins out at higher N, typically $\gtrsim 2000$.

A useful optimisation for walking the tree is to walk it for groups of particles rather than individual particles. Since all the particles in the same leaf cell are likely to require an identical (or almost identical) walk of the tree, applying the distant particle-cell interactions to all particles in a leaf cell at the same time improves efficiency by eliminating the need for repeated calculations.

Figure 5.1 shows how the tree performs on a simple gravitational calculation such as the freefall collapse of a uniform density sphere. Each simulation was performed for 90% of the freefall time and the run time was then normalised by the number of integration steps required. The behaviour is as predicted (N^2 for direct

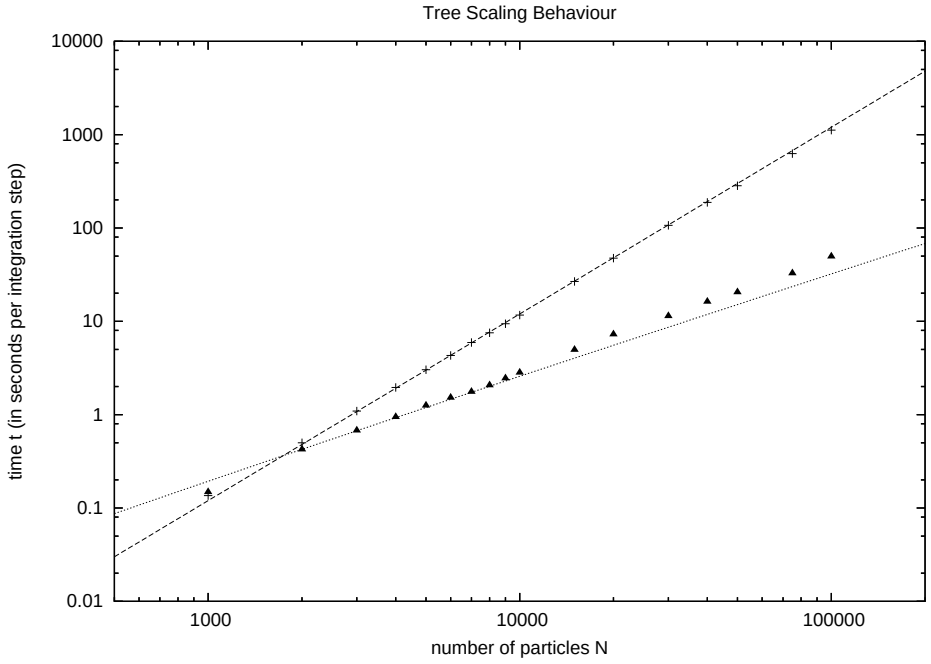


Figure 5.1: Comparison of performance (processing time per integration step) as a function of the number of particles N between direct summation (crosses) and the tree code (triangles). The lines indicate N^2 and $N \log N$ behaviour.

summation and $N \log N$ for the tree). The gradual deviation seen for the tree at higher N is essentially due to memory threshing when walking a tree that is larger than the cache size.

5.3.10 Multipole Moments

Higher-order multipole terms may be calculated (and stored in the gravity tree) to more accurately calculate the gravitational force of a particle-cell interaction. Multipole expansion can be made to any order in principle, but in practice it is optimal to truncate after only a few terms.

The monopole term is the centre-of-mass term for the cell, and the dipole term is always zero if calculated with respect to the centre-of-mass of the cell. The first non-zero correction terms are therefore the quadrupole moments. These are followed by the octupole moments, hexadecipole moments and so on.

Calculating the multipole moments in the centre-of-mass frame of the cell, the components of the quadrupole moment tensor Q for a leaf cell c in terms of the particles it contains are given by

$$Q_{ab,c} = \sum_i m_i \left(3 x_{a,i} x_{b,i} - r_i^2 \delta_{ab}\right) \tag{5.1}$$

where i is summed over all particles in the leaf cell. For a cell that is not a leaf cell, the quadrupole moment tensor can be expressed in terms of the properties of the

child cells.

$$Q_{ab,c} = \sum_d m_d \left(3x_{a,d}x_{b,d} - r_d^2\delta_{ab}\right) + \sum_d Q_{ab,d} \tag{5.2}$$

where d is summed over all child cells.

The octupole moment tensor S for a leaf cell c is given by

$$S_{ab,c} = \sum_i m_i \left[5\left(3-2\delta_{ab}\right)x_{a,i}^2 - 3r_i^2\right] x_{b,i} \tag{5.3}$$

$$S_{123} = 15\sum_i m_i x_{1,i}x_{2,i}x_{3,i} \tag{5.4}$$

where i is summed over all particles in the leaf cell. For non-leaf cells the octupole moment in terms of the child cells is given by

$$\begin{aligned} S_{ab,c} &= \sum_d m_i \left[5\left(3-2\delta_{ab}\right)x_{a,d}^2 - 3r_d^2\right] x_{b,d} \\ &+ \sum_d \left[5\left(1-\delta_{ab}\right)x_{a,d}Q_{ab,d} + \tfrac{5}{2}x_{b,d}Q_{aa,d} - x_{l,d}Q_{bl,d} + S_{ab,d}\right] \end{aligned} \tag{5.5}$$

$$\begin{aligned} S_{123} &= 15\sum_d m_i x_{1,d}x_{2,d}x_{3,d} \\ &+ \sum_d \left[\tfrac{5}{3}\left(x_{1,d}Q_{23,d} + x_{2,d}Q_{31,d} + x_{3,d}Q_{12,d}\right) + S_{123,d}\right] \end{aligned} \tag{5.6}$$

where d is summed over all child cells.

The quadrupole moment tensor Q is a traceless symmetric matrix so there are 5 independent terms that must be stored in memory for each cell. The octupole moment tensor S is a more complicated rank-3 tensor whose symmetries result in 10 independent terms that must be stored in memory.

The gravitational potential at the position of particle i due to a cell c, up to octupole order, is given by

$$\phi_{\text{grav,c}} = -\frac{GM_c}{|\mathbf{r}|} - \frac{Q_{ab,c}r_a r_b}{2|\mathbf{r}|^5} - \frac{S_{ab,c}r_a^2 r_b + S_{123,c}r_1 r_2 r_3}{2|\mathbf{r}|^7} \tag{5.7}$$

where $\mathbf{r}$ is the position of the particle relative to the cell, r_1, r_2 and r_3 are the components in each dimension and the Einstein summation convention is employed where repeated indices are summed.

The gravitational acceleration due to a cell c, up to octupole order, is given by

$$\begin{aligned} \left(\tfrac{d\mathbf{v}}{dt}\right)_{\text{grav,c}} &= -\tfrac{GM_c}{|\mathbf{r}|^3}\mathbf{r} + \tfrac{Q_{ab,c}r_a}{2|\mathbf{r}|^5}\mathbf{e}_b - \tfrac{5}{2}\tfrac{Q_{ab,c}r_a r_b}{2|\mathbf{r}|^7}\mathbf{r} \\ &+ \tfrac{S_{ab,c}r_a r_b}{|\mathbf{r}|^7}\mathbf{e}_a + \tfrac{S_{ab,c}r_a^2}{|\mathbf{r}|^7}\mathbf{e}_b - \tfrac{7S_{ab,c}r_a^2 r_b}{|\mathbf{r}|^9}\mathbf{r} - \tfrac{7S_{123,c}r_1 r_2 r_3}{2|\mathbf{r}|^9}\mathbf{r} \\ &+ \tfrac{S_{123,c}}{|\mathbf{r}|^7}\left(r_2 r_3\mathbf{e}_1 + r_3 r_1\mathbf{e}_2 + r_1 r_2\mathbf{e}_3\right) \end{aligned} \tag{5.8}$$

where $\mathbf{e}_a$ is the unit vector in the a^{th} direction.

The optimal expansion to use may be problem-dependent. The quadrupole moment expansion is typically the fastest for a given accuracy but there may be some cases, particularly where high accuracy is required, when expanding to octupole order is the optimal choice in terms of speed.

When walking the gravity tree the code must interrogate each cell to determine whether to use the multipole expansion of that cell or to open the cell and use the child cells instead. This is determined by a multipole acceptance criterion (MAC). The effectiveness of the multipole expansion is highly dependent on the MAC chosen for the simulation, and a variety of MACs are possible.

Geometric MAC

The geometric MAC is the original MAC used in the Barnes-Hut tree. It uses the geometric size l_c of the cell (its longest corner-to-corner length) and its distance r from the particle to calculate the angle that the cell subtends relative to the particle: $\theta = l_c/r$. If this angle is smaller than some tolerance parameter

$$\frac{l_c}{r} \leq \theta_{\mathrm{MAC}} \tag{5.9}$$

then the gravitational force due to the cell is given by the multipole expansion (Equation 5.8). If this criterion is not satisfied then the cell is opened and the sub-cells on the next level are interrogated. If a leaf cell is opened then direct particle-particle interactions occur so no multipole terms are needed. For computational efficiency, square root operations are avoided by formulating the criterion in terms of lengths squared:

$$r^2 \geq r^2_{\mathrm{MAC}} \equiv \frac{l_c^2}{\theta^2_{\mathrm{MAC}}} \tag{5.10}$$

GADGET MAC

It is possible for the geometric MAC to result in large force errors and non-conservation of energy, particularly if the opening angle is not sufficiently small ($\theta \leq \sqrt{1/3}$) (Hernquist, Hut & Makino 1993), so it would be better to use higher-order multipole error terms or to employ some other opening criterion to prevent any catastrophic errors when interrogating cells.

In this vein the GADGET SPH code uses a MAC that approximates the leading error term in the multipole expansion to calculate the closest particle-cell distance at which the multipole expansion can be used (Springel, Yoshida & White 2001). Since GADGET employs quadrupole moment corrections the leading error terms are the octupole moments, but since these are small in a homogeneous density field the hexadecipole moments are likely to be the largest error terms and are used instead.

For a cell c of total mass M_c and linear size l_c at a distance r from particle i, a crude approximation to the acceleration of i due to the hexadecipole term is

$$a_{\mathrm{hex}} \approx \frac{GM_c l_c^4}{r^6} \tag{5.11}$$

If a_{hex} is less than some pre-defined fraction of the total gravitational acceleration of the particle ($a_{\mathrm{hex}} < \alpha_{\mathrm{MAC}}|\mathbf{a}_{\mathrm{grav}}|$) then the multipole expansion is used, otherwise the cell is opened and its sub-cells are interrogated. Since the total acceleration is not known in advance (it is what is being calculated, after all) the total

acceleration from the previous timestep is used as an approximation. Using Equation 5.11 the criterion can be expressed in terms of lengths squared:

$$r^2 \geq r_{\mathrm{MAC}}^2 \equiv \left(\frac{G M_c l_c^4}{\alpha_{\mathrm{MAC}} |\mathbf{a}_{\mathrm{grav}}|} \right)^{1/3} \tag{5.12}$$

Storing the quantity $(G M_c l_c^4 / \alpha_{\mathrm{MAC}})^{1/3}$ in memory for each cell allows for rapid processing of the MAC for each particle.

If no quadrupole or octupole moment terms are used then the leading error term is the quadrupole moment term. In this case an approximation to the acceleration of the quadrupole term is applicable: $a_{\mathrm{quad}} = G M_c l_c^2 / r^4$. The criterion that the quadrupole moment acceleration be smaller than some fraction of the total acceleration ($a_{\mathrm{quad}} < \alpha_{\mathrm{MAC}} |\mathbf{a}_{\mathrm{grav}}|$) leads to

$$r^2 \geq r_{\mathrm{MAC}}^2 \equiv \left(\frac{G M_c l_c^2}{\alpha_{\mathrm{MAC}} |\mathbf{a}_{\mathrm{grav}}|} \right)^{1/2} \tag{5.13}$$

Here the quantity $(G M_c l_c^2 / \alpha_{\mathrm{MAC}})^{1/2}$ is stored in memory. The GADGET 2 SPH code (Springel 2005) uses this criterion since it does not implement any multipole correction terms in the gravity calculation. In essence the criterion allows some of the accuracy benefits of multipole correction terms to be captured (by crude approximation) without suffering any of their costs in either calculation time or storage requirements (memory usage).

A downside to such approximation is that cells may appear to have a much higher quadrupole term than they actually possess. A cell containing a spherically symmetric distribution of particles should only possess a monopole term (able to be treated as a point mass by Gauss's theorem) but under approximation could appear to have large quadrupole terms. This would lead to the cell being opened unnecessarily and suffering a cost in speed without any gain in accuracy. It is possible to employ a more sophisticated MAC in order to avoid this problem of being overly conservative. One such approach involves basing the MAC upon λ_{MAX}, the maximum of the 3 eigenvalues of the traceless symmetric quadrupole moment tensor (Hubber, Batty, McLeod & Whitworth 2011), resulting in the criterion

$$r^2 \geq r_{\mathrm{MAC}}^2 \equiv \left(\frac{G \lambda_{c,\mathrm{MAX}}}{\alpha_{\mathrm{MAC}} |\mathbf{a}_{\mathrm{grav}}|} \right)^{1/2} \tag{5.14}$$

SPH Neighbour Criterion for MACs

A complication for tree gravity in an SPH code arises due to kernel-softened gravity, where the gravitational force due to SPH neighbours is modified from the analytic result (see section 4.3.8). If the chosen MAC results in SPH neighbours being included in the multipole expansion of a cell then this value will be incorrect since the multipole expansion does not account for softening. It must therefore be ensured that the gravitational force due to SPH neighbours is always added by direct summation, so any cells that could potentially include SPH neighbours must be opened.

As the tree is built then the maximum extent d_c for each cell c can be recorded, where the extent is essentially the distance plus smoothing radius of a particle j from the centre of mass of the cell ($d_c = \mathrm{MAX}_j \left\{|\mathbf{r}_j - \mathbf{r}_c| + 2h_j\right\}$, assuming a smoothing radius of $2h$, applicable to the M4 kernel). The criterion to not open a cell based on SPH neighbours is then

$$r^2 \geq \mathrm{MAX}\left\{(2h_i)^2, d_c^2, r_{\mathrm{MAC}}^2\right\} \tag{5.15}$$

This is similar to the MACs used in the VINE SPH code (Nelson, Wetzstein & Naab 2009). The additional criterion for SPH adds an extra overhead to stocking and walking the gravity tree, with two cell-opening criteria needing to be checked for each cell, but in the highly clustered geometries that are frequently found in gravitational collapse problems this extra effort can bring important accuracy benefits, and even speed benefits too depending on the other MAC used. Certainly it should prevent any large catastrophic errors from developing since this additional MAC precludes the multipole expansion from being used in the cells which include the particle itself.

5.3.11 Multipole Expansion Test

The accuracy of the gravity tree and the multipole moment correction terms can be tested by comparison with the gravitational force obtained from direct summation. The accuracy is dependent on the chosen MAC and the chosen accuracy parameter for that MAC. We will use the Geometric MAC since it has been used in previous tests of this type (McMillan & Aarseth 1993), and define the root-mean-squared fractional force error ϵ as

$$\epsilon = \left(\frac{\sum_{i=1}^{N} |F_i^{tree} - F_i^{direct}|^2}{\sum_{i=1}^{N} |F_i^{tree}|^2} \right)^{1/2} \tag{5.16}$$

where $\mathbf{F}_i^{tree}$ is the force of particle i evalulated using the tree and $\mathbf{F}_i^{direct}$ is the force of particle i evalulated by direct summation over all particles.

We consider the accuracy of the gravity tree by calculating this fractional force error ϵ for a uniform density sphere containing $100,000$ particles. We treat the sphere as an N-body system so kernel-softened gravity is not applied (this removes the effect of h-finding computations from our analysis). We calculate ϵ as a function of the opening angle θ_{MAC} for values between 0.1 and 1.0 up to monopole, quadrupole and octupole order (Figure 5.2). As expected the fractional error is a monotonically increasing function of the opening angle (McMillan & Aarseth 1993) and the errors decrease as we increase the multipole order. The largest decrease in error comes by increasing from monopole to quadrupole order, where the error decreases by more than an order of magnitude for some values of θ_{MAC}. Increasing to octupole order also decreases the error for all values of θ_{MAC} but by only half an order of magnitude at most. For most scenarios and values of θ_{MAC}, quadrupole order will therefore be the optimal choice in terms of speed and memory usage against accuracy. This is emphasised in Figure 5.3, which shows the relative time taken to perform a tree calculation (as compared to direct summation) at a given error level.

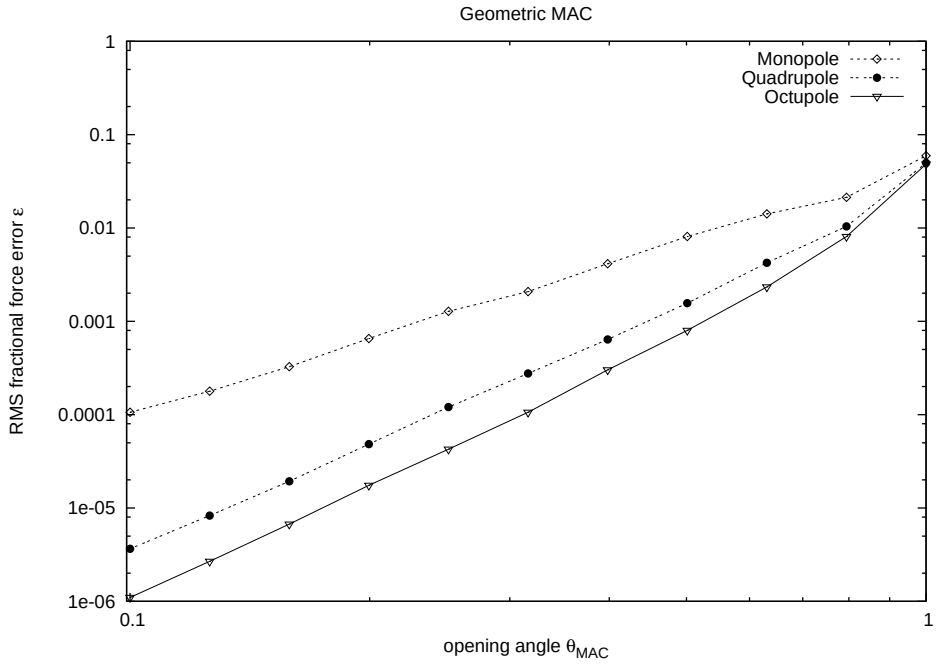


Figure 5.2: The RMS fractional force error as a function of the maximum opening angle θ_{MAC} between 0.1 and 1 for a uniform density sphere (100,000 particles), shown in logarithmic scale.

A value of $\theta_{\rm MAC} = 0.2$ using quadrupole order should therefore speed up the gravitational calculations by an order of magnitude at a cost of $< 0.01\%$ error.

5.4 Block Timestepping

Individual block timesteps (Aarseth 2003) can be used to reduce the run-time of simulations where the particles have a large range of timesteps. In gravitational collapse simulations a small fraction of the particles near the centre of the collapse obtain small timesteps, and these will slow down all other particles when a global timestep is employed. Likewise, in the case of a disc, the orbital period (and consequently the optimal timestep) increases quite rapidly with radius ($T \sim R^{3/2}$ for Keplerian rotation).

By instead allowing particles to have their own timesteps, each can be integrated with a close to optimal step size and efficiency can be gained. In block timestepping the timesteps are constrained to a hierarchy of possible values: $t_{_n} = 2^n t_{\rm MIN}$, where $n = 0, 1, 2, ..., n_{\rm MAX}$. Integer powers of 2 are used so that the particles are automatically synchronised at the end of the largest timestep $\Delta t_{\rm MAX} = 2^{n_{\rm MAX}} t_{\rm MIN}$.

During the simulation the positions and velocities of all longer-timestep particles will be predicted simply by extrapolation using the existing integration scheme. The accelerations of particles will only need to be calculated according to their chosen timestep (on their halfstep with the modified Leapfrog scheme) and this reduction in

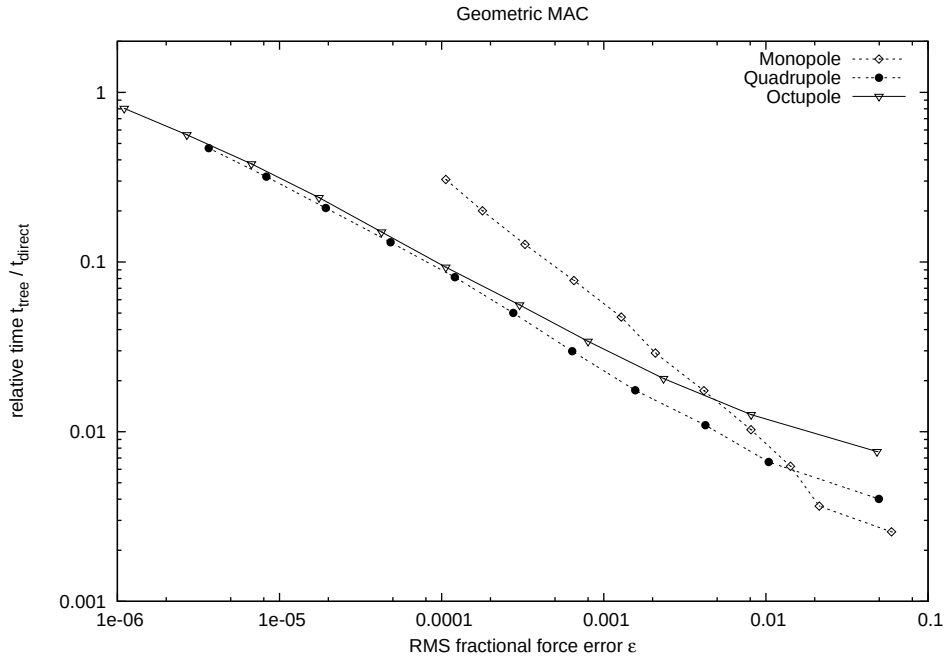


Figure 5.3: The performance (time taken for the tree calculation relative to direct summation) as a function of RMS fractional force error ϵ for a uniform density sphere (100,000 particles), shown in logarithmic scale.

expensive calculations can provide a significant gain in speed. The impact on accuracy is often minimal since the timestep of each particle is tailored to satisfy its local stability conditions. This timestep is recalculated immediately upon completion of a full integration step for that particle, ensuring that stable behaviour is maintained in the case where the optimal timestep decreases, and that efficiency is maintained in the case where the optimal timestep increases.

When the optimal timestep of a particle decreases it can fall directly into any level below it, because any lower timestep level will automatically be synchronised with the higher level. This is not necessarily the case, however, when the optimal timestep increases, so the lower timestep must continue to be used until it synchronises with the higher level, at which point the particle can move up into that level. For reasons of simplicity and safety it is recommended that a particle only be allowed to ascend by one level upon synchronisation, rather than letting it jump up multiple levels. While mildly hampering efficiency this serves to smooth the transition upwards, thus minimising any potential problems with numerical errors.

A potential drawback of the block timestepping scheme occurs when the difference in timesteps between neighbouring particles becomes very large, which might happen in simulations where there is a high Mach-number shock. This issue can be resolved by routinely checking the particle timesteps and reducing them as needs be if the difference from their neighbours is too large (Saitoh & Makino 2009), ideally restricting differences to only one or two timestep levels. This can be done by comparing the timestep of a particle to all of its neighbours when a search for neighbours

is performed. If the minimum timestep of those neighbours is too small then the timestep of the particle can be reduced, provided that it is synchronised at this point in time with the lower timestep level that it is moving to. To robustly ensure that a particle always has an appropriate timestep relative to its neighbours, a dedicated tree walk to compute this appropriate timestep should be employed (Springel 2009), but the greater expense makes its inclusion less desirable; it is only worth employing in simulations where this issue is likely to be important, such as those involving sources of ionising radiation which heat regions of gas to high temperatures (and thus low timesteps).

5.4.1 Data Structures

New variables will be needed to handle the block timestepping scheme, such as the total number of quantised timestep levels, the number of spare levels (included so that those particles whose timesteps decrease will have available levels to fall into), the number of timesteps needed to resynchronise (the number of minimum timesteps in a maximum timestep), a reference timestep for creating levels (which should be chosen to divide exactly into the time intervals at which data output is desired) and the real timestep size of the minimum timestep (in physical units). As far as possible the timestepping variables should be handled as integers (measured as integer multiples of the minimum timestep) to maximise calculation efficiency.

The particle data will also require additional timestepping variables. Each particle will require a boolean logical flag for performing acceleration calculations (set to true when these are required, or false otherwise), its individual timestep size n_p (in terms of the integer multiple of the minimum timestep) and the time that it last completed a full integration step n_{last}.

This will require additional memory allocation, but it might be possible to use smaller variable types for the logical and integer variables in order to minimise memory usage.

5.4.2 Compiler Flags

The use of the block timestepping scheme should be implemented as a compiler flag option since this will allow comparison tests with global timestepping. In principle, however, block timestepping using a single timestep level should be equivalent to using a global timestep in terms of accuracy, but the handling of the additional data structures will incur some overhead costs in speed.

5.4.3 Parameters

Two useful parameters for the block timestepping scheme are the maximum number of levels to use and the number of spare levels to include within them, since different values may be appropriate to different problems, with less being needed for fairly quiet and homogeneous simulations and more being needed for highly dynamic simulations containing dramatically contrasting regions.

5.4.4 Integration

The integration routine within the main program loop must now interface with the block timestepping scheme; these additions have been marked ($\star$) and *highlighted* to stand out from the previous pseudocode.

Pseudocode (integration):

```
⋆ calculate total energy and momentum of system [optional diagnostic]
⋆ (only on synchronisation steps)
⋆ calculate block timesteps
FOR each particle
      apply first part of integration scheme
      (predict position)
ENDFOR
FOR each particle
      calculate smoothing length and density [and any correction terms]
ENDFOR
FOR each particle
      calculate pressure using the equation of state
ENDFOR
⋆ calculate gravitational accelerations [and potentials]
⋆ calculate hydrodynamic accelerations
⋆ (only for flagged particles)
FOR each particle
      apply second part of integration scheme
      (advance velocity and correct position)
ENDFOR
⋆ advance time t = t + n_step Δt_MIN
⋆ (also advance integer timestep counter n = n + n_step)
```

The principal addition is the subroutine to calculate the new timesteps for each individual particle.

Energy and momentum need only be calculated on a synchronisation step since an evaluation at any other time will be based upon extrapolated positions and velocities.

5.4.5 Block Timestepping Routine

The routine for block timestepping will be quite complicated, involving a sizeable amount of auxiliary variables (local to the routine itself), but the overall process is summarised below.

Pseudocode (calculate block timesteps):

```
IF it is a synchronisation timestep (n = n_MAX) THEN
      reset integer timestep counter (n = 0)
      FOR each particle p
```

```
            calculate ideal timestep Δt_p
      ENDFOR p
      use minimum timestep (Δt_MIN) to build block timestepping scheme
      (to desired number of levels)
      FOR each particle p
            assign to appropriate level according to ideal timestep Δt_p
            (determine its integer timestep n_p)
      ENDFOR p
ELSE
      FOR each particle p that has completed a full integration step
            recalculate ideal timestep Δt_p
            reassign to appropriate level
      ENDFOR p
ENDIF
```

When the simulation begins at $t = 0$, $n_{\rm MAX}$ should be set to zero to ensure that the first step is treated as a synchronisation timestep. The time at which each particle last completed a full integration step n_{last} should likewise be initialised to zero.

Synchronisation Timesteps

If it is a synchronisation timestep (when $n = n_{\rm MAX}$) then the integer timestep counter can be reset ($n = 0$) and the ideal timesteps can be calculated for each particle using the usual conditions (an appropriate Courant-Friedrichs-Lewy condition and an acceleration condition to maintain stability).

The minimum timestep among all the particles will be used to form the basis of the block timestepping scheme. To allow for S spare levels this minimum timestep must be divided by 2^S to determine $\Delta t_{\rm MIN}$. With L total levels the synchronisation timestep will be $\Delta t_{\rm MAX} = 2^{L-1}\Delta t_{\rm MIN}$.

If using a reference timestep to create the levels (in order to synchronise them with the time intervals Δt_{int} at which data output is desired) then before dividing by 2^S the minimum timestep must be adjusted to exactly divide into this time interval. First the required division depth must be determined using $X = \frac{1}{\log_e 2}\log_e\left(\frac{\Delta t_{\rm MIN}}{\Delta t_{int}}\right) - 1$ and rounding X down to the nearest integer. Then the minimum timestep can be adjusted to $\Delta t_{\rm MIN} = 2^X \Delta t_{int}$.

If a fixed (rather than adaptive) timestepping scheme is required then $\Delta t_{\rm MAX}$ can be fixed at some value, and then $\Delta t_{\rm MIN} = \frac{\Delta t_{\rm MAX}}{2^{L-1}}$. This will maintain the same block timestepping structure throughout the simulation.

With L levels in the block timestepping scheme, the number of integer timesteps until the next synchronisation will be $n_{\rm MAX} = 2^{L-1}$.

Now that the level structure of the block timestepping scheme has been determined, each particle p can be assigned to the appropriate level according to its ideal timestep Δt_p. This level can be determined using $X = \frac{1}{\log_e 2}\log_e\left(\frac{1.01\times\Delta t_p}{\Delta t_{\rm MIN}}\right) + 1$ and rounding X down to the nearest integer, with the factor of 1.01 being used to

tip borderline cases into the higher level to maximise efficiency. If X ever happens to exceed L then X must be set to L in order to place the particle within the level structure, but any such occurrence will suggest that increasing the number of levels may be beneficial. The individual timestep of the particle can now be set as $n_p = 2^{X-1}$ in terms of its integer multiple of the minimum timestep $\Delta t_{\rm MIN}$.

The lowest value of n_p will determine n_{step}, which is the number of integer timesteps that correspond to a full integration step on the lowest occupied level.

After each integration step the integer timestep counter n is increased by n_{step} and the real time t by $n_{step}\Delta t_{\rm MIN}$.

Non-Synchronisation Timesteps

On the other hand if it is not a synchronisation step then not all the particles will need their timesteps recalculated. A particle only needs to do this at the end of its individual timestep. Since n_{last} will be set to n whenever a particle completes a full integration step, the condition $n = n_{last}$ can be used to determine which particles require recalculation.

If a particle requires recalculation then its previous value of n_p is first recorded as n_{prev}. Then its ideal timestep Δt_p can be calculated. It is useful to check that this is not lower than the minimum timestep $\Delta t_{\rm MIN}$ since such an occurrence risks instability and highlights the need for more spare levels in the scheme. The appropriate level can be determined as before to obtain a new value of n_p.

If $n_p > n_{prev}$ then the particle can be raised one level (such that $n_p = 2n_{prev}$) if it is synchronised with the higher level. This requires that $n_{prev} < \frac{n_{\rm MAX}}{2}$ (so that a higher level exists) and also that the remainder of $\frac{n}{2n_{prev}}$ is zero (so that it is not halfway between levels). If it is not synchronised then it must remain on its current level (such that $n_p = n_{prev}$).

In the other case, when $n_p \leq n_{prev}$, all lower levels are necessarily synchronised with those above so there is no need to check. The exception, of course, is the case mentioned above where $\Delta t_p < \Delta t_{\rm MIN}$, when the lowest level $n_p = 1$ will have to suffice (since no level exists below this).

The lowest value of n_p among all recalculated particles will again determine n_{step}, the number of integer timesteps that correspond to a full integration step on the lowest occupied level, but first the previous value of n_{step} should be recorded as n_{old}.

If $n_{step} > n_{old}$ then the whole integration can be speeded up by one level (such that $n_{step} = 2n_{old}$) if it is synchronised at this higher level. This requires that $n_{old} < \frac{n_{\rm MAX}}{2}$ (so that the higher level exists) and also that the remainder of $\frac{n}{2n_{old}}$ is zero (so that it is not halfway between levels). If it is not synchronised then the integration step must remain at its current level (such that $n_{step} = n_{old}$).

In the other case, when $n_{step} \leq n_{old}$, integration can always continue at a lower level if required, since lower levels are necessarily synchronised with those above.

As before, after each integration step the integer timestep counter n will be increased by n_{step} and the real time t by $n_{step}\Delta t_{\rm MIN}$.

Integration Scheme Modifications

Modifications will also need to be made to the integration scheme regarding particle advancement. The details of the implementation will depend upon the integration scheme, but for the modified Leapfrog scheme the following approach applies.

After the preliminary acceleration calculation the flag for whether a particle requires an acceleration calculation is set to false. For any particular particle at any given integration time n we can define $\Delta n = n - n_{last}$ (which gives the integer time since the last fullstep), and a full integration step for the particle is given by n_p.

When $\Delta n < n_p$ the first part of the integration scheme is used to extrapolate positions and velocities by Δt, where $\Delta t = \Delta n \Delta t_{\mathrm{MIN}}$.

On the halfstep ($\Delta n = \frac{n_p}{2}$) the flag for an acceleration calculation is set to true.

On the fullstep (when $\Delta n = n_p$) the second part of the integration scheme is implemented, after which the positions and velocities are stored (since these non-extrapolated values will be needed at each extrapolation step) and n_{last} is set to n.

When calculating accelerations only flagged particles will require a calculation, and this reduction in calculations is where the significant gain in efficiency is made. After the acceleration calculation the flag is reset to false, and remains so until the next halfstep is reached.

5.4.6 Performance Tests

The performance of the block timestepping scheme can be tested using a simple gravitational calculation. A simulation of a low-mass disc orbiting a central star was performed for one outer rotation period, with the run-time normalised by the number of integration steps required. A disc was chosen because in Keplerian rotation ($T \sim r^{3/2}$) an order of magnitude difference in radial orbit r corresponds to one and a half orders of magnitude difference in orbital period T, providing good reason to implement a block timestepping scheme. A low-mass disc was used (0.01% of the central star mass) in order to remain stable against gravitational fragmentation, which would needlessly complicate the timing results.

The behaviour of a disc represented by $10,000$ particles arranged in an annulus with orbital radii between 100 AU and 1000 AU is used to demonstrate the effect of the block timestepping scheme in Figure 5.4. Plotting the speed-up factor (reciprocal of the time taken relative to global timestepping, which is shown as $L = 1$) against the number of levels L in the block timestepping scheme shows the speed increasing up to $L = 6$, at which point the performance gain is saturated. This is to be expected since in Keplerian rotation an order of magnitude difference in orbital radius corresponds to a 32-fold increase in orbital period ($10^{3/2} \simeq 32$), which requires a 5-level structure to span it ($2^5 = 32$). Accounting for the possibility of some particles straying marginally inside the inner radius (or outside the outer radius) on slightly elliptical orbits, an additional level ($L = 6$) provides the safety margin to handle these, but any further levels prove to be unnecessary.

The speed-up factor for the disc simulation is ~ 5, which is a significant gain

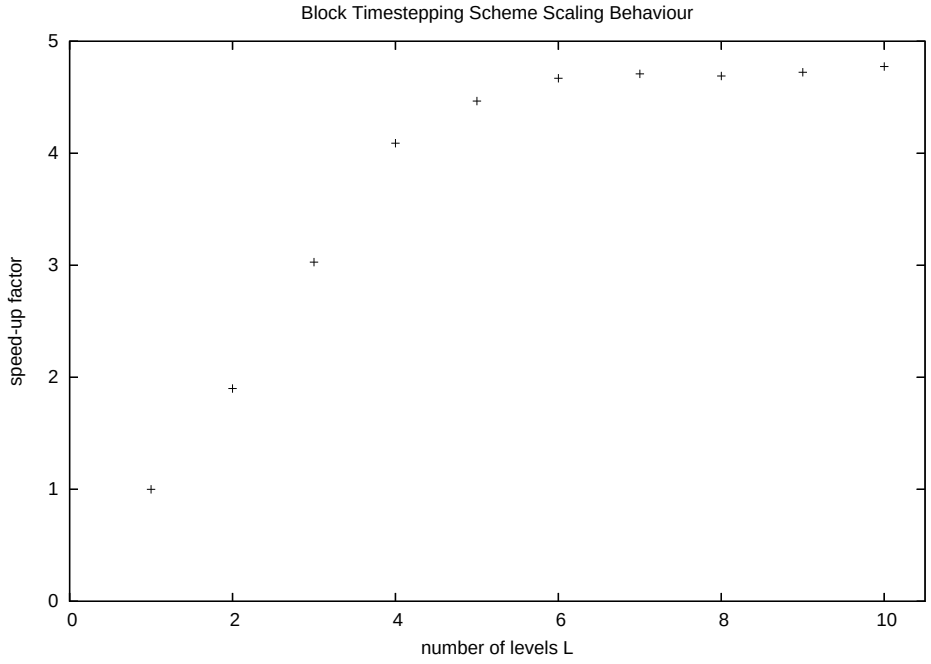


Figure 5.4: Comparison of speed-up factor as a function of the number of levels L in the block timestepping scheme for a disc of 10,000 particles.

in efficiency. The cost in accuracy is shown in Figure 5.5, where the breakdown in the conservation of angular momentum has been used as a global gauge of accuracy. There is negligible degradation in accuracy up to 4 levels, but more levels incurs a small error in accuracy since the outermost particles (which contribute most to the angular momentum, since $L \sim R^{1/2}$) are now integrated on longer timesteps. Even with a large block timestepping structure (10 or more occupied levels) this error remains at less than 0.2% after a full outer rotation period, so the significant gain in efficiency comes at minimal cost to accuracy.

For a stable disc there is no need for any spare levels but collapse problems require their inclusion. Several spare levels are recommended to handle freefall collapse since the timesteps of the central particles can rapidly decrease in orders of magnitude before a synchronisation step is reached and the block timestepping structure can be rebuilt. In general, 7 spare levels will handle a decrease of 2 orders of magnitude ($2^7 = 128$).

The number of spare levels for such collapse problems is far more important than the total number of levels since the initial uniform density sphere will begin with all particles on the same timestep level. With sufficient spare levels provided, the speed-up factor for the freefall collapse simulation is ~ 3.7, a respectable gain in efficiency. Care must be taken to ensure that there are not too many levels above the spare levels (that L is not too much greater than S) in order to ensure that synchronisation steps occur relatively frequently (before the spare levels run out).

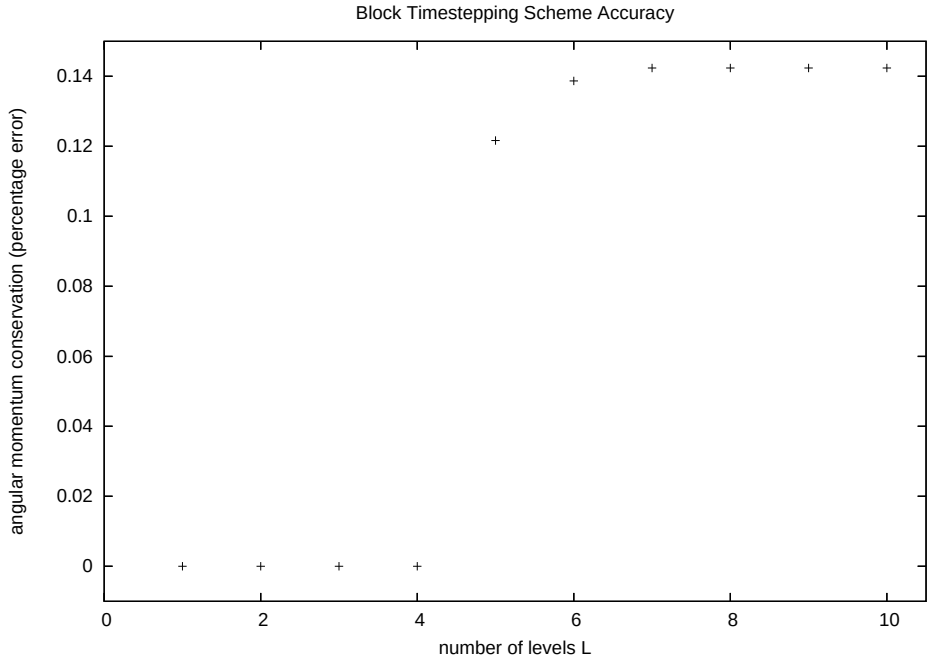


Figure 5.5: Comparison of accuracy in terms of angular momentum conservation as a function of the number of levels L in the block timestepping scheme for a disc of 10,000 particles.

5.5 High Resolution Tests

Using a tree for the gravitational calculation together with a block timestepping scheme enables simulations using greater numbers of particles to be performed in a reasonable length of time. This allows higher spatial resolution to be achieved with only a small loss of accuracy. Some of the tests performed in Chapter 4 are shown again here at higher resolution (in Figures 5.6, 5.7 and 5.8), demonstrating behaviour that is in very good agreement with the analytic solutions.

5.6 Sink Particles

Sink particles allow SPH simulations to be followed beyond the formation of the first dense objects (Bate, Bonnell & Price 1995). Gravitational collapse inevitably leads to high densities, high sound speeds, short smoothing lengths and therefore short timesteps (from the timestep conditions), which translates to long run-times. Once a gravitationally bound condensation has collapsed beyond some density threshold it can often be safely assumed that it will continue to collapse to form a protostar (or an unresolved protostellar multiple system, since the condensation may hierarchically fragment further at even higher densities: see Section 1.2.1). At such a point the particles composing the condensation can then be removed from the simulation and replaced by a sink particle.

This sink particle has a finite radius (called the sink radius, or the accretion radius, $r_{\rm sink}$) and any particle which enters this radius will be accreted by the sink

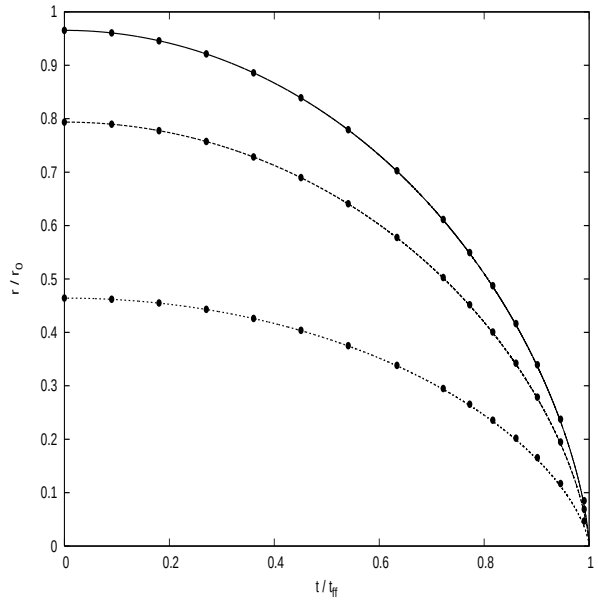


Figure 5.6: Freefall collapse for $N = 100,000$ particles with kernel-softened gravity. The lines show the analytic solutions for the sphere radius and the 90%, 50% and 10% mass radii inside it. The points are the numerical solution values at periodic intervals.

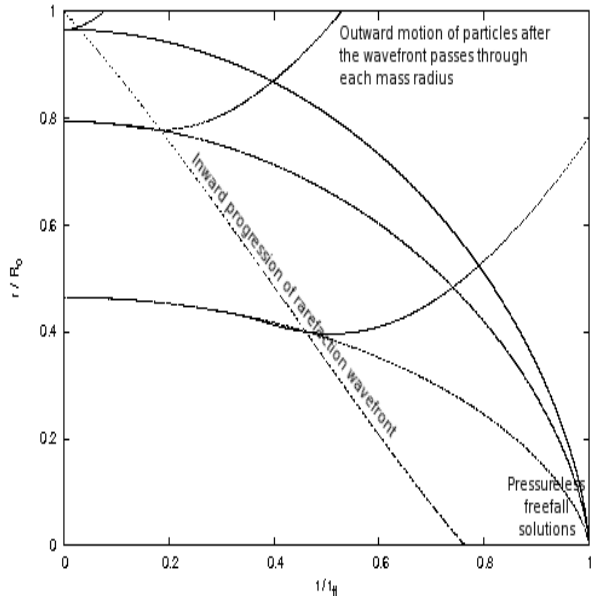


Figure 5.7: Isothermal collapse for $N = 100,000$ particles with SPH and kernel-softened gravity. The downward-curving lines show the freefall (pressureless) analytic solutions for the sphere radius and the 90%, 50% and 10% mass radii inside it. The upward-curving lines are the numerical solution values. The sloping near-diagonal line is the analytic solution for the inward propagation of the rarefaction wave. As the wavefront passes through each mass radius the solutions begin to deviate from the pressureless form.

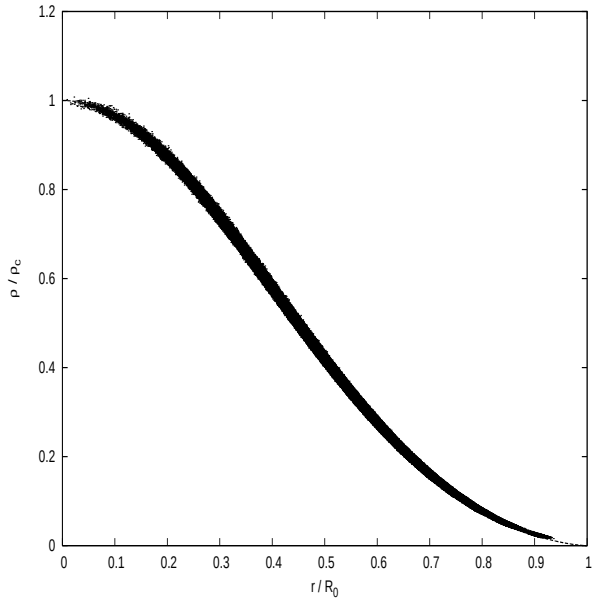


Figure 5.8: Relaxation of a uniform density sphere (composed of 100,000 particles) to a polytrope with exponent 5/3. The figure shows the resulting radial density profile. The dashed line (mostly hidden beneath the points) indicates the analytic solution.

particle if it is gravitationally bound: its mass, linear and angular momentum are added to that of the sink, and the particle itself is removed from the simulation.

The sink particle interacts gravitationally with all other sink and SPH particles, but does not interact hydrodynamically with any other objects. It is possible to implement boundary conditions to estimate an appropriate hydrodynamic interaction (Bate, Bonnell & Price 1995), but such schemes tend to be rather ad-hoc and cumbersome to implement and are therefore not often used.

5.6.1 Creation Criteria

A sink particle is created once an SPH particle satisfies a number of specific criteria. Bate, Bonnell & Price (1995) suggest a large number of tests to be performed on each particle; building upon these suggestions, the following criteria are recommended:

1. the mean density of a particle plus its neighbours is greater than some critical sink formation density ($\overline{\rho} > \rho_{\rm sink}$), where the mean is used to prevent a spuriously high-density "rogue" particle from forming a sink;

2. the SPH velocity divergence of a particle is negative ($\nabla \cdot \mathbf{v} < 0$), ensuring that there is a local convergence of the neighbouring particles;

3. the SPH acceleration divergence of a particle is negative ($\nabla \cdot \mathbf{a} < 0$), ensuring that there are no external tidal or shearing forces that might act to break up the condensation;

4. the total mechanical energy of a particle (kinetic plus potential energy) is negative, ensuring that it lies within a gravitationally bound region;

5. there is no other sink particle within $2r_{\rm sink}$ of the candidate SPH particle, preventing a sink particle from being formed where its radius would overlap with an existing sink particle.

The thermal energy is ignored in the fourth criterion because it can act to dominate over the other energy terms when only a particle and its neighbours are considered. A clump-finding algorithm might be used, allowing the thermal energy to be included in calculating the total energy of the entire clump, or surface terms may be calculated to include thermal energy in the virial expression, if the inclusion of thermal energy is deemed to be important.

Although the criteria listed here are ultimately subjective, they are nonetheless reasonable indicators of the presence of a gravitationally bound object, and hence the formation of a sink.

Constructing sink creation criteria is rarely straightforward. Consider a simple criterion: that a particle and its neighbours must be gravitationally bound. This does not guard against the possibility of the clump itself being bound when its core is not. Compromises must eventually be reached between the guiding principles of practicality and caution.

5.6.2 Accretion Criteria

An SPH particle is accreted by a sink if the following criteria are satisfied:

1. the particle lies within the sink radius ($|\mathbf{r}| \leq r_{\rm sink}$);

2. the 2-body energy (kinetic plus gravitational) of the sink-particle system is negative, ensuring that it is gravitationally bound.

Note that when determining which particles are accreted by which sinks, the above criteria are used to determine a list of particles that are candidates for accretion for each sink. Only after all particles have been checked should the particle properties (mass, momentum, angular momentum) actually be added to the sink; if they were added immediately then the accretion process would be dependent on the order in which the particles were checked.

5.6.3 Code Routines

The inclusion of sink particles should be handled with a compiler flag.

A subroutine will be needed to check for the formation of sinks at each timestep; it is recommended that only one sink be allowed to form each timestep for reasons of simplicity. Another subroutine will be needed to check for the accretion of particles onto sinks each timestep. Both will require a dedicated subroutine to remove SPH particles from the simulation.

Sink particles that are formed will also need to be integrated in time alongside SPH particles, with their gravitational accelerations likewise calculated; direct summation is recommended, avoiding the need to incorporate them into the gravity tree, which would then require a different structure to the hydrodynamic tree. For time integration, the timestep of a sink may be determined based upon the sink radius: $\Delta t \leq r_{\rm sink}/|v|$ and $\Delta t \leq \sqrt{r_{\rm sink}/|a|}$. For simplicity it is recommended that all sinks are integrated according to a single lowest global sink timestep, and placed on the lowest occupied level in a block timestepping scheme. Any particles that have a sink within their smoothing radius could also benefit from being placed on this lowest level.

New parameters will need to be incorporated for the sink formation density threshold, and the sink radius (which might be an absolute value, or based upon the smoothing length of the SPH particle from which it is formed: $r_{\rm sink} = 2h$).

New global variables will be needed to record the total number of sinks in the simulation and for sink timestepping (to incorporate them within the block timestepping scheme), and a new data structure will be needed to store the sink data. This will need to store similar data to an SPH particle (mass, position, velocity, acceleration, and so on) but the softening length h will not necessarily be the same as the accretion radius $r_{\rm sink}$, so an extra variable will be needed. While hydrodynamic quantities will not be needed, angular momentum will need to be recorded for conservation purposes as accreted SPH particles are removed. Other useful quantities to record include the time of formation, the internal angular momentum (present at formation, separate to the accreted angular momentum), the gravitational potential (and potential energy) and the mass accretion rate.

The data structures, subroutines and modifications necessary to incorporate sink particles could easily be extended to incorporate other particle types, such as inter-cloud medium particles (which interact hydrodynamically but not gravitationally) or boundary particles (which interact hydrodynamically but are not integrated in time normally, either remaining stationary or being integrated according to boundary conditions).

5.6.4 The Boss & Bodenheimer Test

A useful test for the sink routines is the evolution of a uniform-density isothermal sphere which is given an azimuthal density perturbation (of mode $m = 2$) and set in solid body rotation (Boss & Bodenheimer 1979). This configuration was devised to test the convergence and resolution of hydrodynamic codes in lieu of any simple analytic test for the Jeans instability. The density perturbation and rotation cause the sphere to collapse towards not one but two points, so two sink particles should be formed.

A $1{\rm M}_\odot$ sphere of radius 3×10^{14}m (~ 0.01pc) and uniform density ρ_0 is subjected to a non-axisymmetric density perturbation along the azimuthal angle ϕ,

$$\rho = \rho_0 \left(1 + A\cos(m\phi)\right) \tag{5.17}$$

using a mode of $m = 2$ and a fractional amplitude of $A = 0.5$. It is set in solid body rotation with angular speed $\Omega = 1.56 \times 10^{-12}{\rm rads}^{-1}$, giving the ratio of rotational to

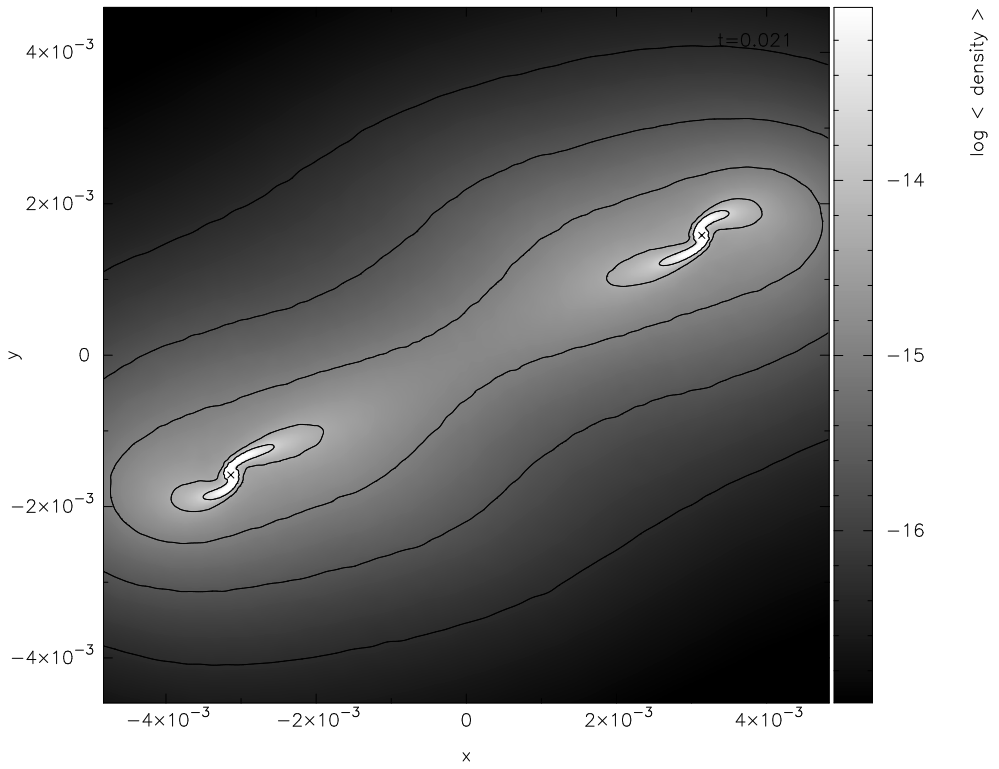


Figure 5.9: The isothermal evolution of a uniform density sphere (composed of 100,000 particles) subjected to an azimuthal density perturbation (mode $m = 2$, amplitude $A = 0.5$) and moving in solid body rotation, shown after 1.2 freefall times. Two sink particles (indicated by crosses) have formed at either end of a central bar.

gravitational energy as $\beta \simeq 0.2$. An initial uniform temperature of $T_0 = 10\text{K}$ gives the ratio of thermal to gravitational energy as $\alpha \simeq 0.25$. Support from rotation and from thermal pressure is therefore insufficient to prevent collapse.

The sink creation density $\rho_{\text{sink}} = 10^{-9}\text{kg m}^{-3}$, which is almost 5 orders of magnitude higher than the initial density $\rho_0 \simeq 1.44 \times 10^{-14}\text{kg m}^{-3}$.

The initial freefall time is $t_{ff} = 5.529 \times 10^{11}\text{s}$ ($\sim 17,500\text{years}$), and two sinks form at $\sim 1.166 t_{ff}$. Figure 5.9 shows the evolution after $1.2 t_{ff}$, with the two sinks at either end of a central bar, each accreting material that is in relative rotation around it.

The sink creation density effectively enforces a minimum smoothing length. Without such a minimum, and therefore no enforced gravitational softening at some minimum length scale, the isothermal collapse would proceed towards infinite density at $\sim 1.26 t_{ff}$ (Bate & Burkert 1997). This singularity makes the evolution difficult to follow computationally when $t \rightarrow 1.26 t_{ff}$ unless sink particles are used.

It is entirely possible for additional sink particles to form at either end of the bar, and then along the bar itself, if further collapsing material reaches the sink creation density before it has a chance to be accreted onto one of the two original sinks. One way to address this issue is to modify the isothermal equation using a more physically-informed approach.

Before applying the isothermal equation of state ($P = \hat{P}\rho T$), we can convert it into a more general barotropic form by first modifying the gas temperature.

$$T = T_0 \left(1 + \left(\frac{\rho}{\rho_{\rm crit}} \right)^{\gamma - 1} \right) \tag{5.18}$$

We will use a ratio of specific heats (adiabatic index) $\gamma = 7/5$. This is representative of molecular hydrogen, a diatomic gas with five degrees of freedom (3 in translation and 2 in rotation); in general $\gamma = \frac{f+2}{f}$, where f is the number of degrees of freedom. Strictly speaking, $\gamma = 7/5$ only applies above $T \sim 200$ K; at low temperatures it is very sensitive to the ortho/para ratio and effectively falls to $\gamma = 5/3$ for a ratio of 1.

We will use a critical density threshold $\rho_{\rm crit} = 10^{-11}{\rm kg\,m^{-3}}$. Below this density the gas responds essentially isothermally, but above it the gas will heat up more rapidly, approximately adiabatically. Since the sink creation density $\rho_{\rm sink} = 10^{-9}{\rm kg\,m^{-3}}$ is 2 orders of magnitude above this critical threshold, this heating will begin to inhibit the collapse just before sink formation, acting to prevent any spurious sink particles from being formed.

This equation of state more accurately models interstellar gas in a molecular cloud, since the low density gas is optically thin to its cooling radiation but at higher densities becomes optically thick. We use $\rho_{\rm crit} = 10^{-11}{\rm kg\,m^{-3}}$ because this is the peak density above which the temperature begins to rise above 10K during the collapse of a uniform density cloud (Stamatellos et al. 2007b).

Using this equation of state the two sinks form slightly later than in the isothermal case, at $\sim 1.28 t_{ff}$. Figure 5.10 shows the evolution after $1.5 t_{ff}$, again showing the two sinks at either end of a central bar, each surrounded by a small accretion disc.

Using this test in an essentially qualitative form is sufficient for the purpose of verifying the sink routines. It is worth noting, however, that more stringent tests of this kind have been used to deduce resolution conditions for simulations. For AMR codes, fragmentation is only modelled correctly if the grid spacing d is less than a quarter of the local Jeans length λ_J. If this Jeans condition ($d < \frac{\lambda_J}{4}$) is not satisfied then artificial fragmentation can occur, where clumps of gas that should be Jeans stable fragment due to numerical rather than physical effects (Truelove et al. 1997). For SPH codes, the Jeans condition requires that the Jeans mass is greater than the minimum resolvable mass, which is given by the mass contained within a smoothing sphere. This constrains the smoothing length h to be less than about a fifth of the Jeans length λ_J (Bate & Burkert 1997), or more precisely $h \leq \frac{\lambda_J}{4.58}$ (Bate, Bonnell & Bromm 2002).

5.7 Cache Optimisation

All particle data should be ordered in memory in the order of traversal when the gravity tree is walked. For an octal-spatial tree of uniform depth (where all branches of the tree terminate at the same level), the positions of the particles in memory will

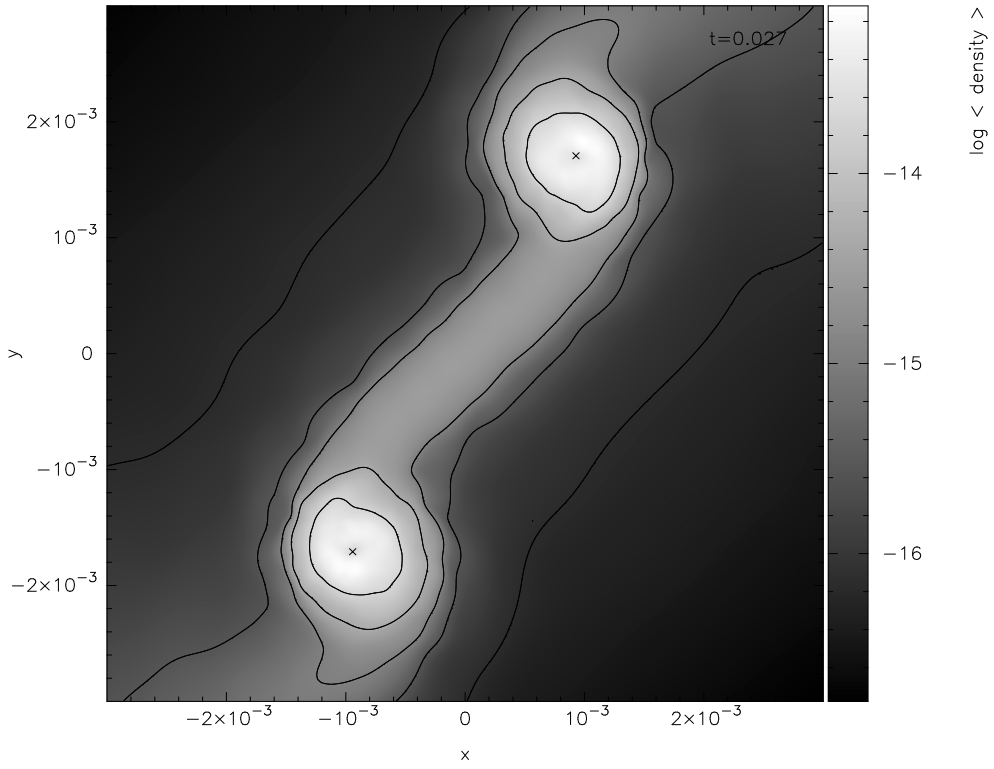


Figure 5.10: The evolution (using a barotropic equation of state) of a uniform density sphere (composed of 100,000 particles) subjected to an azimuthal density perturbation (mode $m = 2$, amplitude $A = 0.5$) and moving in solid body rotation, shown after 1.5 freefall times. Two sink particles (indicated by crosses) have formed at either end of a central bar.

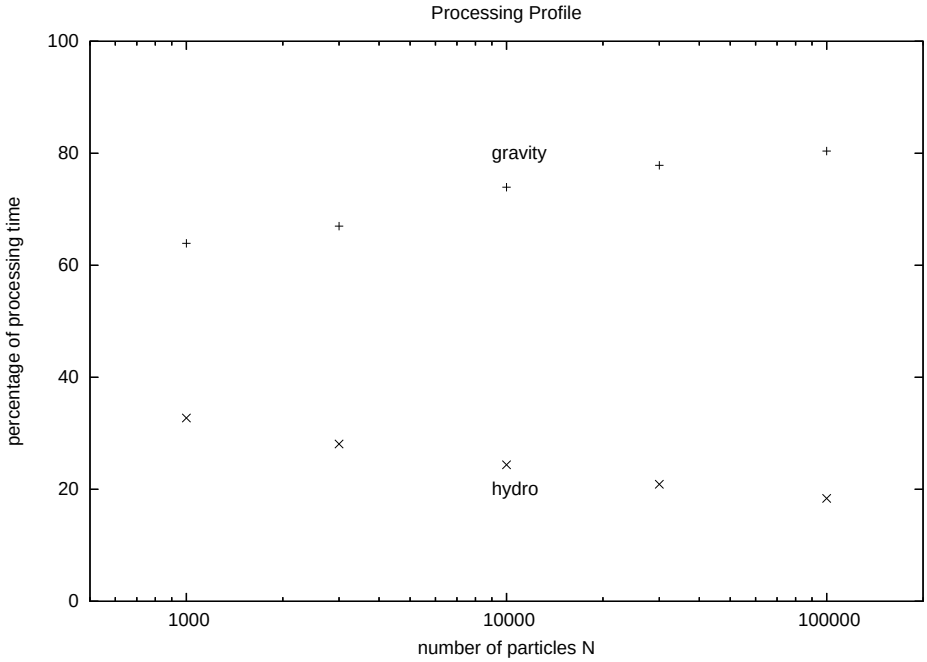


Figure 5.11: Performance profile of a disc simulation (using tree gravity and block timestepping), showing how gravitational calculations steadily dominate as the number of particles N increases, since gravity is a long range force while hydrodynamic forces are local.

trace a space-filling curve known as a Morton curve (or Z-curve, due to the Z-like shape traced within a 2D quad tree). In general the tree will have variable depth across different branches, particularly in the case of gravitational collapse, but this asymmetric space-filling curve is still essentially a Morton curve.

Morton space-filling curves have been used in other hydrodynamical codes, such as in the FLASH grid-based code (Abel, Bryan & Norman 2002), to determine how the grid cells are arranged in memory. This becomes important when parallelising the code on distributed memory machines since the space-filling curve determines how the cells will be distributed across the various processors.

In the GADGET 2 SPH code (Springel 2005) a Peano-Hilbert space-filling curve is used to order the particles in memory, which groups neighbouring particles together in an optimal manner such that they can be distributed to the same processor on paralellisation, in order to minimise the amount of communication needed between processors when calculating SPH quantities. This is useful for distributed-memory parallelisation, but for shared-memory parallelisation the primary concern is simply to optimise the cache usage when performing a tree walk. With the Morton curve, when searching for SPH neighbours all particles in a leaf cell will be contiguous in memory, and particles in nearby branches will likely be close (if not contiguous) in memory.

The data structure for particles should be packaged so as to optimise the gravitational calculations, since these dominate the processing time (see Figure 5.11).

This requires grouping the position r, mass m and softening length h of a particle together in memory (and also the ζ correction term for the energy-conserving formulation). When walking the gravity tree the SPH neighbour criterion (Equation 5.15) ensures that nearby leaf cells are opened and we will have to compute the direct particle-particle interactions for a number of neighbouring particles. Grouping all the required variables for these calculations together in memory, in addition to ordering these groups themselves along a Morton curve, will serve to minimise cache misses when calculating the particle-particle contributions.

5.8 Parallelisation

There are two main paradigms for parallelisation: shared-memory (where each processor shares the same memory space) and distributed-memory (where each processor has its own memory space).

A code can be parallelised for shared-memory multi-processor machines using OpenMP, which allows any loops within the code to be separated into chunks for each processor, spreading the work of calculation across the multiple processors. If the result of each looped calculation is independent of the others in that loop then OpenMP parallelisation is trivial, merely requiring a compiler directive at the start and end of the loop. An SPH code contains many such loops over all particles when various properties need to be calculated, such as SPH quantities or accelerations.

5.8.1 Load Balancing

When a loop of N elements is parallelised across P processors using OpenMP, $\frac{N}{P}$ elements will be performed by each processor. If each loop element requires the same number of processing operations then the work is divided evenly and parallelisation efficiency is optimal. Some loops, however, will contain elements that require significantly varying numbers of operations to complete, such as walking the gravity tree for each particle, where some particles may need to walk the tree down to much greater depths than others. This could lead to $P-1$ processors idly waiting for the remaining processor with the heaviest work-load to finish its calculations.

In such cases the work needs to be spread more evenly across the processors, and this task is referred to as load balancing. It can be achieved using the dynamic sheduling option of OpenMP, where instead of each processor taking $\frac{N}{P}$ elements of the loop, it takes only a few at a time (a chunk). Once it has finished processing this chunk it takes another one. In this way all processors can be kept as active as possible. Determining the size of the chunks is a compromise since each chunk incurs a small overhead cost due to the communication requirement between processors (keeping them all informed of which elements have been taken, to prevent any from taking overlapping chunks). While a chunk size of 1 element would minimise any processor idleness it maximises this communication cost, so a suitable middle ground must be chosen between the extremes of 1 and $\frac{N}{P}$. Using a chunk size of $\sim \frac{N}{10^3 P}$ has proven to be a reasonable compromise in practice.

When block timestepping is used, further care must be taken to ensure load balancing. If only a subset of the particles are deemed active (flagged to calculate their

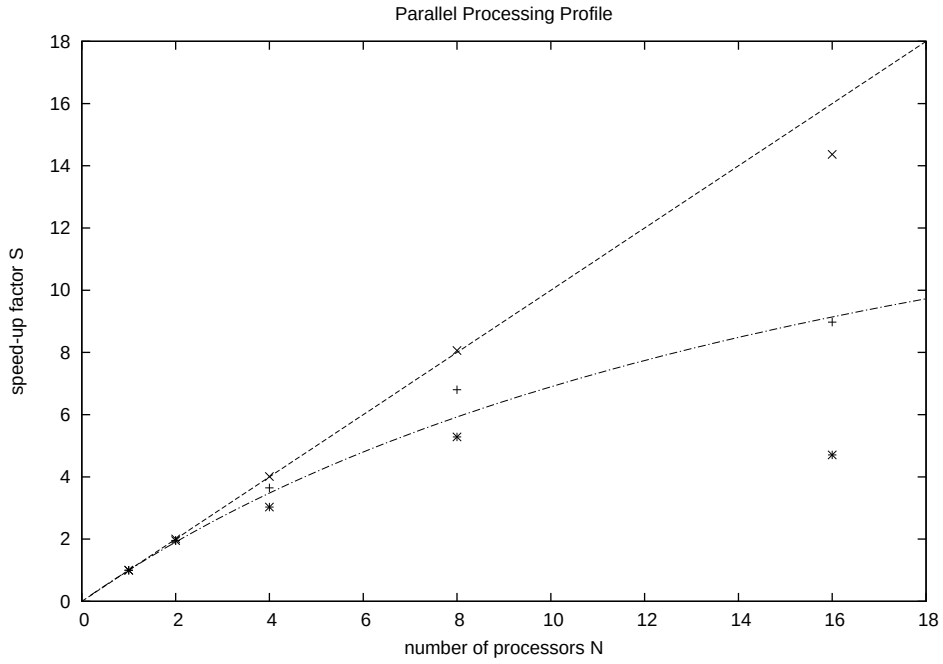


Figure 5.12: Parallel performance profile of a disc simulation showing how the speed-up factor S scales with the number of processors N. The steep line shows ideal scaling (perfectly parallelised $P = 1$). The gravitational calculations (crosses), when considered in isolation, perform close to this ideal. The hydrodynamic calculations (asterisks), however, do not, and slow the overall performance (pluses) down. The result is effectively that of a 95% parallelised code (indicated by the lower curve).

SPH properties and accelerations on this timestep) then the loops should be performed only over the active particles rather than all particles. This avoids the undesirable scenarios of processors taking chunks that contain mostly inactive particles, or in the worst case all active particles being situated within a single chunk, by replacing the loop over N total particles with a loop over a sublist of A active particles (and reducing the chunk size correspondingly to $\sim \frac{A}{10^3 P}$).

5.8.2 Scaling Tests

Amdahl's Law of Parallelisation states that the speed-up factor $S = \frac{1}{(1-P)+(P/N)}$, where P is the proportion of the code that is parallelised and N is number of processors used in parallel. A perfectly parallelised program ($P = 1$) would therefore garner a speed-up factor of N while a completely serial program ($P = 0$) would have a speed-up factor of unity. A half-parallelised program ($P = \frac{1}{2}$) would only have a speed-up factor of $\frac{2N}{N+1}$, which tends towards 2 as N goes to infinity, so it is clearly important to ensure that as much of the program is parallelised as possible. Fortunately, as was shown in Figure 5.11, gravitational calculations dominate the program, and these can be parallelised.

Figure 5.12 shows how parallel performance (speed-up factor S) scales with the

number of processors N. The gravitational calculations dominate and perform very well. Unfortunately the second largest component of the code, the hydrodynamic force calculations, do not perform so well, and damp the speed-up factor for higher numbers of processors. When considered in isolation the speed-up factor for the hydrodynamic force calculations is actually lower for 16 processors than it is for 8. This is due to the bottle-necking that occurs when multiple processors attempt to access the same area of shared memory at the same time. Even the gravitational calculation suffers this effect at higher N, but the effect is far more pronounced with the hydrodynamic forces due to their local nature. The long-range gravitational calculations thresh though the shared memory in a smooth and orderly fashion, while the short-range hydrodynamic calculations access memory in a more fragmented manner, seeking out only their spatial neighbours. Memory bottle-necks therefore enforce the law of diminishing returns on the number of processors. Nonetheless a speed-up factor of ~ 7 when using 8 processors is a respectable return and demonstrates the power of parallelisation.

Also note that high-N shared-memory machines ($N > 8$) are not commonplace (and therefore very expensive), while low-N shared-memory machines ($N = 2 - 4$) have now become the norm (the default architecture of a desktop computer), so good performance in this lower range is of particular significance.

5.8.3 Machine Architecture

Distributed Memory Architecture

While OpenMP can be used for shared-memory parallel processing, distributed memory is another matter entirely. Since there is no single global memory space, the only way for the multiple processors to collate their efforts is to communicate with each other when necessary.

A code can be parallelised for distributed-memory multi-processor machines using MPI (Message Passing Interface), which allows communication between the constituent processors, but such parallelisation is far from trivial and can require significant modification. A simulation must effectively be divided into P sub-simulations, one for each processor, and the difficulties arise when particles need to interact with particles (or tree cells) on other processors, since this requires the passing of required information between processors in an efficient manner; the various dependencies involved in supplying such information can make load-balancing a far more intricate task.

While the effort to parallelise for distributed-memory is significantly greater, there are corresponding potential benefits. Shared-memory machines tend to be limited in the number of processors they possess, since having many processors all reading and writing to a single global memory space creates serious bottle-necking issues. With distributed-memory machines there is no such limit to the number of processors; instead the only real concern is that they all have an efficient means of communicating with each other, such as Myrinet interconnection. As more processors are added, bottle-necking will eventually occur in the intercommunications, so the performance

scaling with P will start to drop off, entering the domain of diminishing returns. At this point the latency (the overhead cost of communications) tends to become less of an issue than bandwidth (the amount of communications traffic), so while a small number of large communications are usually optimal for low P, this may need adjustment to larger numbers of smaller communications at higher P to maintain performance.

Not only does distributed-memory allow for much more potential processing power to be available than shared-memory (allowing for faster run-times), it also allows for much more memory to be available, since managing the memory of X machines each with Y gigabytes of individually-addressed memory is easier than managing the memory of one machine with $X \times Y$ gigabytes of globally-addressed memory. More available memory extends the limit on the number of particles N that can be included in a simulation, which allows for higher resolution simulations. Since spatial resolution improves as $N^{1/D}$ in D dimensions, high-resolution 3D simulations can have dramatic memory requirements that only distributed-memory machines are able to satisfy.

Hybrid Architecture

A typical computer processor now contains multiple cores, making it equivalent to a shared-memory multi-processor system. Each processor in a distributed-memory system is therefore able to parallelise its sub-simulation across its multiple cores using OpenMP, while communicating with the sub-simulations of other processors using MPI. This MPI/OpenMP combination should maximise the efficiency of this hybrid architecture which is prevalent in modern supercomputing.

Graphics Processing Units

The core on a typical computer processor can only handle one floating-point operation per machine cycle. Most of the hardware is dedicated to handling large amounts of integer (and character string) operations because the industry is guided by consumer demand, and the typical computer user is more interested in the efficient browsing of web pages than in the efficient calculation of gravitational accelerations.

Fortunately there is one popular area of the computer industry that requires intense floating-point processing power: 3D computer graphics. Since a typical processor is poor at this job, hardware exists in the form of a dedicated supplementary processor, the graphics processing unit (GPU), to handle all the floating-point arithmetic. The need to render the many polygons of high-resolution 3D graphics at a respectable frame-rate has led to GPUs that can handling over a hundred floating-point operations per machine cycle. Conveniently for astrophysics, the need to reduce the cost of operations such as normalising 3D vectors means that the hardware is already greatly optimised for $\frac{1}{r^2}$ gravitational calculations. This raises the possibility of harnessing the power of GPUs to increase the efficiency of gravitational calculations by about two orders of magnitude (Nyland, Harris & Prins 2007).

5.9 Summary

Methods to optimise the performance of a self-gravitating hydrodynamics code have been described, allowing us to further our aim of modelling self-gravitating gas by achieving higher resolution without any considerable cost to efficiency.

A tree algorithm has been developed which replaces the direct summation of particle-particle gravitational interactions (an N^2 calculation) with particle-cell interactions (an $N \log N$ calculation). Multipole terms may be used to improve the approximation, and an order-of-magnitude efficiency gain is possible with very little cost in accuracy.

A block timestepping scheme has been developed which allows each particle to operate close to its local ideal timestep, using a hierarchy of possible timesteps ($t_n = 2^n t_{\rm MIN}$) to ensure synchronisation. In simulations with widely-varying timescales (such as Keplerian discs, where $T = r^{3/2}$) an overall speed-up factor of ~ 5 is possible with only a small accuracy cost.

High-resolution simulations of freefall collapse, isothermal collapse and a polytrope clearly demonstrate the accuracy of an optimised self-gravitating hydrodynamics code.

Sink particles have been incorporated in order to handle highly dense regions in collapse simulations. Creation and accretion criteria have been addressed, and the Boss & Bodenheimer test demonstrates their performance.

Parallelisation, allowing the code to run on multiple processors, has also been discussed. It provides a speed-up factor close to optimal for $2 - 4$ processors before suffering from diminishing returns.

In the next chapter we will discover that we need to further refine our optimised SPH code in order to handle problems involving shocks and hydrodynamic instabilities.

"Spending is quick,
earning is slow."

Russian proverb

Chapter 6

Shocks and Instabilities

In this chapter we introduce some of the common problems associated with hydrodynamic simulations, such as resolving shocks and hydrodynamic instabilities, together with solutions that can be implemented, such as artificial viscosity and thermal conductivity.

We demonstrate the effectiveness of the various solutions using a variety of tests: supersonic collisions, the Sod shock tube, the Kelvin-Helmholtz instability and the Sedov blast wave.

6.1 The Need for Viscosity

The general solutions to the equations of fluid dynamics are waves, and it is possible for these to form shock waves where the solution becomes discontinous. As described in section 4.2.3 this breaches the assumptions used to formulate the differential equations. In reality, physical quantities are rarely discontinuous since any discontinuity would be smoothed out through the action of viscosity. This is not the case, however, when the inviscid Euler momentum equation is used. Shock waves in SPH simulations will therefore not be handled correctly, since without viscosity there will be no dissipation: particles will not be able to form a layer to represent the shock front since they will pass each other without slowing down. Figure 6.1 shows this effect when two volumes of gas collide. An SPH simulation such as this does in fact provide a model of real viscosity: being a particle-based method, a shock front will form with a width given by the particle mean free path. An SPH particle is essentially a large ensemble of real fluid particles, however, so its mean free path is much too long, allowing the excessive interpenetration to occur.

The handling of shocks can be remedied by using the Navier-Stokes equation to incorporate viscosity (Watkins 1996), but this can be difficult to implement in SPH so an artificial viscosity is often introduced instead (Gingold & Monaghan 1983). Grid-based codes also often use artificial viscosity to handle shocks (Mathews & Bodenheimer 1964), although it is possible to instead implement a piecewise parabolic method (Collela & Woodward 1984) in order to use the solution of a Riemann problem to calculate the fluxes at each cell interface. The related approach in particle-based hydrodynamics is Godunov particle hydrodynamics (Inutsuka 1994; Cha & Whitworth 2003), which uses a Riemann solver to calculate accelerations and

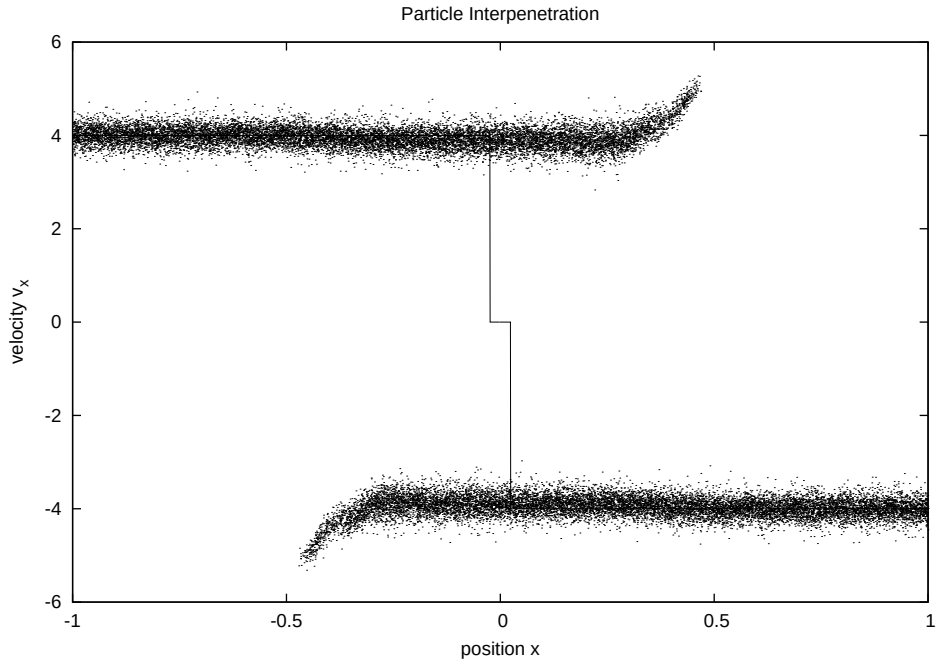


Figure 6.1: Velocity profile of a supersonic collision between two flows (see Section 6.5.1 for details) at time $t = 0.1$. The dots indicate the SPH particles, while the line inidicates the Riemann solution (see Section 6.2). Upon collision a thin shock layer should have developed but instead the flows have interpenetrated each other.

resolves shocks adequately without the need to resort to artificial viscosity.

6.2 The Riemann Solution

A Riemann problem consists of a conservation law with piecewise constant data that possesses a single discontinuity. Its solution allows us to determine the properties of shocks (and rarefaction waves) in the Euler equations.

A one-dimensional hydrodynamic Riemann problem can be represented as a tube of gas extending from x_L to x_R with the discontinuity located at the mid-point x_0. To the left of x_0 the gas properties are (ρ_L, v_L) for density and velocity, and to the right of x_0 they are (ρ_R, v_R).

An exact Riemann solution can be obtained for the properties $(\rho_\star, v_\star)$ at the discontinuity x_0. The Lagrangian continuity equation in one dimension is

$$\frac{D\rho}{Dt} + \rho\frac{\partial v}{\partial x} = 0 \tag{6.1}$$

and the Euler momentum equation (in the absence of gravity) is

$$\frac{Dv}{Dt} + \frac{1}{\rho}\frac{\partial P}{\partial x} = 0 \tag{6.2}$$

which, using the isothermal equation of state $P = c^2\rho$ (where c is the isothermal sound speed), becomes

$$\frac{Dv}{Dt} + \frac{c^2}{\rho}\frac{\partial \rho}{\partial x} = 0 \tag{6.3}$$

These two equations can be combined into one conservative equation

$$\frac{DU}{Dt} + A\frac{\partial U}{\partial x} = 0 \tag{6.4}$$

using a state vector $U = \begin{pmatrix} \rho \\ v \end{pmatrix}$ and matrix $A = \begin{pmatrix} 0 & \rho \\ \frac{c^2}{\rho} & 0 \end{pmatrix}$.

The eigenvalues of A are $\pm c$, and give the propagation speed of the gas (including that of any discontinuity), in this case the isothermal sound speed. The eigenvectors are $\begin{pmatrix} \rho \\ c \end{pmatrix}$ and $\begin{pmatrix} \rho \\ -c \end{pmatrix}$.

Decomposing the left state in terms of these eigenvectors,

$$U_L = \begin{pmatrix} \rho_L \\ v_L \end{pmatrix} = \alpha_L \begin{pmatrix} \rho \\ c \end{pmatrix} + \beta_L \begin{pmatrix} \rho \\ -c \end{pmatrix} \tag{6.5}$$

gives the solutions

$$\alpha_L = \frac{c\rho_L + \rho v_L}{2c\rho} \tag{6.6}$$

$$\beta_L = \frac{c\rho_L - \rho v_L}{2c\rho} \tag{6.7}$$

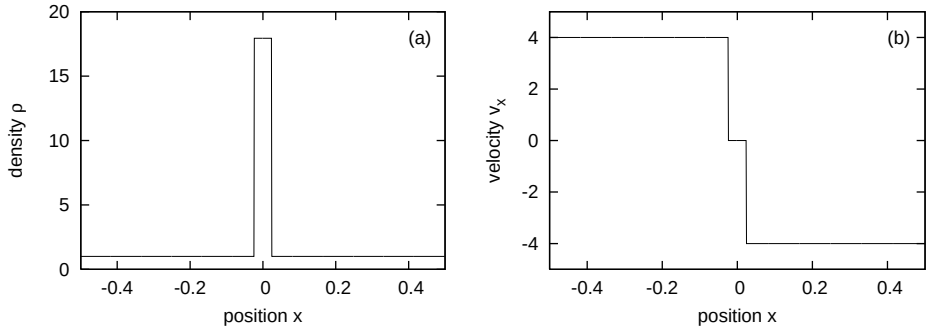


Figure 6.2: The Riemann solution for density ρ and velocity v_x at time $t = 0.1$ for the supersonic collision between two flows.

Similarly the right state,

$$U_R = \begin{pmatrix} \rho_R \\ v_R \end{pmatrix} = \alpha_R \begin{pmatrix} \rho \\ c \end{pmatrix} + \beta_R \begin{pmatrix} \rho \\ -c \end{pmatrix} \tag{6.8}$$

gives the solutions

$$\alpha_R = \frac{c\rho_R + \rho v_R}{2c\rho} \tag{6.9}$$

$$\beta_R = \frac{c\rho_R - \rho v_R}{2c\rho} \tag{6.10}$$

The solution at the interface between U_L and U_R can now be determined as

$$U_\star = \begin{pmatrix} \rho_\star \\ v_\star \end{pmatrix} = \alpha_L \begin{pmatrix} \rho \\ c \end{pmatrix} + \beta_R \begin{pmatrix} \rho \\ -c \end{pmatrix} \tag{6.11}$$

This forms the basis for numerical Riemann solvers, which find a solution for $U_\star$ (by iteration from an initial educated guess using the Newton-Raphson method) and then use this to determine a solution for $U(x)$ at time t across a discretised domain of grid spacings Δx between x_L and x_R (Toro 1989). Figure 6.2 shows the solution obtained for a supersonic collision, demonstrating the formation of a shock layer.

6.3 Artificial Viscosity

Fluid properties in SPH are assumed to vary smoothly on the smallest length scale (the smoothing length h) so discontinuities cannot be resolved at such scales. By introducing a small amount of viscosity the shock front will be able to spread out until it can be sufficiently resolved (Von Neumann & Richtmyer 1950). The drawback of this artificial viscosity is that it can produce excess heating elsewhere in the simulation. Its greatest advantage is its simplicity, allowing it to be easily implemented and combined with other constitutive physics at very little computational expense, unlike Riemann solvers.

Implementing artificial viscosity involves adding an extra term to the SPH momentum equation. This artificial viscous pressure term will ensure that kinetic energy is converted to thermal energy at the shock front, thus generating entropy and forming the contact discontinuity. The viscous term $\left(\frac{d\mathbf{v}_i}{dt}\right)_{\text{visc}}$ is simply added to the normal hydrodynamic acceleration $\left(\frac{d\mathbf{v}_i}{dt}\right)$ obtained from the equation of motion.

The nature of the extra dissipative term can be based upon the Riemann solver (Monaghan 1997).

$$\left(\frac{d\mathbf{v}_i}{dt}\right)_{\text{visc}} = \sum_{j=1}^{N} m_j \frac{\alpha v_{\text{sig}} \mathbf{v}_{ij} \cdot \hat{\mathbf{r}}_{ij}}{\overline{\rho}_{ij}} \overline{\nabla_i W}(\mathbf{r}_{ij}, h_i, h_j) \tag{6.12}$$

where the mean kernel derivative is

$$\overline{\nabla_i W}(\mathbf{r}_{ij}, h_i, h_j) = \frac{\nabla_i W(\mathbf{r}_{ij}, h_i) + \nabla_i W(\mathbf{r}_{ij}, h_j)}{2} \tag{6.13}$$

and $v_{\text{sig}} = c_i + c_j - \mathbf{v}_{ij} \cdot \hat{\mathbf{r}}_{ij}$ is the maximum signal velocity between the two particles, with c_i being the sound speed.

Here the symmetrisation between particles has been achieved via mean kernel W rather than mean smoothing length h. This should match the symmetrisation used in the kernel-softened gravitational interactions between particles (see Section 4.3.8). The mean smoothing length method would instead use $\nabla_i W\left(\mathbf{r}_{ij}, \frac{h_i+h_j}{2}\right)$.

The new parameter α is a numerical factor that determines how much artificial viscosity is applied. A value of $\alpha = 0.5$ is typical for handling shocks.

In the past, artificial viscosity has often been implemented as a two-parameter $\alpha\beta$ formulation (Monaghan & Gingold 1983). While this works in resolving shocks it has less physical basis than the more recent formulation above, which can also be generalised to model dissipation in other quantities such as thermal conductivity (see Section 6.4).

For comparison, the $\alpha\beta$ formulation uses

$$\left(\frac{d\mathbf{v}_i}{dt}\right)_{\text{visc}} = -\sum_{j=1}^{N} m_j \frac{-\alpha \overline{c}_{ij} \mu_{ij} + \beta \mu_{ij}^2}{\overline{\rho}_{ij}} \overline{\nabla_i W}(\mathbf{r}_{ij}, h_i, h_j) \tag{6.14}$$

where

$$\mu_{ij} = \frac{\overline{h}_{ij} \mathbf{v}_{ij} \cdot \hat{\mathbf{r}}_{ij}}{r_{ij}^2 + \eta^2} \tag{6.15}$$

and $\eta = 0.1\overline{h}_{ij}$. It is often only applied when the particles are approaching each other ($\mathbf{v}_{ij} \cdot \hat{\mathbf{r}}_{ij} < 0$) and set to zero otherwise. Values of $\alpha = 1$ and $\beta = 2$ are typical for handling shocks.

6.3.1 Artificial Viscosity Switches

In order to resolve shock fronts, artificial viscosity must be highly dissipative; it therefore converts kinetic energy into thermal energy more rapidly than would be

the case with physical (molecular) viscosity. This is a problem when the evolution of velocity is critical to the outcome of a simulation, as is the case in turbulent simulations. In the case of disc simulations it will also result in the transportation of material and angular momentum through the effective shear viscosity (Bate 2000).

Methods have been developed to limit the artificial viscosity when it is not needed while still preserving its capacity to resolve shocks.

One such method is a "switch" (Balsara 1995). A dimensionless factor f is calculated for each particle, and this is symmetrically incorporated into the dissipative term by multiplying the viscosity parameter α by $\frac{f_i+f_j}{2}$.

$$f_i = \frac{|\nabla \cdot \mathbf{v}|_i}{|\nabla \cdot \mathbf{v}|_i + |\nabla \times \mathbf{v}|_i + 0.001(c_i/h_i)} \tag{6.16}$$

In strong compression regions (shocks) the divergence terms ($|\nabla \cdot \mathbf{v}|$) dominate over vorticity terms ($|\nabla \times \mathbf{v}|$) so $f_i \to 1$ and artificial viscosity is invoked to resolve the shock front. In vortex regions (such as shear flows) the situation is reversed and $f_i \to 0$ switches off the artificial viscosity.

Another method, which can be used in combination with the Balsara switch, is time-dependent viscosity (Morris & Monaghan 1997). Each particle now possesses its own value of α which is permitted to vary in time. This will typically be a low value ($\alpha_{\rm MIN} \simeq 0.1$) in order reduce unwanted dissipation, but non-zero in order to maintain particle order (the presence of some artificial viscosity is often required to damp out high-frequency particle noise). The value of α_i then evolves according to the differential equation

$$\frac{d\alpha_i}{dt} = -\frac{\alpha_i - \alpha_{\rm MIN}}{\tau_i} + S_i \tag{6.17}$$

where $\tau_i = \mathcal{C}h_i/c_i$ is a decay time for particle i (typically $\mathcal{C} \sim 5$, making τ approximately the sound-crossing time through the smoothing sphere). The source term S_i is of the form

$$S_i = {\rm MAX}\left\{-\langle \nabla \cdot \mathbf{v} \rangle_i, 0\right\}(\alpha_{\rm MAX} - \alpha_i) \tag{6.18}$$

This ensures that the source term is zero for static or expanding fluids but has a positive value for converging fluids ($\nabla \cdot \mathbf{v} < 0$). In strong compression regions α_i will therefore increase towards $\alpha_{\rm MAX}$ (which is typically set to 1), while when the source term is zero or negligible the value of α will decay exponentially towards the minimum value $\alpha_{\rm MIN}$. When the $\alpha\beta$ formulation is used, β is not varied independently but is set to $\beta_i = 2\alpha_i$.

Another possible method is a pattern-matching switch (Cartwright & Stamatellos 2010) designed to counteract any tendency for SPH estimates of $\nabla \cdot \mathbf{v}$ to find particle convergence in regions of steady shear flow. This switch is very effective in pure Keplerian discs (non-self-gravitating equilibrium discs modelled in the frame of reference of the central star) but has not yet been adapted to work in more general situations.

6.3.2 The Courant-Friedrichs-Lewy Condition

The local conditions on the timestep based upon acceleration ($\Delta t = \sqrt{h/(|a| + \epsilon)}$) and, when integrating the energy equation directly, internal energy ($\Delta t = u/(\frac{du}{dt} + \epsilon)$),

remain unchanged. The small number ϵ is included simply to prevent division by zero.

The Courant-Friedrichs-Lewy condition based upon velocity ($\Delta t < \frac{h}{v_{sig}}$), however, requires some modification when artificial viscosity is used to account for the effects of this dissipation term.

Previously the signal velocity $v_{sig} = c_s + h|\nabla \cdot \mathbf{v}|$ was used, accounting for both the local sound speed and the relative motion of the particles. When applying artificial viscosity an additional term should be included:

$$\begin{aligned} v_{sig} &= c_s + h|\nabla \cdot \mathbf{v}| + F_{visc}\alpha c_s \\ &= (1 + F_{visc}\alpha)\, c_s + h|\nabla \cdot \mathbf{v}| \end{aligned} \tag{6.19}$$

where the scaling parameter $F_{visc} = 1.2$ is typically used.

When using the $\alpha\beta$ formulation:

$$\begin{aligned} v_{sig} &= c_s + h|\nabla \cdot \mathbf{v}| + F_{visc}(\alpha c_s + \beta h|\nabla \cdot \mathbf{v}|) \\ &= (1 + F_{visc}\alpha)\, c_s + (1 + F_{visc}\beta)\, h|\nabla \cdot \mathbf{v}| \end{aligned} \tag{6.20}$$

6.4 Artificial Thermal Conductivity

An extra dissipation term can be included in the SPH energy equation in a similar manner to artificial viscosity, where it has the effect of introducing artificial thermal conductivity.

$$\left(\frac{du_i}{dt}\right)_{\text{cond}} = -\sum_{j=1}^{N} \frac{m_j}{\overline{\rho}_{ij}} \left[\frac{\alpha(\mathbf{v}_{ij} \cdot \hat{\mathbf{r}}_{ij})^2}{2} + \alpha' v'_{\text{sig}}(u_i - u_j)\right] \hat{\mathbf{r}}_{ij} \cdot \overline{\nabla_i W}(\mathbf{r}_{ij}, h_i, h_j) \tag{6.21}$$

where the mean kernel derivative $\overline{\nabla_i W}$ is used as before (see Section 6.3).

This dissipation formulation contains two terms: the first due to differences in momentum between particles (a viscous heating term that converts kinetic energy to thermal energy) and the second due to differences in specific internal energy between particles (an artificial thermal conductivity term). This conductivity term will act to smooth out discontinuities in specific internal energy, conducting energy from high energy regions to low energy regions.

The new parameter α' is a numerical factor of order unity that determines how much artificial thermal conductivity is applied.

A complication introduced by the artificial conductivity term is that the choice of signal velocity v_{sig} for the artificial viscosity term may not be applicable to the artificial conductivity, so it is advisable to use different signal velocities for the viscosity and conductivity terms (v_{sig} and v'_{sig}).

For purely hydrodynamic simulations the recommended signal velocity for the artificial conductivity term is (Price 2008)

$$v'_{\text{sig}} = \sqrt{\frac{|P_i - P_j|}{\overline{\rho}_{ij}}} \tag{6.22}$$

Equation 6.22 can cause problems in gravitational simulations, however, where thermal energy may be conducted away from dense condensations faster than expected, causing any gravitational collapse to be artificially accelerated. A more suitable alternative when self-gravity is included is therefore (Wadsley et al. 2008)

$$v'_{\rm sig} = |\mathbf{v}_{ij} \cdot \hat{\mathbf{r}}_{ij}| \tag{6.23}$$

In order to ensure that artificial thermal conductivity is only applied at steep thermal energy gradients a time-dependent formulation may be applied (Price 2004), in a similar manner to artificial viscosity but using a source term

$$S' = |\nabla\sqrt{u}| = \left|\frac{\nabla u}{2\sqrt{u}}\right| \tag{6.24}$$

where

$$\nabla u_i = \frac{1}{\rho_i}\sum_j m_j(u_i - u_j)\nabla_i W_{ij}(h_i) \tag{6.25}$$

6.5 Tests of Artificial Dissipation

We can now test the effectiveness of these dissipation terms. The artificial viscosity term will be tested using supersonic collisions and the Sod shock tube. The artificial thermal conductivity term will be tested using the Sod shock tube and the Kelvin-Helmholtz instability. We will also use the Sedov blast wave to demonstrate another problem and its solution.

6.5.1 Supersonic Collisions

A simple shock tube test to examine the ability of a code to capture strong (high Mach number) shocks is a colliding flows test. It essentially demonstrates how effective the artificial viscosity is in resolving strong shocks. This is particularly important in SPH since an inadequate prescription of artificial viscosity can lead to particle penetration and thus no shock layer (see Section 6.1).

In this test an isothermal equation of state is imposed with a dimensionless sound speed $c_s = 1$. We therefore do not need to solve the energy equation or include artificial conductivity.

The initial conditions used for the colliding flows test consist of a shock tube containing two flows of uniform density gas which collide and form a shock at the contact boundary located at $x = 0$. The dimensions of the shock tube are $8 \times 1 \times 1$, stretching from $x = -4$ to $x = +4$ and from 0 to +1 in the y and z dimensions. Periodic boundary conditions are employed in the y and z directions but not the x direction. Each side of the shock tube contains $40,000$ equal-mass particles, with the left-hand side moving with an initial velocity of $v_x = +4$ and the right-hand side at $v_x = -4$, thus initiating a supersonic (Mach number $\mathcal{M} \sim 8$) collision shock. The particles are placed using a Halton sequence to reduce the effect of noise on the density and forces. The simulation uses global timesteps and is terminated at time $t = 0.6$.

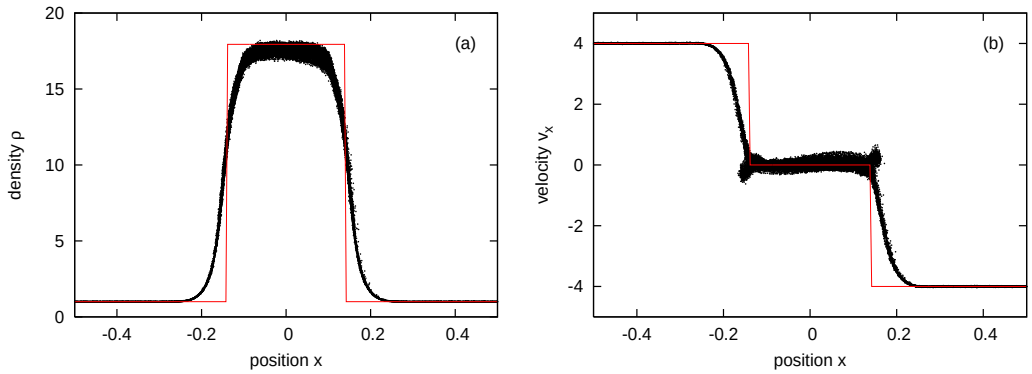


Figure 6.3: The colliding flows test using artificial viscosity ($\alpha = 1.0$) showing (a) the density ρ and (b) the velocity v_x after a time $t = 0.6$. The dots represent the results from the SPH simulation and the line shows the Riemann solution.

Figure 6.3 shows the SPH density ρ and velocity v_x as a function of x together with the one-dimensional Riemann solution. The peak density and the width of the shock is in good agreement with the Riemann solution, but the density discontinuity is not strictly resolved and is spread over several smoothing lengths. This is also visible in the velocity profile. This happens because the SPH artificial viscosity is computed from other smoothed quantities, which effectively smoothes out the deceleration over several smoothing lengths rather than providing the instantaneous deceleration that the shock jump conditions demand. The particles do not decelerate to exactly zero in the shock, having a residual velocity dispersion of $\sim 0.2c_s$ about zero due to errors from the SPH summation approximation and the time integration scheme. Overall, however, the test gives good results (favourable, in fact, when compared with results obtained from Godunov particle hydrodynamic simulations), which provides us with confidence in the ability of artificial viscosity to capture a shock.

Reasonably similar, although somewhat noiser, results can be obtained with $\alpha = 0.5$, but $\alpha = 0.1$ only marginally resolves the shock layer, spreading it over twice its expected width and thus providing more of a qualitative rather than quantitative capture of the shock behaviour. With $\alpha = 0.05$ the flows interpenetrate but are gradually decelerated, while with $\alpha = 0.01$ the flows interpenetrate almost freely. A minimum value of $\alpha = 0.5$ is therefore recommended for the quantitative capture of shocks.

6.5.2 The Sod Shock Tube

The adiabatic Sod test (Sod 1978) is a version of the classic shock tube test used to validate the ability of hydrodynamical codes to resolve shocks.

This shock tube test consists of two volumes of gas, one at a higher density and pressure than the other, which then interact at their interface to form a shock. The gas is assumed to behave adiabatically at the interface (compressional heating occurs and no energy is lost from the system) so both the SPH momentum and energy equations must be solved, which requires the use of both artificial viscosity and artificial thermal

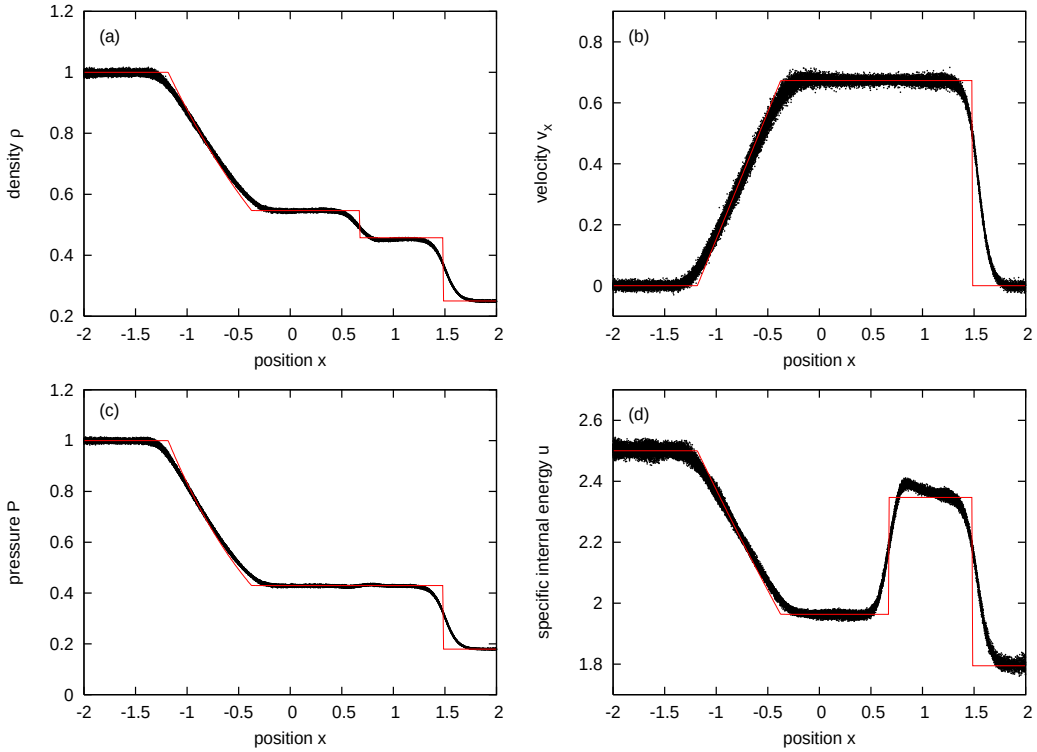


Figure 6.4: The adiabatic shock test using artificial viscosity and thermal conductivity, showing (a) the density ρ, (b) the velocity v_x, (c) the thermal pressure P and (d) the specific internal energy u after a time $t = 1.0$. The dots represent the results from the SPH simulation and the line shows the Riemann solution.

conductivity (to handle the discontinuities in velocity and temperature respectively).

The initial conditions used for the adiabatic Sod test again consist of a shock tube of dimensions $8 \times 1 \times 1$, stretching from $x = -4$ to $x = +4$ and 0 to $+1$ in the y and z directions. Periodic boundary conditions are employed in all directions, so a shock will be formed at both $x = 0$ and $x = \pm 4$, with a shock wave propagating into the low-density gas and a rarefaction wave propagating into the high-density gas. The left-hand side of the shock tube ($x < 0$) contains $64,000$ equal-mass particles and the right-hand side ($x > 0$) contains $16,000$ equal-mass particles (giving one quarter density of the left-hand side). The ratio of specific heats for the gas on both sides of the shock tube is $\gamma = 7/5$ (diatomic gas). The pressure on the right-hand side is set to be lower than the pressure on the left-hand side by a factor of 0.1795, with the initial gas temperature T on each side given by $T = P/\rho$. The test is dimensionless and is run until a time of $t = 1.0$ using global timesteps.

Figure 6.4 shows the profiles of the SPH density ρ, velocity v_x, thermal pressure P and specific internal energy u as a function of x together with the one-dimensional Riemann solution. The SPH results follow the Riemann solution very closely but, due to the fundamental role of smoothing in SPH, the discontinuities are spread across several smoothing lengths.

Figure 6.5 shows that, without artificial thermal conductivity, the SPH results

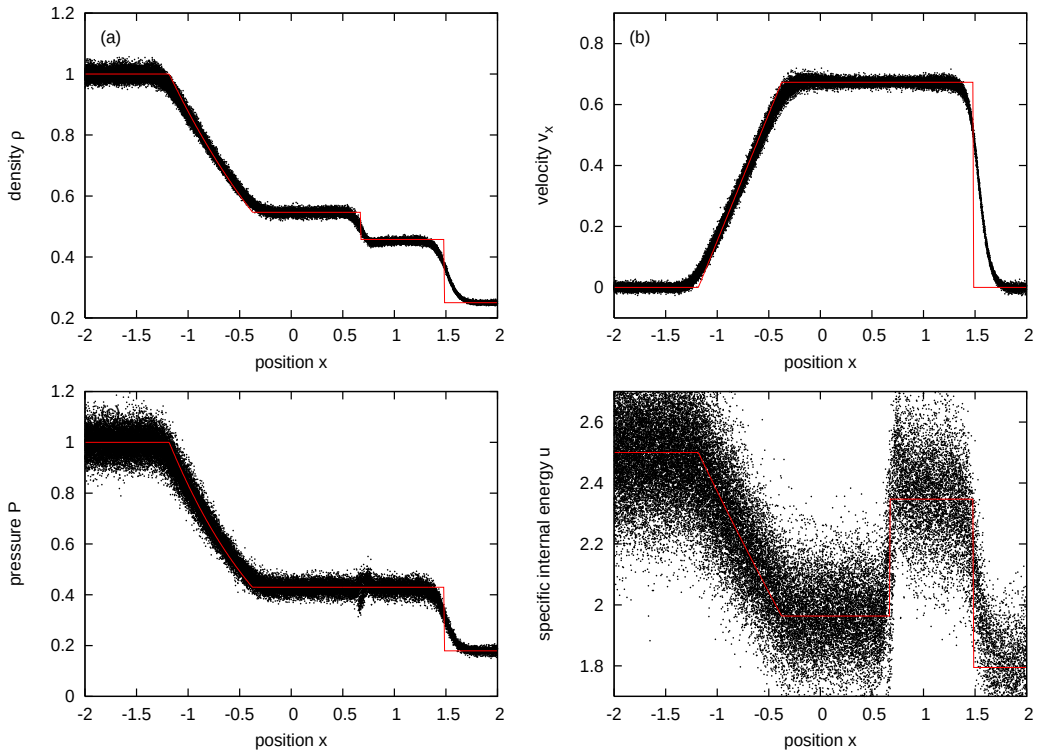


Figure 6.5: The adiabatic shock test using artificial viscosity without thermal conductivity, showing (a) the density ρ, (b) the velocity v_x, (c) the thermal pressure P and (d) the specific internal energy u after a time $t = 1.0$. The dots represent the results from the SPH simulation and the line shows the Riemann solution.

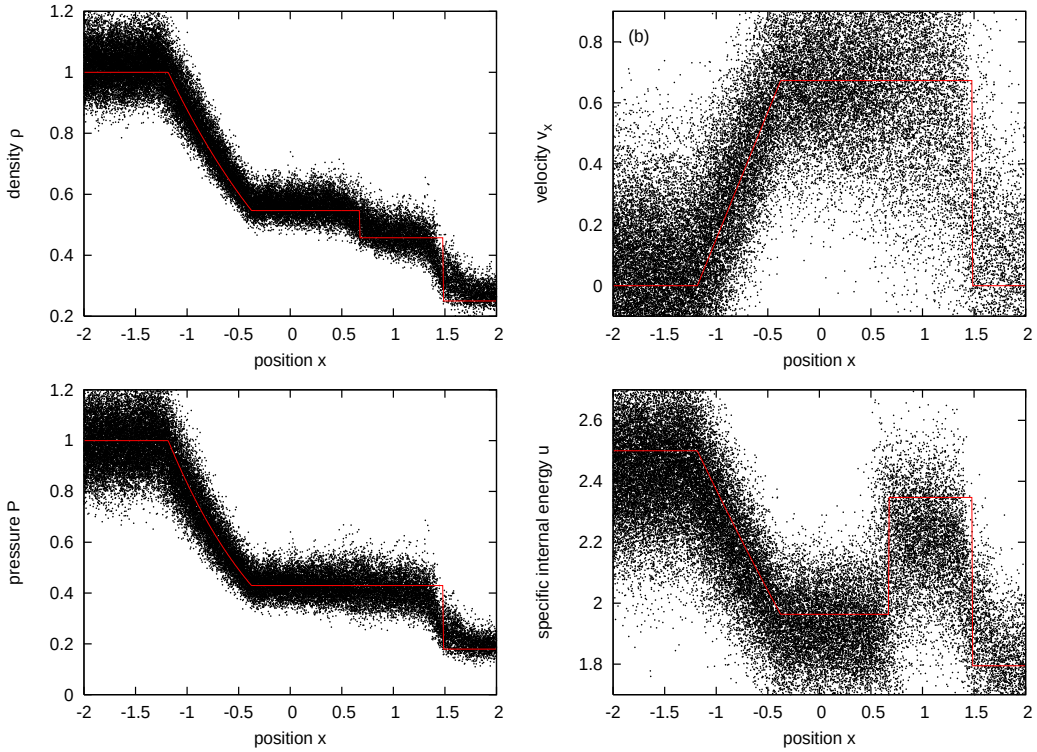


Figure 6.6: The adiabatic shock test without artificial viscosity or thermal conductivity, showing (a) the density ρ, (b) the velocity v_x, (c) the thermal pressure P and (d) the specific internal energy u after a time $t = 1.0$. The dots represent the results from the SPH simulation and the line shows the Riemann solution.

are noiser, particularly the specific internal energy u, and there is a small anomaly in the pressure P at position $x = 0.6$. This is essentially due to the noiser and more substantial overshoot in the thermal energy at the position of the contact discontinuity (Price 2008). The artificial conductivity corrects this, providing almost constant pressure across the contact discontinuity.

Figure 6.6 shows, for completeness, that without artificial viscosity or thermal conductivity the SPH results are exceptionally noisy.

6.5.3 The Kelvin-Helmholtz Instability

The Kelvin-Helmholtz instability occurs, in the simplest case, between two bulk flows that are shearing past each another. It has been extensively studied in recent years as a diagnostic to compare the ability of SPH and grid codes to model the mixing of interacting fluids. In particular this test highlighted an intrinsic problem in the SPH method (Agertz et al. 2007), namely spurious pressure forces in regions with steep density gradients leading to overly-damped interactions, and this has led to several suggested improvements of SPH (Price 2008; Read et al. 2010).

We use similiar initial conditions to Price (2008), with equal-mass particles in a two-dimensional periodic box extending from -0.5 to $+0.5$, arranged on cubic

lattices such that an outer flow ($|y| > 0.25$, density $\rho_1 = 1$, velocity $v_x = -0.5$) is shearing against a thicker central flow ($|y| < 0.25$, density $\rho_2 = 2$, velocity $v_x = 0.5$). The two fluids are set in pressure balance ($P = 2.5$) and have a ratio of specific heats $\gamma = 5/3$ (monatomic gas), so there is a discontinuity in the specific internal energy $u = P/(\gamma - 1)\rho$ at the interfaces. This configuration is unstable to the Kelvin-Helmholtz instability at all wavelengths, and can be seeded at a wavelength λ by applying a small velocity perturbation in the y direction:

$$\begin{array}{rcll} v_y(x,y) & = & A\sin\left[-2\pi(x+0.5)/\lambda\right] & \text{for} \quad |y-0.25| < 0.025 \\ v_y(x,y) & = & A\sin\left[2\pi(x+0.5)/\lambda\right] & \text{for} \quad |y+0.25| < 0.025 \end{array} \tag{6.26}$$

A wavelength of $\lambda = 1/6$ and an amplitude of $A = 0.025$ are chosen.

At these subsonic velocities the growth timescale of the instability between the two shearing layers is

$$\tau_{KH} = \frac{2\pi}{\omega} = \frac{\rho_1 + \rho_2}{\sqrt{\rho_1 \rho_2}} \frac{\lambda}{|\Delta v_x|} \tag{6.27}$$

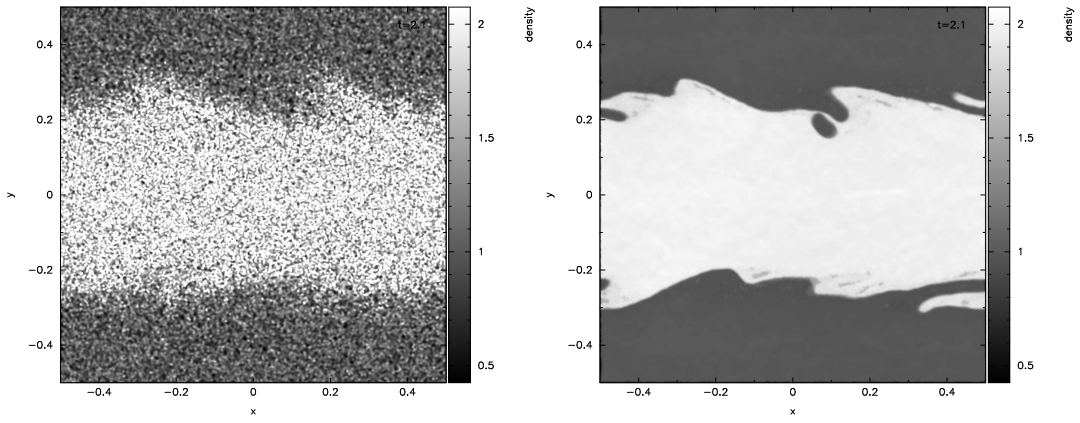
and since $|\Delta v_x| = 1$, $\tau_{KH} = 0.35$. We perform simulations using $73,670$ particles, which does not provide very high resolution but is sufficient to demonstrate the overall behaviour of the code. Improved results can be obtained at higher resolutions, and by using a quintic kernel in place of the cubic M4 kernel (Hubber, Batty, McLeod & Whitworth 2011).

We show the evolution of the density field and the development of the instability at $t = 6\tau_{KH}$, well into the non-linear phase where significant vorticity and mixing should occur near the shearing interface. In Figure 6.7 (a), without artificial viscosity or conductivity, the shearing motion has no effect and mixing occurs by outward diffusion alone since the instability cannot be captured. In Figure 6.7 (b), with artificial viscosity ($\alpha = 0.1$) but not conductivity, there is growth of the perturbation during the linear phase through the action of shear viscosity but the non-linear phase demonstrates that viscosity alone is insufficient to capture the instability. In Figure 6.7 (c), with both artificial viscosity and conductivity ($\alpha' = 1$), the characteristic vortices of the instability are captured. Figure 6.7 (d) demonstrates that artificial conductivity alone is sufficient to capture the instability, which suffers from damping when viscosity is present.

Similar tests may be performed to similar effect for other hydrodynamic instabilities, such as the Rayleigh-Taylor instability (at the interface of two fluid volumes of different densities that are being forced through each other, resulting in interpenetrating fingers) or the Richtmyer-Meshkov instability (when such an interface experiences the passage of a shock wave, resulting in penetrating spikes or bubbles as the heavier or lighter fluid penetrates the other respectively).

6.5.4 The Sedov Blast Wave

The Sedov blast wave test (Sedov 1959) demonstrates how well a code handles large energy contrasts originating from a point source of energy (an explosion). A point source of energy causes the formation of a shock front that travels radially outwards



(a) Without artificial viscosity or thermal conductivity.

(b) Including artificial viscosity ($\alpha = 0.1$) but not thermal conductivity.

(c) Including both artificial thermal conductivity ($\alpha' = 1$) and viscosity ($\alpha = 0.1$).

(d) Including artificial thermal conductivity ($\alpha' = 1$) but not viscosity.

Figure 6.7: The Kelvin-Helmholtz instability after time $t = 6\tau_{KH}$.

and builds up a dense layer of gas as it is swept up in the explosion. Sedov (1959) provides a semi-analytic solution to such problems.

A problem of SPH is that it can perform poorly when neighbouring particles have very different timesteps (Saitoh & Makino 2009). In a high Mach-number shock, for example, the particles may interpenetrate because particles from the low-density pre-shock gas have much longer timesteps than those in the high-density post-shock gas, and therefore in a single timestep they advance deep into the shocked region.

This effect can be mitigated by broadcasting the timestep of each particle (upon allocation) to all it neighbours. If one of the neighbours j of particle i has an allocated timestep which is more than two levels higher in the hierarchy ($t_j > 4t_i$) then their timestep is automatically reduced to $t_j = 4t_i$ as soon as the timestep hierarchy is correctly synchronised.

The Sedov blast wave test can demonstrate the robustness of the block timestepping scheme (See Section 5.4) and, in particular, that it can correctly simulate scenarios that have very large differences in timesteps between neighbouring particles. We set up the test in a similar manner to Saitoh & Makino (2009), with the SPH particles initially relaxed into a glass-like configuration. The problem assumes a central point source of energy which then propagates outwards. In SPH it is more appropriate to spread the energy amongst all particles within the smoothing kernel volume surrounding the source. We also effectively "pre-smooth" the energies of the central particles by distributing the energy amongst the particles in a weighted manner using the smoothing kernel. This removes the discontinuity in specific internal energy between the central "hot" particles (total thermal energy $\Sigma U = 1$) and the surrounding "cold" particles (total thermal energy $\Sigma U < 10^{-6}$). Otherwise we would require artificial conductivity, which in effect smooths the discontinuity during the simulation but also further reduces the minimum timestep.

Three simulations of the Sedov blast wave are performed using different timestep algorithms (Hubber, Batty, McLeod & Whitworth 2011): (a) global timesteps, (b) block timesteps and (c) block timesteps using the neighbour-checking procedure (Saitoh & Makino 2009).

The density profiles of the three simulations at time $t = 0.02$ are shown in Figure 6.8. The simulation with global timesteps (Figure 6.8a) shows agreement with the semi-analytic solution. The position of the advancing shock front is at approximately the same radius as the semi-analytic solution and its width is comparable. The peak density is smaller than the semi-analytic solution, however, partly due to the fact that SPH smoothing on any density peak tends to produce a value smaller than the true peak density. When block timesteps are used (Figure 6.8b) the density profile is incorrect. This is because the timesteps of the hot and cold gases are very different (by a factor of $\sim 10^3$) so the cold particles cannot respond to the pressure of the rapidly advancing shock front (since it is operating on an inadequate timestep and therefore does not compute a new acceleration until it is too late), allowing the hot particles to penetrate through them. Introducing the neighbour-checking algorithm (Figure 6.8c) provides a timestep limiter (no particle is allowed to have a timestep more than four times longer than its neighbours) which corrects this problem. The density profile now strongly resembles the global timestep case but benefits from the speed-up factor of block timestepping (over an order of magnitude faster than a global

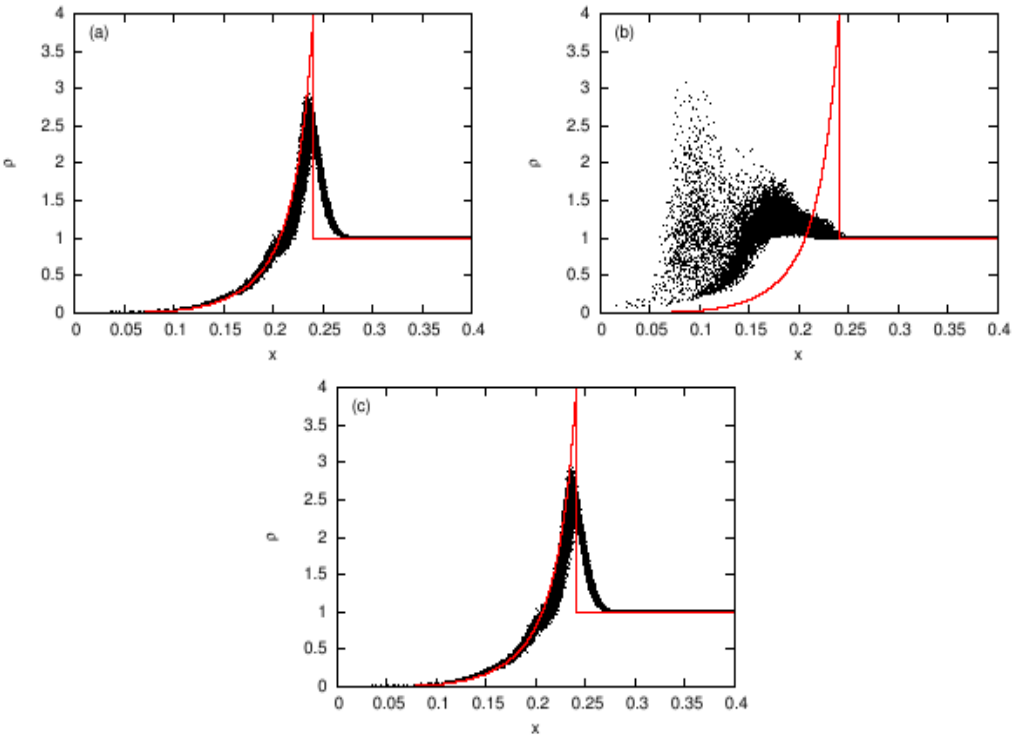


Figure 6.8: The Sedov blast wave test using (a) global timesteps, (b) block timesteps and (c) block timesteps with neighbour-checking (Saitoh & Makino 2009). The dots represent the results from the SPH simulation and the line shows the semi-analytic solution (Sedov 1959).

timestep).

6.6 Summary

Some of the common problems associated with hydrodynamic simulations, such as resolving shocks and hydrodynamic instabilities, can be addressed with suitable modifications such as artificial viscosity, artificial thermal conductivity and neighbour timestep checking.

The effectiveness of these modifications has been demonstrated in various tests: supersonic collisions, the adiabatic Sod shock tube, the Kelvin-Helmholtz instability and the Sedov blast wave.

In the next chapter we will look in detail at how artificial viscosity in SPH models the effects of shear viscosity, since this has particular relevance to disc simulations.

"Do not push the river,
it will flow by itself."

Polish proverb

Chapter 7

Shear Flows

In this chapter we look in detail at how artificial viscosity in SPH models the effects of shear viscosity.

Differential rotation within a circumstellar disc results in the occurrence of shear flows, so ensuring that SPH correctly evolves such flows is a critical issue for disc simulations.

It has been claimed that SPH does not accurately evolve shear flows and rapidly results in the emergence of large density errors (Imaeda & Inutsuka 2002), but this claim of a fundamental flaw in SPH has been refuted with a study of several representative shear flows (Monaghan 2006).

We demonstrate correct viscous behaviour using a variety of tests: periodic shear flow, Couette flow, Poiseuille flow, cylindrical spin-down and ring spreading.

7.1 Incompressible Fluids

Certain problems in classical fluid dynamics that involve the shear flow of incompressible fluids have exact solutions; they therefore provide ideal tests for SPH simulations.

An incompressible fluid may be approximated by a slightly compressible fluid through the use of a stiff equation of state (Monaghan 2006).

$$P = \frac{c_s^2 \rho_0}{\gamma} \left[\left(\frac{\rho}{\rho_0} \right)^\gamma - 1 \right] \tag{7.1}$$

where the pressure P is essentially due to interatomic forces rather than thermal effects, c_s is the sound speed, ρ_0 is a reference density (normalised to 1) and γ is a constant (set to 7 to provide the stiffness for approximate incompressibility). This provides a reasonable approximation to a liquid such as water.

Setting the initial density uniformly to $\rho = 1.02$ provides a non-zero initial pressure. This is achieved by setting the mass of each particle $m_p = 1.02\frac{C}{N}$ where C is the computational volume and N is the number of particles.

Normalising the maximum flow speed to $V = 1$ and the sound speed to $c_s = \sqrt{50}$ gives a low Mach number $\mathcal{M} \sim 0.14$ and correspondingly small density fluctuations of $\frac{|\delta\rho|}{\rho_0} \sim \mathcal{M}^2 \sim 0.02$.

Any boundaries are simulated using fluid particles rather than free surfaces so that SPH density summation is possible.

7.2 Periodic Shear Flow

The chosen system for periodic shear flow is similar to the Cartesian shearing box considered by Imaeda & Inutsuka (2002) but the Mach number is lower (0.14 as opposed to 20). The flow is periodic in x and y using a rectangular 2×1 box, and the particle spacing is 0.04 on a square lattice. The initial velocity is $v_x = \sin(2\pi y)$ and $v_y = 0$. Since this velocity field maintains a steady state the density should remain at its initial value. Artificial viscosity is applied at a small value ($\alpha = 0.01$) to prevent excessive dissipation.

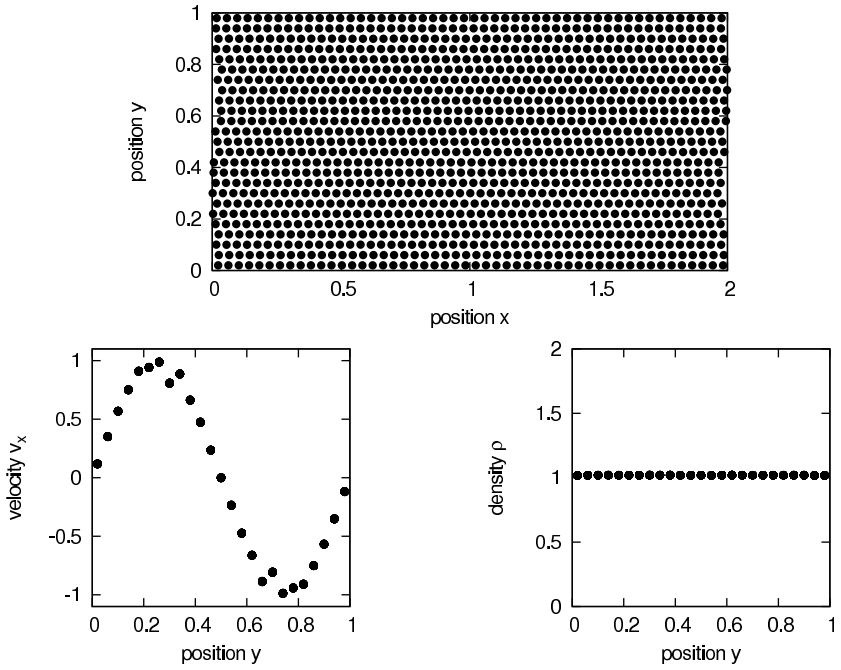


Figure 7.1: Low Mach number periodic shear flow at time $t = 2$, showing the particle positions (top), velocity profile $v_x(y)$ (bottom-left) and density profile $\rho(y)$ (bottom-right).

Figure 7.1 shows the particle positions, the variation of velocity v_x with y and the variation of density with y after one crossing of the box. The smoothing length is $h \sim 0.046$.

The particles have shifted from their square lattice arrangement, resulting in shear. Their velocities remain close to their initial values but the effect of viscosity has flattened the peaks. Their densities also remain very close to their initial uniform value of 1.02, with no evidence of any large fluctuations. Over several crossing times the flow gradually slows and settles through the action of viscosity, but the largest density fluctations are ~ 0.02 as expected. They settle towards a value slightly below the initial value, which is reasonable since the initial state of the system is not in equilibrium and so the particles settle into a lower energy configuration.

Imaeda & Inutsuka (2002) found large errors of $\Delta\rho \geq \rho$. When the details of their model are compared with our well-behaved sub-sonic simulation, however, these

large errors seem less surprising. Rather than modelling sub-sonic incompressible fluid, they instead use a super-sonic polytropic fluid (with exponent $\eta = 5/3$) and apply no artificial viscosity. Large density fluctuations can develop from instabilities growing within such a simulation.

Since the kinematic shear viscosity is of order $\nu \sim \frac{1}{8}\alpha c_s h$ (Monaghan 2005), the Reynolds number $\mathcal{R} = \frac{VL}{\nu}$ where V and L are characteristic velocity and length scales. In our simulation $\mathcal{R} \sim 2460$ and the flow is laminar. Applying no artificial viscosity (relying upon numerical viscosity alone) in the super-sonic simulation gives $\mathcal{R} > 10^6$ which allows turbulent motions to cascade down to very small scales. High artificial viscosity ($\alpha = 1.42$) is required to lower the Reynolds number to match our simulation. This is highly dissipative and after the initial shock waves are damped the particles settle towards a glass-like structure with a uniform density close to the initial value.

It is questionable to expect to model shear flow using highly super-sonic and turbulent initial conditions. Substantial artificial viscosity is required to reduce the Reynolds number below the critical turbulent threshold so that laminar viscous flow is possible. This is highly dissipative and resolves the shock fronts. The shock waves will first disrupt the flow and the dissipation will then damp the particles towards a glass-like configuration.

7.3 Couette Flow

Couette flow occurs within a viscous incompressible fluid that is initially stationary but lies between two parallel boundaries that move with different constant velocities. In this case $v_x = +1$ for the upper boundary and $v_x = -1$ for the lower boundary. The particles are again set up on a square lattice but with spacing 0.05. The boundaries are formed by three rows of particles and confine the system in the y-direction, but the system is periodic in the x-direction. Artificial viscosity is applied at a high value ($\alpha = 0.5$) in order to give strong initial response to shear.

The boundary particles must be given special treatment in the integration scheme, being restricted to integration in the x-direction with constant v_x. They must also be given special treatment when their smoothing lengths and densities are calculated. Since they lack neighbours on one side in the y-direction, their values should be set to match the fluid.

Figure 7.2 shows the particle positions, the variation of velocity v_x with y and the variation of density with y at time $t = 1$. The smoothing length is $h \sim 0.0575$.

The particles have again shifted from their square lattice arrangement, and shear is clearly present. The velocity profile is as expected, with the boundary velocities propagating into the initially stationary fluid through the action of viscosity. The density profile also remains very close to the initial uniform value of 1.02, with no evidence of any large fluctuations.

Couette flow can be solved analytically for its velocity evolution (Batchelor 1967) and for the system described above

$$v_x = -V(1-2y) + \frac{4V}{\pi} \sum_{j=2,4,6,\ldots}^{\infty} \frac{1}{j} \sin(j\pi y) \exp(-j^2 \nu \pi^2 t) \tag{7.2}$$

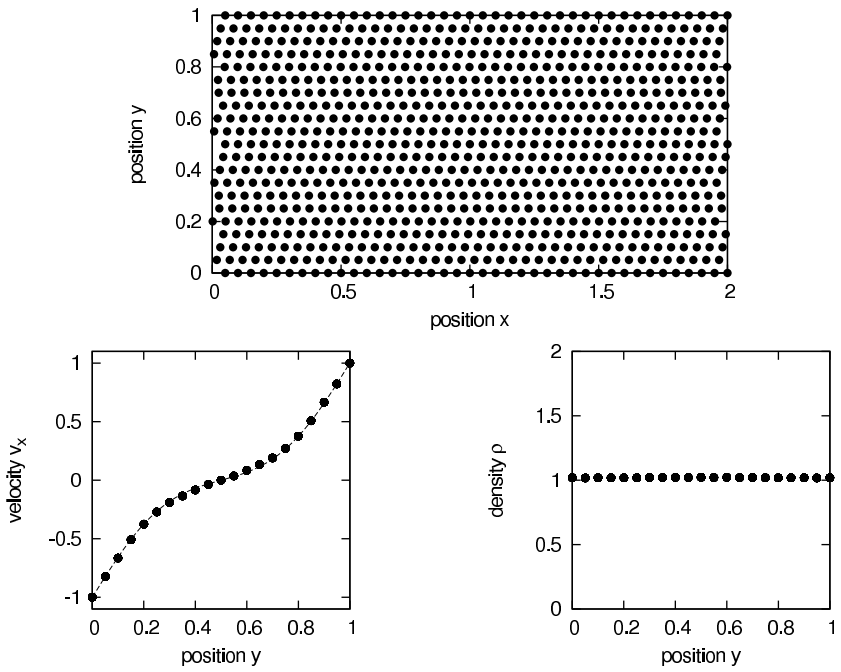


Figure 7.2: Low Mach number Couette flow at time $t = 1$, showing the particle positions (top), velocity profile $v_x(y)$ (bottom-left, with the line showing the analytic solution) and density profile $\rho(y)$ (bottom-right).

where $\nu = \frac{1}{8}\alpha c_s h$ is used for the kinematic shear viscosity (Monaghan 2005). The simulation is in very good agreement with this analytic solution. Lower and higher values of artificial viscosity ($\alpha = 0.25$ and 1) do not produce any unexpected result, and their agreement with the analytic solution is shown in Figure 7.3.

If the boundary velocities are highly super-sonic ($\mathcal{M} = 20$) then density errors develop and the analytic solution is not followed. Given that $\mathcal{R} \sim 5570$ the system is likely to be in turbulent flow or at least in transition flow (where both laminar and turbulent flows are possible) which helps to explain the results shown in Figure 7.4.

The original simulation is shown in Figure 7.5 at four different times ($t = 0.1$, 0.5, 1 and 1.5) to demonstrate that the analytic behaviour is followed throughout the time evolution. For the summation over j in Equation 7.2 the early time solutions require the inclusion of more terms ($j = 8$ is used for $t = 0.1$).

7.4 Poiseuille Flow

Poiseuille flow occurs within a viscous incompressible fluid that flows between two static parallel boundaries under the action of a constant uniform acceleration g in the direction of the flow. The initial conditions are the same as those for Couette flow.

The boundary particles are now held stationary in the integration scheme, and the fluid particles are subject to an additional acceleration term g in the x-direction. With g normalised to 1 to provide a reasonable flow velocity, the artificial viscosity must be applied at a very high value ($\alpha = 10$) in order to reduce the Reynolds number

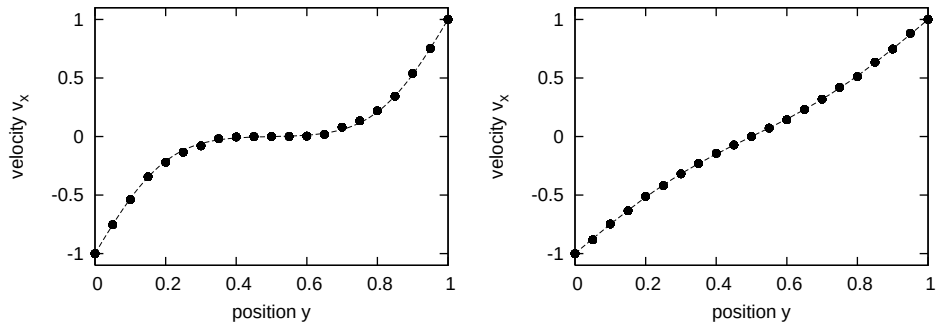


Figure 7.3: Low Mach number Couette flow at time $t = 1$, showing the velocity profile $v_x(y)$ for lower viscosity ($\alpha = 0.25$ on left) and higher viscosity ($\alpha = 1$ on right) with the lines showing the analytic solutions.

since turbulence can occur for $\mathcal{R} > 10$ in Poiseuille flow (Watkins 1996). Using $g = 1$ and $\alpha = 10$ ensures laminar flow ($\mathcal{R} < 1$) and prevents turbulent motions from affecting the results. Lower viscosity ($\alpha = 1$) initially provides similar results but gradually develops turbulent features that are indicative of transition flow.

Figure 7.6 shows the particle positions, the variation of velocity v_x with y and the variation of density with y at time $t = 10$. The smoothing length is $h \sim 0.0575$.

The particles have again shifted from their square lattice arrangement, and shear is clearly present. The velocity profile is as expected, with the stationary boundaries slowing the fluid flow through the action of viscosity. The density profile also remains very close to the initial uniform value of 1.02.

Poiseuille flow can be solved analytically for its steady-state velocity (Batchelor 1967) and for the system described above

$$v_x = \frac{g}{2\nu}\left[\left(\frac{1}{2}\right)^2 - \left(y - \frac{1}{2}\right)^2\right] \tag{7.3}$$

The simulation is in good agreement with this analytic steady-state solution, although slight oscillation occurs about it due to the competing influences of the driving acceleration g and very strong artificial viscosity.

7.5 Cylindrical Spin-Down

This system consists of a fluid in a cylindrical container that is initially rigidly rotating ($v_\theta = \Omega r$, with Ω set to 1).

The fluid particles are placed on 24 rings with a spacing of 0.05 between each ring (and approximately the same spacing between particles around each ring), with the cylindrical boundary represented by the four outermost rings of boundary particles which provide an inward pressure on the fluid enclosed within a radius $R = 1$. By setting the velocity of these boundary particles to zero in the integration scheme the spin-down process of the fluid is initiated. Artificial viscosity is again applied at a high value ($\alpha = 0.5$) for strong initial shear.

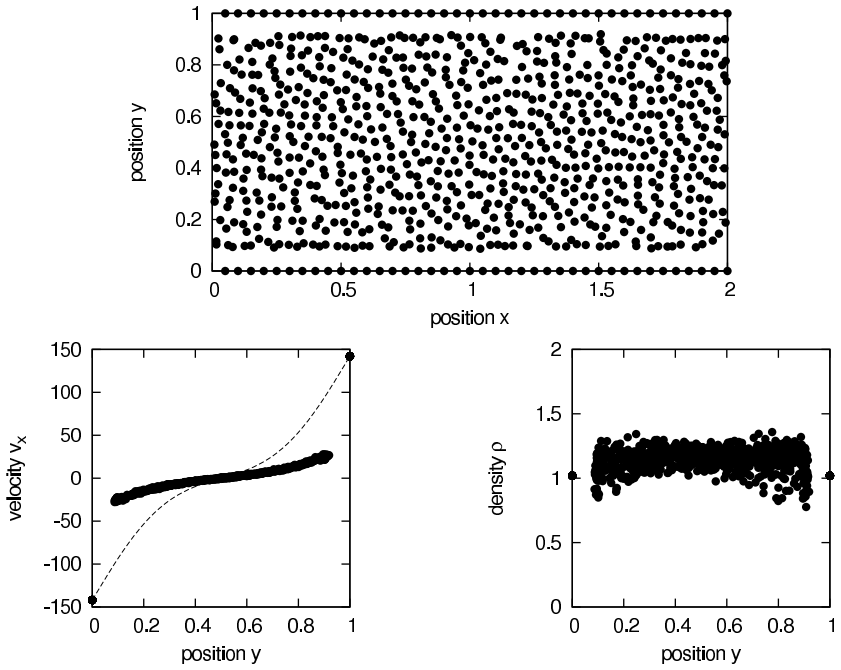


Figure 7.4: High Mach number Couette flow at time $t = 1$, showing the particle positions (top), velocity profile $v_x(y)$ (bottom-left, with the line showing the analytic solution) and density profile $\rho(y)$ (bottom-right).

Figure 7.7 shows the particle positions, the variation of velocity v_θ with r and the variation of density with r at time $t = 1$. The smoothing length is $h \sim 0.0575$. The velocity profile is as expected and the density profile remains very close to the initial uniform value of 1.02.

There is an analytical solution for the velocity evolution (Batchelor 1967) and for this system

$$v_\theta = -2\Omega R \sum_j \frac{J_1(\lambda_j r/R)}{\lambda_j J_0(\lambda_j)} \exp\left(-\frac{\lambda_j^2 \nu t}{R^2}\right) \tag{7.4}$$

where J_1 and J_0 are Bessel functions and λ_j is the jth zero of J_1.

The simulation is shown in Figure 7.8 at four different times ($t = 1$, 1.5, 2 and 2.5) to demonstrate that the analytic behaviour is closely followed throughout the time evolution.

7.6 Further Tests for Incompressible Fluids

There are not many exact solutions for the Navier-Stokes equations. Besides Couette flow and Poiseuille flow, another degenerate case (zero non-linear terms) is the oscillatory Stokes boundary layer (Batchelor 1967). This is formed close to a static boundary in an oscillatory flow, or by an oscillating boundary driving a stationary fluid, and can be solved for laminar flow (low Reynolds number). It might therefore be used as an additional test of viscous behaviour.

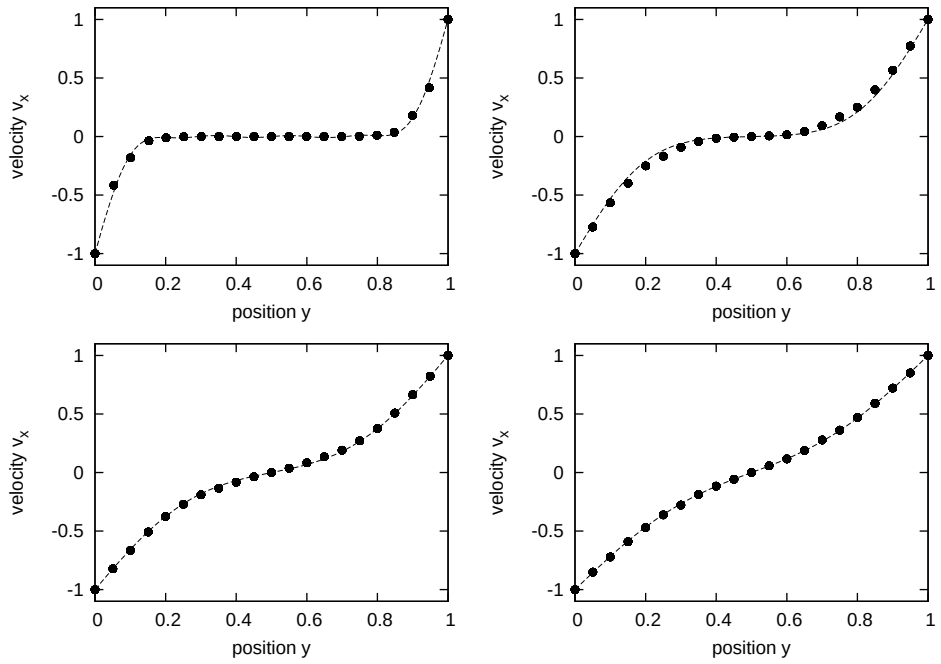


Figure 7.5: Low Mach number Couette flow at times $t = 0.1$ (top-left), 0.5 (top-right), 1.0 (bottom-left) and 1.5 (bottom-right), showing the velocity profile $v_x(y)$, with the lines showing the analytic solutions.

Solutions to the full non-linear equations do exist, such as the Taylor-Green vortex (Taylor & Green 1937). This is the two-dimensional unsteady flow of a decaying vortex; the non-linear terms balance the pressure exactly, so there is only viscous decay and no growth of harmonics. Its exact solution can be used to validate viscous behaviour.

Note that the existence of an exact solution does not imply stability. Turbulence can develop above a critical Reynolds number, disrupting the viscous behaviour. Nonetheless in the laminar regime these exact solutions provide the essential testing ground for viscosity algorithms.

7.7 Ring Spreading

There is a self-similar solution for a ring of material orbiting a central object (Lynden-Bell & Pringle 1974). This provides a shearing arrangement similar to a circumstellar disc and so provides a useful test.

For a ring of mass m at radius r_0 the surface density Σ is

$$\Sigma(x,\tau) = \left(\frac{m}{\pi r_0^2}\right) \tau^{-1} x^{-1/4} \exp\left(-\frac{(1+x^2)}{\tau}\right) I_{1/4}\left(\frac{2x}{\tau}\right) \tag{7.5}$$

where $x = r/r_0$ is the dimensionless radius, $\tau = 12\nu t/r_0^2$ is the dimensionless time and $I_{1/4}$ is a modified Bessel function (Pringle 1981).

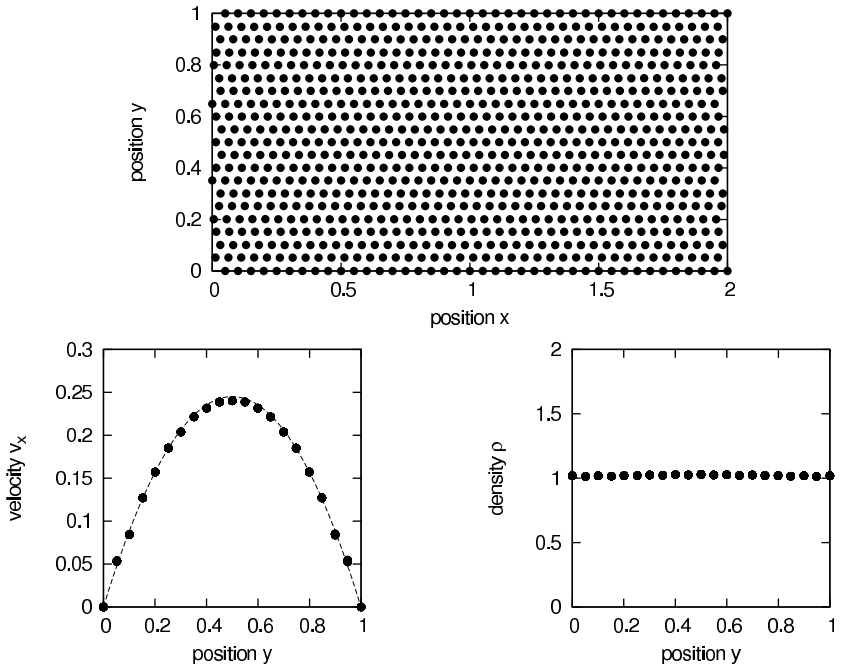


Figure 7.6: Low Mach number Poiseuille flow at time $t = 10$, showing the particle positions (top), velocity profile $v_x(y)$ (bottom-left, with the line showing the analytic solution) and density profile $\rho(y)$ (bottom-right).

A simulation can be performed by using an initial quasi-Monte Carlo distribution of particles in a thin annulus centred upon r_0 with a Gaussian surface density profile. This Gaussian profile can be obtained using a Box-Muller transform (see Equation 8.18 in Section 8.4.2). Since the ring begins with some thickness Δr the simulation is effectively beginning at some time $t > 0$, avoiding the singularity at $t = 0$.

The particle velocities are set in Keplerian rotation about a central sink particle of mass M. The mass of the ring $m \ll M$ so that the central gravity dominates and the self-gravity of the ring is negligible. The gas is isothermal and its temperature is set very low to limit the pressure forces. These conditions restrict the dominant forces acting on the particles to the central gravity and the artificial viscosity.

Figure 7.9 shows the analytic evolution (top-left) and the SPH simulation results (top-right). The SPH densities are binned by radius to reduce the noise from the quasi-Monte Carlo distribution of $10,000$ particles.

The qualitative evolution is quite similar but some differences exist. One reason for this is that the smoothing length h in our SPH formulation is not fixed but variable.

As the ring spreads the density decreases which therefore increases the smoothing length. This consequently increases the kinematic viscosity ν and the effective time τ. Later evolution therefore appears to occur more rapidly, making the peak densities lower at later times.

Since h also varies with r, this results in ν and τ increasing away from the

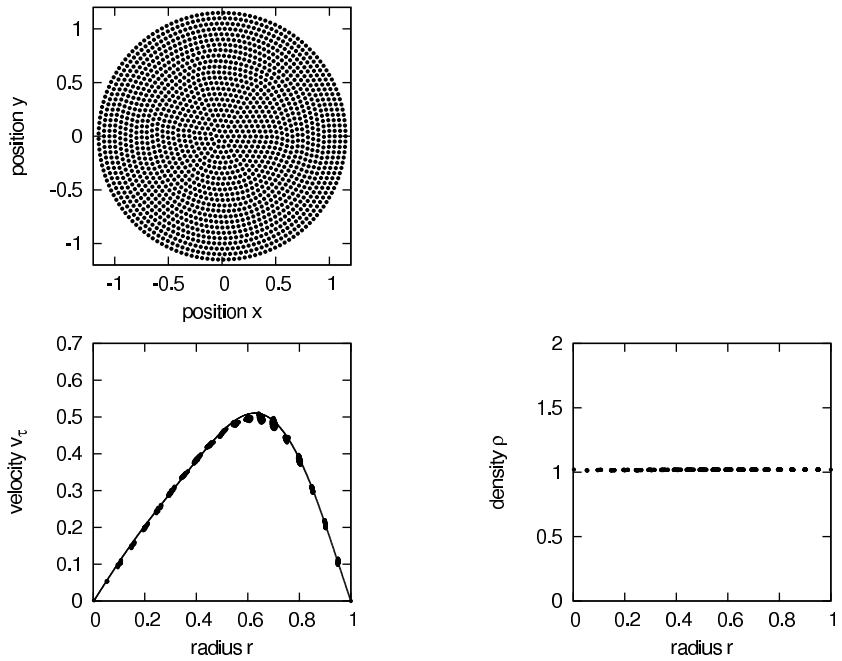


Figure 7.7: Low Mach number spin-down flow at time $t = 1$, showing the particle positions (top), velocity profile $v_\theta(r)$ (bottom-left, with the line showing the analytic solution) and density profile $\rho(r)$ (bottom-right).

centre of the ring. Densities will therefore appear to fall away more rapidly, making the width of the SPH solution narrower.

If variations in h are accounted for by normalising the SPH results in time and width at later t then the results improve (bottom-right). There are still minor discrepancies but these might be explained by the noisy Gaussian particle distribution, spreading due to pressure forces and the influence of edge effects (see Section 8.6). The overall evolution is nonetheless well-behaved, with no large density fluctuations despite high Mach numbers caused by the gas being very cold and by the presence of free boundaries (where the sound speed falls to zero so the Mach number will always be large).

7.8 Circumstellar Discs

All these results carry into simulations of differentially-rotating self-gravitating discs (Monaghan 2006). These are more complex simulations involving a self-gravitating compressible fluid with a non-uniform density profile and therefore do not possess analytic solutions. They nonetheless remain well-behaved, with the notable absence of large density fluctuations despite high Mach numbers due to edge effects.

Keplerian rotation in a circumstellar disc gives $V = 2\pi r^{-1/2}\mathrm{AUyr}^{-1}$. For a Sun-like star with a temperature profile of $T = 300r^{-1/2}\mathrm{K}$ the isothermal sound speed is $c_s = \sqrt{kT/\bar{m}} \simeq 0.22r^{-1/4}\mathrm{AUyr}^{-1}$ (assuming a solar nebula: $\bar{m} = 2.29m_H$). The Mach number is therefore $\mathcal{M} \simeq 28.6r^{-1/4}$, falling from ~ 9 at 100 AU to ~ 5 at

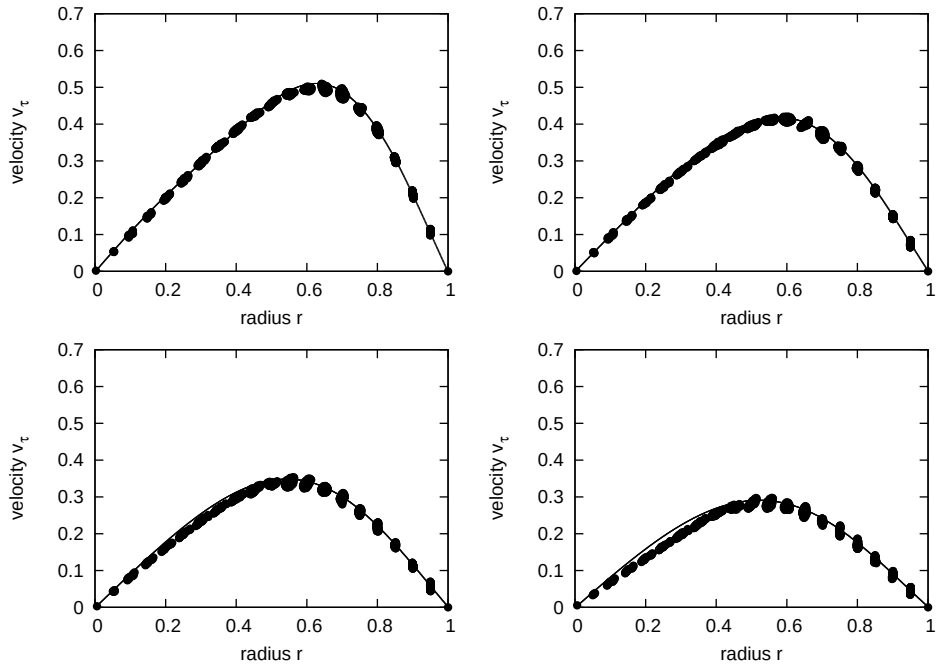


Figure 7.8: Low Mach number spin-down flow at times $t = 1$ (top-left), 1.5 (top-right), 2 (bottom-left) and 2.5 (bottom-right), showing the velocity profile $v_\theta(r)$, with the lines showing the analytic solutions.

1000 AU.

In three dimensions the Reynolds number will scale as $\mathcal{R} \sim \frac{10}{\alpha}\frac{V}{c_s}\frac{L}{h}$ (Monaghan 2005). Since α will typically range between 0.1 and 1, the Reynolds number will range between $50\frac{L}{h}$ and $900\frac{L}{h}$. The spatial resolution could therefore play a role in the onset of turbulence.

7.9 Summary

SPH is capable of accurately evolving shear flow in general, with large density errors only emerging in extreme cases such as highly super-sonic and turbulent regimes.

Correct viscous evolution has been demonstrated in various low Mach number laminar flows: periodic shear flow, Couette flow, Poiseuille flow and cylindrical spin-down.

The viscous spreading of a ring of cold gas is also well-behaved, and differentially rotating circumstellar discs are not subject to large density fluctuations despite high Mach numbers.

Hydrodynamic and viscous fidelity is maintained by SPH, but turbulent flow will introduce problems for any numerical method. The well-behaved evolution of disc density profiles suggests that physical processes are essentially being modelled correctly.

With the handling of shear viscosity we complete the inclusion of constitutive physics into our self-gravitating hydrodynamics code. Additional factors such as

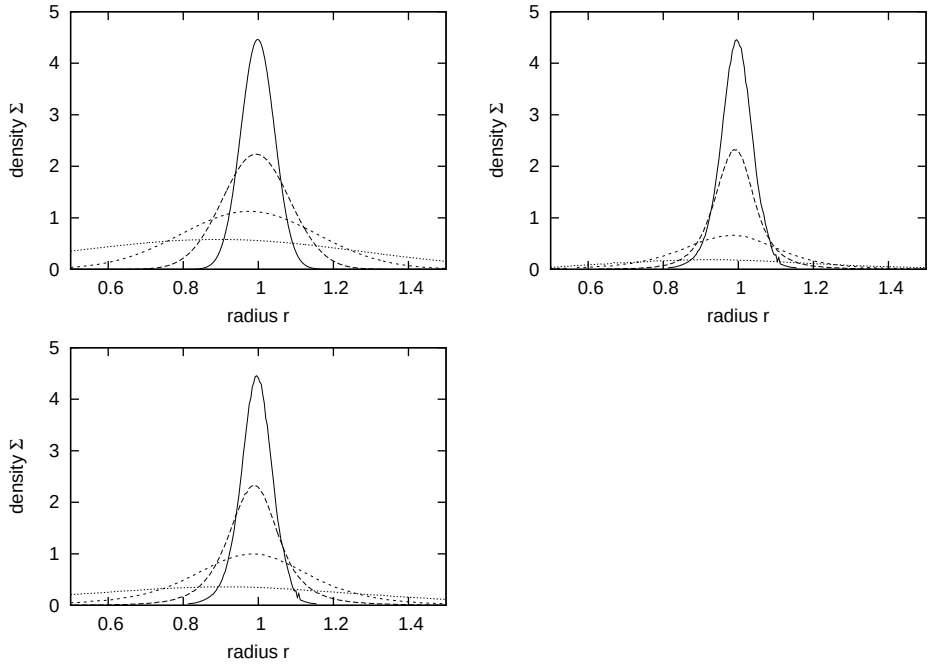


Figure 7.9: Viscous evolution of a ring of cold gas, with the analytic surface density profile shown at times $\tau = 0.004$, 0.016, 0.064 and 0.256 (top-left) and the SPH surface density profile at the same ratio of times $t = 1$, 4, 16 and 64 in central rotation periods (top-right). Accounting for the variation in smoothing length h improves the results (bottom-left).

radiation transport, magnetohydrodynamics and chemistry might play a significant role at various stages but ultimately will not be incorporated. We will, however, need to refine our treatment of the thermal physics to account for stellar radiation.

In the next chapter we will perform self-gravitating hydrodynamic simulations of circumstellar discs.

"All dreams spin out
from the same web."

Hopi proverb

Chapter 8

Circumstellar Discs

In this chapter we perform self-gravitating hydrodynamic simulations of circumstellar discs.

Beginning with a brief summary of disc observations, we perform a preliminary simulation of disc formation before looking in detail at the process of disc fragmentation.

We then describe a method of disc initialisation to generate initial conditions for simulations. After consideration of resolution requirements and edge effects, we investigate the influence of physical and computational factors upon the evolution of disc simulations.

We finally refine our treatment of the thermal physics and consider the implications of our simulations in the context of star and planet formation, concluding with a review of work in this field.

8.1 Disc Observations

More than half of young stars are surrounded by a disc of dust and gas (O'Dell 1995). Such discs have been observed directly for nearby stars, and for more distant stars their presence has been inferred by features in the spectral energy distributions (SEDs) such as infra-red excess (see Section 1.5.1).

It is estimated that 80% of young stars initially have discs (Haisch, Lada & Lada 2001). Many stars lose their discs within 3 Myr, and those existing beyond 6 Myr are typically debris discs or planetary discs. The lifetime of a disc seems to be a function of stellar mass: a massive star loses its disc more rapidly.

Direct visual observations indicate radial extents from 0.1 to 1000 AU, with flared profiles. Masses are assumed to be 0.01 to $0.1 M_{\odot}$, based upon the minimum mass of the solar nebula (Weidenschilling 1977) and the required dust mass to produce observed infra-red signatures (Beckwith et al. 1990). Hundreds of slightly more massive discs (0.13 to $0.39 M_{\odot}$) have been observed, with radii $\sim$ 400 AU (Eisner & Carpenter 2006).

Large massive discs will be simulated, with a radial extent of 1000 AU and masses from 0.45 to $0.68 \mathrm{M}_{\odot}$. These masses represent a range going from stable to unstable against gravitational fragmentation. While the values are higher than those provided by observations of discs around Class II objects, they might be reasonable

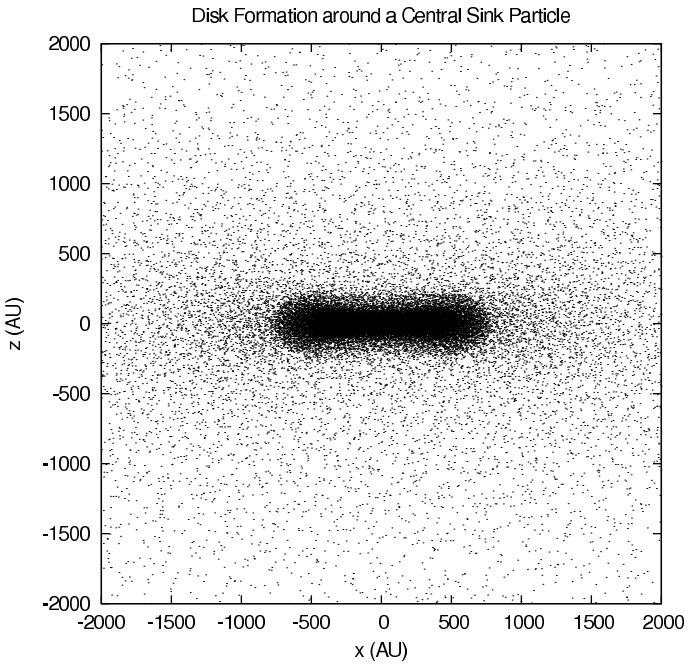


Figure 8.1: Disc formation around a central sink particle at time $t = 2700$ years.

for embedded discs around Class I protostars.

The fragmentation process also has implications for disc observations, since unstable discs rapidly ($< 10^5$ years) end up with masses $< 0.1\mathrm{M}_\odot$ and sizes < 100 AU (Stamatellos et al. 2010), which might explain the lack of observed massive extended discs.

8.2 Disc Formation

The formation of a disc can be simulated in a similar manner to the Boss & Bodenheimer test (see Section 5.6.4), starting with the gravitational collapse of a uniform density cloud that is in solid body rotation. In this case, however, no perturbation is applied and a $1M_\odot$ sink particle (with an accretion radius of 2.424×10^5 pc, or 5 AU) is seeded at the centre of the cloud (Bate, Bonnell & Price 1995). The cloud has a mass of $0.001M_\odot$, a radius of 0.01037 pc (3.2×10^{16} cm), an angular velocity of 1.56×10^{-12} rad/s, an isothermal temperature of 10 K and is composed of $100,000$ particles. High artificial viscosity is applied ($\alpha = 1.0$) to enhance dissipation and therefore accelerate the formation process.

Figure 8.1 shows that the cloud collapses from over 2000 AU in radius to form a disc of ~ 750 AU radius and aspect ratio ~ 0.2. Figure 8.2 shows the surface density profile, which is fit quite well by a $\Sigma \sim r^{-7/4}$ power law in the outer region beyond 100 AU. This power law flattens to $-3/2$ in the inner region within 100 AU, and then $-5/4$ before turning over as the density falls away due to SPH particles accreting onto the central sink. The profile steepens at the outer edge, where the SPH density drops away as it comes into contact with the more diffuse cloud envelope. This power law applies for any initial cloud mass since the solution simply scales in density

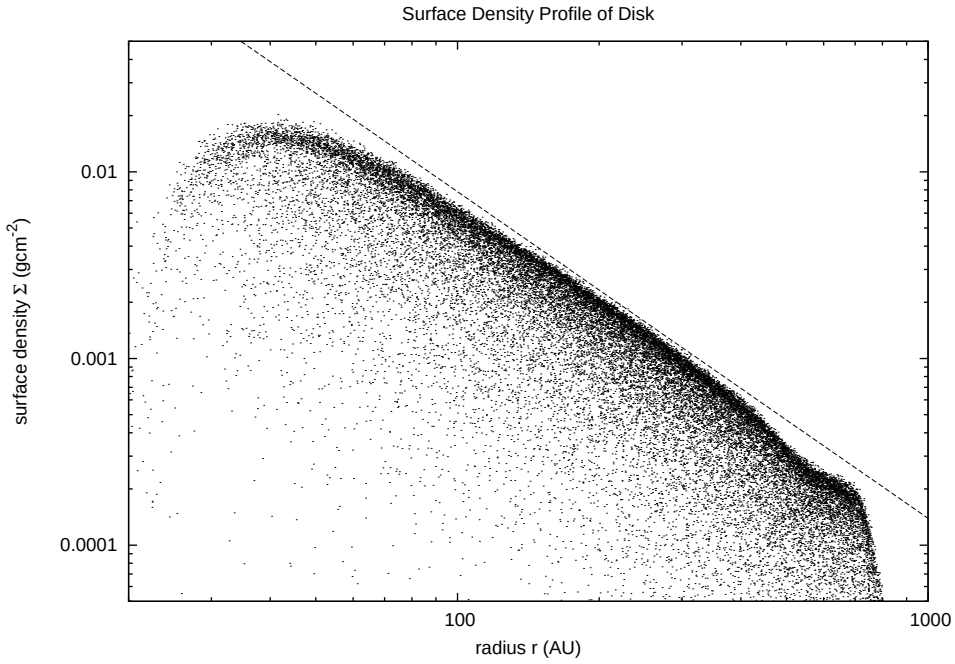


Figure 8.2: The surface density profile of a disc formed around around a central sink particle. The line indicates a power law of $\Sigma \sim r^{-7/4}$.

accordingly. It also holds at lower values of artificial viscosity ($\alpha = 0.1$ and 0.01) although the disc takes longer to form as α is reduced. In the absence of artificial viscosity the disc forms much more slowly and tends towards a slightly flatter power law of $-3/2$.

Semi-analytic investigation of cloud collapse leading to disc formation has suggested that the surface density profile of the disc follows a power law $\Sigma(r) \sim r^{-\sigma}$ with exponent $1 \leq \sigma \leq \frac{3}{2}$, and steepening at the edge of the disc (Lin & Pringle 1990). If viscous processes play little role during infall then $\sigma \sim 1$, steepening to $\sigma \sim \frac{3}{2}$ as viscous evolution dominates. Our 3D simulation suggests an even steeper relationship. Although artificial viscosity plays a significant role in this, self-gravitation alone does provide a pseudo-viscous effect.

The circular velocity is found to be $v_\phi \sim r^{-1/2}$ in the simulation, and since $\Omega = \frac{v_\phi}{r}$ the disc is therefore in Keplerian rotation with $\Omega \sim r^{-3/2}$.

The scale height of a disc $H(r) \sim \frac{c_s(r)}{\Omega(r)}$ (assuming a light disc; see Section 8.4.2) where c_s is the isothermal sound speed (here a constant) and Ω is the angular frequency of rotation. The scale height therefore increases with radius as $H \sim r^{3/2}$; this is a fairly steep relation, hence the large aspect ratio of ~ 0.2.

8.3 Disc Fragmentation

Arguably the most important parameter regarding self-gravity in discs is the Toomre stability parameter (Toomre 1964),

$$Q = \frac{c_s \kappa}{\pi G \Sigma} \tag{8.1}$$

where c_s is the sound speed, κ is the epicyclic frequency (with which a fluid element oscillates when perturbed from circular motion), G is the gravitational constant and Σ is the surface density.

Instability against fragmentation requires $Q \lesssim 1$, and the equation indicates that this is achieved more readily at lower temperature (and pressure, given by c_s), slower rotation speed (given by κ) and higher surface density.

8.3.1 The Fragmentation Process

A more detailed physical interpretation of the Toomre Q parameter can be found by considering a small "protofragment" at radius r_{disc} in a disc. It has an initial radius r_0 and for simplicity is assumed to remain spherical throughout the process. The question asked is whether or not such a fragment will be able to condense out. The collapse (or expansion) is governed by pressure, rotation and self-gravity, so any change in its radial size r is described by the following equation of motion:

$$\frac{\mathrm{d}^2 r}{\mathrm{d}t^2} \simeq \frac{c_s^2(t)}{r} + \frac{h^2(t)}{r^3} - \frac{2\pi G \Sigma(r_{disc}) r_0^2}{r^2}, \tag{8.2}$$

where c_s is the isothermal sound speed in the fragment (which is essentially determined by its radial position in the disc, i.e. its distance from the central star), and $2h/5$ is the specific angular momentum of the fragment (which has an initial value of $2r_0^2\kappa(r_{disc})/5$). For collapse, $\frac{\mathrm{d}^2 r}{\mathrm{d}t^2} < 0$, and substituting in these initial values shows that for instability against fragmentation:

$$2\pi G \Sigma(r_{disc}) > \frac{c_s^2(r_{disc})}{r_0} + \kappa^2(r_{disc}) r_0 \tag{8.3}$$

By differentiating the right hand side with respect to r_0, it is found to be a minimum when $r_0 = c_s(r_{disc})/\kappa(r_{disc})$. Substituting this into the equation above reveals that

$$\pi G \Sigma(r_{disc}) > c_s(r_{disc})\kappa(r_{disc}) \tag{8.4}$$

which is exactly equivalent to the statement $Q < 1$ defined earlier.

Expressing Equation 8.3 in the form of a quadratic equation

$$r_0^2 - 2\frac{\pi G \Sigma(r_{disc})}{\kappa^2(r_{disc})} r_0 + \frac{c_s^2(r_{disc})}{\kappa^2(r_{disc})} = 0 \tag{8.5}$$

solves to give

$$r_0 = \frac{\pi G \Sigma(r_{disc})}{\kappa^2(r_{disc})} \left(1 \pm \sqrt{1 - Q^2}\right) \tag{8.6}$$

which describes the range of initial proto-fragment radii that are unstable against fragmentation (with $Q < 1$). The fastest growing proto-fragment will lie at the centre of this range, and thus have radius $r_{fast} = \frac{\pi G\Sigma(r_{disc})}{\kappa^2(r_{disc})}$ and mass $m_{fast} = \pi r_{fast}^2 \Sigma = \frac{\pi^3 G^2 \Sigma^3(r_{disc})}{\kappa^4(r_{disc})}$. The timescale of this fragmentation, t_{fast}, can be estimated from dimensional considerations ($\frac{\mathrm{d}^2 r}{\mathrm{d}t^2}(r_{fast}) = \frac{r_{fast}}{t_{fast}^2}$, which is negative in the case of collapse) as

$$t_{fast} = \left(\frac{1}{r_{fast}} \frac{\mathrm{d}^2 r}{\mathrm{d}t^2}(r_{fast}) \right)^{-1/2} = \frac{1}{\kappa(r_{disc})} \left(1 - Q^2\right)^{-1/2} \tag{8.7}$$

To determine whether a fragment is actually able to condense out fast enough in order to survive as a bound object, however, requires that $c_s(t)$ and $h(t)$ be considered fully, so radiative transport within the fragment and torques acting upon the fragment should be taken into account.

8.3.2 Unstable Regimes

When $Q < 1$ instability occurs for axisymmetric (ring-like) disturbances (Toomre 1964). For high values of Q pressure stabilises short wavelengths and rotation stabilises long wavelengths, represented by c_s and κ respectively in the equations. The most unstable wavelength when $Q < 1$ is $\lambda \approx 2\pi^2 G\Sigma/\kappa^2 \equiv 2\pi r_{fast}$.

Numerical simulations have demonstrated that instability occurs when $Q \leq 1.5$ for non-axisymmetric (spiral) disturbances (Boss 2004). Since the instability is linear and dynamic, small perturbations grow exponentially on the timescale of a rotation period (Durisen et al. 2006). Values of $Q \gtrsim 1.5$ are considered stable against gravitational fragmentation.

Non-axisymmetric modes can occur at higher values of Q than axisymmetric modes, so gravitational instabilities within discs are expected to develop from growing multi-armed spiral waves.

Extending into the non-linear regime, where results are obtained from numerical simulations, introduces some additional factors (Durisen et al. 2006):
i) The role of thermodynamics in the disc becomes particularly important, due to heating from shocks, compression and gravitational torques.
ii) Non-linear mode coupling results in "gravitoturbulence", when power is rapidly distributed across modes with a wide range of wavelengths and the resulting self-gravitating turbulence spreads throughout the disc.
iii) Surface distortions may occur from thermodynamic effects in the vertical structure of the disc, such as shock heating.

8.3.3 An Alternative Formulation

The epicyclic frequency is defined as $\kappa^2 = r\frac{d}{dr}\Omega^2 + 4\Omega^2$ (Binney & Tremaine 1987). With constant Ω (solid body rotation), $\kappa = 2\Omega$. For a disc in Keplerian rotation, $\Omega \sim r^{-3/2}$ gives $\kappa = \Omega$; we can therefore simply substitute Ω for κ in the case of a Keplerian disc.

For a disc with scale height $H \simeq c_s/\Omega$, where the disc mass is $M_{disc} = \pi R^2 \Sigma$ and the gas moves in a circular orbit around the central star (so that $R\Omega^2 = \frac{GM_{star}}{R^2}$), the condition $Q \equiv \frac{c_s \Omega}{\pi G \Sigma} < 1$ can be rewritten as

$$M_{disc} > \frac{H}{R} M_{star} \tag{8.8}$$

The aspect ratio of a protostellar disc is typically $H/R \sim 0.1$ (Stapelfeldt et al. 1998) so fragmentation requires $M_{disc} > 0.1 M_{star}$ and results in a typical condensation mass scale of one Jupiter mass for a solar-mass star (Papaloizou & Terquem 2005).

It is possible, however, that a disc never becomes massive enough to satisfy $Q < 1$ (Laughlin & Bodenheimer 1994). As the disc mass increases due to infall from the surrounding envelope the Q value will fall, but spiral modes develop before $Q < 1$. These transport angular momentum outward, allowing additional mass to accrete onto the star, and this restores gravitational stability.

The redistribution of mass occurs with a timescale of a few outer rotation periods, so Q can never fall below unity unless the disc can be cooled (to decrease c_s) or matter added (to increase Σ) on a similar timescale. This seems unlikely to happen in the conventional planet formation region (the inner part of the disc) since it is optically thick, but the optically thin outer regions of the disc present a greater possibility.

Although the implication is that planet formation via gravitational instability is unlikely in optically thick regions, it is worth noting that the development of spiral arms might play a significant role in the core accretion model, since the pressure maxima that they produce would assist in the agglomeration of solid bodies.

8.4 Disc Initialisation

To provide the initial state of a disc system we must determine the masses, positions and velocities of the SPH particles that are used to represent the disc material.

The mass of each particle is simply the mass of the disc M_{disc} divided by the number of particles N, and M_{disc} may either be supplied as an initial parameter or instead derived from the Toomre Q parameter.

The position of each particle is determined in two stages.

Firstly the particle is located in terms of radial position r and azimuthal position ϕ (giving the two-dimensional coordinates x and y) using a quasi-Monte-Carlo method (Halton sequence) to fit the desired disc dimensions (inner radius R_{inner} and outer radius R_{outer}) and surface density profile ($\Sigma(r) \sim r^{-\sigma}$). Using the quasi-Monte-Carlo method rather than the standard Monte-Carlo method (pseudo-random numbers) reduces the number of closely-spaced particle pairs. Since these provide numerical seeds for gravitational instabilities, the quasi-Monte-Carlo method effectively limits the "noise" present in the system. This has a similar effect to the "anti-clustering" placement techniques based upon the inhibition method of Diggle (Diggle 1983; Cartwright 2006). Figure 8.3 compares the quasi-Monte-Carlo placement method to the standard Monte-Carlo method; with $10,000$ particles, 960 have their nearest neighbour within $0.2h$ if placed at random, but only 110 do so if placed using the Halton sequence.

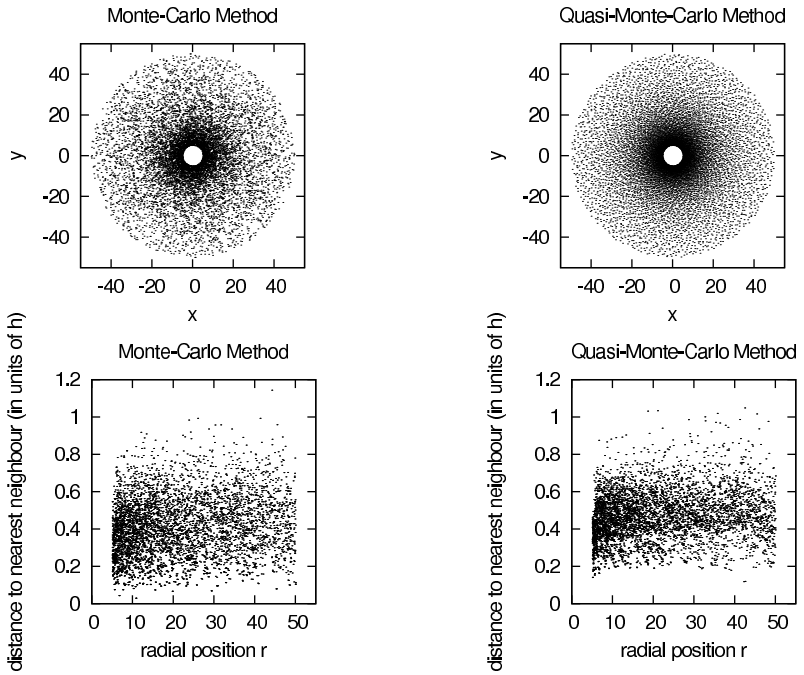


Figure 8.3: The nearest neighbour of each particle in terms of the smoothing length h.

Secondly the particle is located in terms of vertical height z, completing the determination of its three-dimensional position. The thickness of the disc is determined by its scale height, which is essentially determined by the temperature profile $T(r) \sim T_{star} r^{-1/2}$. Instead of distributing particles uniformly within the scale height using a quasi-Monte-Carlo method, they may be distributed in a Gaussian manner by using a Box-Muller transform; this concentrates their placement towards the mid-plane.

In order to maintain the centre of mass at the origin, where the star is located, particle placement is restricted to one half of the disc ($0 < \phi < \pi$) and each particle placed at position (x, y, z) then has a corresponding particle placed in "mirror" fashion at $(-x, -y, -z)$.

The velocity of each particle is determined by Keplerian rotation at its radial position r. Although dominated by the mass of the central star, the contributions to acceleration from the other particles within the disc are calculated in order to correct the orbital velocity of each particle.

Determining the initial state of the disc therefore requires several parameters to be provided: the mass of the disc M_{disc} (or the Toomre Q parameter), the inner radius R_{inner} and outer radius R_{outer}, the surface density profile (specifically the exponent σ), the temperature profile (specifically the factor T_{star}) and the mass of the star M_{star}.

The involvement of these parameters in determining the disc properties will now be examined in detail.

8.4.1 Disc Distribution

If we assume a simple surface density profile for the disc in terms of radial position r, where $R_{inner} \leq r \leq R_{outer}$,

$$\Sigma(r) = \Sigma_0 r^{-\sigma} \tag{8.9}$$

where Σ_0 is the surface density at $r = 1$ then the total mass of the disc is

$$\begin{aligned} M_{disc} &= \int_{R_{inner}}^{R_{outer}} \Sigma 2\pi r \mathrm{d}r \\ &= \frac{2}{(2-\sigma)}\pi\Sigma_0 \left[(R_{outer})^{(2-\sigma)} - (R_{inner})^{(2-\sigma)} \right] \end{aligned} \tag{8.10}$$

The values of R_{inner} and R_{outer}, the inner and outer edges of the disc (which might more appropriately be termed an annulus), are chosen to contain the region of interest within the disc where objects are considered most likely to form. This is likely to be between 100 and 1000 AU.

Each particle in the disc will have mass $m = \frac{M_{disc}}{N}$, where N is the number of particles in the disc and is chosen to satisfy the resolution requirements (see Section 8.5). Calculating M_{disc} may first require Σ_0 to be determined from Q (see Section 8.4.4).

For a random (or quasi-random) number $\mathcal{R}_1$ between 0 and 1, radial positions can then be determined within the desired surface density distribution using the following transformation:

$$r = \left[(R_{inner})^{(2-\sigma)} + \left((R_{outer})^{(2-\sigma)} - (R_{inner})^{(2-\sigma)} \right) \mathcal{R}_1 \right]^{\frac{1}{(2-\sigma)}} \tag{8.11}$$

Azimuthal positions can simply be generated using $\phi = \pi\mathcal{R}_2$, which then gives Cartesian coordinates $x = r\cos(\phi)$ and $y = r\sin(\phi)$. This particle position can then be "mirrored" with a particle at $(-x, -y)$.

8.4.2 Disc Thickness

The initial temperature T in the disc midplane is presumed to vary with distance from the central star (the radial distance r) according to

$$T(r) = T_{star} r^{-1/2} \tag{8.12}$$

where T_{star} is the temperature at $r = 1$ AU and the exponent $(-1/2)$ describes the geometric dilution of heating from the star (since luminosity $L \sim T^4$ scales as $\frac{1}{r^2}$). The isothermal sound speed scales as the square root of temperature so $c_s(r) \sim r^{-1/4}$.

Thin accretion discs, whether active (having an intrinsic luminosity) or passive (only reprocessing radiation from the central star), have a temperature profile $T \sim r^{-3/4}$ (Natta 1993). A flatter profile of $T \sim r^{-1/2}$, however, seems appropriate for a flared accretion disc, and observations of the spectral energy distributions of circumstellar discs support this (Beckwith et al. 1990).

The thickness or scale height H of the disc can be determined by considering an approximate hydrostatic balance between the vertical component of gravity (from

both the star and the disc material), acting towards the midplane, and the thermal pressure opposing it.

$$\frac{GM_{star}}{r^2}\frac{H(r)}{r} + \pi G\Sigma(r) \approx -\frac{1}{\rho}\frac{dP}{dH} \simeq -\frac{1}{\rho}\frac{P}{H} \simeq \frac{c_s^2(r)}{H(r)} \tag{8.13}$$

This can be rearranged into a quadratic equation

$$\frac{GM_{star}}{r^3}H^2(r) + \pi G\Sigma(r)H(r) - c_s^2(r) = 0 \tag{8.14}$$

which can be solved, taking the positive root, to give

$$H(r) = -\frac{\pi\Sigma(r)r^3}{2M_{star}} + \sqrt{\left(\frac{\pi\Sigma(r)r^3}{2M_{star}}\right)^2 + \frac{r^3}{GM_{star}}c_s^2(r)} \tag{8.15}$$

The disc thickness therefore depends upon the surface density profile and the temperature profile. Assuming that the disc is much lighter than the star ($M_{disc} \ll M_{star}$) then

$$H(r) \simeq \sqrt{\frac{r^3}{GM_{star}}}c_s(r) \equiv \frac{c_s(r)}{\Omega(r)} \tag{8.16}$$

Since the density of the disc is likely to decrease with distance from the midplane according to a Gaussian profile (Frank, King & Raine 1992), the vertical position

$$z = H(r)\frac{1}{2}BM_1(\mathcal{R}_3, \mathcal{R}_4) \tag{8.17}$$

can be determined using a Box-Muller transformation

$$\begin{aligned} BM_1(\mathcal{R}_3, \mathcal{R}_4) &= \sqrt{-2\log(\mathcal{R}_3)}\cos(2\pi\mathcal{R}_4) \\ BM_2(\mathcal{R}_3, \mathcal{R}_4) &= \sqrt{-2\log(\mathcal{R}_3)}\sin(2\pi\mathcal{R}_4) \end{aligned} \tag{8.18}$$

where only one of the components (BM_1) is required and the factor of $\frac{1}{2}$ is introduced to ensure that 95% of particles (2 standard deviations) lie within a distance H from the midplane.

Figure 8.4 shows the cross-section of the disc. The scale height $H = \frac{c_s}{\Omega} \sim r^{5/4}$, and the resulting aspect ratio is ~ 0.1.

8.4.3 Disc Rotation

The particle velocities are determined by assuming that the disc is in Keplerian rotation with circular speed $v_\phi(r) = 2\pi r^{-1/2}$ and angular frequency $\Omega(r) = 2\pi r^{-3/2}$. For Keplerian rotation the epicyclic frequency $\kappa = \Omega$ and the orbital period $P(r) = r^{3/2}$.

The circular speed v_ϕ must be related to the radial component of the acceleration a_r such that $a_r = -\frac{v_\phi}{r}$ in order to maintain a stable circular orbit. By calculating the contributions of gravity (and optionally pressure) from other particles in the disc towards a_r, a corrected value for the circular speed can be determined using $v_\phi = \sqrt{|a_r|r}$.

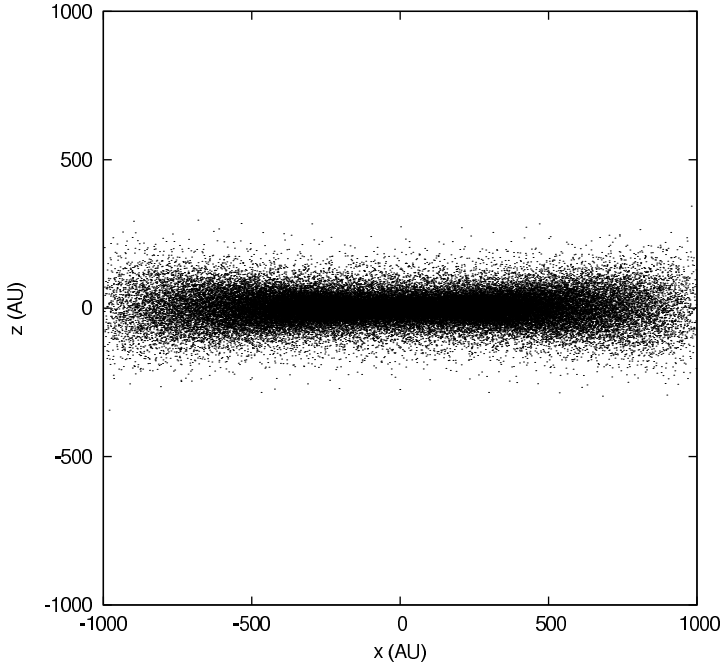


Figure 8.4: Disc of 100,000 particles viewed in the x-z plane, with an aspect ratio of ~ 0.1.

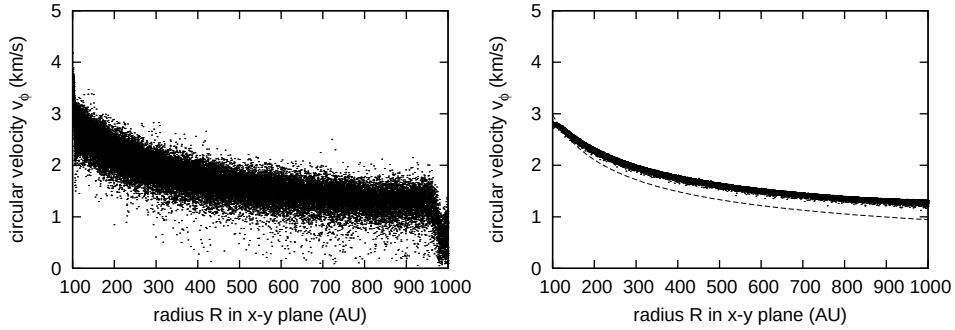


Figure 8.5: Rotation velocity profiles for a disc of 100,000 particles with $Q = 1$. The left-hand plot uses the total radial acceleration, while the right-hand plot includes only the gravitational contribution. The thin lower line indicates the Keplerian rotation curve.

Figure 8.5 shows the effect of correcting the rotation curve. Although the total radial acceleration a_r from both gravity and pressure can be used (Cartwright 2006), the hydrodynamic contribution tends to incorporate a lot of noise (Poisson noise increases with more particles as $\sqrt{N}$), and sub-Keplerian velocities occur at the outer edge where the SPH density falls away (left-hand plot). We will therefore use only the gravitational contribution, and restrict this to the component in the $x-y$ plane (Stamatellos 2003). This leads to a smoother correction which is consistently super-Keplerian (right-hand plot) except at the extreme inner edge of the disc where the self-gravitational force is outwards (Mestel 1963). The corrections are shown for a relatively massive disc ($0.68M_{\odot}$ with $Q = 1$) and so represent a fairly dramatic correction case.

8.4.4 Disc Profiles

The Toomre Q parameter is given by

$$Q = \frac{c_s(r)\kappa(r)}{\pi G\Sigma(r)} \tag{8.19}$$

For a disc in Keplerian rotation $\kappa(r) = \Omega(r) \sim r^{-3/2}$. For simple geometric dilution of heating from the central star, the isothermal sound speed $c_s(r) \sim r^{-1/4}$. The surface density is given by $\Sigma(r) \sim r^{-\sigma}$. The scaling of Q with radial position r is therefore $Q(r) \sim r^{(-7/4+\sigma)}$. Choosing $\sigma = 7/4$ will therefore produce a disc with a uniform value of Q.

Since the super-Keplerian velocity correction effectively increases the epicyclic frequency, there will be a stabilising effect upon the outer regions of the disc; this will result in Q increasing slightly at greater radial position.

The value of Q is essentially determined by the choice of Σ_0 and the isothermal sound speed c_0, where $c_s = c_0 r^{-1/4}$, since in Keplerian rotation $\Omega = 2\pi r^{-3/2}$.

$$Q = \frac{2c_0}{G\Sigma_0} \tag{8.20}$$

Since $M_{disc} \propto \Sigma_0$ and $T_{star} \propto c_0$, these two parameters ultimately determine the stability of the disc.

Temperatures of 31 K or more have been inferred at radius 100 AU in observed discs (Beckwith et al. 1990; Dutrey & Guilloteau 2006), and solar system observations within the asteroid belt indicate temperatures of 145 to 170 K (the "snow line") at radius 2.7 AU in the solar nebula (Lecar et al. 2006). These results imply temperatures of ~ 300 K at radius 1 AU.

Fixing $T_{star} = 300$ K (typical for a solar-mass star) gives $c_0 = 0.22\mathrm{AUyr}^{-1}$ and since $G = 39.48\mathrm{AU}^3\mathrm{M}_{\odot}^{-1}\mathrm{yr}^{-2}$ this results in $\Sigma_0 = \frac{0.0111}{Q}\mathrm{M}_{\odot}\mathrm{AU}^{-2}$. Supplying a value of Q therefore provides Σ_0 which allows M_{disc} to be calculated. When $Q = 1$ this produces a massive disc ($0.68\mathrm{M}_{\mathrm{odot}}$) which is comparable to the mass of the central star ($1\mathrm{M}_{\odot}$).

8.4.5 The Locally Isothermal Equation of State

The temperature of an SPH particle in the disc is determined by its distance from the central star r according to the relation $T = 300r^{-1/2}$ K. The temperature drops below 10 K when $r > 900$ AU in this equation, however, which is not physically reasonable since background radiation from nearby stars should maintain a temperature of at least 10 K. In order to take this into account, a temperature can be calculated by considering the sources of luminosity (central star and background) incident upon the particle. Since luminosity L is related to temperature as $L \sim T^4$, the temperature is given by

$$T = \left(300^4 r^{-2} + 10^4\right)^{1/4} \text{K} \tag{8.21}$$

This temperature can then be used to determine the pressure $P \sim \rho T$ and the isothermal sound speed $c_s = \sqrt{P/\rho}$.

8.5 Resolution Requirements

The primary constraint to properly resolve fragmentation is the Jeans condition (Truelove et al. 1997). Its expression in SPH terms is that the diameter of an SPH particle should always be less than the Jeans length: $4h < R_{Jeans}$ (Hubber, Goodwin & Whitworth 2006).

Contrary to some claims (Klein, Fisher & McKee 2004), under-resolved SPH simulations (using too few particles to resolve the Jeans mass) should not cause multiple fragmentations, but will instead act to suppress or slow down fragmentation (Hubber, Goodwin & Whitworth 2006).

High resolution SPH simulations can also suffer from a lack of convergence (Meru & Bate 2011) when a density gradient is being modelled, as is the case for our disc, and this is discussed under Spatial Resolution in Section 8.7.3.

8.5.1 Resolving the Jeans Mass

The Jeans mass (see Section 1.2.1) is

$$M_{Jeans} = \left(\frac{375}{4\pi}\right)^{1/2} \frac{c_s^3}{G^{3/2}\rho^{1/2}} \tag{8.22}$$

Using the scale height $H(r) = \frac{c_s(r)}{\Omega(r)} = \frac{c_0}{2\pi} r^{5/4}$, the midplane density is

$$\rho(r, H = 0) = \frac{\Sigma(r)}{2H(r)} = \frac{\pi \Sigma_0}{c_0} r^{-3} \tag{8.23}$$

Substituting the midplane density $\rho(r, H = 0)$ and sound speed $c_s(r) = c_0 r^{-1/4}$ gives

$$M_{Jeans} = \frac{1}{\pi}\left(\frac{375}{4}\right)^{1/2} \frac{c_0^{7/2}}{G^{3/2}\Sigma_0^{1/2}} r^{3/4} \tag{8.24}$$

Using $\Sigma_0 = \frac{0.0111}{Q}\mathrm{M}_\odot\mathrm{AU}^{-2}$ and temperature $T_{star} = 300\,\mathrm{K}$ (so $c_0 = 0.22\mathrm{AUyr}^{-1}$) gives $M_{Jeans} \simeq 5.9 \times 10^{-4} Q^{1/2} r^{3/4}$.

For a disc extending from 100 to 1000 AU, $M_{Jeans}/Q^{1/2}$ ranges from ~ 0.019 to $\sim 0.1\mathrm{M}_\odot$. For a $Q = 1$ disc (mass $0.68\mathrm{M}_\odot$) this requires ~ 36 particles to represent the Jeans mass with a single particle, or more appropriately ~ 1900 particles to represent it with the particle and its ~ 50 neighbours within the smoothing sphere.

8.5.2 Resolving the Jeans Radius

The Jeans radius is $R_{Jeans} = (15/4\pi G\rho)^{1/2}\, c_s$, so substitution gives

$$R_{Jeans} = \left(\frac{15}{4\pi G}\right)^{1/2} \left(\frac{c_0^3}{\pi\Sigma_0}\right)^{1/2} r^{5/4} \tag{8.25}$$

This results in $R_{Jeans} \simeq 0.096 Q^{1/2} r^{5/4}$. For a disc extending from 100 to 1000 AU, $R_{Jeans}/Q^{1/2}$ ranges from ~ 30 to ~ 540AU.

The spatial resolution in SPH is determined by the total number of particles N, since having more particles leads to lower mass particles that have shorter smoothing lengths.

When the gas particles in the disc have equal mass, the mass of one such particle is

$$m = \frac{M_{disc}}{N} = \frac{2\pi\Sigma_0}{(2-\sigma)N}\left[(R_{outer})^{(2-\sigma)} - (R_{inner})^{(2-\sigma)}\right] \tag{8.26}$$

The smoothing length in 3 dimensions is therefore given by

$$h(r) = \eta\left(\frac{m}{\rho}\right)^{1/3} = \eta\left(\frac{2c_0}{(2-\sigma)N}\right)^{1/3}\left[(R_{outer})^{(2-\sigma)} - (R_{inner})^{(2-\sigma)}\right]^{1/3} r \tag{8.27}$$

The Jeans condition $4h(r) < R_{Jeans}$ determines the critical resolution

$$\begin{aligned} \mathcal{C}_{Jeans}(r) &= \frac{4h(r,H=0)}{R_{Jeans}} \\ &= 4\pi\eta\left(\frac{4G\Sigma_0}{15c_0^3}\right)^{1/2}\left(\frac{2c_0}{(2-\sigma)N}\right)^{1/3}\left[(R_{outer})^{(2-\sigma)} - (R_{inner})^{(2-\sigma)}\right]^{1/3} r^{-1/4} \end{aligned} \tag{8.28}$$

The resolution condition $\mathcal{C}_{Jeans}(r) < 1$ therefore requires the total number of particles N to be

$$N > \frac{128\pi^3\eta^3}{(2-\sigma)c_0^{7/2}}\left(\frac{4G\Sigma_0}{15}\right)^{3/2}\left[(R_{outer})^{(2-\sigma)} - (R_{inner})^{(2-\sigma)}\right] r^{-3/4} \tag{8.29}$$

Using $\eta = 1.15$ (~ 50 neighbours), temperature $T_{star} = 300$ K (so $c_0 \simeq 0.22\mathrm{AUyr}^{-1}$) and uniform stability Q (so $\sigma = 7/4$) for a disc extending from $R_{inner} = 100$ AU to $R_{outer} = 1000$ AU gives $N > 475000 Q^{-3/2} r^{-3/4}$. For a $Q = 1$ disc this requires $N > 2700$ particles to resolve the outer edge of the disc and $N > 15,000$ particles to resolve the inner edge. Smaller discs require higher particle numbers ($N > 48,000$ for a disc extending between 10 and 100 AU).

8.5.3 Resolving the Toomre Wavelength

In a similar manner, the diameter of an SPH particle should always be less than the Toomre wavelength: $4h < \lambda_{Toomre}$ (Nelson 2006).

$$\lambda_{Toomre} = \frac{2c_s^2}{G\Sigma} \tag{8.30}$$

Within a Keplerian disc $\lambda_{Toomre} \approx \sqrt{2Q}R_{Jeans}$, so in practice this only becomes more restrictive than the Jeans condition when $Q < 0.5$, which might apply to regions of a disc as they begin to fragment.

8.5.4 Resolving the Disc Thickness

For a disc an additional resolution requirement is that the diameter of an SPH particle should always be less than the thickness of the disc: $4h \leq 2H(r)$ (Nelson 2006).

The critical resolution is

$$\begin{aligned} \mathcal{C}_{disc}(r) &= \frac{4h(r,H=0)}{2H(R)} \\ &= \frac{4\pi}{c_0}\eta\left(\frac{2c_0}{(2-\sigma)N}\right)^{1/3}\left[(R_{outer})^{(2-\sigma)} - (R_{inner})^{(2-\sigma)}\right]^{1/3} r^{-1/4} \end{aligned} \tag{8.31}$$

The resolution condition $\mathcal{C}_{disc}(r) < 1$ therefore requires the total number of particles N to be

$$N > \frac{128\pi^3\eta^3}{(2-\sigma)c_0^2}\left[(R_{outer})^{(2-\sigma)} - (R_{inner})^{(2-\sigma)}\right] r^{-3/4} \tag{8.32}$$

Using $\eta = 1.15$, temperature $T_{star} = 300$ K and uniform stability Q for a disc extending from 100 to 1000 AU requires $N > 7000$ particles to resolve the outer edge and $N > 39,000$ particles to resolve the inner edge. Smaller discs require higher particle numbers ($N > 123,000$ for a disc extending between 10 and 100 AU).

The disc thickness condition is therefore a more conservative requirement than the Jeans condition.

This is only a minimal requirement, however, and may not be sufficient; having the disc thickness represented by several SPH particles would be more appropriate. Doubling the resolution (halving $\mathcal{C}(r)$) requires 8 times as many particles, which dramatically increases the length of time required to run a simulation by an order of magnitude. Simulations of small discs can rapidly become very computationally expensive.

8.6 Edge Effects

SPH does not easily cope with an edge or boundary within a system. This is a particular problem for discs since most of the particles lie close to an edge. At an edge a particle only has neighbours to one side. This results in it having a larger smoothing length, and in inevitably experiencing an outward hydrodynamic acceleration. As in

the isothermal collapse scenario (see Section 4.5) a rarefaction wave will propagate inwards, which might affect the evolution of the simulation to some degree (Martin, Pearce & Thomas 1993).

In order to prevent these edge effects, boundary particles might be located at the edges (to a depth of at least two smoothing lengths) and evolved according to the boundary conditions, as in the Couette flow scenario (see Section 7.3). There is no reason, however, why a particle could not percolate through such a boundary layer and eventually escape from confinement within the system. For a disc system it would also be computationally expensive to maintain such a layer, since many particles would be needed to provide hydrodynamic confinement.

Alternatively an analytic function could be applied directly to any particles that overlap the boundary, providing a corrective contribution (Boyd 2003). Again there is no inherent reason why a particle could not eventually overcome such a contribution, and both procedures become problematic when disc-disc interactions are considered.

Since our region of interest is away from the edges, in the midplane and at a middling radial distance, close enough to the star to be dense enough but far enough away to be cool enough to fragment, the edge effects will simply be endured rather than addressed directly.

While the outermost particles might tend to spread into the surrounding vacuum, in doing so they will act to provide some representation of the residual envelope in which the disc system is embedded. The mass effectively lost to this envelope will also bring our high-mass disc simulations closer towards observed disc masses.

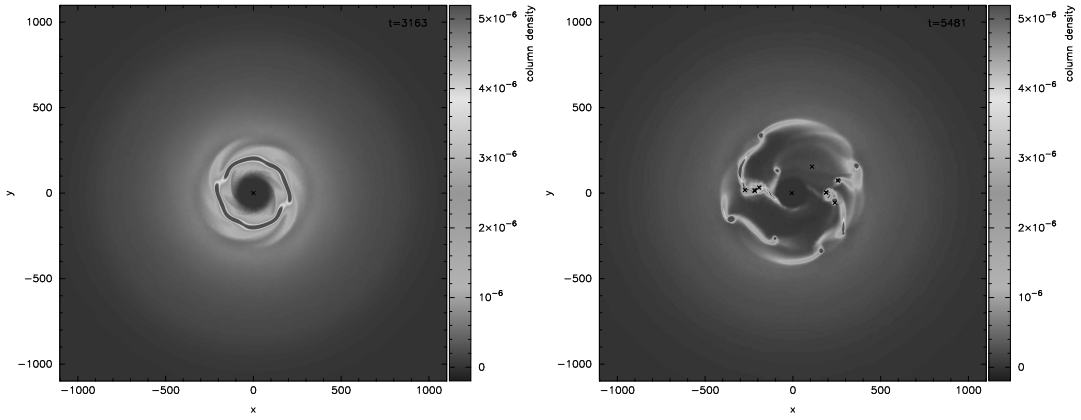
When representing a disc as an annulus the innermost particles will experience an outward self-gravitational force (Mestel 1963). This is most pronounced for dense inner edges (steep surface density profiles) but the effect diminishes more slowly with increasing radii for shallower surface density profiles (Cartwright 2006). This results in a sub-Keplerian correction to rotation velocity at the inner edge (see Section 8.4.3).

The reason for representing a disc as an annulus is to eliminate the need to evolve particles on very fast orbits close to the central star. Any tendency for particles to drift inwards must therefore be countered in order to retain this benefit, and the simplest method is to accrete them onto the central sink. Mass falling inward is likely to accrete onto the star eventually, so this method seems reasonable, but the central sink now essentially represents not just the star but also an inner disc region, so the total mass and accretion rate of the sink will no longer directly correspond to that of the star itself. This immediate accretion onto the sink nonetheless provides upper limits of the values for the star (or likely values given sufficient accretion time). Even if the absolute values become somewhat obscured by this method, like-for-like comparison allows any relative differences to be observed.

8.7 Disc Evolution

The parameter space of disc evolution is investigated to determine its dependence upon both physical parameters and computational parameters.

The physical parameters include the Toomre stability parameter, the viscosity



(a) Instability developing at $t = 3183$ years. (b) Objects fragmenting at $t = 5481$ years.

Figure 8.6: Rapid formation of objects in a disc with Toomre stability parameter $Q = 1$.

parameter, the surface density profile, the temperature profile and the equation of state.

The computational parameters include the sink creation density, the spatial resolution, the particle distribution, the integration scheme and the smoothing factor.

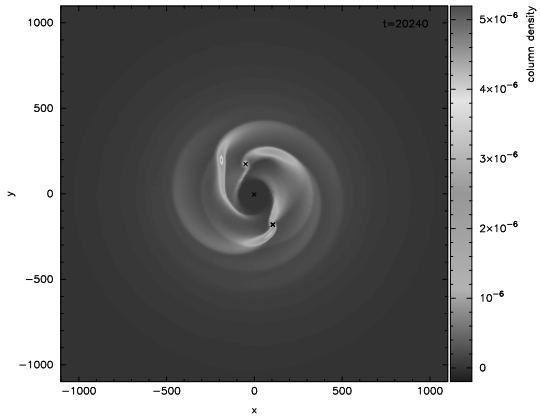
8.7.1 Dependence on the Toomre Stability Parameter

Using a surface density profile of $\Sigma \sim r^{-7/4}$ we can generate a disc with a fairly uniform Q value. From Section 8.3.2 we expect instability to occur for axisymmetric (ring-like) disturbances when $Q < 1$ and for non-axisymmetric (spiral) disturbances when $Q < 1.5$, with values of $Q \geq 1.5$ considered stable against gravitational fragmentation. We therefore investigate the marginally unstable range of values between 1 and 1.5.

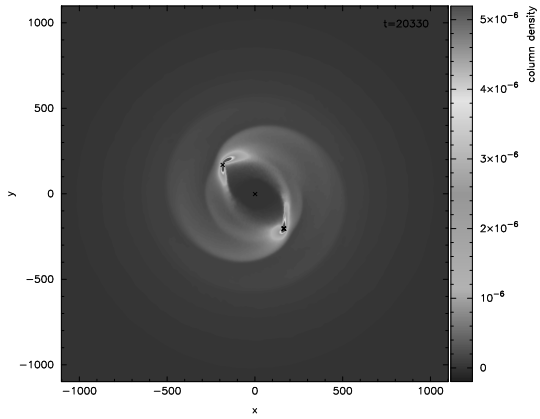
Figure 8.6 shows that a disc with $Q = 1$ is gravitationally unstable within a ring that rapidly develops an $m = 2$ spiral mode. Figure 8.7 shows that discs with $Q = 1.1$ to 1.4 are gravitationally unstable in the non-axisymmetric (spiral) mode, forming sinks in strong spiral arms. Figure 8.8 shows that a disc with $Q = 1.5$ is marginally gravitationally stable; spiral arms develop but do not form sinks and eventually stabilise the disc.

The disc with $Q = 1$ is extremely unstable and forms many sinks very rapidly. A dense ring develops at $r \sim 200$ AU (where the orbital period is ~ 2800 years) and the sinks form in the knots that develop within this ring, seeded in a symmetric manner by the initial quasi-random mirrored particle distribution (see Section 8.4).

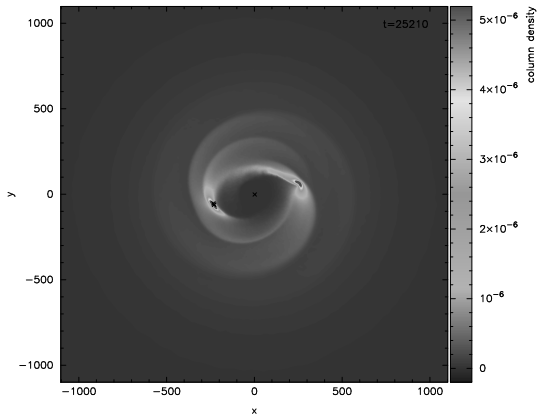
The discs with $Q = 1.1$ to 1.4 are also unstable and form sinks. Spiral arms develop and the sinks form in the clumps that develop along them at $r \sim 200$ to 300 AU, again following the mirror symmetry. Table 8.1 shows that the spiral modes develop on a longer timescale than the ring-like mode, and the fragmentation timescale t_{frag} increases as the stability Q increases. The fragment mass m_{frag} is significantly lower for the fragments formed in spiral arms, and decreases as Q increases. The fragment size is also smaller for the fragments formed in spiral arms, being ~ 0.22



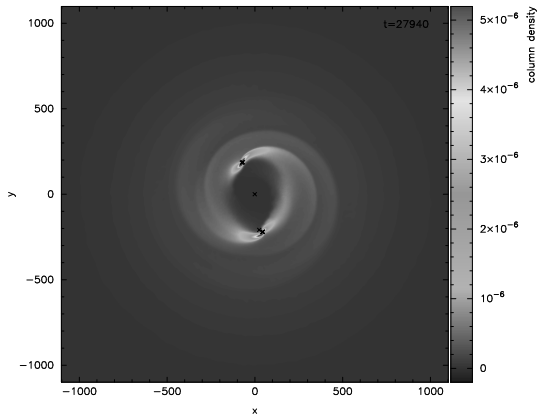
(a) Disc with $Q = 1.1$ at $t = 20,240$ years.



(b) Disc with $Q = 1.2$ at $t = 20,330$ years.



(c) Disc with $Q = 1.3$ at $t = 25,210$ years.



(d) Disc with $Q = 1.4$ at $t = 27,940$ years.

Figure 8.7: Formation of objects in discs with $Q = 1.1$ to 1.4 at various times.

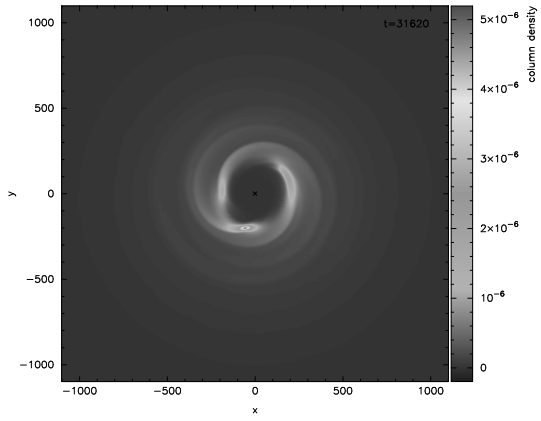


Figure 8.8: Stable disc with $Q = 1.5$ at $t = 31,620$ years (1 outer rotation period).

Q	M_{disc} (in $M_\odot$)	t_{frag} (in years)	r_{disc} (in AU)	m_{frag} (in $10^{-3}M_\odot$)
1.0	0.68	4261	197	1.80
1.1	0.62	19,784	195	0.421
1.2	0.57	20,294	263	0.342
1.3	0.52	25,159	238	0.241
1.4	0.49	27,840	225	0.0758
1.5	0.45	n/a	n/a	n/a

Table 8.1: Fragmentation parameters for discs of various stability Q, showing the disc mass M_{disc}, the fragmentation time t_{frag} at which the first sink forms, its radial position r_{disc} within the disc and its initial mass m_{frag}.

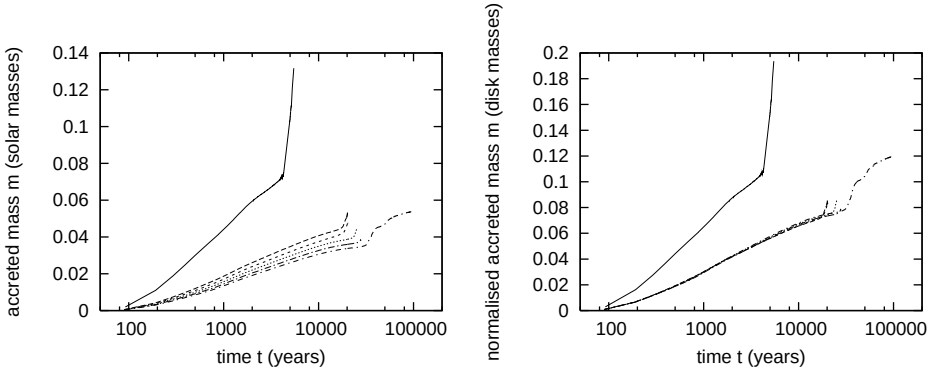


Figure 8.9: The accretion rate of the central star, shown by mass accreted plotted against logarithmic time, for 6 discs from $Q = 1$ to $Q = 1.5$ (lines from left to right). In the right-hand plot the accretion is normalised by the disc mass.

AU in radius compared with ~ 0.33 AU for the fragment formed in the ring-like mode, and does not vary with Q.

The disc with $Q = 1.5$ is stable and forms no sinks. Spiral arms develop and enhance accretion onto the central sink, peaking after ~ 1 outer rotation period, but the disc then stabilises and evolves quiescently.

Figure 8.9 shows the accretion from the disc onto the central star. The $Q = 1$ disc has a rapid accretion rate ($\sim 10^{-5}M_\odot yr^{-1}$), which is comparable to that of a Class 0 protostar (Ward-Thompson et al. 2006; Whitworth & Ward-Thompson 2001). In reality, however, a Class 0 protostar is still accreting from an envelope of greater mass than the star itself rather than from a disc of comparable mass. Sink formation induces an accretion episode where the rate increases by a factor ~ 2.75.

This disc which is unstable in the ring-like mode ($Q = 1$) has an accretion rate one order of magnitude higher than the marginally unstable and stable discs ($Q = 1.1$ to 1.5). These all have a similar accretion rate ($\sim 10^{-6}M_\odot yr^{-1}$) which is closer to that of a Class I protostar (White et al. 2006; Whitworth & Ward-Thompson 2001). The right-hand plot normalises the accretion (dividing it by the disc mass), demonstrating that these discs have basically the same behaviour until the marginally unstable discs

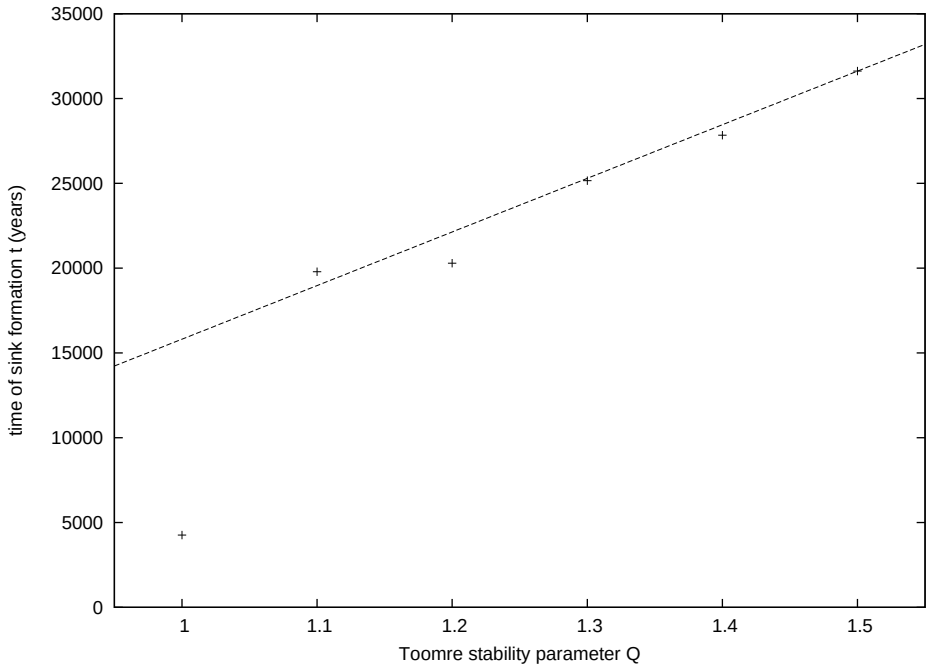


Figure 8.10: Sink formation time for unstable discs with $Q = 1.0$ to 1.4. For the stable disc ($Q = 1.5$) the time at which spiral arms cause enhanced accretion onto the central sink is plotted instead.

($Q = 1.1$ to 1.4) form sinks. Sink formation leads to a burst of accretion where the rate increases by a factor ~ 1.5. Even the stable disc has this accretion enhancement after ~ 1 outer rotation period; spiral arms act to feed accretion enhancement even if they are not strong enough to form sinks.

The sinks form more slowly with increasing stability Q in an approximately linear manner (see Figure 8.10). The disc with $Q = 1$ is initially unstable in an axisymmetric manner and an exception to the linear trend, rapidly forming a ring of sinks. The discs with higher Q all develop non-axisymmetric spiral arms, and these fit reasonably close to a line $t = fQ - \frac{f}{2}$ with gradient f equal to the outer rotation period ($\sim 32,000$ years). This basically means that the marginally unstable discs ($Q = 1.1$ to 1.4) begin to form sinks between 0.5 and 1 outer rotation periods, corresponding to about 5 to 10 rotation periods at a typical radial position of formation ($r_{disc} \sim 220$ AU).

8.7.2 Dependence on the Viscosity Parameter

The evolution of a rotating disc, with inner material losing angular momentum and accreting onto the star, requires viscous forces to transfer angular momentum outwards (Lynden-Bell & Pringle 1974). The exact mechanism that supplies these forces remains unknown. In our simulations it is effectively parameterised through the action of artificial viscosity, making it something of a hybrid between a physical and computational parameter.

Time-dependent viscosity is used so that shocks are captured correctly while excessive dissipation is prevented (see Section 6.3.1). The Balsara switch is not used because it tends to under-estimate the viscosity in the high-vorticity regions of the inner disc and over-estimate it in the low density regions of the disc (Cartwright 2006). Both methods suffer from an alignment artefact due to the SPH evaluation of the velocity divergence in a Keplerian disc (Cartwright 2006) so some improvement to the source term S in the time-dependent formulation would ultimately be desirable.

Time-dependent artificial viscosity requires two values of the viscosity parameter α, and the values chosen are $\alpha_{MAX} = 1$ and $\alpha_{MIN} = 0.1$.

The kinematic shear viscosity is $\nu = \frac{1}{10}\alpha c_s h$ (Monaghan 2005) and the Shakura & Sunyaev viscosity parameter is $\alpha_{SS} = \frac{\nu}{c_s H}$ (Shakura & Sunyaev 1973) so combining these expressions gives

$$\alpha_{SS} = \frac{1}{10}\frac{h}{H}\alpha \tag{8.33}$$

Strictly speaking the α_{SS} prescription is only applicable in the thin-disc approximation, up to aspect ratio ≤ 0.1 or equivalently $M_{disc} \leq 0.5 M_{star}$ (Forgan et al. 2011), but nonetheless forms a useful basis of comparison for our artificial viscosity parameterisation.

The smoothing-length to scale-height ratio $\frac{h}{H}$ for our disc ranges from ~ 0.15 at the outer edge to ~ 0.27 at the inner edge. This assumes standard parameters and $250,000$ particles; higher resolution would decrease these values. For α_{MIN} this gives $\alpha_{SS} = 0.0015$ to 0.0027, and α_{MAX} gives $\alpha_{SS} = 0.015$ to 0.027.

Observations of discs suggest that $\alpha_{SS} \simeq 0.01$ is typical (Andrews & Williams 2007), which gives us an artificial viscosity factor $\alpha = 0.37$ to 0.67. This range lies between α_{MIN} and α_{MAX} so our viscosity parameterisation appears to be reasonable. It will be slightly stronger than observed when capturing shocks and slightly weaker in any regions without particle convergence.

The viscosity parameter α affects the transfer of angular momentum within the disc, so it has significant influence over its evolution. Increasing viscosity is also believed to enhance the survival of clumps by reducing the violence of collisions (Pickett & Durisen 2007).

In simulations at marginal instability, uniformly high viscosity ($\alpha = 1$) is highly dissipative and suppresses both fragmentation and the development of spiral arms. Uniformly low viscosity ($\alpha = 0.1$) only marginally resolves shocks and slightly enhances fragmentation. Sink formation occurs earlier (by about one rotation period at the formation radius r_{disc}) and the fragment mass m_{frag} and size r_{frag} are greater.

Reducing time-dependent viscosity by an order of magnitude ($\alpha_{MAX} = 0.1$ and $\alpha_{MIN} = 0.01$) reduces the sink formation threshold from $Q < 1.5$ to $Q < 1.3$. The sinks form in the same region $r_{disc} = 220 - 230$ AU, but on a faster timescale t_{frag} and with lower mass m_{frag}. The lower viscosity seems to allow the spiral arms to develop more quickly but with less material accumulated in them.

Reducing time-dependent viscosity by two orders of magnitude no longer resolves shocks and prevents the formation of spiral arms, since faster-moving disc material is able to pass through slower-moving material essentially unhindered and does not pile up. When $Q = 1$, however, collapse proceeds towards a ring which then develops dense knots in strong spiral arms, albeit on a slower timescale. While

self-gravity alone is sufficient for fragmentation when $Q \leq 1$, a suitable viscous mechanism is required for non-axisymmetric modes when $Q > 1$. Slightly higher viscosity ($\alpha \sim 0.05$) allows spiral arms to develop very slowly, but the implication may be that low disc viscosity ($\alpha_{SS} \ll 10^{-3}$) will prevent fragmentation via non-axisymmetric modes for $Q > 1$.

Since $\nu \sim c_s h \sim r^{3/4}$ in our parameterisation, the viscous timescale $t_{visc} \sim \frac{r^2}{\nu} \sim r^{5/4}$ indicates that viscous evolution is slightly slower at larger radii. Dividing this by the dynamical time Ω^{-1} essentially gives the Reynolds number $\mathcal{R} \sim r^{-1/4}$, which is lower at small radii (laminar flow dominated by viscous forces, leading to smooth motion, becomes more likely) and higher at large radii (turbulent flow dominated by inertial forces, leading to eddies, vortices and instabilities, becomes more likely).

Turbulent viscosity has actually been proposed in place of molecular (collisional) viscosity (which would be negligible in an astrophysical disc) in order to allow accretion to occur from a disc on a realistic timescale. It was one of the original inspirations for the Shakura-Sunyaev α parameterisation, although the parameterisation itself is simply an application of dimensional analysis and independent of the viscous mechanism; another strong candidate is magnetic viscosity.

8.7.3 Dependence on the Computational Parameters

Assessing the results of numerical simulations when computational parameters are varied is an important step in judging their accuracy. Convergence of the results should ideally occur.

Sink Creation Density

The standard value chosen for the density threshold at which sinks are created is $\rho_{\rm sink} = 10^{-6}{\rm kg\,m^{-3}}$, which is ~ 5 to 6 orders of magnitude higher than the initial density of a typical disc particle.

Varying this from 10^{-8} to $10^{-4}{\rm kg\,m^{-3}}$ does not affect the fragmentation time t_{frag} and position r_{disc}; these are convergent properties.

It does, however, affect the fragment size r_{frag}, which increases as $\rho_{\rm sink}$ decreases ($r_{frag} \sim \rho_{\rm sink}^{-1/3}$). A larger size will enhance initial accretion, which will delay the formation of any subsequent fragments and reduce their potential mass. A smaller size will reduce initial accretion, so any subsequent fragments will form sooner.

The system details may therefore be unresolved at lower values of $\rho_{\rm sink}$, which favours the formation of a massive primary object, while higher values of $\rho_{\rm sink}$ may allow hierarchical fragmentation into multiple objects. These hierarchical cascades tend to prevent precise convergence, but the total system mass does remain reasonably consistent.

We can therefore reliably determine the time and position of fragmentation, and have a reasonable idea of the system mass formed, but cannot be certain of the details of this system.

These values of ρ_{sink} can be associated with pre-main sequence stars in the quasi-static phase of Kelvin-Helmholtz contraction from $\sim 10^{-10}{\rm kg\,m^{-3}}$ (where objects becomes opaque to emitted radiation and enter an adiabatic phase) up to

$\sim 10^{-4}\mathrm{kg\,m^{-3}}$ (at ~ 2000 K when near-isothermal evolution is restored) as described in Section 1.3. In reality, therefore, this adiabatic phase should prevent hierarchical fragmentation cascades.

Too low a value of $\rho_{\rm sink}$ may lead to spurious fragmentation (fragments forming which would later be destroyed by tidal forces, shear stresses, collisions or shocks) but higher values require longer run-times. The chosen value of $10^{-6}\mathrm{kg\,m^{-3}}$ is essentially a compromise between these two factors.

Spatial Resolution

The spatial resolution is determined by the number of particles N, and the standard value chosen is $250,000$. This is sufficient to resolve the disc thickness with more than one particle at the formation radius (see Section 8.5: $\sim 24,000$ particles are needed at ~ 200 AU), and each particle represents less than one Earth mass.

At very low resolution ($N = 31,250$), where the disc thickness is explicitly unresolved at the inner region (≤ 135 AU), spiral arms do not form.

At low resolution ($N = 62,500$), where the disc thickness is only marginally resolved, spiral arms form but there is no accretion enhancement onto the central sink. There must therefore be some condensation mechanism that acts to enhance accretion which is not being resolved.

At fairly low resolution ($N = 125,000$) spiral arms form and act to enhance accretion onto the central sink. In the marginal cases, however, sink formation may be suppressed despite the presence of strong spiral arms. When the disc thickness is represented by a single particle then it is more difficult for sinks to condense out, due to there being no neighbours in the z-direction.

At fairly high resolution ($N = 500,000$) general behaviour similar to standard resolution is observed. While the details are not convergent, since the instabilities are seeded by a different particle distribution, the overall behaviour is reasonably consistent, producing systems of similar total mass after similar times.

At high resolution ($N = 1,000,000$) the general behaviour is also fairly consistent but system masses appear to be much lower than expected.

At very high resolution ($N = 2,000,000$) sink formation may be suppressed completely, despite the presence of spiral arms. Poisson noise (shot noise) in the SPH density calculation increases with the number of particles as $\sim \sqrt{N}$. This only affects the hydrodynamic force contributions, however, not the gravitational contributions (Cartwright, Stamatellos & Whitworth 2009). The hydrodynamic noise therefore effectively begins to overwhelm the self-gravitational "signal" and acts to suppress fragmentation. This effect is so pronounced that it can even suppress fragmentation when $Q = 1$. Careful disc settling is therefore required for very high-resolution simulations.

A lack of convergence at higher resolution has also been found in simulations of disc cooling (Meru & Bate 2011; see Section 8.8.1). At higher resolution the shorter smoothing length effectively reduces the extent of hydrodynamic smoothing in the SPH summation approximation, increasing its sensitivity to the effects of noise and resulting in a less accurate representation of any underlying density gradient. Using a quintic kernel function with larger compact support is one approach that begins to

address this issue (Hubber, Falle & Goodwin 2011).

This reveals a quandary for our simulations. Too few particles fails to fully resolve the disc and its physical behaviour, but too many particles can suppress fragmentation due to hydrodynamic noise. Using the results above as a guide, a region of compromise (and relative convergence) appears to exist between $250,000$ and $500,000$ particles, with $125,000$ particles appearing to be too few and 1 million slightly too many.

A natural compromise arises between the competing aims of increasing the spatial resolution and reducing the simulation run-time. Ensuring sufficient resolution is of course a vital requirement, but over-resolution incurs a cost in both time and hydrodynamic noise.

Also note that higher resolution effectively reduces α_{SS} (see Equation 8.33) so the standard viscosity parameterisation may no longer be appropriate. An eightfold increase in particle number (doubling the spatial resolution) will halve α_{SS}.

Particle Distribution

The standard method used is quasi-random particle placement using the Halton sequence, since this reduces the number of close neighbours compared to random placement (see Section 8.4). In fact, when random placement is used the resultant increase in hydrodynamic noise can actually suppress fragmentation to the point where no sinks form despite the presence of strong spiral arms. The increased noise also delays the time at which accretion enhancement onto the central sink occurs.

Integration Scheme

The standard method used is the modified Leapfrog scheme, a drift-kick-drift formulation.

Switching this to a kick-drift-kick formulation changes the detailed evolution of the system, although the general behaviour is consistent. This highlights the chaotic nature of the processes involved, where small deviations at early times rapidly lead to diverging outcomes.

Switching to a Runge-Kutta scheme (which is not symplectic) can result in suppression of fragmentation in the marginal cases, such that no sinks form despite the presence of strong spiral arms. These act to enhance accretion onto the central sink around the time when sinks would be expected to form.

This ultimate lack of precise convergence between integration schemes implies that chaotic processes are involved, and highlights a requirement for like-for-like comparisons between simulations: only one parameter should be varied at a time in order to investigate underlying trends. Detailed comparisons of results produced by different codes (those using different algorithms and/or parameters) is effectively impossible under these circumstances (although code comparisons are of course very useful for standard tests: those with analytic solutions, such as freefall collapse, or more well-behaved arrangements, such as the Boss & Bodenheimer test).

Any results obtained from computer simulations must therefore be considered with appropriate caution. While their worth in terms of absolutes is questionable, and

little confidence can be placed in a simulation providing the physical truth, their value in relative terms is still very useful. Varying a single parameter between simulations may allow trends to be uncovered, and sets of simulations can provide the basis of statistical arguments as to the likely outcomes of a particular scenario.

Smoothing

The standard value chosen for the smoothing factor η, which is essentially equivalent to the number of SPH neighbours within a smoothing sphere, is 1.15 ($\sim$ 50 neighbours).

Varying this from 1 to 1.3 ($\sim$ 33 to $\sim$ 72 neighbours) affects the fragmentation time t_{frag}, which increases as η increases. This is to be expected since increasing η increases the smoothing length h, which is also the gravitational softening length; softening gravity over a larger distance slows the collapse process. The larger smoothing length will also tend to increase the fragment size r_{frag}.

High values of η can increase the smoothing length to the point where collapse can no longer be spatially resolved, suppressing fragmentation. In the marginal cases with $250,000$ particles this occurs when moving from $\eta = 1.2$ to 1.3. The disc thickness goes from being resolved by more than one particle to a single particle in the formation region and sink formation is suppressed by the lack of neighbours in the z-direction. Strong spiral arms remain and enhance accretion onto the central sink on a timescale that is delayed by the increased gravitational softening.

The combination of a longer timescale t_{frag} and a larger smoothing length will also tend towards fragmentation at larger radial positions r_{disc}.

The benefit of increasing η to reducing the SPH noise must therefore be carefully considered against the cost of reducing the spatial resolution. Larger values of η become a more appropriate choice for larger numbers of particles N (which provide greater spatial resolution but increase the Poisson noise) but the combination of high η and high N has the greatest computational cost.

8.7.4 Dependence on the Surface Density Profile

Using a surface density profile of $\Sigma = \Sigma_0 r^{-\sigma}$ we can vary σ between 1 and 2 to investigate how this affects the disc stability. Unfortunately, only our standard value of $\sigma = 7/4$ generates a disc with a uniform Toomre stability parameter Q. We therefore compare simulations using different σ values for discs with the same mass M_{disc}.

Figure 8.11 shows the stability profile of 5 discs with a fixed disc mass. The stability Q at the inner edge decreases as σ increases. The lower σ is below 7/4 (the shallower the profile), however, the more rapidly the stability Q decreases as radius increases. Conversely, when σ is higher than 7/4 (a steeper profile), the stability Q increases with radius. A value of 1.99 is used instead of 2 in order to avoid division by zero when determining the value of Σ_0 from M_{disc} (see Section 8.4.1).

The shallower profiles should be more unstable in the outer disc region beyond 400 or 500 AU, while the steeper profile should be more unstable in the inner disc region inside 300 AU.

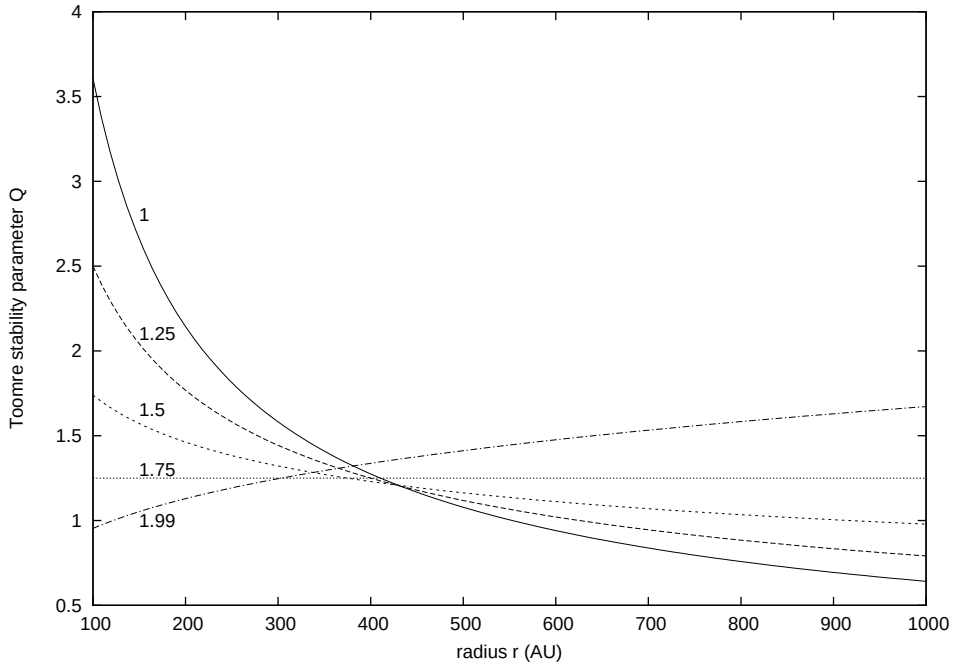


Figure 8.11: Stability against gravitational fragmentation of discs with surface density profiles $\Sigma \sim r^{-\sigma}$ in the range $\sigma = 1$ to 2.

In simulations at marginal instability, the shallower profiles suppress fragmentation and delay the development of spiral arms. The spiral arms are strongest for $\sigma = 3/2$ and eventually enhance accretion onto the central sink, but are weaker for $\sigma = 5/4$ and no clear arms develop for $\sigma = 1$.

The steeper profile ($\sigma \simeq 2$) enhances fragmentation. Sinks form more rapidly, and the mass m_{frag} and size r_{frag} is greater.

A significant drawback here is that the most unstable regions are at the edges of the discs, where edge effects come into play (see Section 8.6). The lower spatial resolution (due to larger smoothing lengths) and the propagation of rarefaction waves would act to suppress fragmentation, and this is the case for the shallower profiles. Fragmentation in the steeper profile may have been encouraged by the stronger outward self-gravitational force experienced at the inner edge of the disc.

In general, however, as the surface density profile steepens, objects form closer to the star on a faster timescale and with greater mass. Conversely, as a surface density profile becomes shallower it appears to become more stable (or more easily stabilised), and any objects would form further from the star on a slower timescale.

8.7.5 Dependence on the Temperature Profile

Using a temperature profile of $T = T_0 r^{-\beta}$ we can change β from 1/2 to 3/4 to investigate how this affects the disc stability. In order to generate a disc with a uniform Toomre stability parameter Q, this temperature profile requires the surface density profile to be steepened from $\sigma = 7/4$ to 15/8. While $\beta = 1/2$ seems appropriate

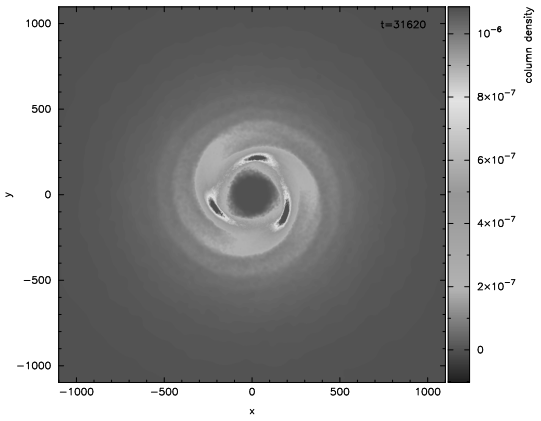


Figure 8.12: Three spiral arms shown at 1 outer rotation period in a disc with a steeper temperature profile ($\beta = 3/4$) and surface density profile ($\sigma = 15/8$).

for flared discs, $\beta = 3/4$ would apply to thin discs. With $\beta = 1/2$ the scale height $H \sim r^{5/4}$, while with $\beta = 3/4$ the scale height is shallower ($H \sim r^{9/8}$), which reduces the aspect ratio from ~ 0.1 to < 0.05.

Using the steeper temperature profile results in a disc of lower mass for a given Q value: $M_{disc}(\beta = 3/4) = 0.48 M_{disc}(\beta = 1/2)$. If instead the disc mass is fixed, the resulting Q value will be lower by the same factor.

In simulations at marginal instability, the steeper temperature profile suppresses fragmentation. Although prominent spiral arms eventually develop, accretion onto the central sink is only enhanced slightly before the disc stabilises.

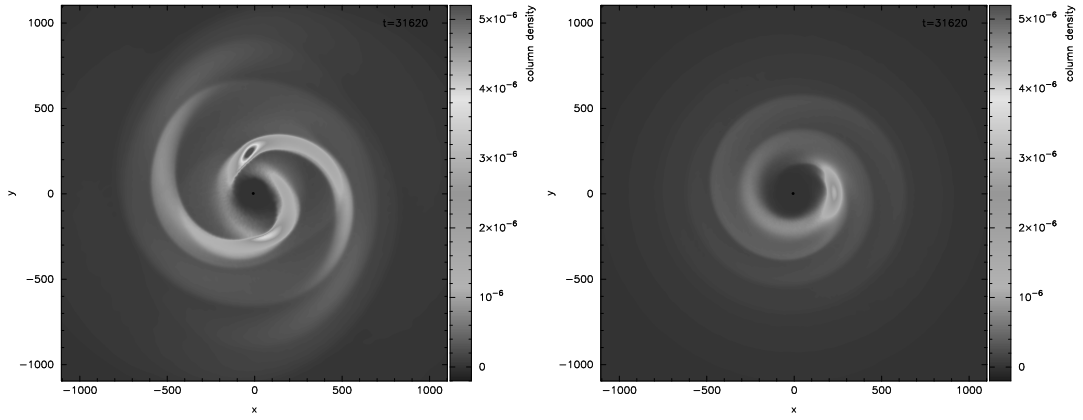
Unstable discs ($Q < 1$) fragment on a slower timescale with the steeper temperature profile.

Although the steeper temperature profile results in a cooler outer disc, and the surface density profile is also steeper, the disc nonetheless appears to be more easily stabilised. The disc is also thinner, however, and the critical resolution $\mathcal{C}_{disc}$ scales more slowly as $\sim r^{-1/8}$. For a given number of particles N the disc thickness is effectively less well resolved, which may contribute a stabilising influence.

Figure 8.12 shows three spiral arms developing with the steeper temperature profile. Two spiral arms tend to be the dominant mode in the simulations, which is probably encouraged by the mirror symmetry of the disc. Comparison with the surface density profile simulations suggests that steeper surface density profiles ($\sigma > 7/4$) may favour the three spiral arm mode.

8.7.6 Dependence on the Equation of State

Using a locally isothermal equation of state (see Section 8.4.5) provides a best-case scenario for fragmentation, since the temperature of a condensing fragment is solely determined by its distance from the central star. Any gain in temperature through its own collapse (PdV work) is ignored, with the implicit assumption being that it cools instantly by radiating away any thermal energy gained in this manner.



(a) $Q = 1.0$ disc. (b) $Q = 1.1$ disc.

Figure 8.13: Suppression of fragmentation in discs shown after 1 outer rotation period with the barotropic equation of state.

In other words the cooling time $t_{cool} = 0$, which is only a reasonable approximation if $t_{cool} \ll t_{dyn} \sim \Omega^{-1}$, and the optical thickness of the disc is likely to prevent this from being true.

Once a fragment reaches a density of $\sim 10^{-10}\mathrm{kg\,m^{-3}}$ it becomes opaque to its own cooling radiation and behaves approximately adiabatically, so allowing it to continue to behave isothermally is unrealistic.

The locally isothermal equation of state can be replaced with a locally barotropic equation of state (see Section 5.6.4), so that a fragment will behave adiabatically beyond a critical density threshold $\rho_{crit} = 10^{-11}\mathrm{kg\,m^{-3}}$ (the density at which the temperature begins to rise above 10K). In this case $t_{cool} = \infty$, as the thermal energy gained during collapse (from PdV work) is never lost. This is only a reasonable approximation if $t_{cool} \gg t_{dyn}$ and is no longer appropriate once the gas reaches ~ 2000 K (energy can then go into hydrogen dissociation rather than heating). It can nonetheless provide a worst-case scenario for fragmentation.

Using this equation of state (with adiabatic index $\gamma = 7/5$ for molecular hydrogen) suppresses fragmentation even for $Q \leq 1$. Strong spiral arms form and enhance accretion onto the central sink, but when swept-up material condenses its increasing thermal energy causes it to adiabatically "bounce", re-expanding before being swept up again. Realistic thermal physics would be even stricter in the cold early stages of collapse, since at low temperatures $\gamma = 5/3$ for an ortho/para ratio of 1.

Figure 8.13 shows the typical behaviour. For $Q > 1$ discs, material is swept up into a single condensation fairly quickly (right-hand plot), and this oscillates between collapse and re-expansion as it orbits but never fragments. For $Q \leq 1$ discs, however, while no longer achieving rapid fragmention they maintain strong condensations in two distinct spiral arms (left-hand plot). It is possible for these two clumps to eventually wind into each other, and a violent collision may result in a sink forming.

These circumstances were contrived for a $Q = 1$ disc by lowering the viscosity

by an order of magnitude. This only marginally resolves shocks ($\alpha_{MAX} = 0.1$) but speeds up the fragmentation timescale. After several episodes of enhanced accretion onto the central sink, a violent collision formed a sink after 53,478 years at radius 382 AU with mass $0.887 \times 10^{-3}\mathrm{M}_{\odot}$.

Although contrived, the fact that a Jupiter-mass fragment can reach a sink density of $10^{-6}\mathrm{kg\,m^{-3}}$ despite adiabatic behaviour beyond $10^{-11}\mathrm{kg\,m^{-3}}$ is quite remarkable. The initial conditions (a $0.68\mathrm{M}_{\odot}$ disc with $\alpha_{SS} \sim 10^{-3}$) are extreme but not ridiculous. Even in this worst-case scenario there exists a region of parameter space that is unstable to disc fragmentation.

8.8 Disc Cooling

Besides the Toomre stability parameter, the cooling time is another important factor for fragmentation. The outcome of gravitational instability is sensitive to the thermal physics of the gas, and if condensations cannot cool sufficiently rapidly then fragmentation is suppressed.

Examining the extremes, with isothermal behaviour ($t_{cool} = 0$) a fragment may form while with adiabatic behaviour ($t_{cool} = \infty$) it may not. We find that with isothermal behaviour fragmentation occurs up to $Q < 1.5$, but with adiabatic behaviour fragmentation is difficult to achieve even if $Q \leq 1$.

The equation of state plays a crucial role in disc fragmentation (Stamatellos & Whitworth 2009b), and the correct treatment of molecular hydrogen in the temperature region from 30 to 50 K might be particularly important (Boley et al. 2007).

8.8.1 The Cooling Criterion

A concise expression of this general concept is the following criterion for fragmentation (Gammie 2001, which assumes a two-dimensional adiabatic index of $\gamma = 2$):

$$t_{cool}\Omega \leq 3 \tag{8.34}$$

The cooling time must therefore be less than half an orbital period. If it is not then the disc reaches a steady gravitoturbulent state in which the cooling is balanced by heating from the dissipation of turbulence.

Although determined from two-dimensional investigations, the criterion has been shown to be applicable in three-dimensional SPH simulations, with the finite thickness introducing a small stabilising influence (Rice et al. 2003). In these simulations the disc was heated by viscous dissipation and compression with the cooling parameterised as

$$\frac{\mathrm{d}u}{\mathrm{d}t} = -\frac{u}{t_{cool}} \tag{8.35}$$

using a cooling parameter $\beta = t_{cool}\Omega$. The solution to this cooling equation, however, is $u = u_0 \exp\left(-\frac{t}{t_{cool}}\right)$, which allows the disc to cool exponentially towards zero. A more realistic approach would be $\frac{\mathrm{d}u}{\mathrm{d}t} = -\frac{u-u_{MIN}}{t_{cool}}$, so that the specific internal energy u can never fall below some minimum value u_{MIN}.

The cooling time required for fragmentation increases as the adiabatic index γ decreases (Rice, Lodato & Armitage 2005).

The cooling parameter β also regulates the amplitude of the spiral modes in the disc: $\langle \delta\Sigma/\bar{\Sigma} \rangle \sim \beta^{-1/2}$ (Cossins, Lodato & Clarke 2009). The amplitude has only weak dependence on the disc-star mass ratio, although it regulates the azimuthal structure: a higher mass ratio gives a less tightly-wound structure.

Grid-based simulations have also been performed in which the disc is heated via artificial viscosity and cooled according to a fixed cooling time (Mejía et al. 2005). Using values for t_{cool} of 2, 1 and 0.25 outer rotation periods, as the cooling time decreases the spiral instabilities grow stronger and fragmentation only occurs for $t_{cool} = 0.25$ outer rotation periods, but only at high azimuthal resolution.

There are issues of numerical convergence to consider, since the criterion appears to relax at higher resolution. Using more SPH particles allows fragmentation to occur with longer cooling times (Meru & Bate 2011; see Spatial Resolution in Section 8.7.3). The value of the critical cooling time could therefore be open to debate.

While this simple criterion cannot be immediately applied to realistic radiative cooling models, it is possible to use it in the asymptotic limit (Johnson & Gammie 2003).

8.8.2 Radiative Cooling

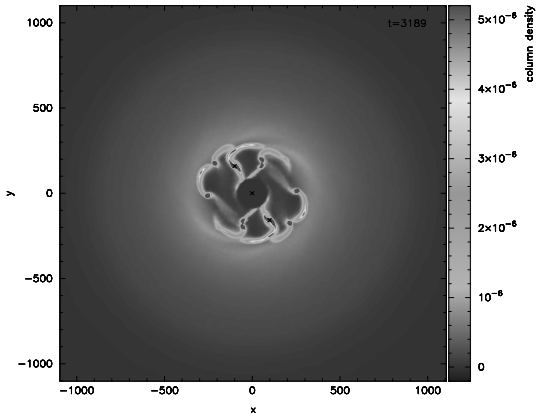
A useful diffusion approximation to radiative cooling (and heating) can be achieved by using a polytropic model (Stamatellos, Whitworth, Bisbas & Goodwin 2007).

Essentially, every SPH particle is assumed to be embedded in a spherically symmetric pseudo-cloud which has a polytropic density and temperature profile. The density, temperature and gravitational potential of the particle can then be used to estimate a mean optical depth. The gravitational potential is effectively used as a gauge of the larger-scale cloud environment in which the particle is embedded, since SPH quantities are local properties. The mean optical depth can then be used to regulate radiative heating and cooling, since it governs how well shielded the particle is from external radiation and how difficult it is for its own cooling radiation to escape. This radiative component can then be applied to the integration of the SPH energy equation.

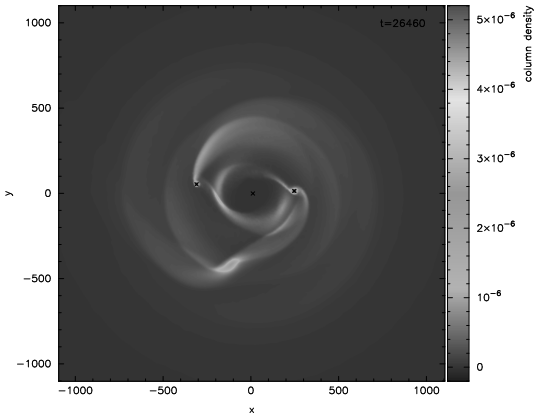
This is a fully three-dimensional method for handling radiative processes, and is surprisingly computationally efficient (only $\sim 3\%$ slower than the general barotropic equation of state). It provides a more reasonable approximation to the thermal physics of the gas in the disc, since the isothermal and adiabatic extremes can only provide upper and lower limits to fragmentation. This method can also be combined with flux-limited diffusion to get the benefits of both components with very little computational cost (Forgan et al. 2009).

Combining this radiative diffusion approximation with ambient heating (using the central star and the 10 K background radiation to determine the temperature as before), simulations at marginal instability undergo fragmentation when $Q \leq 1.3$.

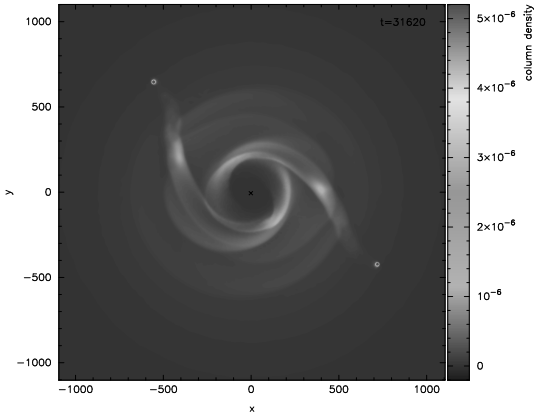
Figure 8.14 shows three scenarios, demonstrating rapid instability for $Q = 1$, the development of an unstable non-axisymmetric mode for $Q = 1.2$ and stability for $Q = 1.4$.



(a) Unstable disc with $Q = 1.0$ at $t = 3189$ years.



(b) Unstable disc with $Q = 1.2$ at $t = 26,460$ years.



(c) Stable disc with $Q = 1.4$ at $t = 31,260$ years.

Figure 8.14: Disc evolution at various times using the radiative diffusion approximation.

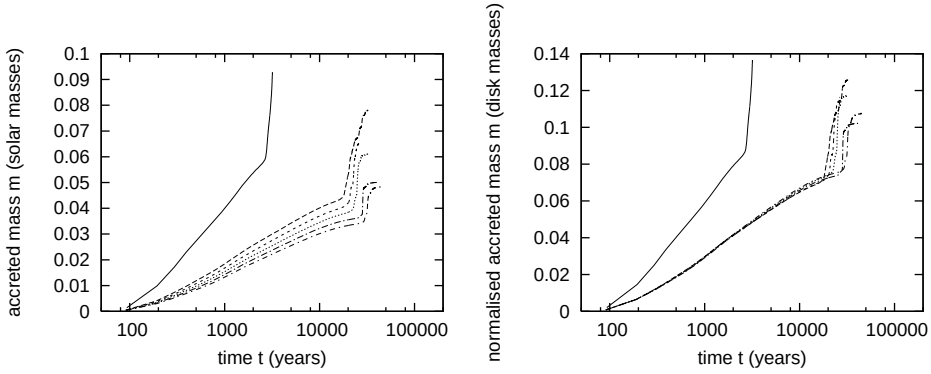


Figure 8.15: The accretion rate of the central star, shown by mass accreted plotted against logarithmic time, for 6 discs from $Q = 1$ to $Q = 1.5$ (lines from left to right) using the radiative diffusion approximation. In the right-hand plot the accretion is normalised by the disc mass.

In the stable case, condensations formed in the spiral arms are ejected before fragmentation occurs. As a condensation forms, its radiative cooling regulates the rate of collapse. In discs with $Q \geq 1.4$ the collapse is slow enough that fragmentation does not occur before the spiral arms wind into each other, and the interaction of the two condensations results in ejection. In discs with $Q < 1.4$ the collapse is fast enough for fragmentation to occur before this happens, although the sink might subsequently be ejected. A violent interaction that induces fragmentation is not seen here since viscous heating is accounted for in the radiative diffusion approximation.

Although the ejected condensations in discs with $Q \geq 1.4$ have not fragmented to form sinks, they are nonetheless bound objects undergoing a quasi-static contraction phase. A disc that appears stable against fragmentation (and the subsequent formation of brown dwarfs) can therefore still be capable of producing free-floating planets (see Figure 8.14 c).

Figure 8.15 shows that the accretion from the disc onto the central star is similar to the locally isothermal case (Figure 8.9). The accretion rate is notably higher for $Q = 1$, and spiral modes enhance accretion onto the sink at later times with increasing Q.

Fewer sinks are formed compared to the locally isothermal case, and the timescale t_{frag} for $Q > 1$ is slightly longer. Fragment masses and sizes are comparable ($m_{frag} \sim 0.24$ to $0.47 \times 10^{-3}\mathrm{M}_{\odot}$ and $r_{frag} \sim 0.22$ AU) although there is no longer a trend of decreasing m_{frag} with increasing Q. There might now be a trend of increasing radial position r_{disc} with increasing Q, but this could simply be due to the increased influence of interactions between spiral arms in the marginal $Q = 1.3$ disc.

It remains open to argument whether real discs ever cool fast enough for fragmentation to occur, but our results using the radiative diffusion approximation suggest that fragmentation is possible in massive discs ($M_{disc} > 0.5 M_{star}$) at large radial positions ($r_{disc} \gtrsim 200$ AU).

8.8.3 Subsequent Evolution

Fragmentation of the disc creates sink particles of up to a Jupiter mass (M_J), but these rapidly accrete further disc material.

For the highly unstable $Q = 1$ disc, two sinks accrete from $1M_J$ to over $9M_J$ in only a few hundred years. Small timesteps, however, prevent the subsequent evolution from being investigated.

For discs unstable in the non-axisymmetric mode ($Q \sim 1.2$) the first sink travels on an eccentric orbit between 200 and 250 AU and accretes beyond the minimum brown-dwarf mass ($13M_J$) after ~ 1 orbit. A subsequent sink forms on a wider orbit that is more eccentric. One such sink, formed at ~ 360 AU, rapidly accretes to $3.7M_J$ after formation. It then gradually grows to $5.4M_J$ over the next 2500 years, as it drifts outwards to $r \sim 760$ AU. This might be ejected as a free-floating planet.

For the marginally unstable $Q = 1.3$ disc, the sink travels on a wider, more eccentric orbit between 250 and 500 AU. It rapidly accretes to $3.5M_J$ after formation and grows to over $7M_J$ over the next 2000 years. Since other condensations have already been ejected, it might struggle to attain brown-dwarf mass.

Disc fragmentation therefore offers a mechanism for the formation of brown dwarfs and high-mass planets. The propensity for ejection also assists in explaining the "brown dwarf desert" and massive free-floating planets.

An important question regards survival: whether fragments last long enough to contract into permanent protoplanets before they are disrupted by tidal forces, shear, collisions or shocks. The interactions of the spiral arms in the disc might destroy fragments as well as create them. The formation of a sink particle neglects this possibility, although the sink creation criteria (see Section 5.6.1) are designed to preclude against such factors at the point of creation.

8.9 Planet Formation

Companion stars ($> 75M_J$ assuming solar metallicity) are less likely to be produced by disc fragmentation, since a considerable amount of material would need to be accreted by the objects that are formed. Perturbations during core collapse (see the Boss & Bodenheimer test in Section 5.6.4) are a more likely mechanism.

The sinks formed in the simulations immediately accrete to a minimum Jeans mass $\sim 3M_J$, so disc fragmentation at large radial position in massive discs cannot explain lower-mass planet formation, but it might explain the observations of massive planets.

8.9.1 Detections of Extra-Solar Planets

The first extra-solar object with a planetary mass was detected around a millisecond pulsar (Wolszczan & Frail 1992), and similar objects have since been detected, although it is not certain whether such planets formed before or after the supernova explosion of their host star.

The first extra-solar planet around a Sun-like star was detected around 51 Pegasi B by Mayor & Queloz (1995) and immediately confirmed from Lick Observatory

(Marcy & Butler 1995). Its orbital properties were surprising: with a period of only 4 days, its proximity to the star meant that the temperature was too high for a solid core to assemble. This led to the suggestion of orbital migration, initiated by tidal interactions between the planet and the disc (Lin, Bodenheimer & Richardson 1996).

Over the next decade, over 150 planets were detected around solar-type stars (Papaloizou & Terquem 2006), with $\sim 10\%$ found to be multiple planetary systems, and the current catalogue has > 500 entries (Schneider 2011).

The most common detection method uses Doppler shift measurements to look for a periodic variation in radial velocity caused by the presence of a planet (Mayor & Queloz 1995), and allows the orbital period and eccentricity to be determined, together with a lower limit for the planet's mass (only a "projected mass" can be determined since the orientation of the orbital plane with respect to the line of sight is unknown). This technique is limited to planets with semi-major axes < 5 AU, but long-term radial velocity surveys have been very productive in finding planets (Schneider 2011).

Transits (Charbonneau et al. 2000) are another promising detection method. These allow an accurate mass determination (since the orbital orientation must be "edge-on" for a transit to be observed) and the planet radius to be measured (from the relative variation of the stellar flux). Although such events are rare, the Kepler mission detected 1235 candidates around 997 stars from May to September 2009 (Borucki et al. 2011) and should therefore greatly expand the current catalogue of > 500 entries.

Other detection methods include:
i) Direct imaging (Deming et al. 2005), using the Spitzer Space Telescope for example, which operates at infra-red wavelengths (the most favourable for planet imaging, since it is where the star/planet flux ratio is lowest).
ii) Microlensing, where the star and planet gravitationally lense the light from a background star, with the presence of the planet causing a distinctive two-spike pattern to be superimposed onto the light curve (Mao & Paczynski 1991; Bond et al. 2004).
iii) Astrometry, using measurements of stellar motion *perpendicular* to the line of sight (Benedict et al. 2002), although this has proved ineffective as an independent method and is therefore used in combination with the radial velocity method.

8.9.2 Properties

About 5% of the stars monitored in radial velocity surveys have been found to host at least one planet, and this serves as a lower limit since low-mass and/or long-period objects cannot be detected using current methods (Papaloizou & Terquem 2005).

Typical masses range from $0.02\mathrm{M_J}$ (6.3 Earth masses, less than half the mass of Uranus) right up to $13\mathrm{M_J}$ (the brown dwarf limit). Although there is an observational bias towards massive planets, 42% are $\leq 1\mathrm{M_J}$ and only 19% are $> 5\mathrm{M_J}$. The most massive short-period planets are found in wide stellar binary systems (Zucker & Mazeh 2002), implying that binarity affects the mass-period distribution of planets through its influence upon the formation mechanism, the orbital evolution of planets, or both.

Typical separations range from 0.02 AU to beyond 5 AU (the detection limit for radial velocity surveys, with 9% found at larger separations using other methods), with a significant number (15%) found between 0.035 and 0.05 AU (orbital periods from 2.5 to 4 days), dubbed "hot Jupiters". Beyond ~ 0.5 AU there seems to be a deficit of sub-Jupiter mass planets, although this might be due to observational biases.

Eccentricities range from zero up to 0.93, although planets within 0.05 AU all have almost circular orbits consistent with tidal circularisation. The planets of our solar system all have nearly circular orbits, so this is a stark contrast.

Differences between the eccentricity distributions of planets and stellar binaries have been used as evidence to suggest that they have different underlying formation mechanisms (Halbwachs et al. 2005). Planets have their orbits circularised much more rapidly after formation than binary companions. Outside the circularisation limit, planetary orbits have significantly smaller eccentricities than binary companions. Since binary systems with small mass ratios have eccentricities more similar to equal-mass binaries than planetary systems, this difference is unlikely to be simply a consequence of low secondary mass, and more likely a result of different formation mechanisms.

Stars hosting planets initially appeared to be metal-rich, although the observed tendency is now less pronounced (Schneider 2011). Although infall of planetary material was suggested as an explanation for this (Sandquist et al. 2002), a primordial origin tended to be favoured (Pinsonneault et al. 2001; Santos et al. 2002; Sadakane et al. 2002).

A population of about 30 young free-floating planetary-mass objects has also been found in the Sigma Orionis open cluster (Zapatero Osorio et al. 2000) and in the Trapezium cluster (Lucas & Roche 2000). These objects introduce some confusion regarding the definition of planets, since they do not orbit a star and they are unlikely to have formed via the core-accretion scenario.

Recent observations using gravitational microlensing suggest that free-floating Jupiter-mass objects might be twice as common as main sequence stars (Sumi et al. 2011).

8.9.3 Formation Mechanisms

Terrestrial planets have long been believed to form via solid body accretion of kilometre-sized "planetesimals", which have themselves been formed by collisional growth of dust grains in a protoplanetary disc (Lissauer 1993).

The formation of giant planets, however, has seen two theories proposed.

In the first theory, giant planets are formed via collapse and fragmentation of protostellar discs. Gravitational instabilities have long been considered as a mechanism in this regard (Kuiper 1949).

In the second theory, known as the core accretion model, a solid core is first assembled (in the same manner as terrestrial planets) which acquires a gaseous envelope once the core has become massive enough to gravitationally bind the gas it is embedded within (typically at 0.1 Earth masses). Once a critical core mass is reached (in the range of $5 - 15$ Earth masses) it is possible for runaway growth to occur via

rapid accretion, leading to the formation of a giant planet (Pollack et al. 1996).

A drawback of the core accretion model is that the timescale for formation is typically a few million years, which is similar to the lifetime of the protostellar disc itself. Gravitational instability, on the other hand, operates on a much shorter timescale, typically a few rotation periods. Over 100 extra-solar planets are $> 5\mathrm{M_J}$, and it seems unlikely that core accretion could form such objects, particularly since gap formation will inhibit the accretion process once a certain critical mass is reached. Observations of transition discs, between the Class II classical disc and the Class III debris disc, should constrain the timescale for planetesimal formation (Cieza 2007).

Our simulations suggest that disc fragmentation is a viable mechanism for massive planet formation, and especially for free-floating planets. There are nine massive planets which have mass $\geq 7\mathrm{M_J}$ and separation > 250 AU, and disc fragmentation is the most plausible explanation for them.

Hybrid mechanisms combining the two theories are possible. Planetesimal growth might thus be speeded up in regions of enhanced density within spiral arms produced by gravitational instability (Rice et al. 2006). Such accelerated growth might in turn allow a gap to be swept out in the disc more rapidly, slowing down the rate of inward planetary migration. Likewise the effects of grain growth within the disc might influence the behaviour of gravitational instability (Boley & Durisen 2010).

8.10 Disc Simulations

In a disc that is unstable for non-axisymmetric perturbations ($1 < Q \leq 1.5$), condensations may form as the density in the spiral arms increases (Boss 1998, using grid-based method; Mayer et al. 2004, using SPH). Clumps that are formed in simulations do not appear to be long-lived, however, and might be transient objects that are soon broken apart by either tidal forces, shear stresses, collisions or shocks (Durisen 2001).

Boss's simulations form clumps at ~ 10 AU after ~ 10 orbital periods at that distance, and he suggests that convective cooling occurs (Boss 2004). This seems highly questionable, however, and the motions observed are more likely due to the "bounce" from pressure forces after the initial collapse rather than circulation due to convection. Stable convection in the presence of spiral arms seems debatable, but even if accepted it is still inherently limited by the radiative cooling from the disc surface (Rafikov 2007). Simulating cooling by convection apparently allows the disc to fragment (Boss 2007) but the cooling seems to be artificially high, with temperatures at the convective surface effectively being set to zero (Cai et al. 2010).

Mayer's simulations also form clumps at similar distances and times, although there appear to be flaws in their approach when examined in detail (Mayer et al. 2004). They use an isothermal gas in which "the thermal energy of a given particle is assigned based on its initial distance from the star and does not vary, irrespective of its subsequent motion through the disc." This assumes that no significant radial departures occur, since it would otherwise be severely non-physical, but to make such an assumption in the regime of non-axisymmetry seems very risky. Also the

use of only 32 SPH neighbours and a fixed gravitational softening length (but a variable smoothing length) are both questionable policies. Simulations using flux-limited diffusion also claim that convective cooling allows fragments of $\sim 1\mathrm{M_J}$ to form at 10 AU (Mayer et al. 2007).

Gravitational instability is sensitive to the thermal physics of the gas. With an adiabatic equation of state, heating of the gas prevents the formation of clumps, while an isothermal equation of state allows condensations to form. If the disc is permitted to expand via gravitational torques then even these clumps tend to dissolve back into the gas (Pickett et al. 2000; Boss 2000). Simulations that model heating by dynamical processes and radiative cooling tend to produce weaker spiral structures that fail to form clumps (Nelson et al. 2000). Simulations also suggest that lower metallicity leads to faster disc cooling and therefore stronger gravitational instability (Cai et al. 2006), but planet-hosting stars are found to be metal-rich.

The effectiveness of gravitational instability in these disc simulations, which all concentrate upon a gas giant planet formation region of order ~ 10 AU, is questionable, and even if clumps are formed then their long-term survival is debatable. Radiative simulations suggest that the formation of giant planets by disc fragmentation within 40 AU is unlikely (Stamatellos & Whitworth 2008).

It has been suggested, given that satisfying both $Q < 1$ and the Gammie criterion requires a sufficiently massive disc that can cool quickly enough, that clump formation is impossible at 1 AU, and highly unlikely at 10 AU, but more feasible at 100 AU (Rafikov 2005). At these distances the clump masses are expected to be higher ($\sim 10\mathrm{M_J}$), and so less applicable to the formation of extra-solar planets in general (since they would be unable to explain the observed mass distribution), but would extend towards a mechanism for brown dwarf formation.

Boss has performed simulations in the region 100 to 200 AU with no clump formation, however, and claims that gas depletion occurs too quickly for their formation (Boss 2006). Other simulations, however, suggest that massive extended discs are capable of forming objects ranging from planetary-mass to low-mass hydrogen-burning stars, but most commonly brown dwarfs (Stamatellos, Hubber & Whitworth 2007). Indeed the fragmentation of massive extended discs can explain the low-mass IMF, the brown dwarf desert, free-floating planets and the binary properties of low-mass stars and brown dwarfs (Stamatellos & Whitworth 2011).

Although gravitational instability faces difficulties in explaining the general formation of gas giant planets, it may play a role in outburst phenomena. Class II objects tend to have a low accretion rate of $10^{-8}\mathrm{M_\odot yr^{-1}}$ that is sometimes punctuated by episodic bursts of $10^{-6}\mathrm{M_\odot yr^{-1}}$, as exhibited by FU Orionis stars. Simulations that are stable against fragmentation still frequently exhibit FU Orionis-like outbursts (Boley et al. 2006; Rice, Mayo & Armitage 2010), and such accretion enhancements were a regular feature of our simulations. Indeed such accretion episodes and the bursts and lulls of luminosity that would be associated with them may regulate the disc fragmentation process itself (Stamatellos, Whitworth & Hubber 2011).

8.11 Summary

We have shown in detail how to construct a model of a circumstellar disc to investigate fragmentation by gravitational instability, exploring the influence of the various physical and computational parameters involved.

Using self-gravitating hydrodynamic simulations with a radiative diffusion approximation, we have shown that the fragmentation of massive extended discs is a viable mechanism for the formation of free-floating planets and brown dwarfs.

In the literature we have seen that gravitational instability struggles to provide a robust mechanism for gas giant planet formation, although it might play an important role in combining with the core accretion model in a hybrid mechanism. We have also seen that extended disc fragmentation can explain many of the properties of brown dwarfs, and that gravitational instability might play a role in outburst phenomena such as that exhibited by FU Orionis stars.

In the next chapter we will use the marginally stable disc ($Q = 1.5$) to investigate the effects of disc-star and disc-disc interactions.

"At high tide the fish eat ants;
at low tide the ants eat fish."

Thai proverb

Chapter 9

Disc Interactions

In this chapter we perform self-gravitating hydrodynamic simulations of disc-star and disc-disc interactions.

We investigate the effects of penetrating, grazing and distant interactions on the evolution of the marginally stable disc ($Q = 1.5$) from the previous chapter for a range of orbital or disc inclinations.

We compare the results of our simulations with other research in this field.

9.1 Disc-Star Interaction

Stellar separation within a cluster is typically more than two orders of magnitude greater than the disc size, and although young embedded stellar densities will be higher since a cluster expands as the gas is dispersed (Bastian et al. 2008), disc lifetimes are relatively short, so intra-cluster disc-star or disc-disc encounters are not likely, except perhaps at the centres of exceptionally dense young clusters.

A binary system might form within the collapse of a rotating or turbulent core, however, where the separation might approach the order of the disc size, so intra-core disc-star or disc-disc encounters are certainly plausible. Indeed the $\sim 7\%$ tilt of the ecliptic plane in our own solar system with respect to the solar equator suggests mild turbulence at formation or perhaps an encounter with another star, and a great many extra-solar planets have very high inclinations (Schneider 2011), lending significant weight to turbulent formation and the possibility of encounters. The Atacama Large Millimeter Array (ALMA) should provide observations of the substructure of cores, allowing us to see whether they are forming binary systems.

9.1.1 Orbital Motion

The orbit of a stellar encounter is expected to be mildly hyperbolic (Larson 1990), but for simplicity we will use a parabolic orbit. This lies on the border between elliptical and hyperbolic orbits, so the difference in orbital energy should be minor.

The locus of a parabolic orbit can be written as

$$r = \frac{2p}{1 + \cos\theta} \tag{9.1}$$

where p is the periastron distance (closest approach), or equivalently in Cartesian form as

$$y = \frac{x^2}{4p} - p \tag{9.2}$$

The velocity along this path is

$$v = \sqrt{\frac{2GM}{r}} \tag{9.3}$$

where M is the total system mass (of the system in orbit plus the system at the focus).

While θ is the polar angle describing the locus of position, the velocity is directed tangentially to the locus, so its components are

$$v_x = v\cos\gamma \tag{9.4}$$
$$v_y = v\sin\gamma \tag{9.5}$$

where $\tan\gamma = \frac{dy}{dx} = \frac{x}{2p}$.

For a three-dimensional orbit, with inclination angle ϕ to the $x-y$ (or $r-\theta$) plane, a locus can be described in Cartesian coordinates (Watkins 1996) as

$$x = -r\sin\theta\cos\phi \tag{9.6}$$
$$y = -r\cos\theta \tag{9.7}$$
$$z = -r\sin\theta\sin\phi \tag{9.8}$$

with velocity

$$v_x = v\cos\gamma\cos\phi \tag{9.9}$$
$$v_y = v\sin\gamma \tag{9.10}$$
$$v_z = v\cos\gamma\sin\phi \tag{9.11}$$

and $\tan\gamma = \frac{-r\sin\theta}{2p}$.

Since our disc lies in the $x-y$ plane and rotates anti-clockwise, $\phi = 0$ will give a prograde coplanar encounter, $\phi = \pi$ a retrograde coplanar encounter and $\phi = \frac{\pi}{2}$ an orthogonal encounter.

We will use the stable $Q = 1.5$ disc from the previous chapter (seen in Figure 8.8) and a $1\mathrm{M}_\odot$ perturbing star (represented by a sink particle).

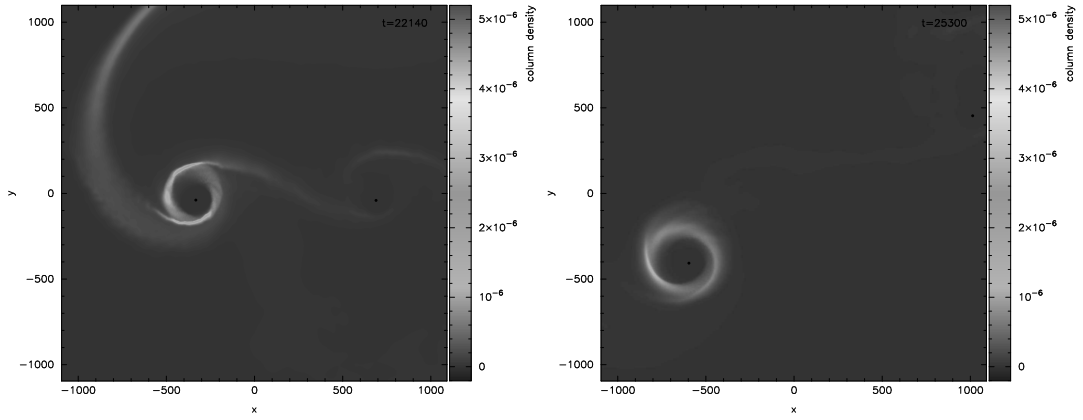
9.1.2 Initial Separation

The magnitude of the gravitational acceleration in the disc due to the central star of mass M is

$$a_{grav} \sim \frac{GM}{r_{disc}^2} \tag{9.12}$$

The magnitude of the tidal acceleration in the disc due to the perturbing star, also of mass M, is

$$a_{tidal} \sim \frac{2GMr_{disc}}{r^3} \tag{9.13}$$



(a) $t = 22,140$ years. (b) $t = 25,300$ years.

Figure 9.1: Penetrating prograde coplanar disc-star encounter after periastron, with no fragmentation.

The condition $a_{tidal} \ll a_{grav}$ therefore requires

$$2\left(\frac{r_{disc}}{r}\right)^3 \ll 1 \tag{9.14}$$

If $r = 5r_{disc}$ then $2\left(\frac{1}{5}\right)^3 = 0.016$ which is comfortably less than 1.

Setting the initial separation $r = 5r_{disc}$ and the periastron p allows θ to be calculated (from Equation 9.1), which then gives the Cartesian position $\mathbf{r} = (x, y, z)$. This allows γ to be calculated, giving the Cartesian velocity $\mathbf{v} = (v_x, v_y, v_z)$.

By placing the perturbing star at position $\frac{\mathbf{r}}{2}$ with velocity $\frac{\mathbf{v}}{2}$ and offsetting the positions and velocities of the disc system by $-\frac{\mathbf{r}}{2}$ and $-\frac{\mathbf{v}}{2}$ we obtain a centre-of-mass reference frame for the interaction.

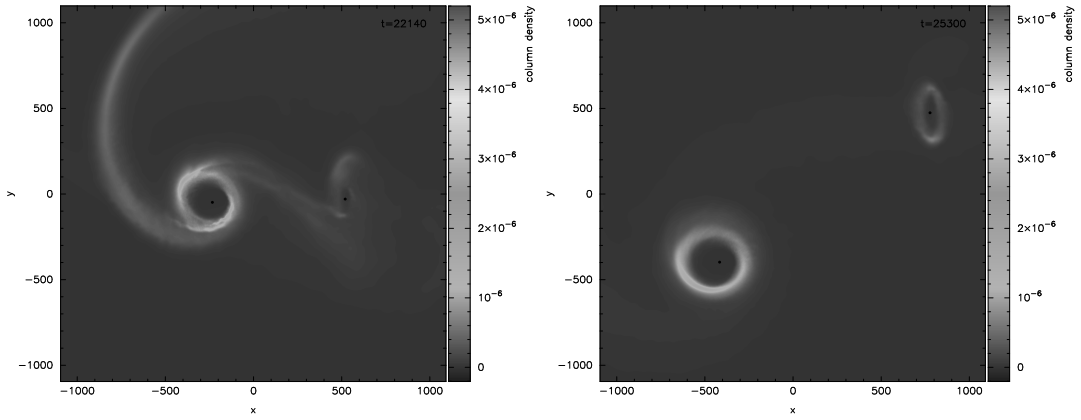
9.1.3 Encounters

Simulations are performed at three periastron distances ($p = 1.5$, 1 and $0.5r_{disc}$), to model encounters where the star simply passes nearby (distant), skims the outer disc (grazing) or deeply penetrates the disc (penetrating).

Five inclination angles are used ($\phi = 0$, $\frac{\pi}{4}$, $\frac{\pi}{2}$, $\frac{3\pi}{4}$ and π), to model encounters that are prograde coplanar, tilted prograde, orthogonal, tilted retrograde and retrograde coplanar.

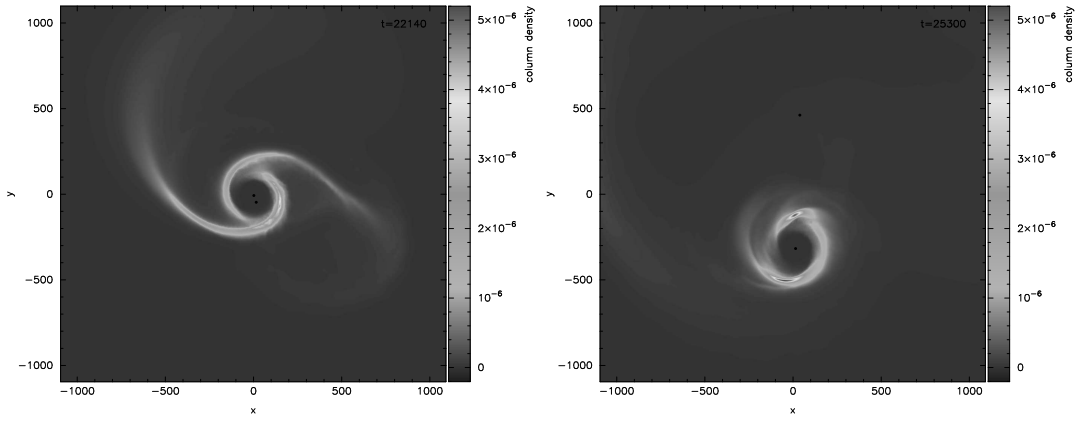
Figures 9.1, 9.2, 9.3, 9.4 and 9.5 show snapshots of the penetrating encounters at the five inclination angles shortly after periastron.

Figures 9.6, 9.7, 9.8, 9.9 and 9.10 show snapshots of the grazing encounters at the five inclination angles shortly after periastron. Snapshots of the distant encounters are not shown; they bear general similarity to the grazing encounters but with weaker effects.



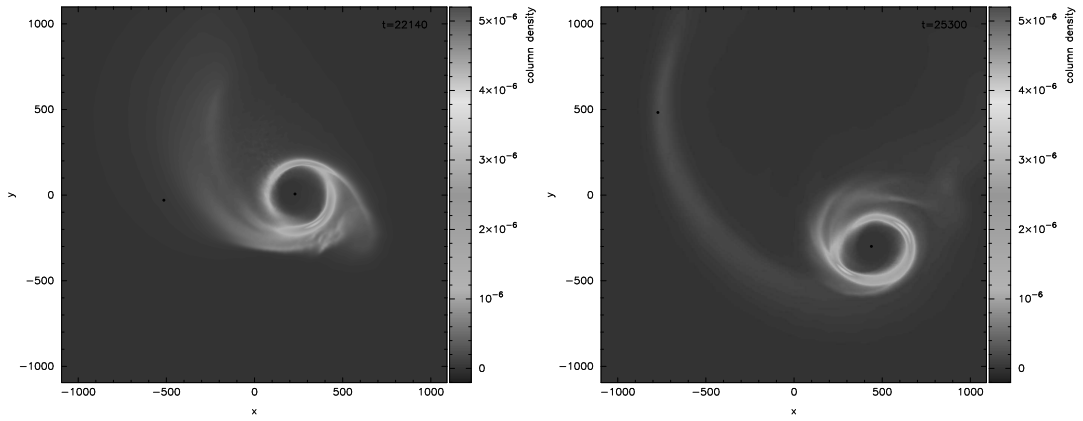
(a) $t = 22,140$ years. (b) $t = 25,300$ years.

Figure 9.2: Penetrating tilted prograde disc-star encounter after periastron, with no fragmentation, showing a prominent disc swept up around the perturbing star.



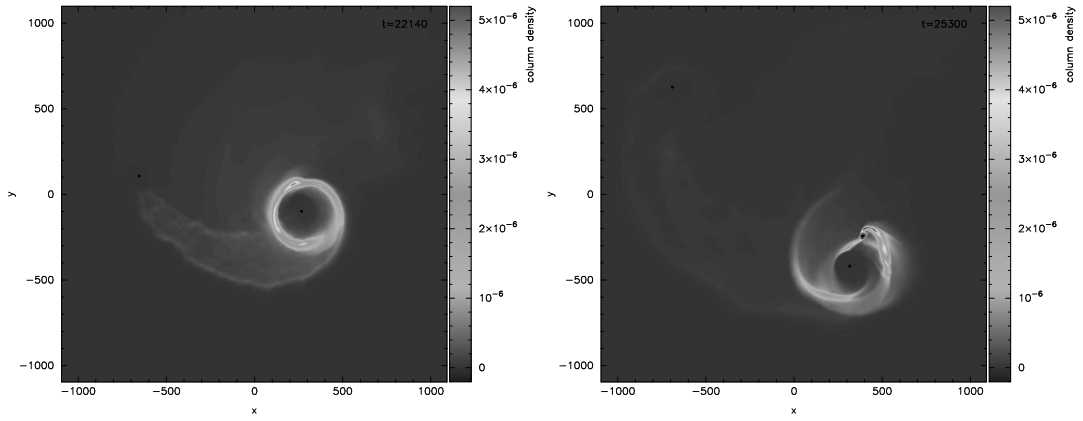
(a) $t = 22,140$ years. (b) $t = 25,300$ years.

Figure 9.3: Penetrating orthogonal disc-star encounter after periastron, with no fragmentation.



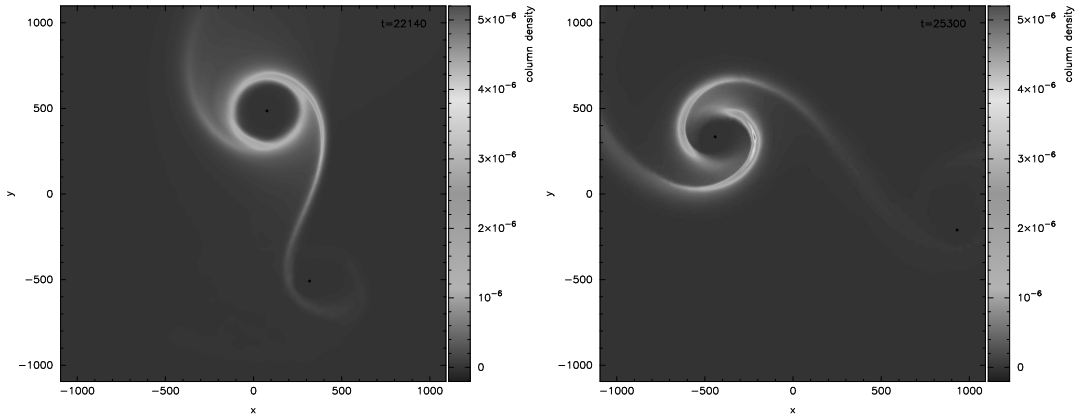
(a) $t = 22,140$ years. (b) $t = 25,300$ years.

Figure 9.4: Penetrating tilted retrograde disc-star encounter after periastron, which will soon fragment to form two free-floating planets (see Figure 9.11a).



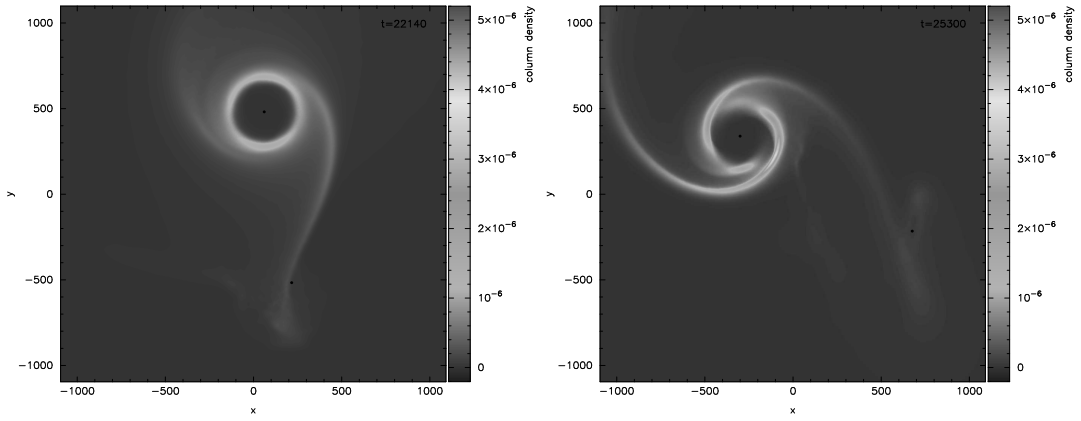
(a) $t = 22,140$ years. (b) $t = 25,300$ years.

Figure 9.5: Penetrating retrograde coplanar disc-star encounter after periastron, which fragments to form a brown dwarf companion.



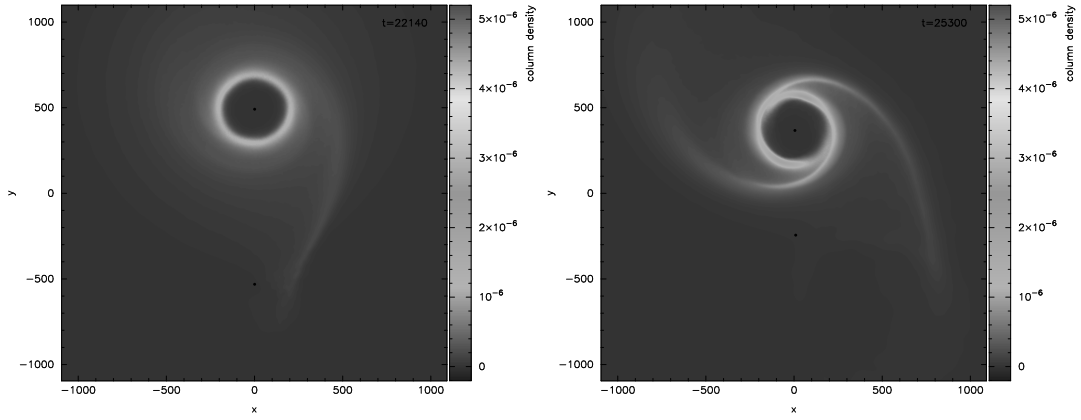
(a) $t = 22,140$ years. (b) $t = 25,300$ years.

Figure 9.6: Grazing prograde coplanar disc-star encounter after periastron, with no fragmentation.



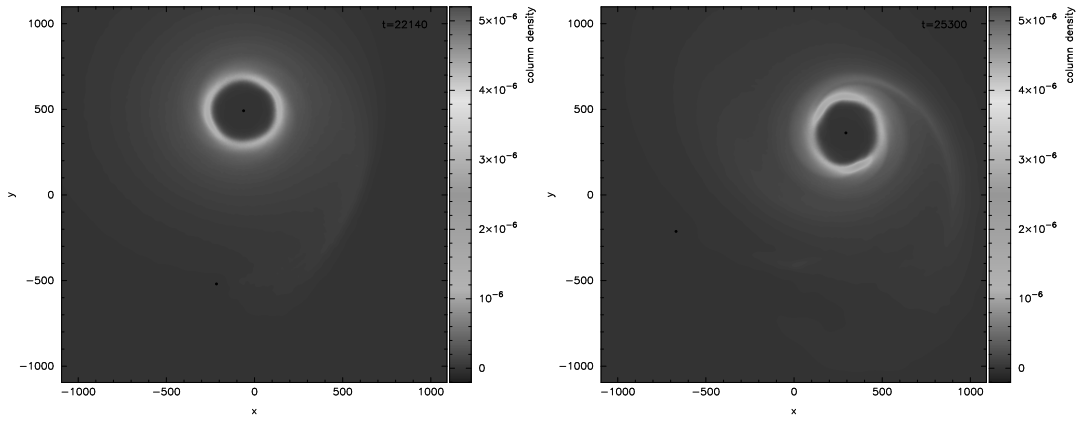
(a) $t = 22,140$ years. (b) $t = 25,300$ years.

Figure 9.7: Grazing tilted prograde disc-star encounter after periastron, with no fragmentation.



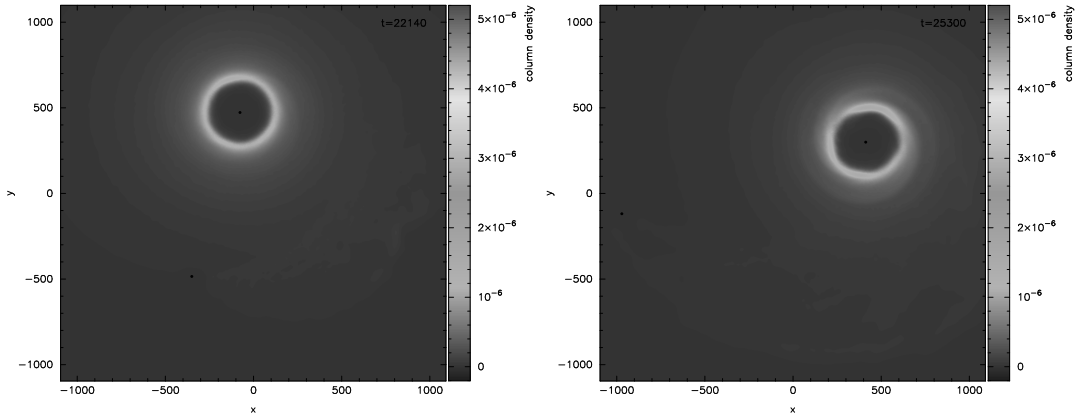
(a) $t = 22,140$ years. (b) $t = 25,300$ years.

Figure 9.8: Grazing orthogonal disc-star encounter after periastron, with no fragmentation.



(a) $t = 22,140$ years. (b) $t = 25,300$ years.

Figure 9.9: Grazing tilted retrograde disc-star encounter after periastron, with no fragmentation.



(a) $t = 22,140$ years. (b) $t = 25,300$ years.

Figure 9.10: Grazing retrograde coplanar disc-star encounter after periastron, with no fragmentation.

9.1.4 Fragmentation

Figure 9.11 shows the two encounters that resulted in disc fragmentation: the retrograde penetrating encounters.

The retrograde coplanar encounter ($\phi = \pi$) formed a sink ~ 5000 years after periastron, at $r_{disc} \sim 212$ AU with mass $m_{frag} = 0.169 \times 10^{-3}\mathrm{M}_{\odot}$ and size $r_{frag} = 0.236$ AU. It rapidly accretes to over $6\mathrm{M}_{\mathrm{J}}$ and then continues to steadily accrete further material, reaching $18\mathrm{M}_{\mathrm{J}}$ after ~ 1500 years of orbital motion to become a brown dwarf companion.

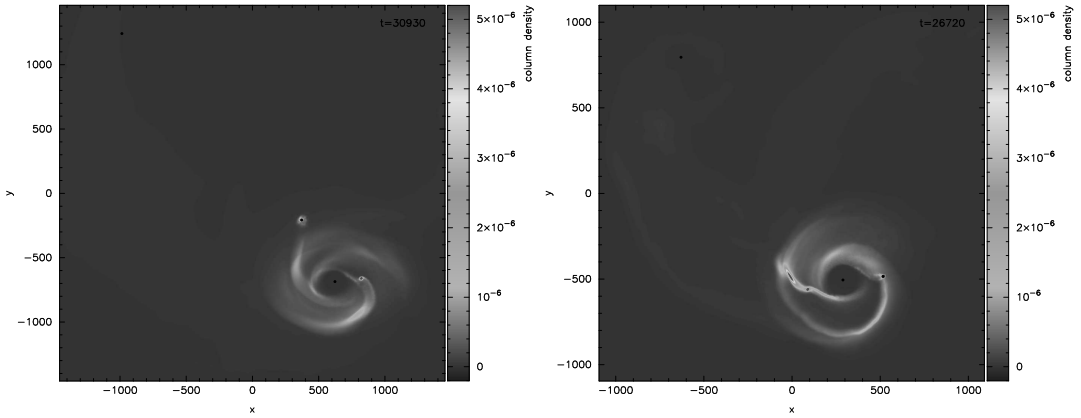
The tilted retrograde encounter ($\phi = \frac{3\pi}{4}$) formed two sinks after $\sim 10,000$ years: one at $r_{disc} \sim 514$ AU with mass $m_{frag} = 0.276 \times 10^{-3}\mathrm{M}_{\odot}$ and size $r_{frag} = 0.132$ AU, and another at $r_{disc} \sim 922$ AU with mass $m_{frag} = 0.149 \times 10^{-3}\mathrm{M}_{\odot}$ and size $r_{frag} = 0.14$ AU. The first rapidly accretes to $4.5\mathrm{M}_{\mathrm{J}}$ and the second to $4.9\mathrm{M}_{\mathrm{J}}$; they look set to become massive free-floating planets.

In the retrograde penetrating encounters the perturbing star opposes the flow of the disc material and sweeps out a trailing arm of material. Its strong disrupting effect appears to be sufficient to destabilise an initially marginally stable disc ($Q = 1.5$).

9.1.5 Condensation

While none of the other encounter modes destabilises the disc to the point of sink creation, condensations form and are ejected. These condensations do not satisfy the criteria required to form a sink particle (see Section 5.6.1), having not quite reached the density threshold $\rho_{\mathrm{sink}} = 10^{-6}\mathrm{kg\,m^{-3}}$. Ejected condensations are gravitationally bound, however, with their collapse slowed as it enters the adiabatic phase. They should therefore survive as free-floating planets.

In the isolated scenario this disc ejects a condensation after $35,000$ years, providing a basis for comparison. In the context of free-floating planet formation,



(a) Tilted retrograde at $t = 30,930$ years, where two free-floating planets have formed.

(b) Retrograde coplanar at $t = 26,720$ years, where a brown dwarf companion has formed in the right-hand spiral arm.

Figure 9.11: Disc fragmentation in penetrating retrograde disc-star encounters.

even $Q = 1.5$ might be considered marginally unstable.

The orthogonal penetrating encounter ejects two condensations by this point, while the prograde penetrating encounters eject no condensations but instead discs are swept up around the perturbing star. Figure 9.2(b) shows a prominent disc formed in the tilted prograde encounter.

In the grazing encounters, the prograde encounter ejects two condensations while the tilted prograde encounter ejects one condensation but forms a more prominent secondary disc. The orthogonal encounter forms four condensations and ejects at least two of them, but on a timescale $\sim 10,000$ years later. The retrograde tilted encounter ejects two condensations while the retrograde coplanar encounter forms and ejects one condensation on a timescale $\sim 25,000$ years later.

In the distant encounters, the prograde coplanar and tilted encounters both eject two condensations. The orthogonal encounter ejects four condensations at $\sim 35,000$ years. The retrograde tilted encounter ejects one condensation, as does the retrograde coplanar encounter but on a timescale $\sim 15,000$ years later.

A tilted prograde encounter appears to form the most prominent disc around the perturbing star; the prograde motion goes with the flow of the disc material while the tilt softens the shock of the encounter compared to the coplanar scenario.

An orthogonal encounter appears to cause the most disruption to the disc evolution (short of fragmentation by retrograde penetrating encounters), producing the largest number of ejected condensations.

9.1.6 Accretion

Table 9.1 summarises the enhancement of the accretion of disc material onto the central sink.

Closer encounters provide greater accretion enhancement. For grazing and distant encounters, prograde encounters result in greater accretion than retrograde encounters. For penetrating encounters the same trend is initially followed, but is

		$p = 0.5r_{disc}$ (penetrating)	$p = r_{disc}$ (grazing)	$p = 1.5r_{disc}$ (distant)
$\phi = 0$	(prograde)	0.110	0.086	0.050
$\phi = \pi/4$		0.108	0.071	0.049
$\phi = \pi/2$	(orthogonal)	0.124	0.052	0.050
$\phi = 3\pi/4$		> 0.081	0.054	0.045
$\phi = \pi$	(retrograde)	> 0.116	0.049	0.042

Table 9.1: Additional mass accreted onto the central sink in $M_\odot$ for disc-star encounters with periastron p and orbital inclination ϕ at $t \sim 37,000$ years.

quickly overridden when the disc is destabilised. Fragmentation results in a sustained enhancement of accretion, as does the formation of several transient condensations in the disruptive orthogonal encounter. The ">" symbol appears in the results for the fragmenting encounters because sink formation prevented the evolution from being followed up to the comparison time of $t \sim 37,000$ years.

Table 9.2 summarises the enhancement of the accretion of disc material onto the perturbing sink.

		$p = 0.5r_{disc}$ (penetrating)	$p = r_{disc}$ (grazing)	$p = 1.5r_{disc}$ (distant)
$\phi = 0$	(prograde)	0.105	0.048	0.014
$\phi = \pi/4$		0.076	0.021	0.004
$\phi = \pi/2$	(orthogonal)	0.019	0.004	0.001
$\phi = 3\pi/4$		0.013	0.004	0.001
$\phi = \pi$	(retrograde)	> 0.066	0.014	0.003

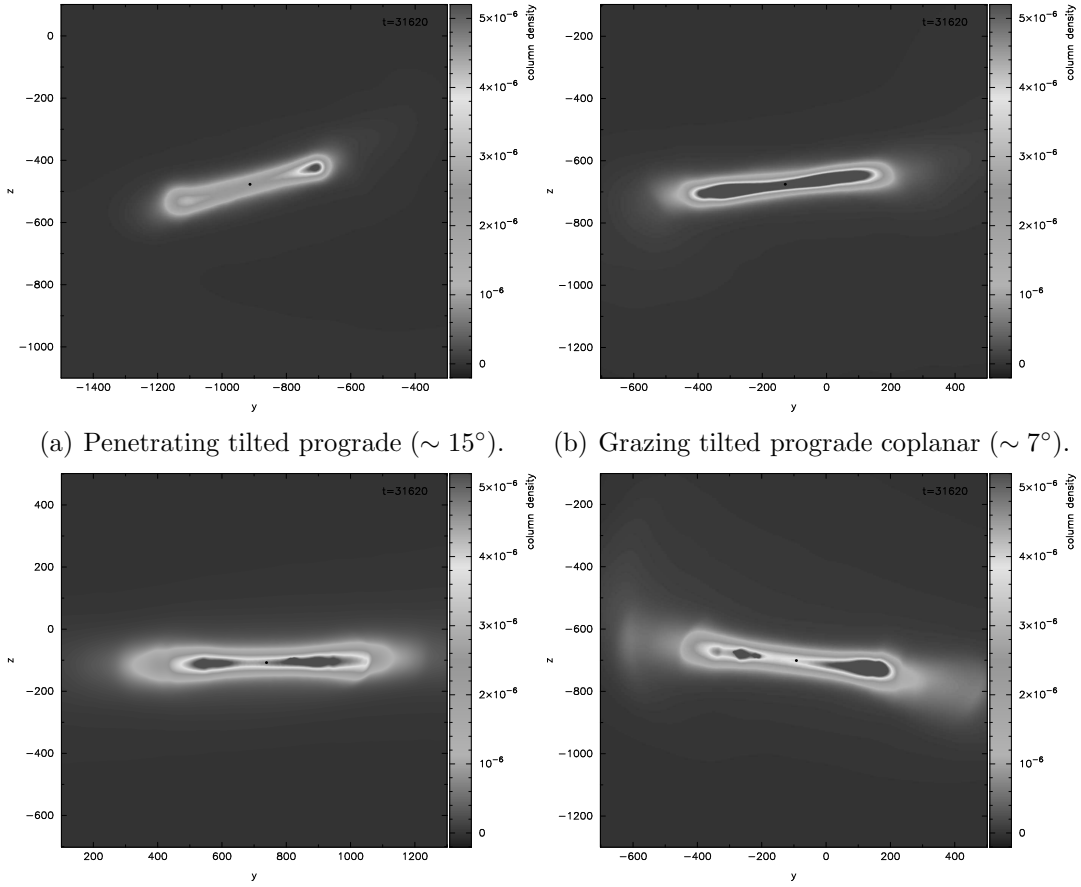
Table 9.2: Additional mass accreted onto the perturbing sink in $M_\odot$ for disc-star encounters with periastron p and orbital inclination ϕ at $t \sim 37,000$ years.

Closer encounters again provide greater accretion enhancement. Prograde encounters result in greater accretion. The accretion declines as ϕ increases until the retrograde encounter ($\phi = \pi$). This is a coplanar encounter so the perturbing star passes through a significant quantity of disc material, enhancing its accretion. Nonetheless it does not accrete as much material as the $\phi = \frac{\pi}{4}$ encounter, which benefits from the accretion of material captured in a secondary disc.

These results are in broad agreement with other parameter studies (Pfalzner et al. 2005).

9.1.7 Disc Tilting

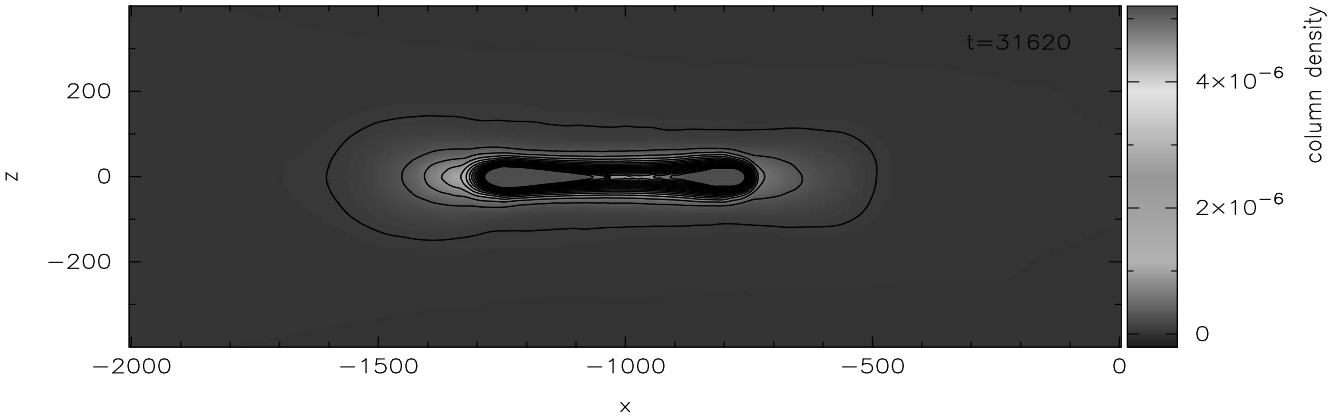
Figure 9.12 shows the disc tilting that results from non-coplanar encounters with $\phi = \frac{\pi}{4}$ and $\frac{3\pi}{4}$. The orthogonal encounters had a smaller tilting effect due to the rapid transit of the perturbing star through the disc.



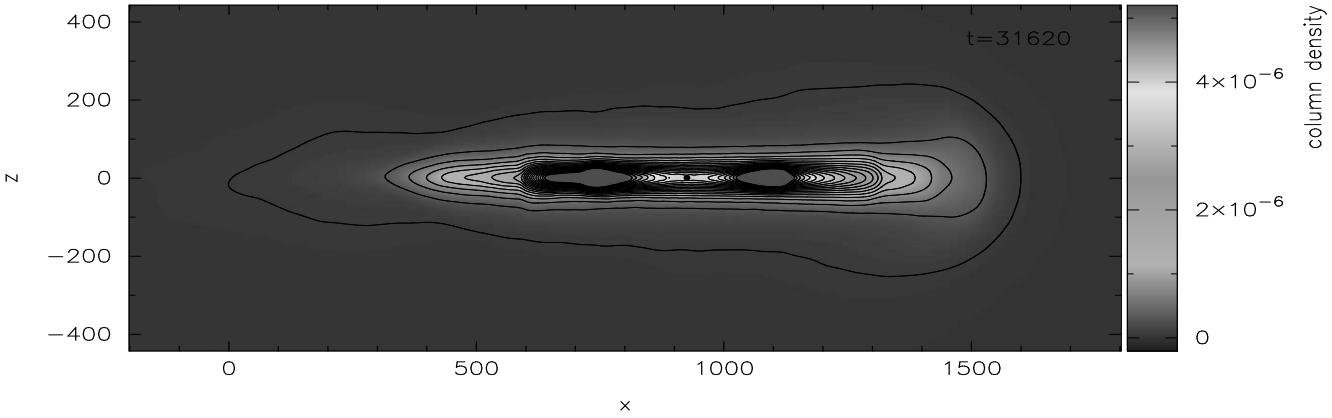
(a) Penetrating tilted prograde ($\sim 15°$). (b) Grazing tilted prograde coplanar ($\sim 7°$).

(c) Distant tilted prograde ($\sim 2°$). (d) Grazing tilted retrograde ($\sim -7°$).

Figure 9.12: Disc tilting in tilted disc-star encounters at $t = 31,620$ years.



(a) Grazing prograde, with significant truncation.



(b) Grazing retrograde.

Figure 9.13: Disc truncation in coplanar disc-star encounters at $t = 31,620$ years.

The tilted prograde encounter ($\phi = \frac{\pi}{4}$, or 45°) results in a disc tilt of $\sim 15°$ for the penetrating encounter, $\sim 7°$ for the grazing encounter and $\sim 2°$ for the distant encounter.

The tilted retrograde encounter ($\phi = \frac{3\pi}{4}$) results in a disc tilt of $\sim -18°$ for the penetrating encounter, $\sim -7°$ for the grazing encounter and $\sim -2°$ for the distant encounter.

A closer encounter has a greater tilting effect, but distant encounters also exert some effect. The retrograde penetrating encounter, which resulted in disc fragmentation, has a slightly enhanced tilting effect.

9.1.8 Disc Truncation

Figure 9.13 shows the disc truncation that results from coplanar encounters with $\phi = 0$ and π, with contours plotted to outline the disc material.

Grazing encounters are shown because no fragmentation occurs. The prograde encounter truncates the disc significantly, as expected from the accretion rates.

9.1.9 Conclusions

1. Retrograde penetrating encounters can fragment a marginally stable disc.
2. Orthogonal penetrating encounters disrupt the disc and result in significant accretion enhancement.
3. For grazing and distant encounters, the prograde encounters result in the greatest accretion enhancement onto both stars, and truncate the disc to the greatest extent.
4. Non-coplanar encounters tilt the disc.

If both stars formed out of similar cloud material then it might be expected that prograde coplanar encounters would be the most common, and strong accretion bursts would be associated with such encounters.

Penetrating retrograde encounters can trigger fragmentation in a marginally stable disc, forming brown dwarf companions (in our coplanar scenario) or free-floating planets (in our tilted scenario), but might be the least likely encounter scenario.

If star formation is coeval then disc-star interactions are less likely and disc-disc interactions more probable. Disc-star interactions might then occur where a system has had its disc stripped by a previous interaction.

9.1.10 Disc-Star Simulations

Early work suggested that stellar encounters with massive extended discs might induce disc fragmentation to produce low-mass companion stars, with 20% of random encounters doing so, rising to 50% for prograde encounters (Boffin et al. 1998). Our results suggest that it produces objects of brown dwarf or planetary mass, and has the most dramatic effect in the retrograde scenario.

It has been suggested that fragmentation is unaffected by encounters, and that any discs that fragment in encounters would do so in isolation (Clarke et al. 2008). Our results suggest that only retrograde encounters might trigger disc fragmentation, so it does not appear to be a high-probability mechanism.

Recent simulations incorporating radiative transfer have indicated that the compressive and shock heating of an encounter will stabilise the disc, with the cooling rate being insufficient for fragmentation to occur (Forgan & Rice 2009a). These simulations focus on compact discs (≤ 40 AU), however, and simulations of massive extended discs suggest that tidal perturbation alone may be sufficient to trigger fragmentation, producing highly eccentric or ejected brown dwarfs (Thies et al. 2010). Our simulations were not so effective in terms of sink formation, but tidal perturbation in grazing encounters did encourage massive transient condensations to form, so the findings are not entirely inconsistent.

Apart from fragmentation, discs might also play a role in capture scenarios through the increased cross-sectional area that they provide, producing binary

and multiple systems. Any triples and quadruples formed by disc-star interactions might eventually be disrupted and account for the excess of binaries among pre-main sequence stars (McDonald & Clarke 1995). While many simulations suggest that capture is not a frequent mechanism in low-mass disc-star interactions, it is much more effective for massive stars (Moeckel & Bally 2007) and might explain the high binarity of massive stars in the Orion Nebula Cluster (Pfalzner & Olczak 2007).

Disc-star encounters could also play a role in explaining FU Orionis-like outbursts, since SPH simulations of encounters with marginally stable discs can initiate such outbursts (Forgan & Rice 2009b). Most FU Orionis stars do not have a companion, however, so this is not a robust explanation. Nonetheless the encounter-driven scenario predicts that such outbursts should occur predominantly close to the centre of young dense clusters (Pfalzner 2008). In such regions these encounters would also affect the evolution of the discs themselves, resulting in observational signatures such as the cluster disc fraction decreasing with increasing cluster density (Olczak et al. 2010).

9.2 Disc-Disc Interaction

To investigate the interactions of discs, we can use a replica of our $Q = 1.5$ disc system in place of the perturbing star. Inverting the relative positions of the particles in the replicated disc ($\mathbf{r} \mapsto -\mathbf{r}$) will preserve the centre-of-mass in our offset reference frame.

9.2.1 Encounter Modes

Simulations are performed at three periastron distances ($p = 3$, 2 and $1r_{disc}$), to model distant, grazing and penetrating encounters.

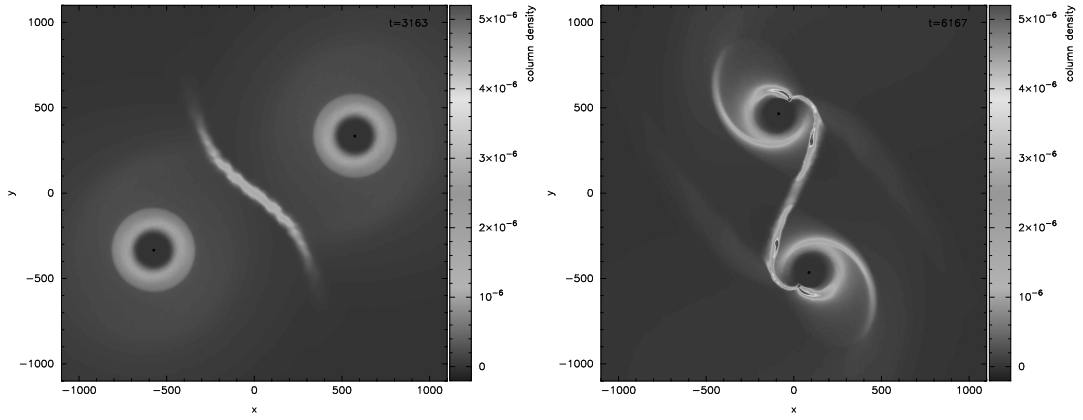
An inclination angle $\phi = 0$ models spin-orbit parallel (SOP) encounters: coplanar encounters where both discs are prograde, spinning in the same direction as the orbital motion. An inclination angle $\phi = \pi$ models spin-orbit mixed (SOM) encounters, where one disc is prograde and the other retrograde. Spin-orbit anti-parallel (SOA) encounters, where both discs are retrograde, can be modelled by reversing the rotations of the discs ($\mathbf{v} \mapsto -\mathbf{v}$) when $\phi = 0$.

Inclination angles of $\phi = \frac{\pi}{4}$, $\frac{\pi}{2}$ and $\frac{3\pi}{4}$ model non-coplanar encounters that are tilted or orthogonal between SOP and SOM.

Applying the inclination angle ϕ in the orbital equation will result in parallel-aligned discs interacting along an inclined orbit. To incline the disc instead, the orbital inclination angle is set to zero and a rotation matrix such as $R_x(\phi)$ is applied to the positions $\mathbf{r}$ and velocities $\mathbf{v}$ of the disc.

$$R_x(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix} \tag{9.15}$$

Since modelling two separate disc systems is computationally expensive and the orbital timescale of the interaction is quite long, the simulations are initiated just



(a) $t = 3163$ years. (b) $t = 6167$ years.

Figure 9.14: Penetrating spin-orbit parallel disc-disc encounter after periastron, showing the development of a strong shock layer.

before the point of periastron. While the disc-star encounters focused on gravitational interaction, this close-contact disc-disc scenario focuses greater attention on hydrodynamic interaction in the extended disc material.

9.2.2 Coplanar Encounters

Figures 9.14, 9.15 and 9.16 show snapshots of the penetrating encounters for SOP, SOM and SOA modes shortly after periastron.

Figure 9.17 shows snapshots of the grazing encounters for SOP, SOM and SOA modes after periastron. Snapshots of the distant encounters are not shown; they bear general similarity to the grazing encounters but with weaker effects.

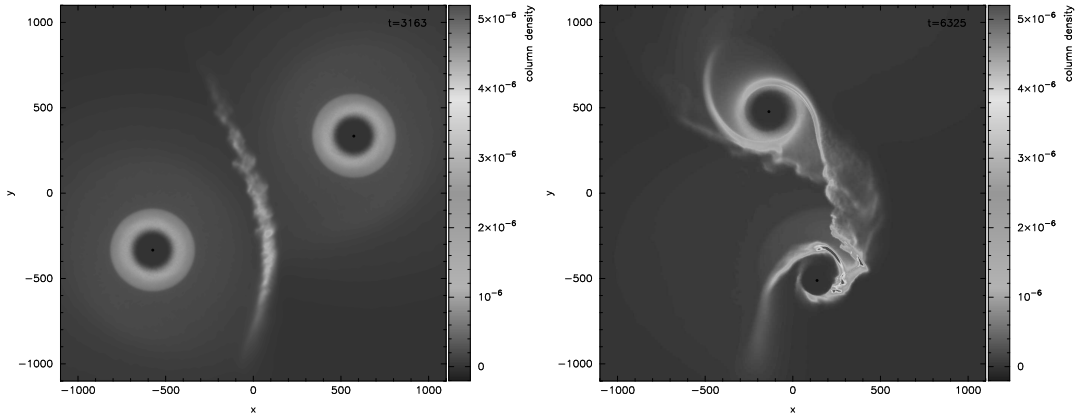
9.2.3 Non-Coplanar Encounters

Figures 9.18, 9.19 and 9.20 show snapshots of the penetrating encounters for disc inclinations $\phi = \frac{\pi}{4}$, $\frac{\pi}{2}$ and $\frac{3\pi}{4}$ shortly after periastron.

Figure 9.21 shows snapshots of the grazing encounters for disc inclinations $\phi = \frac{\pi}{4}$, $\frac{\pi}{2}$ and $\frac{3\pi}{4}$ after periastron. Snapshots of the distant encounters are not shown; they bear general similarity to the grazing encounters but with weaker effects.

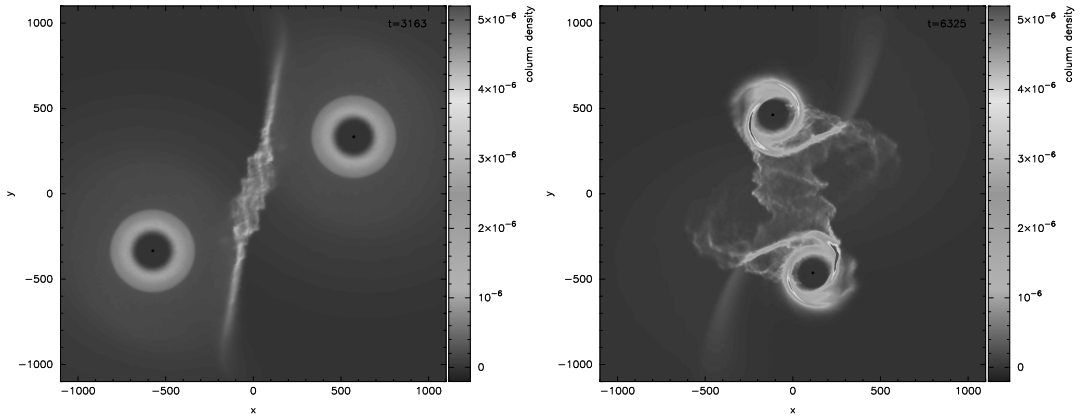
9.2.4 Shearing Encounters

Figures 9.22, 9.23 and 9.24 show snapshots of penetrating encounters for orbital inclinations $\phi = \frac{\pi}{4}$, $\frac{\pi}{2}$ and $\frac{3\pi}{4}$ shortly after periastron. In these encounters, both discs lie in the $x - y$ plane and shear through each other along an inclined orbital path.



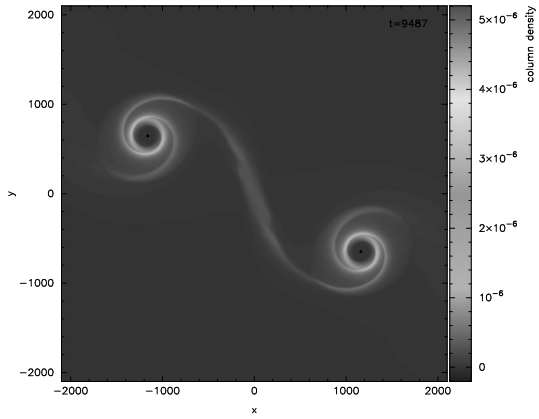
(a) $t = 3163$ years. (b) $t = 6325$ years.

Figure 9.15: Penetrating spin-orbit mixed disc-disc encounter after periastron, showing the development of a stronger shock layer.

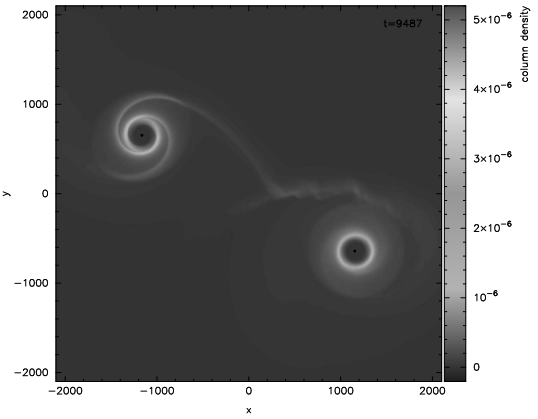


(a) $t = 3163$ years. (b) $t = 6325$ years.

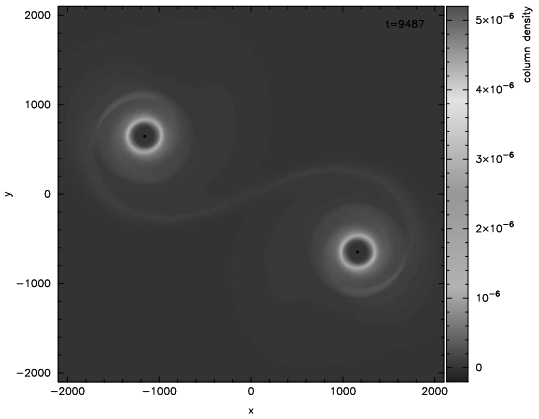
Figure 9.16: Penetrating spin-orbit anti-parallel disc-disc encounter after periastron, showing the development of the strongest shock layer.



(a) Spin-orbit parallel.

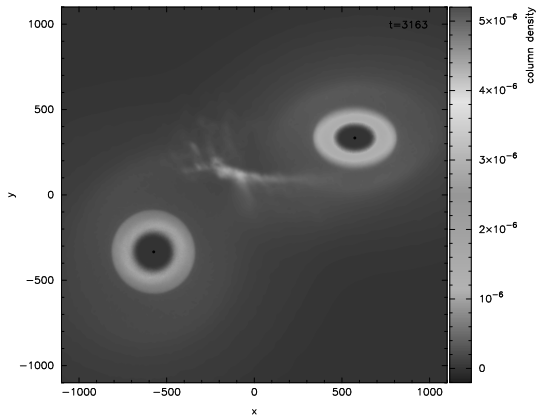


(b) Spin-orbit mixed.

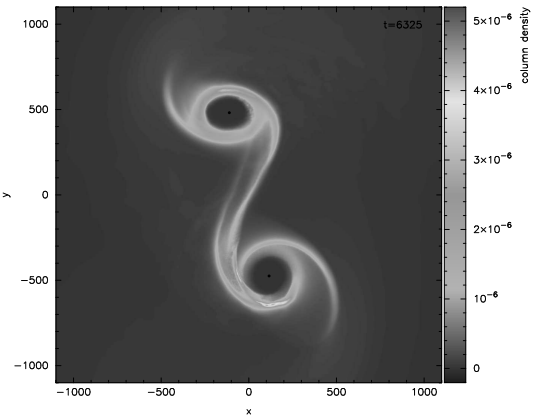


(c) Spin-orbit anti-parallel.

Figure 9.17: Grazing coplanar disc-disc encounters after periastron at $t = 9487$ years, with no fragmentation but enhanced formation of condensations.

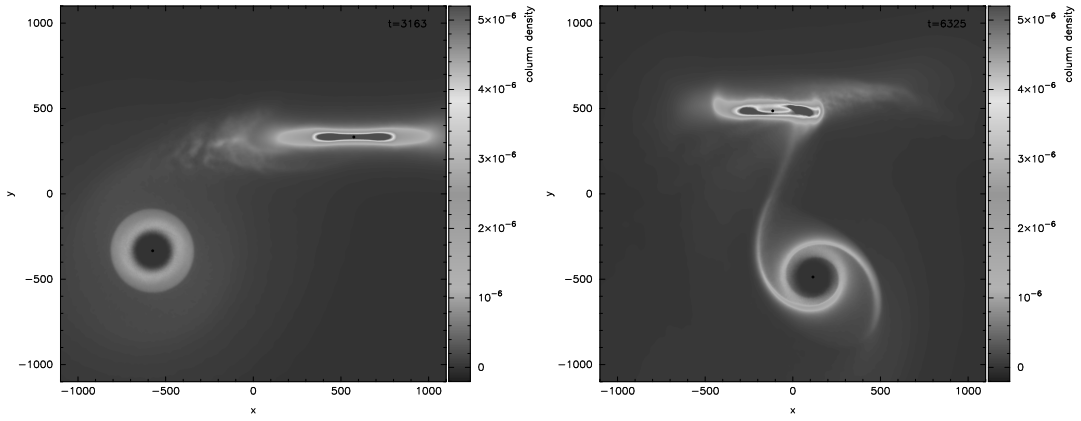


(a) $t = 3163$ years.



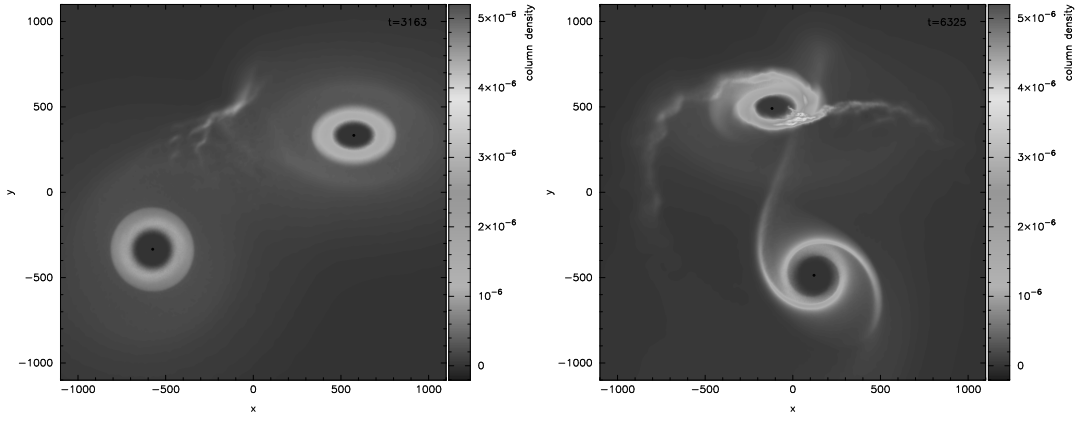
(b) $t = 6325$ years.

Figure 9.18: Penetrating $\phi = \frac{\pi}{4}$ disc-disc encounter after periastron, with shocked material forming condensations.



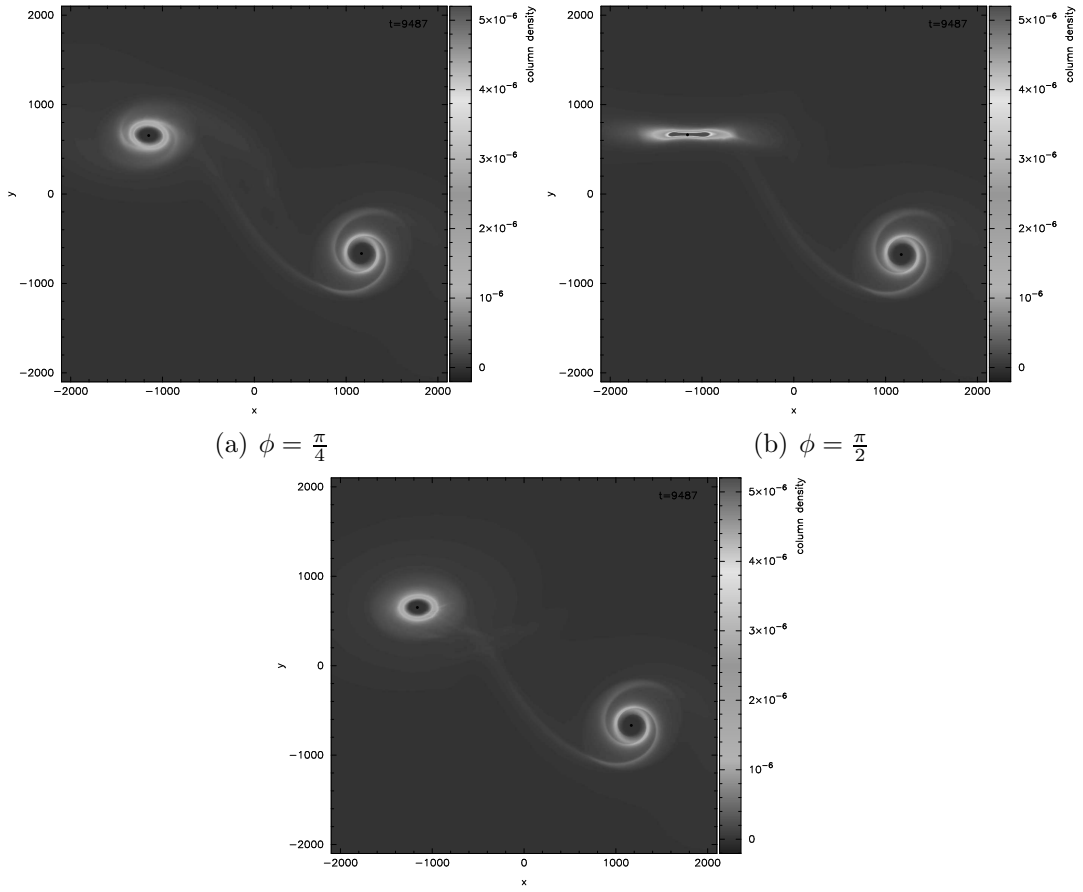
(a) $t = 3163$ years. (b) $t = 6325$ years.

Figure 9.19: Penetrating $\phi = \frac{\pi}{2}$ disc-disc encounter after periastron, with shocked material forming condensations.



(a) $t = 3163$ years. (b) $t = 6325$ years.

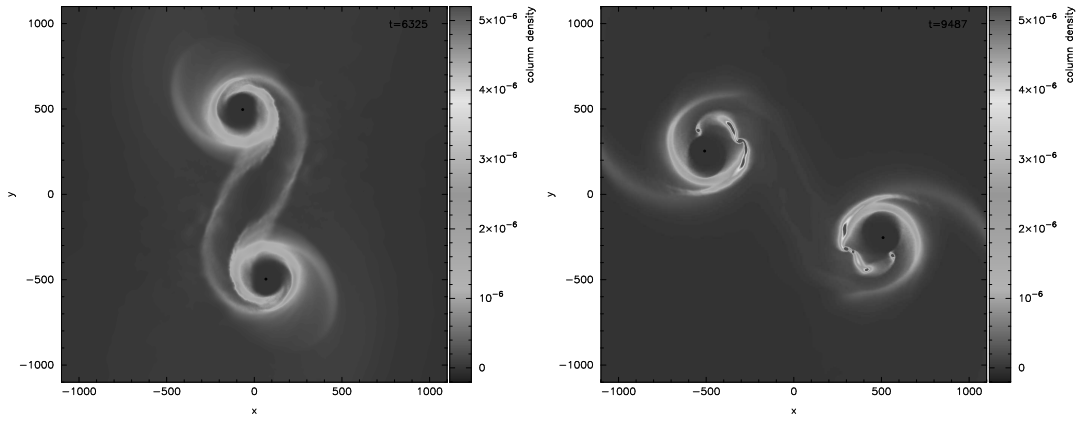
Figure 9.20: Penetrating $\phi = \frac{3\pi}{4}$ disc-disc encounter after periastron, with shocked material forming condensations.



(a) $\phi = \frac{\pi}{4}$ (b) $\phi = \frac{\pi}{2}$

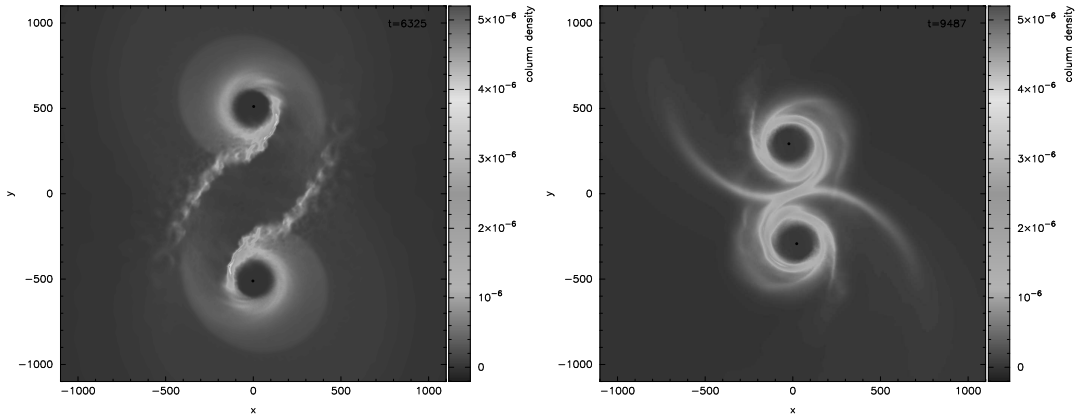
(c) $\phi = \frac{3\pi}{4}$

Figure 9.21: Grazing non-coplanar disc-disc encounters after periastron at $t = 9487$ years, with no fragmentation but enhanced formation of condensations.



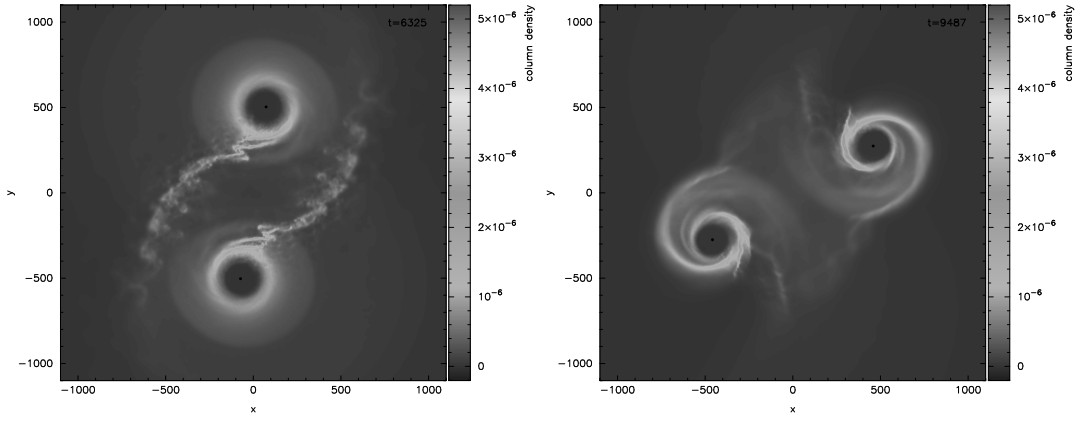
(a) $t = 6325$ years. (b) $t = 9487$ years.

Figure 9.22: Shearing $\phi = \frac{\pi}{4}$ disc-disc encounter after periastron, with very rapid formation of condensations.



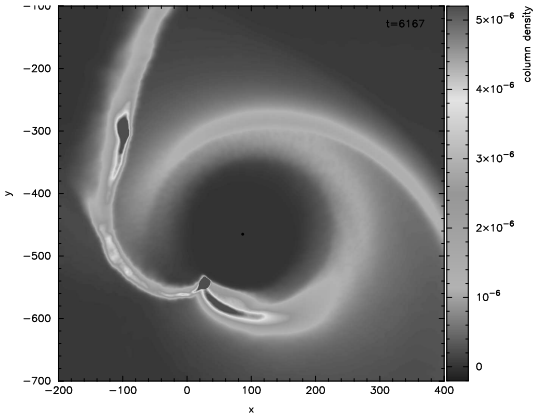
(a) $t = 6325$ years. (b) $t = 9487$ years.

Figure 9.23: Shearing $\phi = \frac{\pi}{2}$ disc-disc encounter after periastron, with rapid formation of condensations.

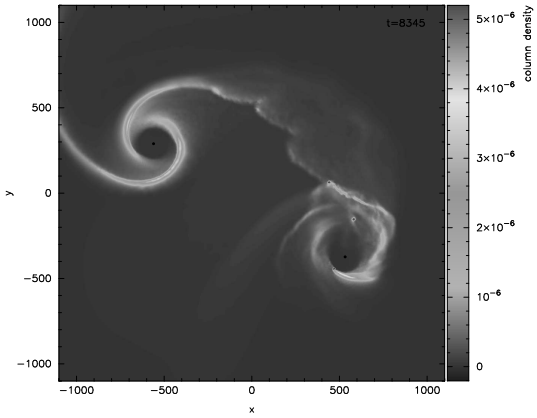


(a) $t = 6325$ years. (b) $t = 9487$ years.

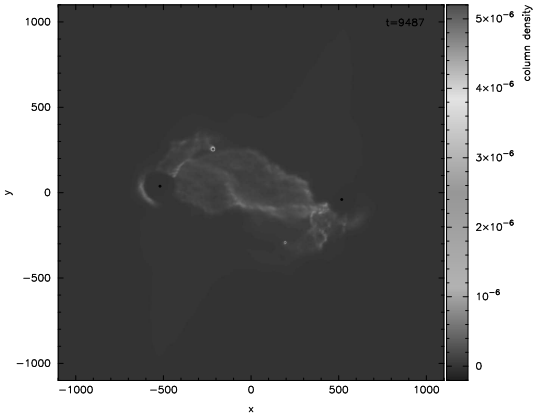
Figure 9.24: Shearing $\phi = \frac{3\pi}{4}$ disc-disc encounter after periastron, with fairly rapid formation of condensations.



(a) SOP close-up at $t = 6167$ years, showing dense shock-heated condensations in the tidal tail.



(b) SOM at $t = 8345$ years, showing rapid formation of condensations in the retrograde disc on the right.



(c) SOA at $t = 9487$ years, showing rapid dissipation of the discs.

Figure 9.25: Coplanar penetrating disc-disc encounters.

9.2.5 Evolution

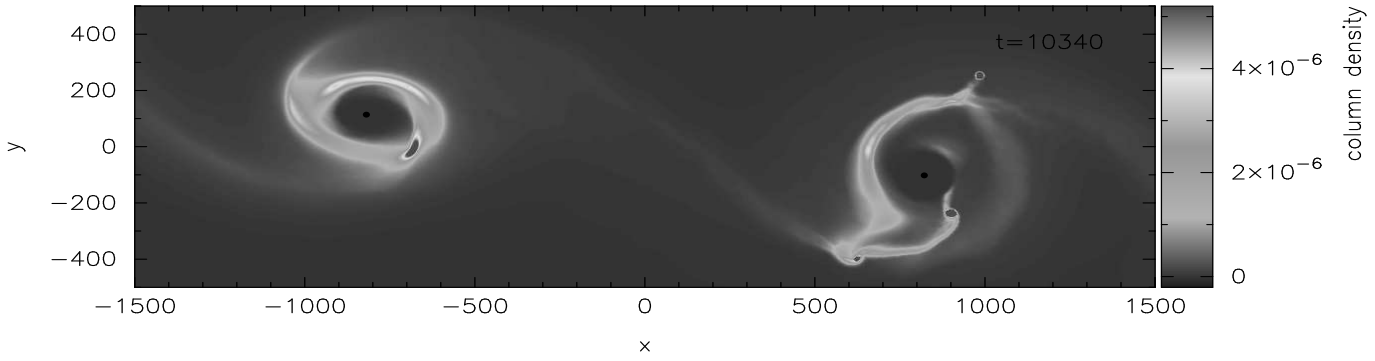
None of the simulations undergoes fragmentation to the point of sink creation, but several form condensations.

Figure 9.14 shows the development of a shock layer in the SOP penetrating encounter. Figure 9.15 shows a stronger shock for the SOM encounter, and Figure 9.16 the strongest for the SOA encounter.

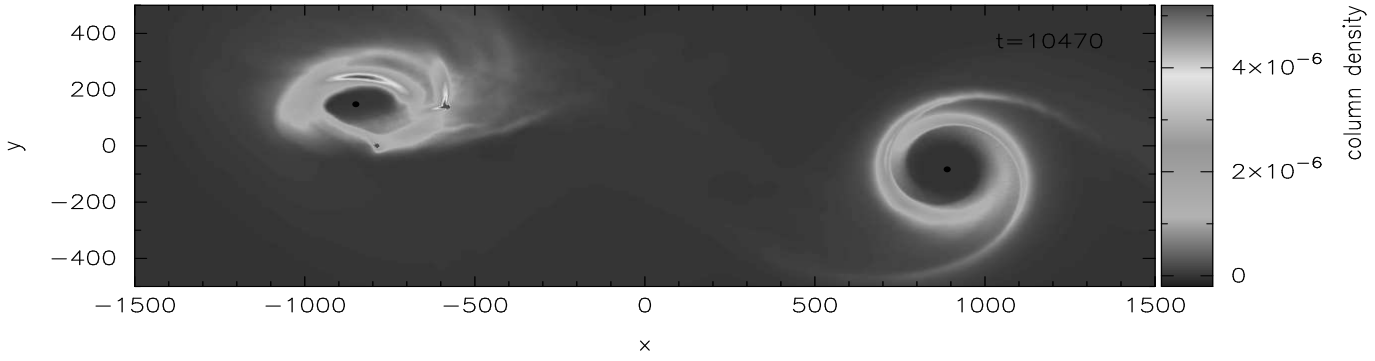
The SOP encounter, with connecting tidal tails, is the strongest candidate for fragmentation. Dense condensations form from material piling into the tidal tail (Figure 9.25 a), but the evolution of the system is crippled by a small timestep due to the shock heating.

The SOM encounter results in the rapid formation of condensations in the retrograde disc (Figure 9.25 b). The clumps are likely to be ejected, but again the timestep cripples the evolution due to the shock heating of the innermost clump.

The SOA encounter is highly dissipative, with two condensations ejected as the discs are rapidly destroyed (Figure 9.25 c). The retrograde angular momentum



(a) $\phi = \frac{\pi}{4}$ (prograde), showing greater destabilisation in the level disc (right).



(b) $\phi = \frac{3\pi}{4}$ (retrograde), showing greater destabilisation in the tilted disc (left).

Figure 9.26: Non-coplanar penetrating disc-disc encounters.

of the discs counters that of the orbit, resulting in a binary capture. Capture is negligible in the SOP mode, but random inclination can result in a 20% capture rate (Watkins et al. 1998b); disc-disc interactions increase the capture rate by up to a factor ~ 6 compared with disc-star interactions (Watkins 1996).

Non-coplanar penetrating encounters (Figures 9.18 to 9.20) also trigger the formation of condensations. The timestep again prevents evolution, but the clumps will probably be ejected or transitory. Figure 9.26 shows that with a prograde inclination (a) the level disc is more destabilised while with a retrograde inclination (b) the tilted disc is more destabilised. In both cases the level disc will have a greater volume of material involved in the interaction because it lies on the orbital plane and therefore collides edge-on. This provides the greater effect in (a), but in (b) the retrograde motion is seen to outweigh this factor. In the orthogonal encounter, the tilted disc destabilises first since it lies in the $x - z$ plane; a disc aligned in the $y - z$ plane should provide a symmetric encounter.

The penetrating shearing encounters (Figures 9.22 to 9.24) result in rapid clump formation, with the shortest timescale for the prograde angle ($\phi = \frac{\pi}{4}$) and the longest for the retrograde angle ($\phi = \frac{3\pi}{4}$).

In the isolated scenario the disc ejects a condensation after $35,000$ years.

In grazing coplanar encounters (Figure 9.17), the SOP encounter forms and ejects two or three clumps from each disc by $32,000$ years. In the SOM encounter, the retrograde disc ejects a clump after $18,000$ years. In the SOA encounter, both

discs are already forming two or three clumps after 31,000 years.

In grazing non-coplanar encounters (Figure 9.21), the $\phi = \frac{\pi}{4}$ encounter forms two or three clumps in each disc before 30,000 years. In the $\phi = \frac{\pi}{2}$ encounter, the tilted disc ejects a clump at 30,000 years. In the $\phi = \frac{3\pi}{2}$ encounter, the tilted (retrograde) disc has formed three clumps at 29,000 years.

In distant coplanar encounters, the SOP encounter forms clumps on a marginally faster timescale than the isolated scenario. In the SOM encounter, the retrograde disc forms several clumps and ejects one by 32,000 years. In the SOA encounter there is little significant effect.

In distant non-coplanar encounters, the $\phi = \frac{\pi}{4}$ encounter forms clumps on a slightly faster timescale, with greater activity in the level disc. The $\phi = \frac{\pi}{2}$ encounter forms clumps on a faster timescale, with greater activity in the tilted disc in the $x-z$ plane. In the $\phi = \frac{3\pi}{4}$ encounter, the tilted (retrograde) disc forms clumps on a faster timescale with greater activity.

9.2.6 Accretion

Table 9.3 summarises the enhancement of the accretion of disc material onto the first central sink.

	$p = r_{disc}$ (penetrating)	$p = 2r_{disc}$ (grazing)	$p = 3r_{disc}$ (distant)
SOP (prograde)	>0.030	0.031	0.030
SOM (retrograde)	>0.038	0.031	0.030
SOA (retrograde)	0.397	0.031	0.030
$\phi_{disc} = \pi/4$	0.056	0.031	0.030
$\phi_{disc} = \pi/2$	0.089	0.034	0.030
$\phi_{disc} = 3\pi/4$	0.136	0.044	0.030
$\phi_{orbit} = \pi/4$	0.078	n/a	n/a
$\phi_{orbit} = \pi/2$	0.087	n/a	n/a
$\phi_{orbit} = 3\pi/4$	0.157	n/a	n/a

Table 9.3: Additional mass accreted onto the first central sink in $M_\odot$ for disc-disc encounters with periastron p and various orbital modes at $t \sim 10,000$ years.

Closer encounters provide greater accretion enhancement. Distant encounters have no discernable initial effect. For grazing encounters, orthogonal and retrograde disc-tilted encounters result in slightly greater accretion onto the level disc. For penetrating encounters the accretion increases as the perturbing disc becomes more retrograde-inclined (as ϕ_{disc} increases), being greatest for the SOA mode. For shearing encounters the accretion is also greatest for retrograde motion.

Table 9.4 summarises the enhancement of the accretion of disc material onto the second central sink.

Closer encounters again provide greater accretion enhancement, with no initial difference for distant or grazing encounters. For penetrating encounters the accretion

	$p = r_{disc}$ (penetrating)	$p = 2r_{disc}$ (grazing)	$p = 3r_{disc}$ (distant)
SOP (prograde)	>0.030	0.031	0.030
SOM (prograde)	>0.260	0.030	0.030
SOA (retrograde)	0.408	0.031	0.030
$\phi_{disc} = \pi/4$	0.050	0.031	0.030
$\phi_{disc} = \pi/2$	0.038	0.031	0.030
$\phi_{disc} = 3\pi/4$	0.038	0.031	0.030
$\phi_{orbit} = \pi/4$	0.079	n/a	n/a
$\phi_{orbit} = \pi/2$	0.087	n/a	n/a
$\phi_{orbit} = 3\pi/4$	0.156	n/a	n/a

Table 9.4: Additional mass accreted onto the second central sink in $M_\odot$ for disc-disc encounters with periastron p and various orbital modes at $t \sim 10,000$ years.

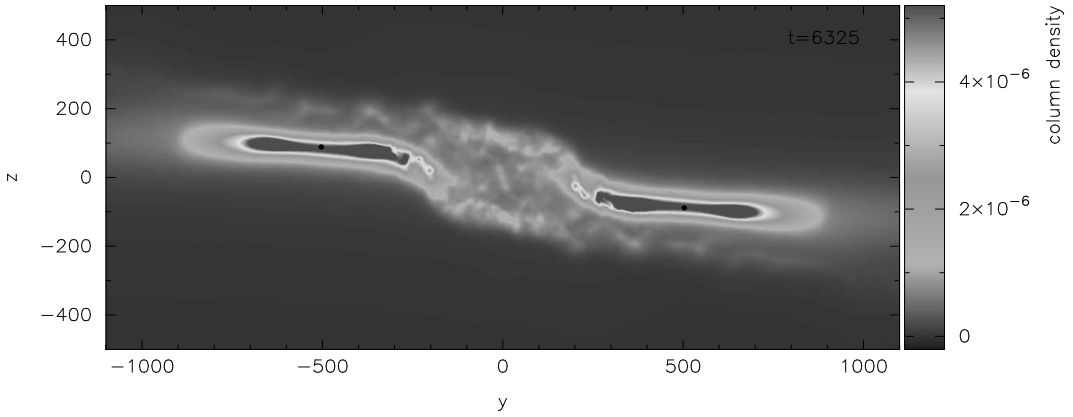


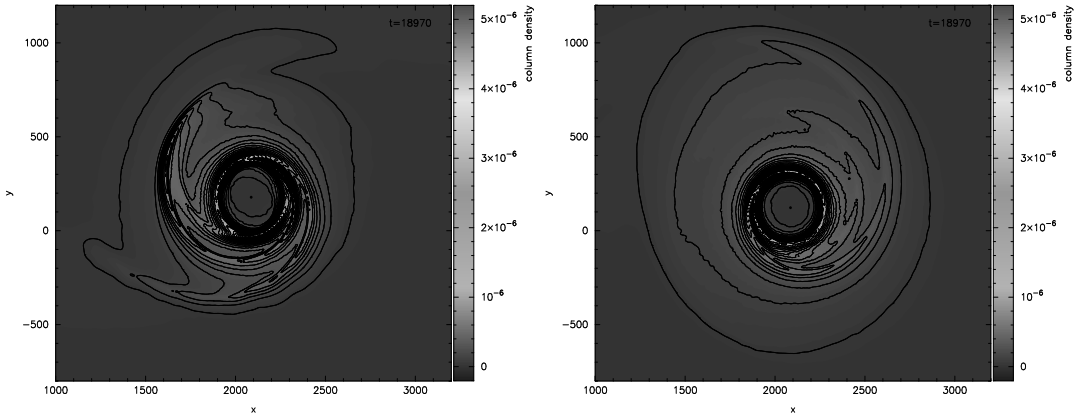
Figure 9.27: Disc tilting and twisting in a penetrating shearing disc-disc encounter.

decreases as the perturbing disc becomes more retrograde-inclined (as ϕ_{disc} increases), but is greatest when in coplanar retrograde motion (in the SOM and especially the SOA mode). For shearing encounters the accretion is again greatest for retrograde motion; being orbitally symmetric, both sinks follow the same accretion history.

In non-coplanar encounters, a prograde inclined disc ($\phi = \frac{\pi}{4}$) triggers greater accretion in the level disc than itself, due to the greater volume of material disrupted. A retrograde inclined disc ($\phi = \frac{3\pi}{4}$) feeds a trail of shocked material into the level disc in the same manner as the SOM mode. Note that, due to the manner in which the coplanar modes were initialised, it is the first disc in the SOM mode that has the retrograde motion and feeds material onto the second disc.

9.2.7 Disc Tilting

An encounter with an inclined disc does not significantly tilt the discs. A penetrating shearing encounter, however, where the orbital path is inclined, results in a tilting effect (see Figure 9.27). It also has a twisting effect on the outer edge of the disc.



(a) Spin-orbit parallel (prograde), with significant truncation.

(b) Spin-orbit anti-parallel (retrograde).

Figure 9.28: Disc truncation in coplanar grazing disc-disc encounters at $t = 18,970$.

9.2.8 Disc Truncation

Figure 9.28 shows the disc truncation that results from coplanar encounters in the SOP and SOA modes, with contours plotted to outline the disc material.

Grazing encounters are shown because penetrating encounters are halted by condensation (SOP) or dissipate the disc (SOA). The prograde SOP encounter truncates the disc to a greater extent, as expected from the results of the disc-star interactions.

9.2.9 Conclusions

1. SOP penetrating encounters produce dense condensations (as does the SOM retrograde disc) but shock heating appears to inhibit fragmentation.
2. SOA penetrating encounters dissipate the disc and result in binary capture.
3. Grazing prograde coplanar encounters truncate the disc to the greatest extent.
4. Shearing encounters tilt and twist the disc.

SOP encounters, which might be more common, would result in disc truncation and negligible capture. Material piling into the tidal tails during penetrating encounters might result in fragmentation, but the strong shock heating may prevent this.

Penetrating SOA encounters do not trigger fragmentation but instead dissipate the discs and result in binary capture, although this might be the least likely encounter mode.

Disc-disc interactions do lead to greater instability, since more condensations do form, and on a shorter timescale, but for our $Q = 1.5$ disc they are transient or ejected and do not result in sink formation. A slightly less stable disc ($Q \sim 1.4$) might allow for induced fragmentation, but disc-disc interaction does not appear to be a strong triggering mechanism.

9.2.10 Disc-Disc Simulations

Early work suggested that interactions between massive extended discs might induce disc fragmentation to produce low-mass companion stars, either via the fragmentation of a shock layer in coplanar encounters (Watkins et al. 1998a) or within spiral arms triggered by non-coplanar encounters (Watkins et al. 1998b). The efficiency of this mechanism generated 0.6 companions on average for randomly inclined encounters, rising to almost double for SOP encounters. Our results show that condensations do form, but do not reach sink creation densities. The SOP mode, however, does appear to be the best candidate for inducing fragmentation.

Some simulations have claimed that disc-disc interactions can actually inhibit the fragmentation of discs that would fragment in isolation, due to tidally induced shock heating (Mayer et al. 2005). These simulations involve massive compact discs ($m_{disc} \sim 0.1\mathrm{M}_{\odot}$, $r_{disc} \sim 20$ AU) in close binaries with a separation of ~ 60 AU. At separation ~ 120 AU fragmentation can proceed unhindered, so they suggest that fewer planets might be expected to form in binary systems with separations < 100 AU. Our simulations suggest that shock heating in disc-disc encounters is likely to inhibit a triggered fragmentation process.

Other simulations have formed disc-like objects of planetary to brown-dwarf mass ($2 - 73\mathrm{M}_{\mathrm{J}}$) involving massive extended discs ($m_{disc} \sim 0.6\mathrm{M}_{\odot}$, $r_{disc} \sim 1500$ AU) that are initially stable ($Q_{MIN} \sim 1.6$) undergoing penetrating disc-disc encounters with $p \sim 1000$ AU (Shen & Wadsley 2006). These objects have a fairly high ejection rate, and are claimed to fragment in the shock layers, the prograde tidal tails and via instabilities triggered in the inner disc. No sink particles are used, however, and their fixed gravitational softening length probably enhances fragmentation (see Section 4.3.8). Their objects correspond to the condensations observed in our simulations.

More recent simulations have claimed the formation of objects from 0.9 to $127\mathrm{M}_{\mathrm{J}}$, suggesting that disc-disc interaction could be a fairly frequent mechanism for brown dwarf formation (Shen et al. 2010). They use a locally isothermal equation of state with no shock heating, however, and claim that cooling would be efficient in the outer regions of the disc (cooling parameter $\beta = 0.3$ to 0.7 beyond 50 AU). Our simulations suggest that more detailed thermal physics inhibits the formation mechanism. They also use only 32 SPH neighbours, which is a low value for 3D simulations.

It is also interesting to note that SPH simulations are used to investigate disc-disc interactions in the mergers of gas-rich disc galaxies (Richard et al. 2010).

9.3 Disc-Planet Interaction

Once a planet (or indeed any other object) forms in a disc, its interaction with the disc in which it is embedded may become the subject of investigation.

It may exert a cooling effect in its locality due to the shadow that it casts in the modelling of radiative transfer (Jang-Condell & Sasselov 2003).

A planet may undergo migration within the disc, in the fast type I regime for terrestrial masses or, if sufficiently massive to open a gap in the disc ($\sim$ 10 Earth masses), in the slower type II regime that provides inward migration on the

accretion timescale and explains the existence of "hot Jupiters" (Masset & Papaloizou 2003). Gravitational scattering might also occur, which may explain the eccentricity distribution of extra-solar planets (Moeckel, Raymond & Armitage 2008).

Planetary accretion and migration within a disc has been studied using SPH simulations (Shaefer et al. 2004). A thin non-self-gravitating disc is modelled (quite unlike our simulations), and gap formation does not inhibit accretion onto the planet.

Regarding the formation of planets via core accretion, a detailed investigation of the dynamics of planetesimals in protoplanetary discs may require the incorporation of magnetohydrodynamics into a code (Nelson & Gressel 2010).

Massive planets ($10-15\mathrm{M_J}$) embedded in discs might also provide a mechanism for triggering FU Orionis outbursts (Lodato & Clarke 2004).

9.4 Summary

We have investigated the effect of disc-star and disc-disc interactions on the evolution of a marginally stable disc ($Q = 1.5$).

Retrograde penetrating disc-star encounters can fragment a marginally stable disc, while orthogonal penetrating disc-star encounters disrupt the disc and result in significant accretion enhancement.

For grazing and distant disc-star encounters, prograde encounters result in the greatest accretion enhancement onto both stars, and truncate the disc to the greatest extent. Non-coplanar encounters will tilt the disc.

Spin-orbit parallel penetrating disc-disc encounters produce dense condensations, but shock heating is likely to inhibit fragmentation. Anti-parallel penetrating encounters dissipate the disc and result in binary capture.

Grazing prograde coplanar disc-disc encounters truncate the disc to the greatest extent, and shearing encounters will tilt and twist the disc.

Interactions do not appear to be a likely mechanism for inducing fragmentation in otherwise stable discs. They are likely to be associated with observational signatures, however, such as accretion bursts, and will significantly affect the evolution of the disc itself.

"If there's one thing I hope
I've taught you, it's that
success is 95 percent blind naïveté
and 5 percent other stuff."

Homer J. Simpson

Chapter 10

Summary

Aware of the powers and pitfalls of numerical methods, we have developed a computer program that can handle self-gravitating hydrodynamic problems. We have used a particle-based approach, Smoothed Particle Hydrodynamics (SPH), because is it particularly suited to the collapse and fragmentation problems of star and planet formation.

We have dealt with various issues along the way, such as numerical stability, adaptive timesteps, gravitational softening and the Thomas & Couchman SPH kernel modification. We have optimised the code with a Barnes-Hut tree, a block timestepping scheme, sink particles and parallelisation. We have handled shocks, instabilities and shear flows, and performed extensive tests to validate the code.

10.1 Disc Fragmentation

Suitably armed with our numerical code, we have modelled the fragmentation of massive extended circumstellar discs via gravitational instability, including a radiative diffusion approximation for more realistic treatment of the thermal physics (Stamatellos et al. 2007b).

This allowed us to investigate the objects that might be formed in discs around young stars, which turned out to be brown dwarfs or massive planets, and to answer some of the questions posed in Section 1.6.

Under what circumstances can such objects form?

The disc must have a Toomre stability parameter of $Q \lesssim 1.3$ to fragment, and the fragment must be able to cool sufficiently quickly. This typically requires a massive extended disc. A disc that is stable against fragmentation may still be capable of ejecting bound condensations to produce free-floating planets.

When and where do they form?

In our discs, which have radial extent $100 < r < 1000$ AU and surface density profile $\Sigma \sim r^{-7/4}$, the objects formed on a timescale of $\sim 20,000 - 30,000$ years in a region $r \sim 200 - 500$ AU. The timescale therefore corresponds to a few rotation periods at the formation radius.

What factors determine their initial properties?

The objects all have minimum Jeans mass $\sim 3M_J$ (Jupiter masses) and eccentric orbits. They all form within spiral arms, and less stable discs might form objects closer to the central star.

What factors influence their subsequent orbital evolution?

If an object is on a close orbital path $r \sim 200-250$ AU then it might accrete to brown dwarf mass ($13M_J$) within an orbit. If an object forms later and/or drifts outwards it might only accrete to $\sim 5-7M_J$. Condensations which do not fragment also form in the spiral arms, and interactions often result in ejections of bound objects. This would explain the "brown dwarf desert" (the lack of brown dwarf companions around main sequence stars) and the frequent formation of free-floating planets (Sumi et al. 2011).

10.2 Disc Interactions

We have also modelled the effect of disc-star and disc-disc interactions on the evolution of a marginally stable disc ($Q = 1.5$). In a turbulent star-forming region, many protostars might experience such an interaction.

Disc-Star Interactions

1. Retrograde penetrating encounters can fragment a marginally stable disc.
2. Orthogonal penetrating encounters disrupt the disc and result in significant accretion enhancement.
3. For grazing and distant encounters, the prograde encounters result in the greatest accretion enhancement onto both stars, and truncate the disc to the greatest extent.
4. Non-coplanar encounters tilt the disc.

Prograde coplanar encounters might be the most common, and would result in strong accretion bursts.

Penetrating retrograde encounters might be the least likely, but are the ones which might trigger fragmentation in a marginally stable disc. They formed brown dwarf companions in the coplanar simulation and free-floating planets in the non-coplanar simulation.

Disc-Disc Interactions

1. SOP penetrating encounters produce dense condensations, but shock heating appears to inhibit fragmentation.
2. SOA penetrating encounters dissipate the disc and result in binary capture.

3. Grazing prograde coplanar encounters truncate the disc to the greatest extent.

4. Shearing encounters tilt and twist the disc.

SOP encounters, which might be more common, would result in disc truncation and negligible capture. Material piles into the tidal tails during penetrating encounters but is strongly shock heated.

Penetrating SOA encounters, which might be least likely, do not trigger fragmentation but instead dissipate the discs and result in binary capture.

Interactions do lead to greater instability, with more condensations forming, and on a shorter timescale, but for our $Q = 1.5$ disc they are transient or ejected and do not result in sink formation. Interaction does not appear to be a likely mechanism for triggering star or brown dwarf formation, but it could provide an increased number of free-floating planets.

10.3 Future Work

In the course of the investigation, several issues have arisen which suggest further work.

The lack of numerical convergence at high resolution (see Section 8.7.3) should be addressed, and replacing the M4 cubic spline kernel function with a quintic spline function (Morris 1996) provides one initial approach (Hubber, Falle & Goodwin 2011). It may ultimately involve finding an optimal trade-off between smoothing and resolution, in particular for simulations that model a density gradient, through convergence testing.

Another numerical issue to consider is noise in the initial conditions. We addressed this using the Halton sequence (based on the Van der Corput sequence in Table 3.2), but alternatives include the Sobol sequence (Press et al. 1992), "anti-clustering" placement methods (Cartwright 2006) or the direct transformation of a settled "glass" configuration to the required profile. Noise reduction should assist in achieving convergence.

An issue combining computational and astrophysical aspects concerns how greatly the evolution of the simulations would be affected by a more sophisticated treatment of viscosity, in terms of disc stability, fragmentation timescales and masses (see Section 8.7.2). This could involve adjusting the source term in the time-dependent artificial viscosity (see Section 6.3.1) or using a pattern-matching switch (Cartwright & Stamatellos 2010).

Greater computing time and processing power will permit longer simulations, allowing the long-term evolution of formed objects to be investigated, such as their accretion histories and final masses. Multiple realisations, using different seeds (or Halton bases) for the initial conditions, could build up a statistical picture of object properties and ejection probabilities.

Longer simulations of disc-disc interactions might even determine the outcome of coplanar penetrating encounters, such as whether the dense condensations in the tidal tails of the SOP encounter can eventually cool and fragment or not.

More simulations would allow the parameter space to be explored in finer detail in order to reveal any underlying trends. In particular, the parameter space for $Q > 1.5$ could be explored to seek the stability threshold for ejected bound condensations (free-floating planet formation).

Distinguishing the parameter space for free-floating planet formation from that of brown dwarf formation might be possible. Objects that form on close orbits are able to accrete to brown dwarf mass after one full circular orbit, but a more distant or eccentric orbit results in only planetary mass. More details of how effective the interactions are in enhancing planet formation could also be determined using simulations with $Q > 1.5$, by examining their effect upon the stability threshold.

Ejected free-floating planets could be removed from the simulations, allowing the long-term evolution of the disc itself to be investigated, to see if the formation process leaves any observational signatures that might be detected by ALMA, for example. Interaction simulations, which result in accretion bursts and enhanced free-floating planet formation, could also be investigated for any observational signatures of the interaction.

Future astronomical observations will allow us to refine the initial conditions of our simulations, and future simulations should highlight any observational signatures of underlying processes. In this way, observations and simulations allow us to converge towards the physical truth.

Bibliography

[1] Aarseth, S.J., 2003, Gravitational N-Body Simulations: Tools and Algorithms, Cambridge University Press, Cambridge, UK

[2] Agertz, O., Moore, B., Stadel, J., Potter, D., et al., 2007, MNRAS, 380, 963

[3] André, P., 1994, in Montmerle, Th., Lada, Ch.J., Mirabel, I.F., Trân Thanh Vân, J., eds, The Cold Universe, Editions Frontières, Gif-sur-Yvette Cedex, France, p. 179

[4] André, P., Ward-Thompson, D., Barsony, M., 1993, ApJ, 406, 122

[5] André, P., Ward-Thompson, D., & Barsony, M., 2000, in Mannings, V., Boss, A.P., Russell, S.S., eds, Protostars and Planets IV, University of Arizona Press, Tucson, AZ, p. 59

[6] Andrews, S.M., Williams, J.P., 2007, ApJ, 659, 705

[7] Attwood, R., 2008, PhD thesis, Cardiff University, ch. 3

[8] Balbus, S.A., Hawley, J.F., 1991, ApJ, 376, 214

[9] Ballesteros-Paredes, J., Klessen, R.S., Vásquez-Semadeni, E., 2003, ApJ, 592, 188

[10] Ballesteros-Paredes, J., Klessen, R.S., Mac Low, M.-M., Vásquez-Semadeni, E., 2006, in Reipurth, B., Jewitt, D., Keil, K., eds, Protostars and Planets V, University of Arizona Press, Tucson, AZ, p. 63

[11] Bally, J., Reipurth, B., Davis, C.J., 2006, in Reipurth, B., Jewitt, D., Keil, K., eds, Protostars and Planets V, University of Arizona Press, Tucson, AZ, p. 215

[12] Balsara, D.S., 1995, J. Comp. Phys., 121, 357

[13] Barnes, J., Hut, P., 1986, Nature, 324, 466

[14] Bastian, N., Gieles, M., Goodwin, S.P., Trancho, G., Smith, L.J., Konstantopoulos, I., Efremov, Yu., 2008, MNRAS, 389, 223

[15] Batchelor, G.K., 1967, An Introduction to Fluid Dynamics, Cambridge University Press, Cambridge, UK

[16] Bate, M.R., 2000, MNRAS, 314, 33

[17] Bate, M.R., Burkert, A., 1997, MNRAS, 288, 1060

[18] Bate, M.R., Bonnell, I.A., Bromm, V., 2002, MNRAS, 332, 65

[19] Bate, M.R., Bonnell, I.A., Price, N.M., 1995, MNRAS, 277, 362

[20] Beckwith, S.V.W., Sargent, A.I., Chini, R.S., Guesten, R., 1990, ApJ, 99, 924

[21] Benedict, G.F., McArthur, B.E., Forveille, T., Delfosse, X., et al., 2002, ApJ, 581, L115

[22] Binney, J., Tremaine, S., 1987, Galactic Dynamics, Princeton University Press, Princeton, NJ

[23] Bodenheimer, P., 1995, ARA&A, 33, 199

[24] Boffin, H.M.J., Watkins, S.J., Bhattal, A.S., Francis, N., Whitworth, A.P., 1998, MNRAS, 300, 1189

[25] Boley, A.C., Durisen, R.H., 2010, ApJ, 724, 618

[26] Boley, A.C., Mejía, A.C., Durisen, R.H., Cai, K., Pickett, M.K., D'Alessio, P., 2006, ApJ, 651, 517

[27] Boley, A.C., Hartquist, T.W., Durisen, R.H., Michael, S., 2007, ApJ, 656, L89

[28] Bond, I.A., Udalski, A., Jaroszyński, M., Rattenbury, N.J., et al., 2004, ApJ, 606, L155

[29] Bonnell, I.A., Bate, M.R., 1994, MNRAS, 271, 999

[30] Bonnell, I.A., Larson, R.B., Zinnecker, H., 2006, in Reipurth, B., Jewitt, D., Keil, K., eds, Protostars and Planets V, University of Arizona Press, Tucson, AZ, p. 149

[31] Borucki, W.J., Koch, D.G., Basri, G., Batalha, N., et al., 2011, ApJ, 736, 19

[32] Boss, A.P., 1998, ApJ, 503, 923

[33] Boss, A.P., 2000, ApJ, 536, L101

[34] Boss, A.P., 2004, ApJ, 610, 456

[35] Boss, A.P., 2006, ApJ, 637, L137

[36] Boss, A.P., 2007, ApJ 661, L73

[37] Boss, A.P., Bodenheimer, P., 1979, ApJ, 234, 289

[38] Bouvier, J., Duchêne, G., Mermilliod, J.C., Simon, T., 2001, A&A, 375, 989

[39] Boyd, D., 2003, PhD thesis, Cardiff University

[40] Boyd, D.F.A., Whitworth, A.P., 2005, A&A, 430, 1059

[41] Burkert, A., Bodenheimer, P., 2000, ApJ, 543, 822

[42] Cai, K., Durisen, R.H., Michael, S., Boley, A.C., Mejía, A.C., Pickett, M.K., D'Alessio, P., 2006, ApJ, 636, L149

[43] Cai, K., Pickett, M.K., Durisen, R.H., Milne, A.M., 2010, ApJ, 716, L176

[44] Calvet, N., Hartmann, L., Stromm, S.E., 2000, in Mannings, V., Boss, A.P., Russell, S.S., eds, Protostars and Planets IV, University of Arizona Press, Tucson, AZ, p. 377

[45] Carter, P.A., 2006, PC Assembly Language, `http://www.drpaulcarter.com/pcasm`, ch. 6

[46] Cartwright, A., 2006, PhD thesis, Cardiff University

[47] Cartwright, A., Stamatellos, D., 2010, A&A, 516, 99

[48] Cartwright, A., Stamatellos, D., Whitworth, A.P., 2009, MNRAS, 395, 2373

[49] Caselli, P., Walmsley, C.M., Terzieva, R., Herbst, E., 1998, ApJ, 499, 234

[50] Cha, S.-H., Whitworth, A.P., 2003, MNRAS, 340, 73

[51] Charbonneau, D., Brown, T.M., Latham, D.W., Mayor, M., 2000, ApJ, 529, L45

[52] Chenciner, A., Montgomery, R., 2000, Ann. Math., 152, 881

[53] Cieza, L.A., 2007, in Frebel, A., Maund, J.R., Shen, H., Siegel, M.H., eds, New Horizons in Astronomy: Frank N. Bash Symposium, ASP Conference Series 393, The Evolution of Primordial Circumstellar Disks, Astronomical Society of the Pacific, San Francisco, CA, p. 35

[54] Clarke, C.J., Harper-Clark, E., Meru, F., Lodato, G., 2008, in Fischer, D., Rasio, F.A., Thorsett, S.E., Wolszczan, A., eds, ASP Conference Series 398, Extreme Solar Systems, Astronomical Society of the Pacific, San Francisco, CA, p. 341

[55] Collela, P., Woodward, P.R., 1984, J. Comp. Phys., 54, 174

[56] Cossins, P., Lodato, G., Clarke, C.J., 2009, MNRAS, 393, 1157

[57] Courant, R., Friedrichs, K.O., Lewy, H., 1928, Math. Ann., 100, 32

[58] Deming, D., Seager, S., Richardson, L.J., Harrington, J., 2005, Nature, 434, 740

[59] Diggle, P.J., 1983, Statistical Analysis of Spatial Point Patterns, Academic Press, London, UK

[60] Dobbs, C.L., Bonnell, I.A., Pringle, J.E., 2006, MNRAS, 371, 1663

[61] Donati, J.-F., Bouvier, J., Walter, F.M., Gregory, S.G., et al., 2011, MNRAS, 412, 2454

[62] Durisen, R.H., 2001, in Zinnecker, H., Mathieu, R.D., eds, Proc. IAU Symp. 200, ASP Conference Series 200, The Formation of Binary Stars, Astronomical Society of the Pacific, San Francisco, CA, p. 381

[63] Durisen, R.H., Boss, A.P., Mayer, L., Nelson, A.F., Quinn, T., Rice, W.K.M., 2006, in Reipurth, B., Jewitt, D., Keil, K., eds, Protostars and Planets V, University of Arizona Press, Tucson, AZ, p. 607

[64] Dutrey, A., Guilloteau, S., 2004, Ap&SS, 292, 407

[65] Dutrey, A., Guilloteau, S., 2006, in Reipurth, B., Jewitt, D., Keil, K., eds, Protostars and Planets V, University of Arizona Press, Tucson, AZ, p. 495

[66] Eckel, B., 2000, Thinking in C++, Volume 1, Prentice Hall, Upper Saddle River, NJ, ch. 15

[67] Eisner, J.A., Carpenter, J.M., 2006, ApJ, 641, 1162

[68] Eisner, J.A., Plambeck, R.L., Carpenter, J.M., Corder, S.A., Qi, C., Wilner, D., 2008, ApJ, 683, 304

[69] Enoch, M.L., Corder, S.T., Duchêne, G., Bock, D.C., et al., 2011, ApJS, 195, 21

[70] Ferrière, K.M., 2001, Rev. Mod. Phys., 73, 1031

[71] Forgan, D., Rice, K., 2009a, MNRAS, 400, 2022

[72] Forgan, D., Rice, K., 2009b, MNRAS, 402, 1349

[73] Forgan, D., Rice, K., Stamatellos, D., Whitworth, A., 2009, MNRAS, 394, 882

[74] Forgan, D., Rice, K., Cossins, P., Lodato, G., 2011, MNRAS, 410, 994

[75] Fox, L., 1964, An Introduction to Numerical Linear Algebra, Clarendon Press, Oxford, UK

[76] Frank, J., King, A.R., Raine, D.J., 1992, Accretion Power in Astrophysics, Cambridge University Press, Cambridge, UK

[77] Fulk, D.A., Quinn, D.W., 1996, J. Comp. Phys., 126, 165

[78] Gammie, C.F., 2001, ApJ, 553, 174

[79] Gammie, C.F., Lin, Y., Stone, J.M., Ostriker, E.C., 2003, ApJ, 592, 203

[80] Gingold, R.A., Monaghan, J.J., 1977, MNRAS, 181, 375

[81] Gingold, R.A., Monaghan, J.J., 1983, MNRAS, 204, 715

[82] Goodwin, S.P., Bastian, N., 2006, MNRAS, 373, 752

[83] Goodwin, S.P., Ward-Thompson, D., Whitworth, A.P., 2003, in De Buizer, J.M., van der Bliek, N.S., eds, ASP Conference Series 287, Galactic Star Formation Across the Stellar Mass Spectrum, Astronomical Society of the Pacific, San Francisco, CA, p. 52

[84] Goodwin, S.P, Whitworth, A.P., Ward-Thompson, D., 2004, A&A, 414, 633

[85] Goodwin, S.P., Kroupa, P., Goodman, A., Burkert, A., 2006, in Reipurth, B., Jewitt, D., Keil, K., eds, Protostars and Planets V, University of Arizona Press, Tucson, AZ, p. 133

[86] Haisch, K.E., Lada, E.A., Lada, C.J., 2001, ApJ, 553, 153

[87] Halbwachs, J.L., Mayor, M., Udry, S., 2005, A&A, 431, 1129

[88] Hartigan, P., Edwards, S., Ghandour, L., 1995, ApJ, 452, 736

[89] Hayashi, C., 1966, Ann. Rev. A&A, 4, 171

[90] Henyey, L.G., Lelevier, R., Levée, R.D., PASP, 67, 154

[91] Hernquist, L., Bouchet, F.R., Suto, Y., 1991, ApJS, 75, 231

[92] Hernquist, L., Hut, P., Makino, J., 1993, ApJ 402, L85

[93] Hollenbach, D.J., Yorke, H.W., Johnstone, D., 2000, in Mannings, V., Boss, A.P., Russell, S.S., eds, Protostars and Planets IV, University of Arizona Press, Tucson, AZ, p. 401

[94] Holmberg, E., 1941, ApJ, 94, 385

[95] Hosking, J.G., 2002, PhD thesis, Cardiff University

[96] Hosking, J.G., Whitworth, A., 2004, MNRAS 347, 994

[97] Hubber, D.A., Goodwin, S.P., Whitworth, A.P., 2006, A&A, 450, 881

[98] Hubber, D.A., Falle, S.A.E.G., Goodwin, S.P., 2011, in Alves, J., Elmegreen, B.G., Girart, J.M., Trimble, V., eds, Proceedings of the International Astronomical Union, IAU Symposium 270, Computational Star Formation, Cambridge University Press, Cambridge, UK, p. 429

[99] Hubber, D.A., Batty, C.P., McLeod, A., Whitworth, A.P., 2011, A&A, 529, A27

[100] Hut, P., Makino, J., 2007, Moving Stars Around (The Maya Open Lab School Series, Volume 1), `http://www.ArtCompSci.org`

[101] Imaeda, Y., Inutsuka, S.-I., 2002, ApJ, 569, 501

[102] Intel, 2006, Quick-Reference Guide to Optimization with Intel Compilers, `http://www.intel.com`

[103] Inutsuka, S.-I., 1994, Mem. S. A. It., 65, 1027

[104] Jang-Condell, H., Sasselov, D.D., 2003, ApJ, 593, 1116

[105] Johnson, B.M., Gammie, C.F., 2003, ApJ, 597, 131

[106] Jørgensen, J.K., van Dishoeck, E.F., Visser, R., Bourke, T.L., et al., 2009, A&A, 507, 861

[107] Kirk, J., 2002, PhD thesis, Cardiff University

[108] Klein, R.I., Fisher, R., McKee, C.F., 2004, in García-Segura, G., Tenorio-Tagle, G., Franco, J., Yorke, H.W., eds, Revisita Mexicana de Astronoma y Astrofisica (Serie de Conferencias) 22, p. 3

[109] Königl, A., Pudritz, R.E., 2000, in Mannings, V., Boss, A.P., Russell, S.S., eds, Protostars and Planets IV, University of Arizona Press, Tucson, AZ, p. 759

[110] Kroupa, P., Petr, M.G., McCaughrean, M.J., 1999, New Astronomy, 4, 495

[111] Kuiper, G.P., 1949, ApJ, 109, 308

[112] Kumar, S.S., 1963, ApJ, 137, 1121

[113] Lada, C.J., 1999, in Lada, C.J., Kylafis, N., eds, The Origin of Stars and Planetary Systems, Kluwer Academic Publishers, Dordrecht, The Netherlands, p. 143

[114] Lada, C.J., Lada, E.A., 2003, ARA&A, 41, 57

[115] Larson, R.B., 1969, MNRAS, 145, 271

[116] Larson, R.B., 1990, in Capuzzo-Dolcetta, R., Chiosi, C., Di Fazio, A., eds, Physical Processes in Fragmentation and Star Formation, Kluwer Academic Publishers, Dordrecht, The Netherlands, p. 389

[117] Larson, R.B., 2001, in Zinnecker, H., Mathieu, R., eds, Proc. IAU Symp. 200, ASP Conference Series 200, The Formation of Binary Stars, Astronomical Society of the Pacific, San Francisco, CA, p. 93

[118] Laughlin, G., Bodenheimer, P., 1994, ApJ, 436, 335

[119] Lax, P.D., 1954, Comm. Pure Appl. Math., 7, 135

[120] Lecar, M., Podolak, M., Sasselov, D., Chiang, E., 2006, ApJ, 640, 1115

[121] Lin, D.N.C., Papaloizou, J., 1980, MNRAS, 191, 37

[122] Lin, D.N.C., Pringle, J.E., 1990, ApJ, 358, 515

[123] Lin, D.N.C., Bodenheimer, P., Richardson, D.C., 1996, Nature, 380, 606

[124] Lissauer, J.J., 1993, Ann. Rev. Astron. Astr., 31, 129

[125] Lodato, G., Clarke, C.J., 2004, MNRAS, 353, 841

[126] Lucas, P.W., Roche, P.F., 2000, MNRAS, 314, 858

[127] Lucy, L.B., 1977, AJ, 82, 1013

[128] Lynden-Bell, D., Pringle, J.E., 1974, MNRAS, 168, 603

[129] McCabe, C., Stapelfeldt, K., Pham, C., 2011, Catalog of Resolved Circumstellar Disks, `http://www.circumstellardisks.org`

[130] McCaughrean, M.J., Stapelfeldt, K.R., Close, L.M., 2000, in Mannings, V., Boss, A.P., Russell, S.S., eds, Protostars and Planets IV, University of Arizona Press, Tucson, AZ, p. 485

[131] McDonald, J.M., Clarke, C.J., 1995, MNRAS, 275, 671

[132] McKee, C.F., Ostriker, E.C., 2007, Ann. Rev. A&A, 45, 565

[133] McMillan, S.L.W., Aarseth, S.J., 1993, ApJ, 414, 200

[134] Mao, S., Paczynski, B., 1991, ApJ., 374, L37

[135] Marcy, G.W., Butler, R.P., 1994, in Tinney, C.G., ed., Proc. ESO Workshop, The Bottom of the Main Sequence - and Beyond, Springer-Verlag, Heidelberg, p. 98

[136] Marcy, G.W., Butler, R.P., 1995, BAAS (Bulletin of the American Astronomical Society), 27, 1379

[137] Martin, T.J., Pearce, F.R., Thomas, P.A., 1993, An Owner's Guide to Smoothed Particle Hydrodynamics, arXiv (astro-ph/9310024)

[138] Masunaga, H., Inutsuka, S., 2000, ApJ, 531, 350

[139] Masunaga, H., Miyama, S.M., Inutsuka, S., 1998, ApJ, 495, 346

[140] Masset, F.S., Papaloizou, J.C.B., 2003, ApJ, 588, 494

[141] Mathews, W.G., Bodenheimer, P., 1964, AJ, 69, 552

[142] Mathieu, R.D., 2007, in Bouvier, J., Appenzeller, I., eds, Proc. IAU Symp. 243, ASP Conference Series 200, Star-Disk Interaction in Young Stars, Cambridge University Press, Cambridge, UK, p. 315

[143] Mathieu, R.D., Ghez, A.M., Jensen, E.L.N., Simon, M., 2000, in Mannings, V., Boss, A.P., Russell, S.S., eds, Protostars and Planets IV, University of Arizona Press, Tucson, AZ, p. 703

[144] Mayer, L., Quinn, T., Wadsley, J., Stadel, J., 2004, ApJ, 609, 1045

[145] Mayer, L., Wadsley, J., Quinn, T., Stadel, J., 2005, MNRAS, 363, 2005

[146] Mayer, L., Lufkin, G., Quinn, T., Wadsley, J., 2007, ApJ, 661, L77

[147] Mayor, M., Queloz, D., 1995, Nature, 378, 355

[148] Mejía, A.C., Durisen, R.H., Pickett, M.K., Cai, K., 2005, ApJ, 619, 1098

[149] Meru, F., Bate, M.R., 2011, MNRAS, 411, L1

[150] Mestel, L., 1963, MNRAS, 126, 557

[151] Mestel, L., 1965, QJRAS, 6, 161-265

[152] Mestel, L., Spitzer, L.Jr., 1956, MNRAS, 116, 503

[153] Meyer, M.R., Adams, F.C., Hillenbrand, L.A., Carpenter, J.M., Larson, R.B., 2000, in Mannings, V., Boss, A.P., Russell, S.S., eds, Protostars and Planets IV, University of Arizona Press, Tucson, AZ, p. 121

[154] Mikkola, S., Aarseth, S. J., 1993, Celestial Mechanics and Dynamical Astronomy, 57, 439

[155] Moeckel, N., Bally, J., 2007, ApJ, 656, 275

[156] Moeckel, N., Raymond, S.N., Armitage, P.J., 2008, ApJ, 688, 1361

[157] Monaghan, J.J., 1997, J. Comp. Phys., 136, 298

[158] Monaghan, J.J., 2005, Rep. Prog. Phys., 68, 1703

[159] Monaghan, J.J., 2006, MNRAS, 365, 199

[160] Monaghan, J.J., Gingold, R.A., 1983, J. Comp. Phys., 52, 374

[161] Monaghan, J.J., Lattanzio, J.C., 1985, A&A, 149, 135

[162] Morris, J.P., 1996, PhD thesis, Monash University, ch. 2

[163] Morris, J.P., Monaghan, J.J., 1997, J. Comp. Phys., 136, 41

[164] Mouschovias, T.C., 1976, ApJ, 207, 141

[165] Myers, P.C., 1999, in Lada, C.J., Kylafis, N., eds, The Origin of Stars and Planetary Systems, Kluwer Academic Publishers, Dordrecht, The Netherlands, p. 67

[166] Natta, A., 1993, ApJ, 412, 761

[167] Nelson, A.F., 2006, MNRAS, 373, 1039

[168] Nelson, A.F., Benz, W., Ruzmaikina, T.V., 2000, ApJ, 529, 357

[169] Nelson, A.F., Wetzstein, M., Naab, T., 2009, ApJS, 184, 326

[170] Nelson, R.P., Gressel, O., 2010, MNRAS, 409, 639

[171] Nyland, L., Harris, M., Prins, J., 2007, in Nguyen, H., ed, GPU Gems 3, Addison-Wesley Professional, Boston, MA, p. 676

[172] O'Dell, C.R., 1995, in Lizano, S., Torrelles, J.M., eds, Revisita Mexicana de Astronoma y Astrofisica (Serie de Conferencias) 1, p. 11

[173] Olczak, C., Pfalzner, S., Eckart, A., 2010, A&A, 509, A63

[174] Oliveira, S., Stewart, D., 2006, Writing Scientific Software: A Guide to Good Style, Cambridge University Press, Cambridge, UK

[175] Padgett, D.L., Stromm, S.E., Ghez, A., 1997, ApJ, 477, 705

[176] Papaloizou, J.C.B., Terquem, C., 2006, Rep. Prog. Phys., 69, 119

[177] Parker, R.J., Goodwin, S.P., Kroupa, P., Kouwenhoven, M.B.N., 2009, MNRAS, 397, 1577

[178] Pfalzner, S., 2008, A&A, 492, 735

[179] Pfalzner, S., Gibbon, S., 1996, Many-Body Tree Methods in Physics, Cambridge University Press, Cambridge, UK

[180] Pfalzner, S., Olczak, C., 2007, A&A, 475, 875

[181] Pfalzner, S., Vogel, P., Scharwächter, J., Olczak, C., 2005, A&A, 437, 967

[182] Pickett, M.K., Durisen, R.H., 2007, ApJ, 654, L155

[183] Pickett, B.K., Cassen, P., Durisen, R.H., Link, R., 2000, ApJ, 529, 1034

[184] Pineda, J.L., Goldsmith, P.F., Chapman, N., Snell, R.L., et al., 2010, ApJ, 721, 686

[185] Pinsonneault, M.H., DePoy, D.L., Coffee, M., 2001, ApJ, 556, L59

[186] Pollack, J.B., Hubickyj, O., Bodenheimer, P., Lissauer, J.J., Podolak, M., Greenzweig, Y., 1996, Icarus, 124, 62

[187] Potter, D., 1973, Computational Physics, John Wiley & Sons, New York

[188] Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P., 1992, Numerical Recipes in Fortran, Cambridge University Press, Cambridge, UK

[189] Price, D.J., 2004, PhD thesis, University of Cambridge, ch. 3

[190] Price, D.J., 2008, J. Comp. Phys., 227, 10040

[191] Price, D.J., Monaghan, J.J., 2007, MNRAS, 374, 1347

[192] Pringle, J.E., 1981, Ann. Rev. Astron. Astrophys., 19, 137

[193] Rafikov, R.R., 2005, ApJ, 621, L69

[194] Rafikov, R.R., 2007, ApJ, 662, 642

[195] Read, J.I., Hayfield, T., Agertz, O., 2010, MNRAS, 405, 1513

[196] Reipurth, B., Clarke, C., 2001, ApJ, 122, 432

[197] Rice, W.K.M., Armitage, P.J., Bate, M.R., Bonnell, I.A., 2003, MNRAS, 339, 1025

[198] Rice, W.K.M., Lodato, G., Armitage, P.J., 2005, MNRAS, 364, L56

[199] Rice, W.K.M., Lodato, G., Pringle, J.E., Armitage, P.J., Bonnell, I.A., 2006, MNRAS, 372, L9

[200] Rice, W.K.M., Mayo, J.H., Armitage, P.J., 2010, MNRAS, 402, 1740

[201] Richard, S.R., Brook, C.B., Martel, H., Kawata, D., Gibson, B.K., Sanchez-Blazquez, P., 2010, MNRAS, 402, 1489

[202] Richer, J.S., Shepherd, D.S., Cabrit, S., Bachiller, R., Churchwell, E., 2000, in Mannings, V., Boss, A.P., Russell, S.S., eds, Protostars and Planets IV, University of Arizona Press, Tucson, AZ, p. 867

[203] Richtmyer, R.D., Morton, K.W., 1967, Difference Methods for Initial-Value Problems, Interscience Publishers, New York

[204] Sadakane, K., Ohkubo, M., Takeda, Y., Sato, B., Kambe, E., Aoki, W., 2002, Publ. Astron. Soc. Jpn, 54, 911

[205] Saitoh, T.R., Makino, J., 2009, ApJ, 697, 99

[206] Salpeter, E.E., 1955, ApJ, 121, 161

[207] Sandquist, E.L., Doktor, J.J., Lin, D.N.C., Mardling, R.A., 2002, ApJ, 572, 1012

[208] Santos, N.C., Garcia Lopez, R.J., Israelian, G., Mayor, M., Rebolo, R., Garcia-Gil, A., Perez de Taoro, M.R., Randich, S., 2002, A&A, 386, 1028

[209] Schaefer, C., Speith, R., Hipp, M., Kley, W., 2004, A&A, 418, 325

[210] Schneider, J., 2011, The Extrasolar Planets Encyclopaedia, `http://exoplanet.eu/catalog.php`

[211] Sedov, L.I., 1959, Similarity and Dimensional Methods in Mechanics, Academic Press, New York

[212] Shakura, N.I., Sunyaev, R.A., 1973, A&A, 24, 337

[213] Shen, S., Wadsley, J., 2006, ApJ, 651, L145

[214] Shen, S., Wadsley, J., Hayfield, T., Ellens, N., 2010, MNRAS, 401, 727

[215] Shu, F.H., Lizano, S., Adams, F.C., 1987, in Peimbert, M., Jugaku, J., eds, Star Forming Regions, IAU Symposium, D. Reidel Publishing Company, Dordrecht, The Netherlands, p. 417

[216] Shu, F.H, Najita, J.R., Shang, H., Li, Z.Y., 2000, in Mannings, V., Boss, A.P., Russell, S.S., eds, Protostars and Planets IV, University of Arizona Press, Tucson, AZ, p. 789

[217] Sod, G.A., 1978, J. Comp. Phys., 27, 1

[218] Springel, V., 2005, MNRAS, 364, 1105

[219] Springel, V., 2010, MNRAS, 401, 791

[220] Springel, V., Hernquist, L., 2002, MNRAS, 333, 649

[221] Springel, V., Yoshida, N., White, S.D.M., 2001, New Astronomy, 6, 79

[222] Stamatellos, D., 2003, PhD thesis, Cardiff University

[223] Stamatellos, D., Whitworth, A.P., 2008, A&A 480, 879

[224] Stamatellos, D., Whitworth, A.P., 2009a, in Stempels, E., ed., AIP Conference Proceedings 1094, Cool Stars, Stellar Systems and the Sun, American Institute of Physics, New York, p. 557

[225] Stamatellos, D., Whitworth, A.P., 2009b, MNRAS, 400, 1563

[226] Stamatellos, D., Whitworth, A., 2011, in Alves, J., Elmegreen, B.G., Girart, J.M., Trimble, V., eds, Proceedings of the International Astronomical Union, IAU Symposium 270, Computational Star Formation, Cambridge University Press, Cambridge, UK, p. 223

[227] Stamatellos, D., Hubber, D.A., Whitworth, A.P., 2007a, MNRAS, 382, L30

[228] Stamatellos, D., Whitworth, A.P., Bisbas, T., Goodwin, S., 2007b, A&A, 475, 37

[229] Stamatellos, D., Maury, A., Whitworth, A., Andre, P., 2010, MNRAS, 413, 1787

[230] Stamatellos, D., Whitworth, A.P., Hubber, D.A., 2011, ApJ, 730, 32

[231] Stapelfeldt, K.R., Krist, J.E., Menard, F., Bouvier, J., Padgett, D.L., Burrows, C.J., 1998, ApJ, 502, 65

[232] Stiefel, E.L., Scheifele, G., 1971, Linear and Regular Celestial Mechanics, Springer-Verlag, Berlin, Germany

[233] Stone, J.M., Gammie, C.F., Balbus, S.A., Hawley, J.F., 2000, in Mannings, V., Boss, A.P., Russell, S.S., eds, Protostars and Planets IV, University of Arizona Press, Tucson, AZ, p. 589

[234] Sumi, T., Kamiya, K., Bennett, D.P., Bond, I.A., et al., 2011, Nature, 473, 349

[235] Taylor, G.I., Green, A.E., 1937, Proc. R. Soc. Lond. A, 158, 499

[236] Thies, I., Kroupa, P., Goodwin, S.P., Stamatellos, D., Whitworth, A.P., 2010, ApJ, 717, 577

[237] Thomas, P.A., Couchman, H.M.P., 1992, MNRAS, 257, 11

[238] Toomre, A., 1964, ApJ, 139, 1217

[239] Toro, E.F., 1989, Intl. J. for Num. Methods in Fluid, 9

[240] Truelove, J.K., Klein, R.I., McKee, C.F., Holliman, J.H., II, Howell, L.H., Greenough, J.A., 1997, ApJ, 489, L197

[241] Truelove, J.K., Klein, R.I., McKee, C.F., Holliman II, J.H., Howell, L.H., Greenough, J.A., Tod Woods, D., 1998, ApJ, 495, 821

[242] von Neumann, J., Richtmyer, R.D., 1950, J. Appl. Phys., 21, 232

[243] Wadsley, J.W., Veeravalli, G., Couchman, H.M.P., 2008, MNRAS, 387, 427

[244] Ward-Thompson, D., Whitworth, A.P., 2011, An Introduction to Star Formation, Cambridge University Press, Cambridge, UK

[245] Ward-Thompson, D., Scott, P.F., Hills, R.E., Andre, P., 1994, MNRAS, 268, 276

[246] Ward-Thompson, D., Nutter, D., Bontemps, S., Whitworth, A., Attwood, R., 2006, MNRAS, 369, 1201

[247] Ward-Thompson, D., André, P., Crutcher, R., Johnstone, D., Onishi, T., Wilson, C., 2006, in Reipurth, B., Jewitt, D., Keil, K., eds, Protostars and Planets V, University of Arizona Press, Tucson, AZ, p. 33

[248] Waters, L.B.F.M., Waelkens, C., 1998, Ann. Rev. A&A, 36, 233

[249] Watkins, S.J., 1996, PhD thesis, Cardiff University

[250] Watkins, S.J., Bhattal, A.S., Boffin, H.M.J., Francis, N., Whitworth, A.P., 1998a, MNRAS, 300, 1205

[251] Watkins, S.J., Bhattal, A.S., Boffin, H.M.J., Francis, N., Whitworth, A.P., 1998b, MNRAS, 300, 1214

[252] Watson, A.M., Stapelfeldt, K.R., Wood, K., Ménard, F., 2006, in Reipurth, B., Jewitt, D., Keil, K., eds, Protostars and Planets V, University of Arizona Press, Tucson, AZ, p. 523

[253] Weidenschilling, S.J., 1977, Astrophys. & Space Sci., 51, 153

[254] White, R.J., Greene, T.P., Doppmann, G.W., Covey, K.R., Hillenbrand, L.A., 2006, in Reipurth, B., Jewitt, D., Keil, K., eds, Protostars and Planets V, University of Arizona Press, Tucson, AZ, p. 117

[255] Whitworth, A.P., Ward-Thompson, D., 2001, ApJ, 547, 317

[256] Whitworth, A.P., Zinnecker, H., 2004, A&A, 427, 299

[257] Whitworth, A.P., Bhattal, A.S., Turner, J.A., Watkins, S.J., 1995, A&A, 301, 929

[258] Whitworth, A., Bate, M.R., Nordlund, A., Reipurth, B., Zinnecker, H., 2006, in Reipurth, B., Jewitt, D., Keil, K., eds, Protostars and Planets V, University of Arizona Press, Tucson, AZ, p. 459

[259] Williams, J.P., Blitz, L., McKee, C.F., 2000, in Mannings, V., Boss, A.P., Russell, S.S., eds, Protostars and Planets IV, University of Arizona Press, Tucson, AZ, p. 97

[260] Wolszczan, A., Frail, D.A., 1992, Nature, 355, 145

[261] Zapatero Osirio, M.R, Béjar, V.J.S., Martín, E.L., Rebolo, R., Barrado y Navascués, D., Bailer-Jones, C.A.L., Mundt, R., 2000, Science, 290, 103

[262] Zinnecker, H., Yorke, H.W., 2007, Ann. Rev. A&A, 45, 481

[263] Zucker, S., Mazeh, T., 2002, ApJ, 568, L113

[264] Zuckerman, B., 2001, ARA&A, 39, 549