

This is an Open Access document downloaded from ORCA, Cardiff University's institutional repository: <https://orca.cardiff.ac.uk/id/eprint/161948/>

This is the author's version of a work that was submitted to / accepted for publication.

Citation for final published version:

Yu, Ruizhi, Gu, Wei, Lu, Hai, Yao, Shuai, Zhang, Suhan, Lu, Shuai, Ding, Shixing and Luo, Enbo 2023. Non-iterative calculation of quasi-dynamic energy flow in the heat and electricity integrated energy systems. IEEE Transactions on Power Systems 38 (5), pp. 4148-4164. 10.1109/TPWRS.2022.3210167

Publishers page: <https://doi.org/10.1109/TPWRS.2022.3210167>

Please note:

Changes made as a result of publishing processes such as copy-editing, formatting and page numbers may not be reflected in this version. For the definitive version of this publication, please refer to the published source. You are advised to consult the publisher's version if you wish to cite this paper.

This version is being made available in accordance with publisher policies. See <http://orca.cf.ac.uk/policies.html> for usage policies. Copyright and moral rights for publications made available in ORCA are retained by the copyright holders.



Non-iterative Calculation of Quasi-Dynamic Energy Flow in the Heat and Electricity Integrated Energy Systems

Ruizhi Yu, *Student Member, IEEE*, Wei Gu, *Senior Member, IEEE*, Hai Lu, Shuai Yao, *Student Member, IEEE*, Suhan Zhang, *Student Member, IEEE*, Shuai Lu, *Member, IEEE*, Shixing Ding, *Student Member, IEEE*, Enbo Luo

Abstract—Quasi-dynamic energy flow calculation is an indispensable tool for the heat and electricity integrated energy system (HE-IES) analysis. One solves the nonlinear partial differential algebraic equations to obtain thermal, hydraulic and electric variations. However, mainstream iteration solvers face the challenges of inefficiency and bad robustness. For one thing, the frequent update and factorization of Jacobian matrices utilize high CPU time. For another, the per-step iteration numbers grow exponentially as the system loading level creeps up. This paper presents a novel non-iterative algorithm for the quasi-dynamic energy flow calculation. The kernel of the proposed algorithm is to transform these nonlinear equations into linear recursive ones, by solving which, we obtain explicit closed-form solutions of unknown variables. In each step, the proposed algorithm requires only one matrix factorization and fixed times of arithmetic operations regardless of the loading levels, so that it achieves small and consistent per-step time costs. A semi-discrete scheme is used in PDE solution to avoid dissipative and dispersive errors that are often overlooked in previous literature. To ensure convergence, we also propose to control the temporal step sizes adaptively by estimating the simulation errors. Case studies showed that the proposed method manifested efficient and robust time performance compared with the iterative algorithms, and meanwhile preserved high accuracy.

Index Terms—Differential Transformation, Integrated Energy Systems, Partial Differential Equations, Quasi-Dynamic Energy Flow, Semi-Analytical Solution

NOMENCLATURE

A. Abbreviation

DHS	District heating system.
DT	Differential transformation.
EPS	Electric power system.
FDM	Finite difference method.
HE-IES	Heat and electricity integrated energy system.
IU	Implicit upwind scheme.
NM	Node method.
ODE	Ordinary differential equation.

This work was supported by the National Key Research and Development Program of China (Grant No. 2020YFE0200400), and in part by the Natural Science Foundation of Jiangsu Province (Grant No. BK20200013).

R.Z. Yu, W. Gu, S. Yao, S.H. Zhang and S. Lu are with the School of Electrical Engineering, Southeast University, Nanjing, Jiangsu, 210096 China. *Corresponding author: Wei Gu*, e-mail: wgu@seu.edu.cn.

H. Lu, E.B. Luo are with the Yunnan Power Grid Co., Ltd, Kunming, Yunnan 650217, China.

S.X. Ding is with the School of Cyber Science and Engineering, Southeast University, Nanjing 210096, China.

PDE	Partial differential equation.
SAS	Semi-analytical solution.
SOE	Second-order explicit scheme.
TVD	Total variation decreasing.

B. Operators

diag(\cdot)	Transform a column vector into a diagonal matrix.
\otimes	Convolution.
$X(0 : k)$	Linear combination of $X(0), X(1), \dots, X(k)$.

C. Variables

The capitalized/bold forms of these variables denote the DT/vector of these variables respectively.

$\dot{m}/\dot{m}^{\text{in}}$	Pipe/node injection mass flow rate.
$\tau^{\text{s/r}}$	Node supply/return temperature.
$\tau^{\text{in/out,s/r}}$	Inlet/outlet temperature of supply/return pipes.
τ^{amb}	Ambient Temperature.
ϕ	Node power.
e/f	Real/imaginary part of bus voltage.
p/q	Active/reactive power.

D. Coefficients

γ	Cross-sectional area.
ρ	Density.
H	Length of pipe.
C_p	Thermal capacity of mass flow.
λ	Overall heat transfer coefficient.
K	Resistance coefficient.
Z	Trade-off between heat supply and electric power.
c_{m1}	Heat-to-power ratio.
η_e	Electrical efficiency.
F_{in}	Fuel input rate of steam turbine.
$\Delta x/\Delta t$	Spatial/temporal step sizes.
θ	Parameter that balances the dissipative/dispersive errors in PDE solutions.
V/L	Node/loop incidence matrix.
G/B	Conductance/susceptance matrix.

E. Superscripts/Subscripts

D	Diagonal matrix generated by diag(\cdot).
PQ/PV/R	Row indices of vector/matrix related to PQ/PV/Slack buses in EPSs.

R/S/L/I	Row indices of vector/matrix related to slack/source/load/intermediate nodes in DHSs.
SP	Specified value.

I. INTRODUCTION

IN the recent decade, the applications of heat and electricity integrated energy systems (HE-IESs) have been prevailing. For one thing, the integration of district heating systems (DHSs) and electric power systems (EPSs) provides more flexibility and helps improve the economy [1]–[3]; for another, state changes and outages are transferred from one system to the other, threatening the overall operation security [4]–[6]. In view of this, quasi-dynamic energy flow calculation, which simulates the real operation scenarios and foresees the upcoming contingencies, becomes indispensable.

In HE-IESs, the aim of quasi-dynamic energy flow calculation is to obtain: 1) in DHS side, the variations of temperature distribution, heat power and mass flow, and 2) in EPS side, the variations of voltage and electric power. This is realized by solving the partial differential algebraic equations which are comprised of the EPS models and the DHS models [7], [8]. Nonlinear power flow equations constitute the EPS models, whereas the complexity of DHS models depends on the regulation modes. In quality regulation mode, the operators fix the mass flow rates and regulate the DHS by adjusting source supply temperatures, which means the hydraulics are predetermined and remain unchanged. Hence, the thermal and energy algebraic equations, and the partial differential equations (PDEs) governing thermal dynamics in pipes are linear. These models are relatively easy to solve and their analytical solutions have been derived in [9]–[12]. In quantity regulation mode, operators fix the source supply temperatures and regulate the DHS by adjust mass flow rates. As a result, couplings between temperature and mass flow variables bring nonlinearity to DHS models.

Nonlinearity is intractable. Traditional iteration solvers cannot guarantee to find exact solutions of these nonlinear equations within prescribed number of steps. Also, in each of the iteration steps, highly dimensional Jacobian matrices are updated and factorized, which is computationally intensive. On the other hand, the convergence of iteration solvers on this complex systems have not been theoretically secured, so that divergence is a common issue. In view of this, this paper focuses on the nonlinear DHS models and aims to develop a new efficient and robust solver for the quasi-dynamic energy flow calculation in HE-IESs.

Typically, PDE in the HE-IES models are first converted into algebraic equations by finite difference methods (FDMs) with different accuracy and stability properties, which include: 1) implicit upwind (IU) scheme [13]; 2) the modified characteristic line methods [13], [14]; 3) Second-order Explicit(SOE) scheme [15]. However, little attention is paid to the decayings of high-frequency components, which are called dissipative errors [16], and the fake oscillations, which are called dispersive errors [16], in solutions by these methods. It is later shown in case studies that the widely used IU scheme has distinct dissipative errors and the SOE scheme has distinct

dispersive errors, which undoubtedly weaken the credibility of the quasi-dynamic energy flow calculation. Choosing proper temporal and spatial step sizes may help reduce these errors, but, as shown in Appendix A, this becomes in vain when mass flow velocities are variable. To reduce the dissipative and dispersive errors regardless of the mass flow velocities and spatial/temporal step sizes, schemes with total variation decreasing (TVD) property have been developed in [16], [17] and work well.

After applying FDMs to the PDEs, we should solve the consequent nonlinear algebraic equations which are composed of the discretized PDEs, the nonlinear thermal/hydraulic/energy equations and the nonlinear power flow equations. Reference [7] has proposed the HE-FBI method to alternatively solve these equations until the convergence criteria are satisfied. However, the alternating solution strategy passes errors between different iteration loops. For example, the values of EPS variables used in DHS model calculation may have errors bigger than convergence criteria and would not be updated in the current iteration loop. These errors can be called alternating errors and increase the risk of divergence. In view of this, reference [18] and [8] have tried to improve the computation efficiency and convergence of the DHS side by uncoupling the DHSs into small simple sub-systems. Reference [19] has applied holomorphic embedding method to eliminate the iterations of the EPS side. However, iterations, which account for most of the computation overhead in quasi-dynamic energy flow calculation, have not been totally eliminated yet. Also, these methods are still prone to divergence because of alternating errors and improper initial guesses.

Recently, differential transformation (DT) has been proposed to solve nonlinear power system models. This method derives time-polynomial solutions, which are called semi-analytical solutions (SASs), of electro-mechanical models [20], [21] and the modified continuation power flow models [22]. This method demonstrates fast and reliable performance compared with traditional numerical-integration methods for the following two reasons: 1) it gets rid of iterations completely; 2) it constructs high-order approximations conveniently. These researches shed light on a new non-iterative solver of the nonlinear partial differential algebraic models of HE-IESs. However, DTs developed in [20]–[22] can only cope with ordinary differential algebraic equations. Therefore, there is a need for converting the PDEs into ordinary differential equations (ODEs).

In this paper, we present a non-iterative algorithm for quasi-dynamic energy flow based on DT. The proposed method is able to derive SASs of all the unknown variables in the nonlinear models, but requires only fixed computation resources in each time window. Furthermore, the proposed method solves EPS and DHS models together and hence alternating errors are avoided. The kernel of the proposed method is to discretize the spatial derivative of the PDE only, which converts the PDE into ODEs. Then we can apply DT to the resulting ordinary differential algebraic equations of the spatially discretized HE-IES models. In this paper, we use a scheme with TVD property to realize the conversion, which reduces dissipative and dispersive errors effectively. Compared with [23] which

has used the same TVD scheme, we achieve higher temporal accuracy because we use DT as the ODE solver instead of simple forward difference of temporal derivative. Additionally, to ensure calculation robustness and efficiency, we develop an easy-implemented adaptive time window control strategy based on the recursive nature of DT.

The major contributions are summarized as follows.

1) A DT-based non-iterative algorithm is proposed to improve the efficiency and robustness of quasi-dynamic energy flow calculation in HE-IESs. With the aid of a semi-discrete TVD scheme, the dissipative and dispersive errors in thermal dynamics are effectively reduced.

2) An adaptive time window control strategy is designed to further accelerate the simulation routines and to avoid non-convergence by selecting appropriate temporal step sizes.

This paper is outlined in the following way: Section II introduces DT; Section III presents the DT-based PDE solver; Section IV derives the DTs of nonlinear algebraic equations; Section V illustrates the generalized DT-based SAS-derivation framework on a small system, introduces the adaptive time window control strategy and gives the overall pseudocode of the proposed method; Section VI gives the case studies; Section VII concludes.

II. INTRODUCE DIFFERENTIAL TRANSFORMATION

DT derives SASs of state variables of ODEs by obtaining the coefficients of their Taylor series. Following [20]–[22], these coefficients are called DT coefficients and are defined as (1). The corresponding SAS can be written as (2). The $K + 1$ -term SAS (2) is called DT- K for short in this paper.

$$\mathbf{X}(k) = \frac{1}{k!} \left[\frac{d^k \mathbf{x}(t)}{dt^k} \right]_{t=0} \quad (1)$$

$$\mathbf{x}(t) = \sum_{k=0}^K \mathbf{X}(k) \cdot t^k \quad (2)$$

To obtain $\mathbf{X}(0)$ to $\mathbf{X}(K)$, DT transforms the original nonlinear ODEs about $\mathbf{x}(t)$ into linear recursive equations about $\mathbf{X}(k)$ with the following rules, where we denote by $\mathbf{Y}(k)$ the DT coefficient of $\mathbf{y}(t)$; $c \in \mathbb{R}$; $\mathbf{x}(t), \mathbf{y}(t), \mathbf{X}(k), \mathbf{Y}(k), \delta(k) \in \mathbb{R}^{n \times 1}$; $\mathbf{X}_D(k) \in \mathbb{R}^{n \times n}$; $\text{diag}(\cdot)$ is an operator which transforms a column vector into a diagonal matrix.

- 1) $\mathbf{x}(0) \rightarrow \mathbf{X}(0)$.
- 2) $c\mathbf{x}(t) \rightarrow c\mathbf{X}(k)$.
- 3) $\mathbf{x}(t) \pm \mathbf{y}(t) \rightarrow \mathbf{X}(k) \pm \mathbf{Y}(k)$.
- 4) $\text{diag}(\mathbf{x}(t))\mathbf{y}(t) \rightarrow$

$$\mathbf{X}_D(k) \otimes \mathbf{Y}(k) = \sum_{m=0}^k \mathbf{X}_D(m) \mathbf{Y}(k-m).$$

- 5) $c \rightarrow c\delta(k)$, where

$$\delta(k) = \begin{cases} \mathbf{1}, & k = 0 \\ \mathbf{0}, & k \neq 0 \end{cases}.$$

- 6) $d\mathbf{x}(t)/dt \rightarrow (k+1)\mathbf{X}(k+1)$.

The readers can refer to [20]–[22], [24, §1.8] for proofs and rules for more complex functions. The next section can be viewed as a first tutorial of DT.

III. DISCRETIZE AND DERIVE DT OF PDE

The PDE governing thermal dynamics is essentially a one-dimensional hyperbolic convection equation with source term [7], which is

$$\frac{\partial \tau}{\partial t} + \frac{\dot{m}}{\gamma \rho} \frac{\partial \tau}{\partial x} + \frac{\lambda}{\gamma \rho C_p} (\tau - \tau^{\text{amb}}) = 0, \quad (3)$$

where τ is the two-dimensional distribution of temperature along time t and position x ; \dot{m} is the mass flow rate; γ is the cross-sectional area of the pipe; ρ is the density of mass flow; λ is the overall heat transfer coefficient; C_p is the thermal capacity of mass flow; τ^{amb} is the ambient temperature, with initial condition

$$\tau(x, 0) = \varphi(x), \quad x \geq 0, \quad (4)$$

and boundary condition

$$\tau(0, t) = \psi(t), \quad t \geq 0. \quad (5)$$

Semi-discrete difference scheme is widely applied to solving the hyperbolic and parabolic PDEs [17], [25]. By replacing the spatial derivative $\partial \tau / \partial x$ by its discrete approximation, the PDEs are converted into ODEs, which can be solved by ODE solvers such as Runge-Kutta and Euler methods. In the simplest case, we can replace $\partial \tau / \partial x$ by the backward difference quotient $(\tau_j - \tau_{j-1}) / \Delta x$ where Δx is the spatial step size. If we assume that total length of the pipe is H , then the PDE becomes $M - 1$ ODEs about temperatures at each discrete node as shown in Fig. 1 where $M = H / \Delta x + 1$. τ_1 corresponds to the boundary conditions and is viewed as a known variable.

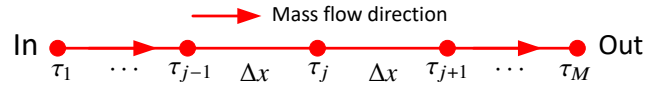


Fig. 1. Semi-discrete difference of pipe.

However, semi-discrete difference scheme based on backward difference quotient suffers dissipative and dispersive errors. In this paper, we discretize the spatial derivative of (3) with the semi-discrete TVD scheme proposed in [17], which derives

$$\frac{d\tau_j}{dt} = \frac{\dot{m}}{\gamma \rho \Delta x} \left(\tau_{j-1} + \frac{\Delta x}{2} (\tau_{j-1})_x - \tau_j - \frac{\Delta x}{2} (\tau_j)_x \right) - \frac{\lambda}{\gamma \rho C_p} (\tau_j - \tau^{\text{amb}}), \quad 2 \leq j \leq M \quad (6)$$

where $(\tau_j)_x$ denotes the spatial derivative of τ_j and its formula is not distinct but dynamically decided by minmod slope limiter

$$\text{minmod}(\chi_1, \chi_2, \chi_3) = \begin{cases} \min(\chi_1, \chi_2, \chi_3) & \text{if } \forall \chi_i > 0, \\ \max(\chi_1, \chi_2, \chi_3) & \text{if } \forall \chi_i < 0, \\ 0 & \text{otherwise.} \end{cases} \quad (7)$$

where

$$\begin{aligned}\chi_1 &= \theta \frac{\tau_j(0) - \tau_{j-1}(0)}{\Delta x}, \\ \chi_2 &= \frac{\tau_{j+1}(0) - \tau_{j-1}(0)}{2\Delta x}, \\ \chi_3 &= \theta \frac{\tau_{j+1}(0) - \tau_j(0)}{\Delta x}.\end{aligned}$$

$\theta \in [1, 2]$ balances the dissipative and dispersive errors in the solutions. Choices of θ are later studied in Section VI. $\tau_j(0)$ denotes the initial value of τ_j in the current time window. We put

$$(\tau_j)_x = \begin{cases} \theta \frac{\tau_j - \tau_{j-1}}{\Delta x} & \text{if } \min\text{mod}(\chi_1, \chi_2, \chi_3) = \chi_1, \\ \frac{\tau_{j+1} - \tau_{j-1}}{2\Delta x} & \text{if } \min\text{mod}(\chi_1, \chi_2, \chi_3) = \chi_2, \\ \theta \frac{\tau_{j+1} - \tau_j}{\Delta x} & \text{if } \min\text{mod}(\chi_1, \chi_2, \chi_3) = \chi_3, \\ 0 & \text{otherwise.} \end{cases}$$

In each time window, we should first give a distinct formula to each $(\tau_j)_x$ according to the output of minmod slope limiter. Then (6) becomes ODE about τ_{j-2} , τ_{j-1} , τ_j and τ_{j+1} . For the $j = 2$ and $j = M$ cases where τ_0 and τ_{M+1} do not exist, we assumes $(\tau_j)_x$ and $(\tau_{j-1})_x$ to be zero in (6).

The scheme (6) will be of second-order spatial accuracy only in the cases where $(\tau_j)_x = (\tau_{j+1} - \tau_{j-1})/(2\Delta x)$ and $(\tau_{j-1})_x = (\tau_j - \tau_{j-2})/(2\Delta x)$ and these cases usually occur on smooth sections of the solutions. Otherwise, it will be spatially first-order accurate and these cases usually occur in discontinuous sections of solutions.

Here, we use DT to solve these ODEs and exemplify the derivation of DT of (6) by assuming $j = 2$ or M . The cases where $2 < j < M$ can be derived in the same way.

In quantity regulation mode, \dot{m} is also variable. Applying transformation rule 6) to left hand side and 2)3)4)5) to right hand side of

$$\frac{d\tau_j}{dt} = \frac{\dot{m}}{\gamma\rho\Delta x}(\tau_{j-1} - \tau_j) - \frac{\lambda}{\gamma\rho C_p}(\tau_j - \tau^{\text{amb}})$$

derives

$$\begin{aligned}(k+1)T_j(k+1) &= \frac{1}{\gamma\rho\Delta x} \dot{M}(k) \otimes (T_{j-1}(k) - T_j(k)) \\ &\quad - \frac{\lambda}{\gamma\rho C_p} (T_j(k) - \tau^{\text{amb}}\delta(k))\end{aligned}\quad (8)$$

where $T_j(k)$ is the DT of τ_j ; $\dot{M}(k)$ is the DT of \dot{m} .

After the semi-discrete difference of PDE (3), the initial condition becomes

$$\tau_j(0) = \varphi(x_j), \quad 2 \leq j \leq M, \quad (9)$$

where x_j is the position of discrete node j , and the boundary condition becomes

$$\tau_1(t) = \psi(t). \quad (10)$$

According to transformation rule 1), we obtain

$$T_j(0) = \varphi(x_j), \quad 1 \leq j \leq M.$$

As for boundary condition $\psi(t)$ with distinct formula such as polynomials, trigonometric functions, etc., we should first

obtain the recursive expression of the DT of $\psi(t)$, which is denoted by $\Psi(k)$. Then derive DT of (10), we obtain

$$T_1(k) = \Psi(k), \quad k \geq 0.$$

It is obvious that right hand side of (8) only contains DT coefficients up to order k . If we are given the variation of \dot{m} , we can derive $\dot{M}(0)$ to $\dot{M}(K)$ first and then obtain, in the following order,

- 1) $T_j(1)$ by $\dot{M}(0)$ and $T_i(0)$;
- 2) $T_j(2)$ by $\dot{M}(0)$ to $\dot{M}(1)$ and $T_i(0)$ to $T_i(1)$;
- 3) $T_j(3)$ by $\dot{M}(0)$ to $\dot{M}(2)$ and $T_i(0)$ to $T_i(2)$;
- 4) ...
- 5) $T_j(K)$ by $\dot{M}(0)$ to $\dot{M}(K-1)$ and $T_i(0)$ to $T_i(K-1)$;

where $2 \leq j \leq M$, $1 \leq i \leq M$ and $T_1(0)$ to $T_1(K)$ are given by the boundary condition.

After obtaining $T_j(0)$ to $T_j(K)$, we can obtain the time polynomial depicting temperature variation as

$$\tau_j(t) = \sum_{k=0}^K T_j(k) \cdot t^k.$$

In this section, we apply DT to the PDE of a single pipe whose initial-boundary conditions and mass flow variation \dot{m} are known. However, in HE-IESs, boundary conditions of pipes and \dot{m} are not explicitly given. In following sections, we will derive the DT of HE-IES models first and illustrate how to deal with this case in Section V-A.

IV. DERIVE DTs OF NONLINEAR ALGEBRAIC EQUATIONS

A. DTs of DHS Models

This paper considers the quantity regulation mode of DHSs and hence follows the DHS models and the node type assumptions adopted by [7], [18], [26]. The known and unknown variables of DHS nodes are shown in Table I where $\phi/\tau^s/\tau^r/\dot{m}^{\text{in}}$ are respectively heat power/node supply temperature/node return temperature/node injection mass flow rate.

TABLE I
NODE TYPE ASSUMPTION IN DHSs

Node Type	Slack (R)	Source (S)	Load (L)	Intermediate (I)
Known	τ^s	ϕ, τ^s	ϕ, τ^r	$\phi = \dot{m}^{\text{in}} = 0$
Unknown	$\phi, \tau^r, \dot{m}^{\text{in}}$	$\tau^r, \dot{m}^{\text{in}}$	$\tau^s, \dot{m}^{\text{in}}$	τ^s, τ^r

Below, we use subscript R, S, L and I to denote the row indices of matrices or vectors related to slack, source, load and intermediate nodes, respectively. The combination of these subscripts denotes the combination of these nodes. For example, subscript R,S denotes the combination of slack and source nodes.

1) *Mass Flow Continuity Equations*: node injection mass flow rate is equal to the mass flow rate that enters into the node minus the mass flow rate leaves from the node, which yields

$$V_{R,S} \dot{m} = -\dot{m}_{R,S}^{\text{in}}, \quad (11)$$

$$V_L \dot{m} = \dot{m}_L^{\text{in}}, \quad (12)$$

$$V_I \dot{m} = 0, \quad (13)$$

where V is the node incidence matrix and its element

$$V_{ij} = \begin{cases} 1, & \text{if pipe } j \text{ flows into node } i \\ -1, & \text{if pipe } j \text{ flows out of node } i \\ 0, & \text{otherwise} \end{cases}$$

V describes the supply networks. In this paper, we consider the case where the mass flow in supply and return networks are symmetric, so we only need to study the hydraulic equations of supply networks here. $\dot{\mathbf{m}}$ is the vector of mass flow rate; $\dot{\mathbf{m}}^{\text{in}}$ is the vector of node injection mass flow rate. There is a minus sign before $\dot{\mathbf{m}}_{\text{R,S}}^{\text{in}}$ to ensure that node injection mass flow rates of slack and source nodes are negative.

Applying transformation rule 2) to both sides of (11), (12) and (13) derives

$$V_{\text{R,S}}\dot{\mathbf{M}}(k) = -\dot{\mathbf{M}}_{\text{R,S}}^{\text{in}}(k), \quad (14)$$

$$V_{\text{L}}\dot{\mathbf{M}}(k) = \dot{\mathbf{M}}_{\text{L}}^{\text{in}}(k), \quad (15)$$

$$V_{\text{I}}\dot{\mathbf{M}}(k) = 0, \quad (16)$$

where $\dot{\mathbf{M}}(k)$ and $\dot{\mathbf{M}}^{\text{in}}(k)$ are DTs of $\dot{\mathbf{m}}$ and $\dot{\mathbf{m}}^{\text{in}}$. (14), (15) and (16) are linear equations for $\dot{\mathbf{M}}(k)$ and $\dot{\mathbf{M}}^{\text{in}}(k)$ because the coefficients of them are elements of node incidence matrix, which are constant.

2) *Loop Pressure Equations*: The loop pressure equation states that the sum of head losses around a closed loop must be equal to zero, which is

$$L\text{diag}(K)\text{diag}(\dot{\mathbf{m}})\dot{\mathbf{m}} = \mathbf{0}, \quad (17)$$

where L is the loop incidence matrix and its element

$$L_{ij} = \begin{cases} 1, & \text{if pipe } j \text{ in loop } i \text{ is clockwise} \\ -1, & \text{if pipe } j \text{ in loop } i \text{ is counterclockwise;} \\ 0, & \text{if pipe } j \text{ is not in loop } i \end{cases}$$

K is the vector of pipe resistance coefficient. The same as V , L also describes the supply networks.

Applying transformation rules 2) and 4) to (17) derives

$$LK_{\text{D}}\dot{\mathbf{M}}_{\text{D}}(k) \otimes \dot{\mathbf{M}}(k) = \mathbf{0} \quad (18)$$

where the subscript D denotes the diagonal matrix transformed by $\text{diag}(\cdot)$. The coefficient of $\dot{\mathbf{M}}(k)$ is a constant coefficient multiple of $\dot{\mathbf{M}}(0)$ while $\dot{\mathbf{M}}(0)$ is the initial value of $\dot{\mathbf{m}}$. Hence, the nonlinear equation (17) for $\dot{\mathbf{m}}$ is transformed into the linear equation (18) for $\dot{\mathbf{M}}(k)$.

3) *Node Temperature Equations*: Node temperature equals the mixture of pipe outlet temperatures, which, according to Table I, yields

$$\text{diag}(\tau_{\text{L,I}}^{\text{s}})V_{\text{L,I}}^+\dot{\mathbf{m}} = V_{\text{L,I}}^+\text{diag}(\tau^{\text{out,s}})\dot{\mathbf{m}}, \quad (19)$$

$$\text{diag}(\tau_{\text{R,S,I}}^{\text{r}})V_{\text{R,S,I}}^-\dot{\mathbf{m}} = V_{\text{R,S,I}}^-\text{diag}(\tau^{\text{out,r}})\dot{\mathbf{m}}, \quad (20)$$

where $\tau^{\text{s/r}}$ is the vector of node supply/return temperatures, $\tau^{\text{out,s/r}}$ is the vector of the outlet temperatures of supply/return pipes; $V^+ = \max(\mathbf{0}, V)$ while $V^- = \min(\mathbf{0}, V)$. Notation explanations to (19) and (20) can be found in Appendix B.

Applying transformation rules 2) and 4) to (19) and (20) respectively derives

$$T_{\text{L,I,D}}^{\text{s}}(k) \otimes V_{\text{L,I}}^+\dot{\mathbf{M}}(k) = V_{\text{L,I}}^+T_{\text{D}}^{\text{out,s}}(k) \otimes \dot{\mathbf{M}}(k), \quad (21)$$

$$T_{\text{R,S,I,D}}^{\text{r}}(k) \otimes V_{\text{R,S,I}}^-\dot{\mathbf{M}}(k) = V_{\text{R,S,I}}^-T_{\text{D}}^{\text{out,r}}(k) \otimes \dot{\mathbf{M}}(k), \quad (22)$$

where $T^{\text{s/r}}(k)$ is the DT of $\tau^{\text{s/r}}$; $T^{\text{out,s/r}}(k)$ is the DT of $\tau^{\text{out,s/r}}$. In (21), the coefficients of $\dot{\mathbf{M}}(k)$ are linear combinations of elements of $T^{\text{s}}(0)$ and $T^{\text{out,s}}(0)$ while the coefficients of elements of $T^{\text{s}}(k)$ and $T^{\text{out,s}}(k)$ are $\dot{\mathbf{M}}(0)$. Hence, the nonlinear equation (19) about $\dot{\mathbf{m}}$, $\tau_{\text{L,I}}^{\text{s}}$ and $\tau^{\text{out,s}}$ is transformed into the linear equation (21) about $\dot{\mathbf{M}}(k)$, $T_{\text{L,I}}^{\text{s}}(k)$ and $T^{\text{out,s}}(k)$. Similar observation also applies to (22).

4) *Node Power Equations*: Heat power produced or consumed at slack/source/load nodes satisfies

$$\phi_{\text{R,S,L}} = C_{\text{p}}\text{diag}(\dot{\mathbf{m}}_{\text{R,S,L}}^{\text{in}})(\tau_{\text{R,S,L}}^{\text{s}} - \tau_{\text{R,S,L}}^{\text{r}}) \quad (23)$$

where ϕ is the vector of node power.

Applying transformation rules 2) and 4) to right hand side of (23) derives

$$\Phi_{\text{R,S,L}}(k) = C_{\text{p}}\dot{\mathbf{M}}_{\text{R,S,L,D}}^{\text{in}}(k) \otimes (T_{\text{R,S,L}}^{\text{s}}(k) - T_{\text{R,S,L}}^{\text{r}}(k)) \quad (24)$$

Likewise, the nonlinear equation (23) about ϕ , $\dot{\mathbf{m}}^{\text{in}}$, τ^{s} and τ^{r} are transformed into the linear equation (24) about $\Phi^*(k)$, $\dot{\mathbf{M}}^{\text{in}*}(k)$, $T^{\text{s}*}(k)$ and $T^{\text{r}*}(k)$, the coefficients of which are linear combinations of constants, $T^{\text{s}}(0)$, $T^{\text{r}}(0)$ and $\dot{\mathbf{M}}^{\text{in}}(0)$.

B. DTs of EPS Models

The power flow models are

$$\mathbf{p} = \text{diag}(\mathbf{e})(G\mathbf{e} - B\mathbf{f}) + \text{diag}(\mathbf{f})(B\mathbf{e} + G\mathbf{f}), \quad (25)$$

$$\mathbf{q} = \text{diag}(\mathbf{f})(G\mathbf{e} - B\mathbf{f}) - \text{diag}(\mathbf{e})(B\mathbf{e} + G\mathbf{f}), \quad (26)$$

$$\text{diag}(\mathbf{e}_{\text{PV}})\mathbf{e}_{\text{PV}} + \text{diag}(\mathbf{f}_{\text{PV}})\mathbf{f}_{\text{PV}} = \text{diag}(\mathbf{U}_{\text{PV}}^{\text{SP}})\mathbf{U}_{\text{PV}}^{\text{SP}}, \quad (27)$$

$$\mathbf{e}_{\text{R}} = \mathbf{e}^{\text{SP}}, \quad \mathbf{f}_{\text{R}} = \mathbf{f}^{\text{SP}}, \quad (28)$$

where \mathbf{p}/\mathbf{q} are respectively the vectors of active/reactive power; \mathbf{e}/\mathbf{f} are respectively the vectors of real/imaginary part of bus voltage; G/B are respectively the conductance/susceptance matrices; \mathbf{U} is the vector of bus voltage magnitude; subscript PV/R denotes the indices of rows in matrices/vectors related to PV/slack buses; superscript SP denotes that the values are specified. We denote by $\mathbf{P}(k)/\mathbf{Q}(k)/\mathbf{E}(k)/\mathbf{F}(k)$ the DT of $\mathbf{p}/\mathbf{q}/\mathbf{e}/\mathbf{f}$ respectively.

DTs of (25) to (28) have been derived in [22] using a set of different notations, which are

$$\mathbf{P}(k) = \mathbf{E}_{\text{D}}(k) \otimes (G\mathbf{E}(k) - B\mathbf{F}(k)) + \mathbf{F}_{\text{D}}(k) \otimes (B\mathbf{E}(k) + G\mathbf{F}(k)). \quad (29)$$

(29) is a linear equation about $\mathbf{P}(k)$, $\mathbf{E}(k)$, $\mathbf{F}(k)$, the coefficients of which are linear combinations of constants, $\mathbf{E}(0)$ and $\mathbf{F}(0)$.

Likewise, we have the DTs of (26), (27) and (28), which are linear equations about $\mathbf{Q}(k)$, $\mathbf{E}(k)$ and $\mathbf{F}(k)$ as follows.

$$\mathbf{Q}(k) = \mathbf{F}_{\text{D}}(k) \otimes (G\mathbf{E}(k) - B\mathbf{F}(k)) - \mathbf{E}_{\text{D}}(k) \otimes (B\mathbf{E}(k) + G\mathbf{F}(k)). \quad (30)$$

$$\mathbf{E}_{\text{PV,D}}(k) \otimes \mathbf{E}_{\text{PV}}(k) + \mathbf{F}_{\text{PV,D}}(k) \otimes \mathbf{F}_{\text{PV}}(k) = \mathbf{U}_{\text{PV,D}}^{\text{SP}}\mathbf{U}_{\text{PV}}^{\text{SP}}\delta(k). \quad (31)$$

$$\mathbf{E}_{\text{R}}(k) = \mathbf{e}^{\text{SP}}\delta(k), \quad \mathbf{F}_{\text{R}}(k) = \mathbf{f}^{\text{SP}}\delta(k). \quad (32)$$

C. DTs of Coupling Components

According to [26], the relationships between heat and electric power of extraction steam turbine and gas turbine respectively satisfy

$$\mathbf{p} = -\phi/Z + \eta_e F_{in}, \quad (33)$$

$$\phi = c_{m1} \mathbf{p}, \quad (34)$$

where Z is the ratio that describes the trade-off between heat supplied to the site and the electric power of the extraction steam turbine; c_{m1} is the heat-to-power ratio of the gas turbine; η_e is the electrical efficiency; F_{in} is the fuel input rate of the steam turbine.

Applying transformation rules 2), 3) and 5) to (33) and (34) derives

$$\mathbf{P}(k) = -\Phi(k)/Z + \eta_e F_{in} \delta(k), \quad (35)$$

$$\Phi(k) = c_{m1} \mathbf{P}(k). \quad (36)$$

which are linear equations about $\mathbf{P}(k)$ and $\Phi(k)$.

V. ILLUSTRATE SAS-DERIVATION FRAMEWORK AND INTRODUCE ADAPTIVE TIME WINDOW CONTROL

A. Illustrating the SAS-derivation framework on a small HE-IES

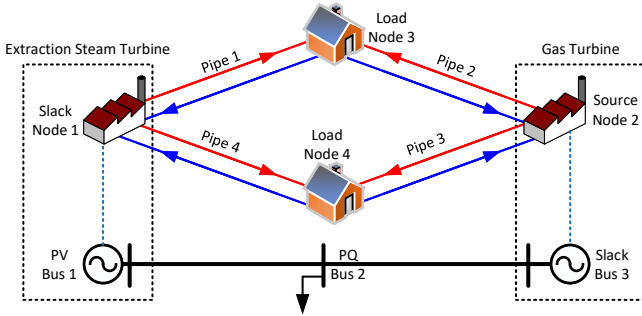


Fig. 2. A small HE-IES with four DHS nodes and three EPS buses.

The proposed method is illustrated on the small system shown in Fig. 2 which is comprised of four DHS nodes and three EPS buses. The node and loop incidence matrices of the DHS are respectively

$$\mathbf{V} = \begin{bmatrix} -1 & 0 & 0 & -1 \\ 0 & -1 & -1 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

and

$$\mathbf{L} = \begin{bmatrix} 1 & -1 & 1 & -1 \end{bmatrix}.$$

We assume the spatial step size Δx to be the length of each pipe. Therefore, PDE of each supply pipe is converted into an ODE about $\tau^{\text{out},s}$ and $\tau^{\text{in},s}$ while PDE of each return pipe is converted into an ODE about $\tau^{\text{out},r}$ and $\tau^{\text{in},r}$. We denote by $\tau^{\text{in},s/r}$ the vector of the inlet temperatures of supply/return pipes. The known and unknown variables in the HE-IES are shown in Table II and III. There are totally thirty-eight unknown variables. Different from Table I and the bus type assumption in power flow model, p_1 is unknown because the

TABLE II
KNOWN AND UNKNOWN VARIABLES OF EPS

Bus Number	1	2	3
Type	PV	PQ	Slack
Known	-	p_2, q_2	e_3, f_3
Unknown	e_1, f_1, p_1, q_1	e_2, f_2	p_3, q_3

electrical output of bus 1 is constrained by the heat output, which is ϕ_1 , of the extraction steam turbine and ϕ_1 is unknown. ϕ_2 is unknown because the heat output of node 2 is constrained by the electrical output, which is p_3 , of the gas turbine and p_3 is unknown.

First, we obtain the zeroth-order DT coefficients, which are the initial values of variables, by steady-state energy flow calculation [26]. Then we should derive the DT of known variables, for example, $T_1^s(0)$ to $T_1^s(K)$ of τ_1^s . Next, we can start calculating the first-order DT coefficients of the above thirty-eight unknown variables based on the following three steps.

Step 1: Calculate elements of vector $\mathbf{T}^{\text{out},s}(1)$ and $\mathbf{T}^{\text{out},r}(1)$ by (8). For example,

$$\begin{aligned} T_1^{\text{out},s}(1) &= \frac{1}{\gamma\rho\Delta x} \dot{M}(0)(T_1^{\text{in},s}(0) - T_1^{\text{out},s}(0)) \\ &\quad - \frac{\lambda}{\gamma\rho C_p} (T_1^{\text{out},s}(0) - \tau^{\text{amb}}\delta(0)/\tau_b) \end{aligned}$$

where the right hand side are all known. The other seven unknown first-order DT coefficients can be calculated directly likewise. That is, there remain thirty unknown first-order DT coefficients.

Step 2: By (14) and (15), the mass flow continuity equations are transformed into four linear equations about $\dot{\mathbf{M}}(1)$ and $\dot{\mathbf{M}}^{\text{in}}(1)$.

By (18), the loop pressure equation is transformed into

$$\begin{bmatrix} 1 & -1 & 1 & -1 \end{bmatrix} \text{diag} \begin{bmatrix} K_1 \\ K_2 \\ K_3 \\ K_4 \end{bmatrix} \cdot 2 \cdot \text{diag} \begin{bmatrix} \dot{M}_1(0) \\ \dot{M}_2(0) \\ \dot{M}_3(0) \\ \dot{M}_4(0) \end{bmatrix} - \text{diag} \begin{bmatrix} \dot{M}_1(1) \\ \dot{M}_2(1) \\ \dot{M}_3(1) \\ \dot{M}_4(1) \end{bmatrix} = \mathbf{0},$$

which is one linear equation about $\dot{\mathbf{M}}(1)$.

By (21) and (22), the node temperature equations are transformed into

$$\begin{aligned} &\text{diag} \begin{bmatrix} T_3^s(0) \\ T_4^s(0) \end{bmatrix} \mathbf{V}_{3,4}^+ \begin{bmatrix} \dot{M}_1(1) \\ \dot{M}_2(1) \\ \dot{M}_3(1) \\ \dot{M}_4(1) \end{bmatrix} + \text{diag} \begin{bmatrix} T_3^s(1) \\ T_4^s(1) \end{bmatrix} \mathbf{V}_{3,4}^+ \begin{bmatrix} \dot{M}_1(0) \\ \dot{M}_2(0) \\ \dot{M}_3(0) \\ \dot{M}_4(0) \end{bmatrix} = \\ &\mathbf{V}_{3,4}^+ \text{diag} \begin{bmatrix} T_1^{\text{out},s}(1) \\ T_2^{\text{out},s}(1) \\ T_3^{\text{out},s}(1) \\ T_4^{\text{out},s}(1) \end{bmatrix} \begin{bmatrix} \dot{M}_1(0) \\ \dot{M}_2(0) \\ \dot{M}_3(0) \\ \dot{M}_4(0) \end{bmatrix} + \mathbf{V}_{3,4}^+ \text{diag} \begin{bmatrix} T_1^{\text{out},s}(0) \\ T_2^{\text{out},s}(0) \\ T_3^{\text{out},s}(0) \\ T_4^{\text{out},s}(0) \end{bmatrix} \begin{bmatrix} \dot{M}_1(1) \\ \dot{M}_2(1) \\ \dot{M}_3(1) \\ \dot{M}_4(1) \end{bmatrix}, \\ &\text{diag} \begin{bmatrix} T_1^r(0) \\ T_2^r(0) \end{bmatrix} \mathbf{V}_{1,2}^- \begin{bmatrix} \dot{M}_1(1) \\ \dot{M}_2(1) \\ \dot{M}_3(1) \\ \dot{M}_4(1) \end{bmatrix} + \text{diag} \begin{bmatrix} T_1^r(1) \\ T_2^r(1) \end{bmatrix} \mathbf{V}_{1,2}^- \begin{bmatrix} \dot{M}_1(0) \\ \dot{M}_2(0) \\ \dot{M}_3(0) \\ \dot{M}_4(0) \end{bmatrix} = \\ &\mathbf{V}_{1,2}^- \text{diag} \begin{bmatrix} T_1^{\text{out},r}(1) \\ T_2^{\text{out},r}(1) \\ T_3^{\text{out},r}(1) \\ T_4^{\text{out},r}(1) \end{bmatrix} \begin{bmatrix} \dot{M}_1(0) \\ \dot{M}_2(0) \\ \dot{M}_3(0) \\ \dot{M}_4(0) \end{bmatrix} + \mathbf{V}_{1,2}^- \text{diag} \begin{bmatrix} T_1^{\text{out},r}(0) \\ T_2^{\text{out},r}(0) \\ T_3^{\text{out},r}(0) \\ T_4^{\text{out},r}(0) \end{bmatrix} \begin{bmatrix} \dot{M}_1(1) \\ \dot{M}_2(1) \\ \dot{M}_3(1) \\ \dot{M}_4(1) \end{bmatrix}, \end{aligned}$$

TABLE III
KNOWN AND UNKNOWN VARIABLES OF DHS

	Node 1	Node 2	Node 3	Node 4	Pipe 1	Pipe 2	Pipe 3	Pipe 4
Type	Slack	Source	Load	Load	-	-	-	-
Known	τ_1^s	τ_2^s, τ_3^r	ϕ_3, τ_3^r	ϕ_4, τ_4^r	-	-	-	-
Unknown	$\phi_1, \tau_1^r, \dot{m}_1^{\text{in}}$	$\phi_2, \tau_2^r, \dot{m}_2^{\text{in}}$	$\tau_3^s, \dot{m}_3^{\text{in}}$	$\tau_4^s, \dot{m}_4^{\text{in}}$	$\tau_1^{\text{out,s/r}}, \tau_1^{\text{in,s/r}}, \dot{m}_1$	$\tau_2^{\text{out,s/r}}, \tau_2^{\text{in,s/r}}, \dot{m}_2$	$\tau_3^{\text{out,s/r}}, \tau_3^{\text{in,s/r}}, \dot{m}_3$	$\tau_4^{\text{out,s/r}}, \tau_4^{\text{in,s/r}}, \dot{m}_4$

$$\begin{aligned} \begin{bmatrix} P_1(1) \\ P_2(1) \\ P_3(1) \end{bmatrix} &= \text{diag} \begin{bmatrix} E_1(1) \\ E_2(1) \\ E_3(1) \end{bmatrix} \left(G \begin{bmatrix} E_1(0) \\ E_2(0) \\ E_3(0) \end{bmatrix} - B \begin{bmatrix} F_1(0) \\ F_2(0) \\ F_3(0) \end{bmatrix} \right) + \text{diag} \begin{bmatrix} F_1(1) \\ F_2(1) \\ F_3(1) \end{bmatrix} \left(B \begin{bmatrix} E_1(0) \\ E_2(0) \\ E_3(0) \end{bmatrix} + G \begin{bmatrix} F_1(0) \\ F_2(0) \\ F_3(0) \end{bmatrix} \right) \\ &+ \text{diag} \begin{bmatrix} E_1(0) \\ E_2(0) \\ E_3(0) \end{bmatrix} \left(G \begin{bmatrix} E_1(1) \\ E_2(1) \\ E_3(1) \end{bmatrix} - B \begin{bmatrix} F_1(1) \\ F_2(1) \\ F_3(1) \end{bmatrix} \right) + \text{diag} \begin{bmatrix} F_1(0) \\ F_2(0) \\ F_3(0) \end{bmatrix} \left(B \begin{bmatrix} E_1(1) \\ E_2(1) \\ E_3(1) \end{bmatrix} + G \begin{bmatrix} F_1(1) \\ F_2(1) \\ F_3(1) \end{bmatrix} \right), \end{aligned} \quad (37)$$

$$\begin{aligned} \begin{bmatrix} Q_1(1) \\ Q_2(1) \\ Q_3(1) \end{bmatrix} &= \text{diag} \begin{bmatrix} F_1(1) \\ F_2(1) \\ F_3(1) \end{bmatrix} \left(G \begin{bmatrix} E_1(0) \\ E_2(0) \\ E_3(0) \end{bmatrix} - B \begin{bmatrix} F_1(0) \\ F_2(0) \\ F_3(0) \end{bmatrix} \right) - \text{diag} \begin{bmatrix} E_1(1) \\ E_2(1) \\ E_3(1) \end{bmatrix} \left(B \begin{bmatrix} E_1(0) \\ E_2(0) \\ E_3(0) \end{bmatrix} + G \begin{bmatrix} F_1(0) \\ F_2(0) \\ F_3(0) \end{bmatrix} \right) \\ &+ \text{diag} \begin{bmatrix} F_1(0) \\ F_2(0) \\ F_3(0) \end{bmatrix} \left(G \begin{bmatrix} E_1(1) \\ E_2(1) \\ E_3(1) \end{bmatrix} - B \begin{bmatrix} F_1(1) \\ F_2(1) \\ F_3(1) \end{bmatrix} \right) - \text{diag} \begin{bmatrix} E_1(0) \\ E_2(0) \\ E_3(0) \end{bmatrix} \left(B \begin{bmatrix} E_1(1) \\ E_2(1) \\ E_3(1) \end{bmatrix} + G \begin{bmatrix} F_1(1) \\ F_2(1) \\ F_3(1) \end{bmatrix} \right), \end{aligned} \quad (38)$$

$$2E_1(0)E_1(1) + 2F_1(0)F_1(1) = 0. \quad (39)$$

which are four linear equations about $\dot{M}(1)$, $T_L^s(1)$, $T_S^r(1)$. $T^{\text{out,s}}(1)$ and $T^{\text{out,r}}(1)$ have been calculated in *Step 1*.

By (24), the node power equations are transformed into

$$\begin{aligned} \begin{bmatrix} \Phi_1(1) \\ \Phi_2(1) \\ \Phi_3(1) \\ \Phi_4(1) \end{bmatrix} &= C_p \text{diag} \begin{bmatrix} \dot{M}_1^{\text{in}}(1) \\ \dot{M}_2^{\text{in}}(1) \\ \dot{M}_3^{\text{in}}(1) \\ \dot{M}_4^{\text{in}}(1) \end{bmatrix} \left(\begin{bmatrix} T_1^s(0) \\ T_2^s(0) \\ T_3^s(0) \\ T_4^s(0) \end{bmatrix} - \begin{bmatrix} T_1^r(0) \\ T_2^r(0) \\ T_3^r(0) \\ T_4^r(0) \end{bmatrix} \right) \\ &+ C_p \text{diag} \begin{bmatrix} \dot{M}_1^{\text{in}}(0) \\ \dot{M}_2^{\text{in}}(0) \\ \dot{M}_3^{\text{in}}(0) \\ \dot{M}_4^{\text{in}}(0) \end{bmatrix} \left(\begin{bmatrix} T_1^s(1) \\ T_2^s(1) \\ T_3^s(1) \\ T_4^s(1) \end{bmatrix} - \begin{bmatrix} T_1^r(1) \\ T_2^r(1) \\ T_3^r(1) \\ T_4^r(1) \end{bmatrix} \right), \end{aligned}$$

which are four linear equations about $\Phi(1)$, $\dot{M}^{\text{in}}(1)$, $T_L^s(1)$ and $T_S^r(1)$.

By (29), (30), (31), the power flow equations are transformed into (37), (38) and (39), which are seven linear equations about $P_{\text{PV,R}}(1)$, $Q_{\text{PV,R}}(1)$, $E_{\text{PV,PQ}}(1)$ and $F_{\text{PV,PQ}}(1)$.

By (35) and (36), the coupling component equations are transformed into

$$P_1(1) = -\Phi_1(1)/Z, \quad \Phi_2(1) = c_{m1}P_3(1),$$

which are two linear equations about $P_1(1)$, $\Phi_1(1)$, $\Phi_2(1)$ and $P_3(1)$. $\delta(1) = 0$ by definition.

There are totally twenty-two linear equations. By combining and solving these linear equations, we obtain the first-order DT coefficients of the above mentioned twenty-two variables. Then there remain eight unknown $T^{\text{in,s/r}}(1)$ which are the vectors of the DTs of $\tau^{\text{in,s/r}}$.

Step 3: Update $T^{\text{in,s/r}}(1)$. Because $\tau^{\text{in,s/r}}$ equal the temperatures of the nodes that the inlets of the pipes are connected to, we have, for example,

$$T_1^{\text{in,s}}(1) = T_1^s(1).$$

Finally, the first-order DT coefficients of all unknown variables are obtained.

The above three steps is the $k = 1$ case of the more generalized notation

$$\text{Step 1: } X(k) = B(X(0:k-1), Y(0:k-1), Z(0:k-1)),$$

$$\text{Step 2: } A_0 Y(k) = C(X(0:k), Y(0:k-1), W(0:k)),$$

$$\text{Step 3: } Z(k) = D(Y(k)),$$

where $X(k)$ is the vector of $T^{\text{out,s}}(k)$ and $T^{\text{out,r}}(k)$. $Y(k)$ is the vector of unknown k th-order DT coefficients of the variables we solve in *Step 2*. $Z(k)$ is the vector of $T^{\text{in,s}}(k)$ and $T^{\text{in,r}}(k)$. $W(k)$ is the vector of the k th DT coefficients of known variables. $X(0:k)$ denotes the combinations of $X(0)$ to $X(k)$. The same notation also applies to $Y(k)$, $Z(k)$ and $W(k)$. We denote by $x(t)$, $y(t)$, $z(t)$ and $w(t)$ the original variables of these DTs. A_0 is the coefficient matrix that relates only to initial values. This follows from the observations in the previous section that the coefficients of the k th-order DT coefficients are the zeroth-order DT coefficients. Therefore, inverse of the matrix, A_0^{-1} , should be calculated only once in each step. $B(\cdot)$ denotes (8). $C(\cdot)$ is the known DT coefficients that we move to the right hand side of equations. $B(\cdot)$ and $C(\cdot)$ are mainly comprised of the convolution operations, \otimes , of matrices/vectors. $D(\cdot)$ assigns the k th order DT coefficients of node temperatures to related $T^{\text{in,s/r}}(k)$.

As illustrated above, we can repeat the above three steps, recursively calculating the DT coefficients from the zeroth-order to the K -th order. Finally, the SASs of all variables are obtained.

B. Adaptive Time Window Control

To strengthen the robustness of the proposed method under big disturbances, an adaptive time window control strategy,

which ensures that the temporal step sizes produce results that satisfy the error tolerances, is developed as follows.

The idea comes from the philosophy of embedded Runge-Kutta formulas [24], which constructs two Runge-Kutta formulas with different numerical accuracy while sharing the same function values. Then the difference of these two Runge-Kutta approximations yield an estimate of the local error which can be used for step size control.

Here, assuming that we have finished the DT- K calculation, we can obtain DT- $K + 1$ recursively by one more evaluation of the three steps in Section V-A. Then we can approximate the local truncation error of DT- K , which is the Lagrange remainder

$$\frac{(\mathbf{x})^{(K+1)}(\xi)}{(K+1)!}(\Delta t)^{K+1}, \quad \text{where } \xi \in [0, \Delta t],$$

by the difference between DT- K and DT- $K + 1$, which is

$$\tilde{\mathbf{x}} = \mathbf{X}(K+1)(\Delta t)^{K+1}.$$

To ensure convergence and accuracy requirements, each component of $\tilde{\mathbf{x}}$ should be within the prescribed error tolerance vector ε , the component of which is defined as

$$\varepsilon_i = \text{Atol} + \min(|x_i(0)|, |x_i(\Delta t)|) \cdot \text{Rtol}$$

where Atol and Rtol are respectively the prescribed absolute and relative error tolerances. Rtol is prescribed to control the number of significant figures in the computed values. $\min(|x_i(0)|, |x_i(\Delta t)|)$ ensures that both ends of the time window $[0, \Delta t]$ satisfy Rtol. Atol is prescribed to prevent endless step size diminishing when true values are close to or equal zero.

We take root mean square error

$$err = \sqrt{\frac{1}{n} \sum_{i=1}^n \left(\frac{\tilde{x}_i}{\varepsilon_i} \right)^2}$$

to measure the overall relative error of the current time window. Then we compare err with 1. If $err \leq 1$, the current time window is accepted and the computation of next time window is advanced with a new temporal step size Δt_{new} . Else, the current time window is rejected and the computation starts again with Δt_{new} until $err \leq 1$.

There comes the problem of the formulation of Δt_{new} . Based on the temporal accuracy order $K + 1$, it is natural to take

$$\Delta t'_{\text{new}} = \Delta t \cdot fac \cdot (1/err)^{1/(K+1)}$$

where fac is a conservative factor smaller than one, which tries to avoid the reject of $\Delta t'_{\text{new}}$. However, it is also observed by numerical experiments that the surge of temporal step sizes increases the probability of the reject of the next time window while the plunge of temporal step sizes to unnecessarily small levels significantly reduces efficiency. Therefore, we should set an upper bound fac_{max} and a lower bound fac_{min} to restrain the violent variation of temporal step size, which finally yields

$$\Delta t_{\text{new}} = \min(fac_{\text{max}} \cdot \Delta t, \max(fac_{\text{min}} \cdot \Delta t, \Delta t'_{\text{new}})).$$

A special case where models containing sin or cos pattern should be further discussed here because the even terms of

Taylor series of sin function at $t = k\pi$ ($k \in \mathbb{N}$) equal zero, making $\mathbf{X}(K) \neq \mathbf{0}$ while $\mathbf{X}(K+1) = \mathbf{0}$ if K is odd. The cos counterpart encounters the same dilemma if K is even. In the cases where $\mathbf{X}(K+1) = \mathbf{0}$, we should perform one more recursive calculation based on DT- $K + 1$, and use $\mathbf{X}(K+2)$ to adjust temporal step sizes.

C. Overall Pseudocode of Quasi-dynamic Energy Flow Calculation under Disturbances

We can set disturbances easily by giving the variation of known variables, $\mathbf{w}(t)$, in Table II and III. The variation can be either distinct functions or discrete time-series. In each time window, we first derive $\mathbf{W}(0) : \mathbf{W}(K)$ and then start the recursive DT calculation. The overall pseudocode of the proposed DT-based quasi-dynamic energy flow calculation method is shown in Algorithm 1.

Algorithm 1: DT-based Quasi-dynamic Energy Flow Calculation

Input: Total simulation time \mathcal{T} , $\mathbf{x}(0)$, $\mathbf{y}(0)$, $\mathbf{z}(0)$, variation of $\mathbf{w}(t)$;

Output: $\mathbf{X}(0) : \mathbf{X}(K)$, $\mathbf{Y}(0) : \mathbf{Y}(K)$ and $\mathbf{Z}(0) : \mathbf{Z}(K)$ in each time window;

```

1 begin
2    $t \leftarrow 0$ ,  $\mathbf{X}(0) \leftarrow \mathbf{x}(0)$ ,  $\mathbf{Y}(0) \leftarrow \mathbf{y}(0)$ ,  $\mathbf{Z}(0) \leftarrow \mathbf{z}(0)$ ;
3   while  $t < \mathcal{T}$  do
4     Update  $\mathbf{A}_0$ , calculate  $\mathbf{A}_0^{-1}$ , and  $err \leftarrow 2$ ;
5     while  $err > 1$  do
6       Derive  $\mathbf{W}(0) : \mathbf{W}(K)$ ;
7       for  $k = 1 : K + 1$  do
8         Perform Step 1, 2, 3;
9       Derive  $[\tilde{\mathbf{x}}, \tilde{\mathbf{y}}]$  and  $[\varepsilon_x, \varepsilon_y]$ ;
10      Update  $err$ ;
11      if  $err \leq 1$  then
12         $t \leftarrow t + \Delta t$ ;
13      Update  $\Delta t_{\text{new}}$  and  $\Delta t \leftarrow \Delta t_{\text{new}}$ ;
14       $\mathbf{X}(0) \leftarrow \sum_{k=0}^K \mathbf{X}(k)t^k$ ,  $\mathbf{Y}(0) \leftarrow \sum_{k=0}^K \mathbf{Y}(k)t^k$ ,  $\mathbf{Z}(0) \leftarrow \sum_{k=0}^K \mathbf{Z}(k)t^k$ 

```

D. Some special cases

1) *Compound node types:* In, for example, the Barry Island system [26], some loads are located in intermediate nodes. Thereby, according to Table I, τ^r of these nodes are both known and unknown, which is contradictory. Actually, the real case is, load nodes are connected to intermediate nodes through *implicit pipes* whose length is 0 m. As shown in Fig. 3, we can extract a virtual load node and then the original compound node becomes an intermediate one.

Since the length of Pipe 3 is 0 m, we can not model Pipe 3 by PDE and hence, $T_3^{\text{out,s}}(k+1)$ and $T_3^{\text{out,r}}(k+1)$ can not be obtained in *Step 1* of the SAS-derivation framework.

Instead, we have two additional equations,

$$\tau_3^{\text{out,s}} = \tau_1^s, \quad \tau_3^{\text{out,r}} = \tau_3^r.$$

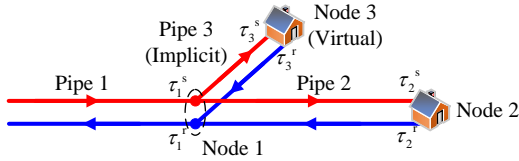


Fig. 3. Dealing with compound node types.

After performing DT, we have

$$T_3^{\text{out},s}(k+1) = T_1^s(k+1), \quad T_3^{\text{out},r}(k+1) = T_3^r(k+1).$$

There are two extra variables and two extra equations. So we still can successfully perform *Step 2* of the SAS-derivation framework by adding the above two equations to $C(\cdot)$.

The cases where source nodes are located in intermediate nodes can be dealt with likewise.

2) *Reverse mass flow rates*: DHSs are directed networks, that is, if mass flow rate of some pipe is reversed at some time during the simulation, then the mass flow continuity equations, loop pressure equations and node temperature equations will have different formulae. Fortunately, these equations are depicted by node/loop incidence matrices in this paper. We can obtain the post-reverse equations by reversing columns related to these reverse pipes in those matrices. As for the semi-discrete PDE, we should exchange the inlet and outlet in Fig. 1, and reverse the computation sequence.

Having finished the calculation in one time window, we judge if, at the end of the current time window $[t_0, t_1]$, some $\dot{m}_i < 0$. If there is, then we should find the zero t' of univariate polynomial equation

$$\sum_{k=0}^K M_i(k)t^k = 0,$$

which can be solved by the Newton-Raphson or bisection method. This is because, after $t > t'$, the equations differ from the original ones.

These special cases were considered in the following Barry Island case study.

VI. CASE STUDIES

In this section, we tested the proposed method on 1) a real DHS with measured data, 2) the Barry Island system and 3) a 225-node-118-bus big system. In the first system, we showed that the proposed method can effectively reduce dissipative and dispersive errors, which improves the accuracy of PDE solutions. In the second system, we compared the proposed method with iteration ones in terms of accuracy and time performance. We analyzed the reason of efficiency improvement of the proposed method in details and studied the impact of tuning parameters of the proposed method on computational performance. In the last system, we mainly tested the robustness and convergence of the proposed method under severe disturbances. All the tests were programmed with MATLAB R2022a on a desktop computer equipped with AMD Ryzen 7 3700x CPU and 64GB RAM.

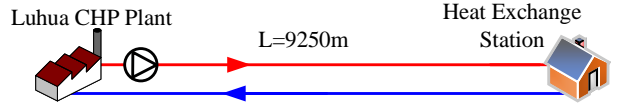


Fig. 4. A real DHS located in Shijiazhuang, Hebei Province, China.

A. A Real DHS with Measured Data

As shown in Fig. 4, the real DHS, located in the suburb of Shijiazhuang, Hebei Province, China, consists of a pair of supply and return pipes that connect the CHP plant and the heat exchange station. The parameters and measured data are given in [27]. The spatial and temporal step sizes were set to be 370 meters and 180 seconds, respectively. IU scheme, SOE scheme, node method (NM) and DT-20 were performed.

We first fixed \dot{m} to be 2543.5 kg s^{-1} and gave step boundary condition. The temperature of the inlet, denoted by τ_1 , increased from 90.1725°C to 92.0000°C at $t=10\text{h}$. The exact solution of (3), denoted by REF below, was obtained by the characteristic line method, which is

$$\tau(x, t) = (1 - e^{-\frac{\lambda}{c_p \dot{m}} x}) \tau^{\text{amb}} + e^{-\frac{\lambda}{c_p \dot{m}} x} \tau(0, t - \frac{\gamma \rho}{\dot{m}} x). \quad (40)$$

As shown in Fig. 5, solution by the IU scheme was severely smeared because of its big dissipative errors. Obviously, high frequency components in the solution were damped out. Meanwhile, serious oscillation ruined the solution by the SOE scheme due to its dispersive errors. The solution by DT with $\theta = 2$ had small dissipative errors while it revealed small dispersive error in front of the rising edge. The solution by DT with $\theta = 1$ had relatively big dissipative error but no dispersive error. The node method performed best in the test, showing no dissipative and dispersive errors.

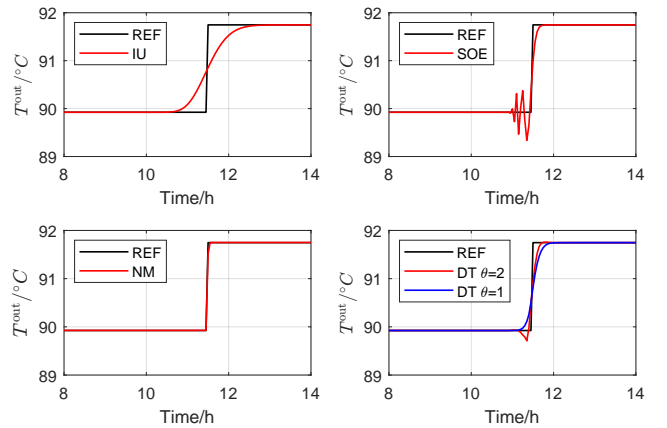


Fig. 5. Accuracy comparison on step function.

Next, we varied both \dot{m} and τ_1 . The results were compared with the measured data in Fig. 6. We first knocked out the IU scheme, because it is of first-order accuracy only and displayed big dissipative errors, omitting most of the details of temperature variation. Solutions by the SOE scheme and NM portrayed and tracked the temperature variations well, but small oscillations could not be avoided. On the contrary, solutions by DT got rid of these fake oscillations effectively. Compared with $\theta = 2$, $\theta = 1$ produced more conservative results. As circled in Fig. 6, the real oscillation was damped

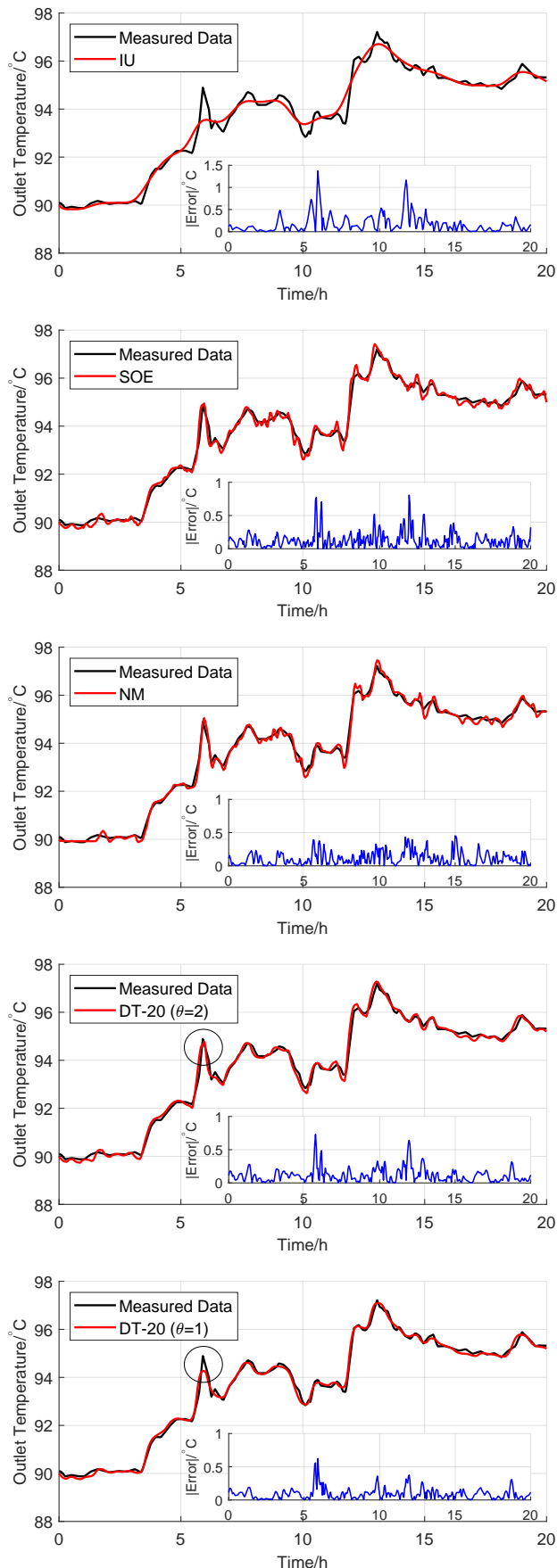


Fig. 6. Accuracy comparison on a real DHS.

with $\theta = 1$. However, the root mean square errors (RMSEs) between the solutions and the measured data, which are computed in Table IV, showed that DT with $\theta = 1$ was the most accurate and was trailed by the NM and DT with $\theta = 2$. The exclusive TVD property granted the proposed method accuracy superiority over others and we think that DT with $\theta = 1$ should be adopted in this case.

TABLE IV
RMSES AGAINST MEASURED DATA

IU	SOE	NM	DT($\theta = 2$)	DT($\theta = 1$)
0.2607	0.1773	0.1484	0.1546	0.1253

B. The Barry Island System

To test the accuracy, efficiency and convergence performance of the proposed method, the famous Barry Island system was used here. Each heating and electric load was given a typical 24-hour variation trajectory, with 15 min as the interval. These load trajectories are classified into four types, including Supermarkets, Guaranteed Savings Buildings, Hotels and Industrial Factories. The reader can refer to [28] for design philosophy of the test bed and [27] for detailed parameters.

The following methods were performed.

- M1—The HE-FBI iteration solver proposed in [7], which solves PDE with IU scheme.
- M2—Replace the IU scheme in M1 by SOE scheme.
- M3—Replace the IU scheme in M1 by NM.
- M4—DT-6 with $\theta = 1$.
- M5—DT-6 with $\theta = 2$.

The spatial step size for M1, M2, M4 and M5 was 20 m. The temporal step size for M1, M2 was 60 s. The convergence performance of M3 is unsatisfactory. For M3, only temporal step size 10 s produced relatively good results, which was adopted here. M4 and M5 were performed with adaptive windows, but results with fixed time windows were obtained from SASs with temporal step size 60 s. Error tolerance was set to be $1e-9$.

The Reference method (REF) generated reference values with spatial and temporal step sizes 2 m and 3 s respectively. The PDE (3) is first spatially discretized by (6) and then temporally discretized by the Dormand-Prince5 Runge-Kutta formula [29]. [17] has verified the accuracy of the PDE solver. The nonlinear algebraic equations and the discretized PDEs were alternatively solved as [7] does in each of the seven stages of Dormand-Prince5.

The EPS power flow calculations in M1, M2, M3 and REF were performed with the state-of-the-art Matpower V7.1 [30].

The simulation results of T^s , \dot{m} and p are shown in Fig. 7-9. The dissipative errors of M1 and the dispersive errors of M2 were still obvious in temperatures. Different from the preceding fixed direction case, M3 could not accurately depict the temperature variations in the limited convergent range. However, M4 and M5 obtained fairly good temperature results. Though mass flow rates were reversed several times in this case, M1, M2, M4 and M5 still produced accurate \dot{m} and p trajectories.

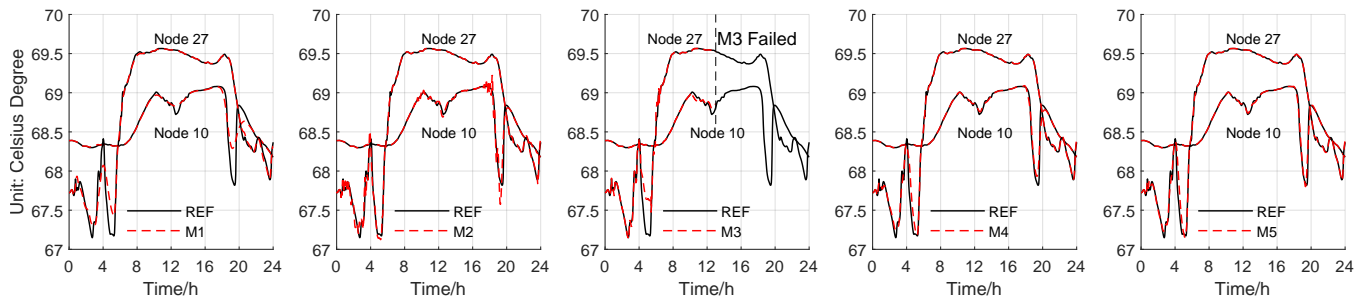


Fig. 7. Node supply temperature in Barry Island case.

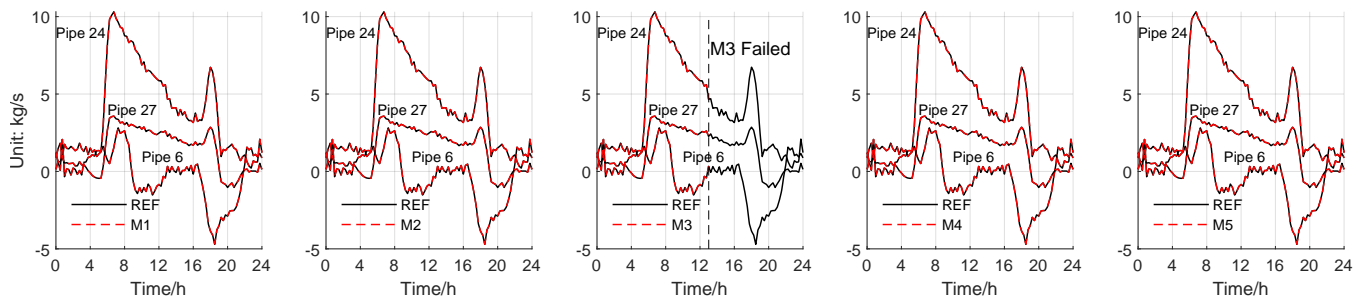


Fig. 8. Mass flow rate in Barry Island case.

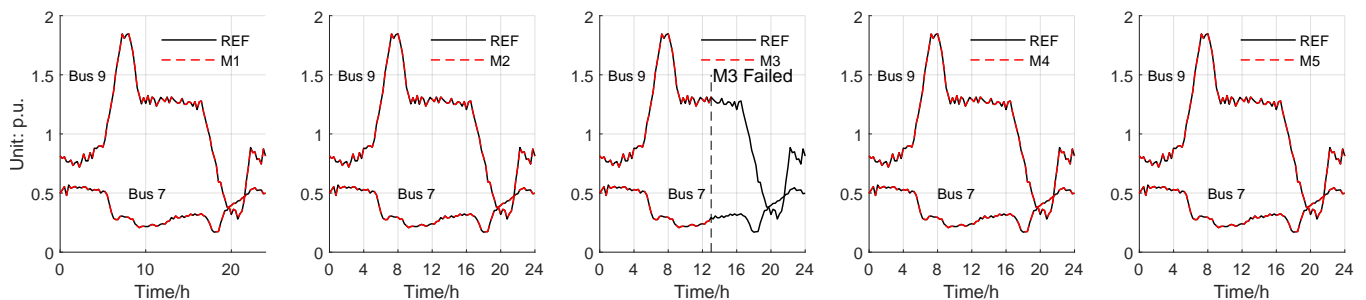


Fig. 9. Electric Power in Barry Island case.

The maximum RMSEs of each variable were listed in Table V. M5 had the smallest errors, which was trailed by M4. On one hand, the proposed method diminished dissipative and dispersive errors effectively with the TVD property. On the other, the high order temporal accuracy, which was 20 here, and the strict time window control strategy impeded the growth of errors. These facts explain the accuracy performance of the proposed method from the theoretical perspective.

TABLE V
MAXIMUM RMSEs

Variable	M1	M2	M3	M4	M5
T^s ($^{\circ}\text{C}$)	1.01e-1	7.57e-2	\	5.64e-2	5.21e-2
T^r ($^{\circ}\text{C}$)	3.41e-2	3.52e-2	\	2.13e-2	1.80e-2
\dot{m} (kg s^{-1})	7.27e-3	4.99e-3	\	3.82e-3	3.17e-3
ϕ (W)	1.60e+3	1.04e+3	\	7.49e+2	5.96e+2
p (p.u.)	1.98e-4	1.28e-4	\	9.30e-5	7.39e-5
e (p.u.)	1.34e-7	8.45e-8	\	6.13e-8	4.89e-8
f (p.u.)	2.87e-6	1.86e-6	\	1.35e-6	1.07e-6

The time performance of M1-M5 were compared in Table VI. M5 spent the shortest time finishing the simulation, trailed first by M4 and then M1, M2. The efficiency improvement not only came from the adaptive time window strategy, i.e. fewer

TABLE VI
TIME PERFORMANCE

Time Cost (s)	M1	M2	M3	M4	M5
Total	36.17	37.64	\	27.30	20.86
Per Step	0.025	0.026	\	0.019	0.018

¹ M1 and M2 took 1440 steps, M4 took 1452 steps, M5 took 1183 steps.

TABLE VII
COMPONENTS OF PER-STEP TIME COST (s)

Simulation Routines	M1	M5
Matrix Update and Factorization	0.0171	0.0011
Arithmetic Operation ¹	0.0068	0.0128
Others	0.0007	0.0037

¹ For M1, it mainly contains the recursive calculation of pipe temperatures. For M5, it mainly contains the convolution operation, \otimes , of matrices/vectors.

total steps, but also from the smaller per-step time cost. To explain this, the components of time cost in each step were profiled and listed in Table VII, with M1 and M5 serving as the representative of iterative and non-iterative solvers respectively. The update and factorization of Jacobian/coefficient matrices occupy the most computation overhead of M1 be-

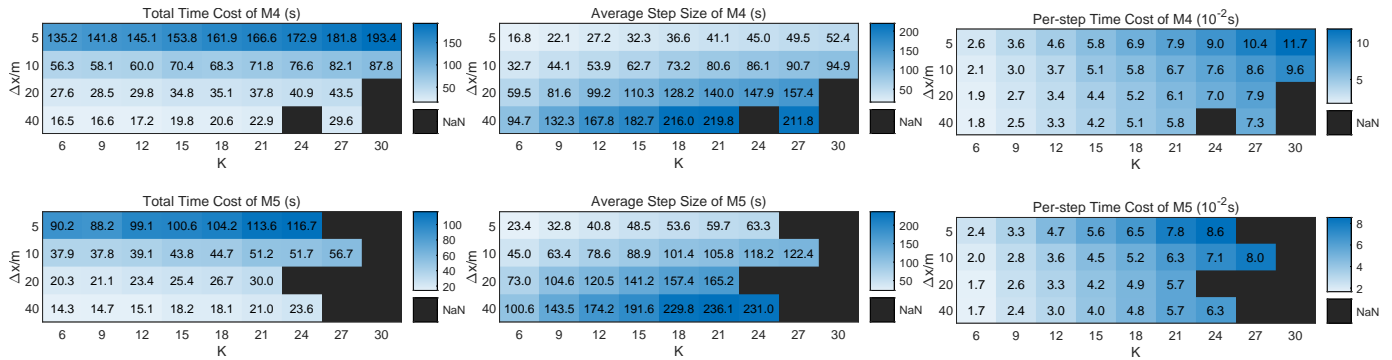


Fig. 10. Computational performance in terms of spatial step size and K .

TABLE VIII
AVERAGE NUMBER OF ITERATIONS OF M1

Iteration Loops	Number of iterations
IES loop	10.60
DHS loop	2.24
Hydraulic loop	1.92

cause, as shown in Table VIII, tens of iterations are required in each step of M1. Different from M1, the time cost of M5 in matrix computation was tiny. This followed from the fact, which has been pointed out in Section V-A, that matrix update and factorization should be performed only once in each step of M5. Although it took M5 much more time to perform the arithmetic operations, which mainly consisted of K times convolution operations of matrices/vectors, the overall computational burden was still small. Therefore, the proposed method (M4) was able to perform more efficiently when the spatial and temporal step sizes resembled M1 and M2's. With the adaptive window strategy, the proposed method (M5) could further improve the overall efficiency.

Even if we varied different Δx , the proposed method still had better per-step time performance as shown in Table IX. We believe that $K = 6$ has the best time performance in this system. Because it was observed from Fig. 10 that bigger K meant bigger average step size, but the smaller number of total steps was offset by the spiking per-step time cost, finally decreasing the overall efficiency. It should be also noted that infinitely enlarging K led to divergence instead of infinite temporal step size. This is because M4 and M5 are conditionally stable and hence, have limited convergence region. Their convergence region, which was reflected by the average temporal step size Δt , depended on the choice of Δx , and we found that bigger Δx allowed bigger Δt . Therefore, the adaptive time window strategy is critical to ensure computation convergence when we do not know what Δt ensures convergence. The choice of θ also influences the computation performance. M4 and M5 had similar per-step time cost, but M5 admitted bigger Δt , which achieved higher overall efficiency in this system.

C. A 225-DHS-Node and 118-EPS-Bus HE-IES

The 225-DHS-Node and 118-EPS-Bus HE-IES was constructed by integrating a real DHS from [31] and an EPS from

TABLE IX
PER-STEP TIME COSTS UNDER DIFFERENT SPATIAL STEP SIZE (10⁻² s)

	5 m	10 m	20 m	40 m
M1	3.9	2.9	2.4	2.2
M2	4.0	3.0	2.5	2.2
M4	2.6	2.1	1.9	1.8
M5	2.4	2.0	1.7	1.7

TABLE X
MAXIMUM RMSES IN SCENARIO 1

Variable	M1	M2	M3	M4	M5
T^s (°C)	1.22e-1	7.69e-2	2.76e-2	4.56e-2	\
T^r (°C)	1.96e-4	1.50e-4	5.59e-3	6.34e-5	\
\dot{m} (kg s ⁻¹)	1.96e-1	8.65e-2	7.00e-2	4.50e-2	\
ϕ (W)	1.71e+4	7.56e+3	1.17e+4	4.95e+3	\
p (p.u.)	1.54e-2	6.80e-3	1.05e-2	4.45e-3	\
e (p.u.)	5.35e-6	2.36e-6	3.64e-6	1.54e-6	\
f (p.u.)	6.30e-5	2.78e-5	4.29e-5	1.82e-5	\

[32]. The original DHS was modified by adding a loop. An extraction steam turbine acts as the slack node 1 of DHS and the PV bus 118 of EPS. A gas turbine acts as the source node 224 of DHS and the slack bus 1 of EPS. The readers can refer to [27] for detailed parameters.

1) *Accuracy and efficiency*: M1-M2 with $\Delta x = 25$ m and $\Delta t = 60$ s, M2 with $\Delta t = 60$ s, M4-M5 with $K = 20$ and $\Delta x = 25$ m, and REF with $\Delta x = 5$ m and $\Delta t = 2$ s were performed in the following two scenarios.

- Scenario 1: $\mathcal{T} = 4$ h. T_1^s increased from 85 °C at $t = 600$ s to 86 °C at $t = 660$ s.

TABLE XI
MAXIMUM RMSES IN SCENARIO 2

Variable	M1	M2	M3	M4	M5
T^s (°C)	2.32e-2	3.32e-3	1.38e-2	5.55e-4	2.62e-4
T^r (°C)	1.95e-4	5.95e-5	5.59e-3	1.25e-5	8.55e-6
\dot{m} (kg s ⁻¹)	3.92e-3	6.56e-4	5.51e-2	1.66e-4	8.67e-5
ϕ (W)	5.10e+2	8.69e+1	8.99e+3	1.97e+1	1.22e+1
p (p.u.)	4.59e-4	7.82e-5	8.10e-3	1.77e-5	1.10e-5
e (p.u.)	1.86e-7	3.18e-8	3.18e-6	6.69e-9	4.28e-9
f (p.u.)	1.85e-6	3.14e-7	3.27e-5	7.23e-8	4.44e-8

TABLE XII
TIME PERFORMANCE

Time Cost (s)	M1	M2	M3	M4	M5
Scenario 1	180.35	179.54	402.15	146.40	\
Scenario 2	264.57	252.04	848.18	216.24	150.18

- Scenario 2: $\mathcal{T} = 4$ h. Active load of PQ bus 63 to 77 and PQ bus 100 to 111 changed sinusoidally with amplitude half of its original value and period $0.35\mathcal{T}$. The disturbance started at $t = \mathcal{T}/40$ and ended at $t = 36\mathcal{T}/40$.

Accuracy and time performance of M1-M5 were compared in Table X to Table XII. M5 failed in scenario 1, so we think that M5 is not suitable for cases where there are steep temperature changes. M3 obtained the most accurate T^s results since, as shown in Fig. 5, it can effectively get rid of dissipative and dispersive errors in this case. Except for T^s , the proposed algorithm obtained more accurate results with smaller total time costs.

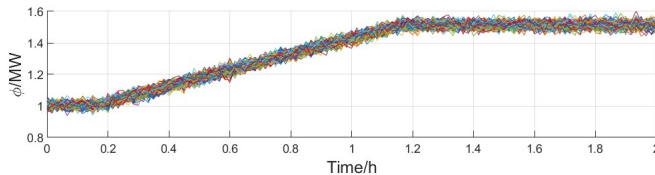


Fig. 11. Load curve when loading level=150%.

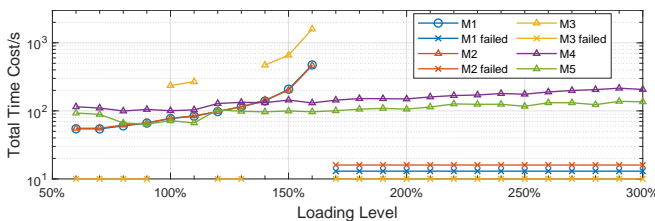


Fig. 12. Total time costs under different loading level.

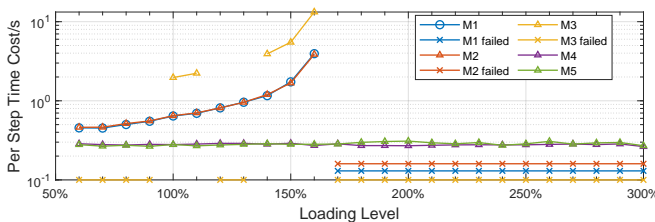


Fig. 13. Per-step time costs under different loading level.

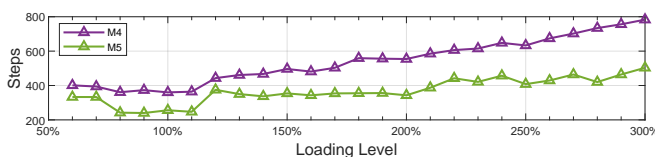


Fig. 14. Number of steps under different loading level.

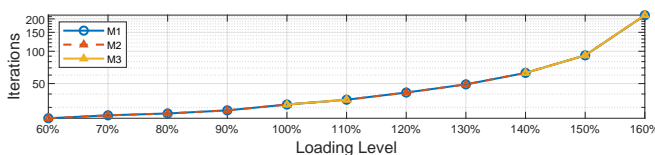


Fig. 15. Average number of IES loop iterations under different loading level.

2) *Robustness*: In DHS, some heat load, for instance, the Guaranteed Saving Buildings (GSBs), spikes over ten times within one hour [28]. To test if the proposed method can produce convergent results and have robust time performance

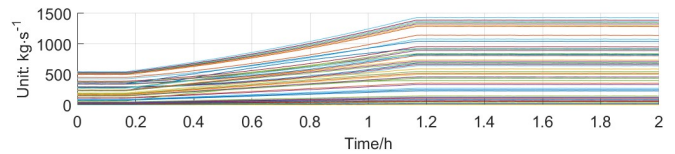


Fig. 16. Mass flow rates by M4 when final loading level=300%.

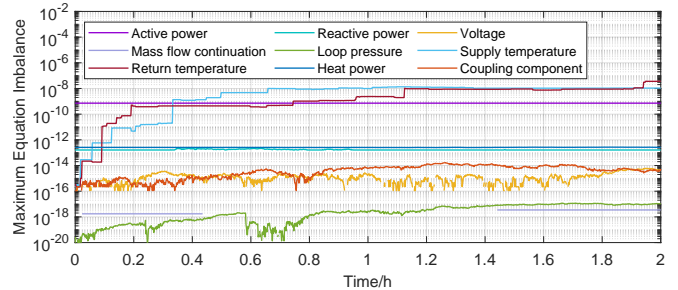


Fig. 17. Maximum equation imbalance when final loading level=300%.

under this kind of severe disturbances, the following scenario is designed.

- Scenario: $\mathcal{T} = 2$ h. Loading level of all load nodes in the DHS changed from loading level 100% at $t = 10$ min to the target loading level, which ranges between 60% and 300% with step size 10%, at $t = 70$ min.

To simulate the real scenario, random noises with uniform distribution were added to the load curves and initial load values. The load curve when target loading level = 150% is shown in Fig. 11. M1-M5 were performed with previous settings.

The proposed method had consistent and robust time performance. As shown in Fig. 13, the per-step time costs of M4 and M5 were nearly the same under all loading levels. As shown in Fig. 12 and 14, the total time costs of M4 and M5 increased linearly as the target loading level grew since $\overline{\Delta t}$ diminished and the number of steps slightly increased.

In most cases, iteration solvers, M1-M3, produced divergent results. The total time costs of M1-M3 grew exponentially as the loading level grew. This is because the spiking loading level significantly increased the number of iterations and hence the per-step time costs. However, M1 and M2 were more efficient than the proposed method when the loading levels were smaller than 80% for the number of iterations was tiny. M1-M3 had similar number of iterations in the convergent cases, but the per-step time cost of M3 was much bigger than that of M1 and M2. This is because M3 performed time-consuming index operations in each step.

In the cases where M1-M3 and even the REF failed, we verified the correctness of the proposed method as follows. Firstly, Fig. 16 is drawn to show that the results are reasonable, that is, they did not converge to meaningless solutions. Secondly, the results were verified by checking the imbalance of the nonlinear algebraic equations. We substituted the solutions into the original algebraic equations and it is shown in Fig. 17 that the maximum equation imbalance is around $1e-8$. Therefore, the proposed method was able to provide reliable results in this case, and hence demonstrated good robustness.

The possible explanations to the robustness superiority of

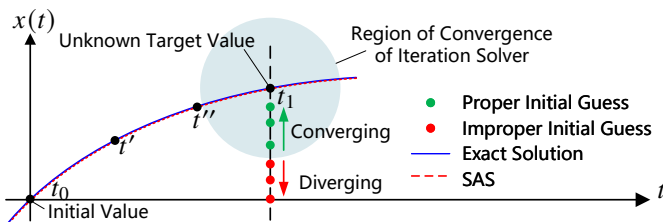


Fig. 18. Graphical explanation to the robustness of the proposed method.

the proposed method are as follows. 1) First, the proposed method solves DHS and EPS models together, getting rid of alternating errors. 2) Second, the adaptive time window control strategy controls the simulation errors within the prescribed error tolerance effectively. 3) Third, the proposed method does not rely on initial guesses while for iteration methods, initial guesses should be provided within the region of convergence. The region of convergence is an area probably centered by the unknown target value. However, the theory of iteration solver can not figure out the probable location of unknown target value, and the shape/area of the region of convergence. Following the common practice of solving differential equations, we assigned the results of the previous time window to the initial guesses when performing iteration methods. As shown in Fig. 18, if the distances between these initial guesses and the target values are longer than the radius of the region of convergence, the iteration solver diverges. This accounts for the failures of iteration methods in the robustness tests where the variables changed violently and hence the target values are far away from the initial guesses. As shown in Fig. 18, the proposed method starts the searching of the target value at $t = t_1$ from the initial value at $t = t_0$ with a definite trajectory. Though the approximation of target value may deviate greatly from the true values, we can use the adaptive time window control strategy to detect these errors, and perform simulation with smaller temporal step sizes again, for example, from $t = t'$ to $t = t''$. The above steps are conducted repeatedly until the value that the proposed method finds at $t = t_1$ is within the error tolerance. Therefore, the proposed method were more likely to ensure convergence in the preceding robustness tests.

VII. CONCLUSION

This paper proposes a DT-based non-iterative method to achieve efficient and robust time performance in HE-IES quasi-dynamic energy flow calculation. A semi-discrete TVD scheme is solved by DT to reduce dissipative and dispersive errors in thermal dynamics. An adaptive time window control strategy is designed to accelerate calculation and avoid non-convergence issues.

The proposed method has better accuracy performance compared with the FDM-based methods. But in cases where there are steep temperature changes, it cannot depict the temperature variations as accurately as NM. The proposed method has small and consistent per-step time costs regardless of loading levels, and thus, displays efficient and robust time performance. Whereas the iteration methods can only converge rapidly in lightly loaded cases. To choose parameter θ and order K should also be careful. In cases where there are

steep temperature changes, we should set $\theta = 1$ while in other cases, setting $\theta = 2$ will be more efficient. Though small K is suggested, we think that the choice of K should be case-specific. Several tests should be performed ahead of simulation to avoid tiny average temporal step sizes. The adaptive time window strategy is critical since the proposed method is conditionally stable.

The proposed method also applies to simplified scenarios where, for example, the DHSs are in quality regulation mode. By further considering the equipment models in HE-IESs, the proposed method is expected to perform efficient dynamic simulation, which will be our future research purpose.

APPENDIX A

STEP SIZES AND DISSIPATIVE/DISPERSIVE ERRORS

We illustrate the relationships between step sizes and dissipative/dispersive errors using the following thermal-dynamics-like PDE

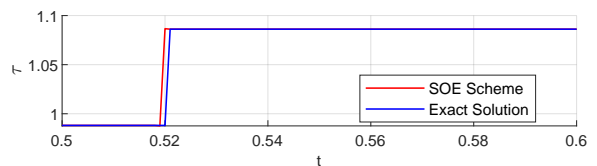
$$\frac{\partial \tau}{\partial t} + v \frac{\partial \tau}{\partial x} + 0.9997(\tau - 0.4) = 0$$

with step boundary condition. v is the mass flow velocity.

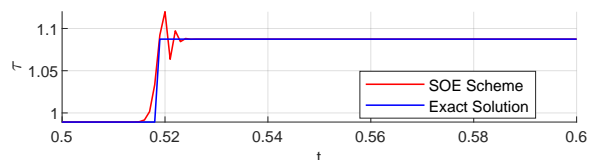
For the SOE scheme [15]

$$\begin{cases} \frac{\partial \tau}{\partial t} = \frac{\tau_k^{n+1} - \tau_k^n + \tau_{k+1}^{n+1} - \tau_{k+1}^n}{2\Delta t} \\ \frac{\partial \tau}{\partial x} = \frac{\tau_{k+1}^{n+1} - \tau_k^{n+1} + \tau_{k+1}^n - \tau_k^n}{2\Delta x} \end{cases}, \quad (\text{A.1})$$

spatial step size Δx and temporal step size Δt satisfying $v\Delta t/\Delta x = 1$ eliminate dissipative and dispersive errors in solutions effectively, which is shown in Fig. 19(a). $R = v\Delta t/\Delta x$ is called Courant-Friedrichs-Lewy (CFL) number in literature and serves as an important stability indicator of difference schemes [16]. However, if we increase v slightly, then the dispersive errors, which are the fake oscillations in Fig. 19(b), become obvious. Therefore, SOE scheme is able to obtain accurate results when mass flow velocities are fixed, but it is difficult for the scheme to avoid dispersive errors when mass flow velocities are variable.



(a) $\Delta x=0.05, \Delta t=0.001, v=50, R=1$



(b) $\Delta x=0.05, \Delta t=0.001, v=55, R=1.1$

Fig. 19. Results by the SOE scheme.

For the IU scheme [13]

$$\begin{cases} \frac{\partial \tau}{\partial t} = \frac{\tau_{k+1}^{n+1} - \tau_k^n}{\Delta t} \\ \frac{\partial \tau}{\partial x} = \frac{\tau_{k+1}^{n+1} - \tau_k^{n+1}}{\Delta x} \end{cases}, \quad (\text{A.2})$$

it can be observed from Fig. 20 that neither putting $R = 1$ nor decreasing Δx can eliminate the decaying of high-frequency components, which are the manifestation of dissipative errors in solutions. Therefore, solutions by the IU scheme are always accompanied by the dissipative errors.

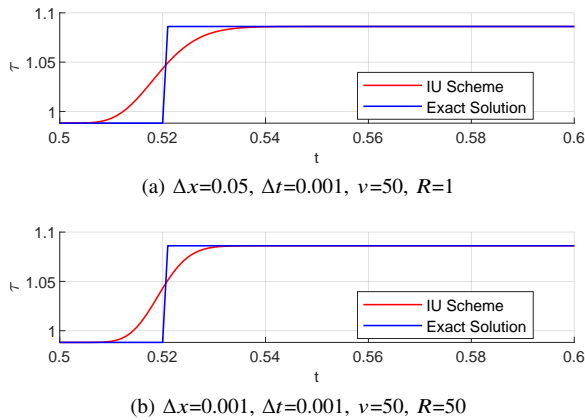


Fig. 20. Results by the IU scheme.

APPENDIX B NOTATION EXPLANATION

Notation of (19) and (20) can be explained using the supply network example in Fig. 21 where we assume Node 1 to be an intermediate node.

τ_1^s can be calculated by node temperature mixture equation [7] as

$$\tau_1^s = \tau_1^{\text{out},s} \frac{\dot{m}_1}{\dot{m}_1 + \dot{m}_2} + \tau_2^{\text{out},s} \frac{\dot{m}_2}{\dot{m}_1 + \dot{m}_2}.$$

Moving $\dot{m}_1 + \dot{m}_2$ in denominator to the left hand side, we have

$$\tau_1^s (\dot{m}_1 + \dot{m}_2) = \tau_1^{\text{out},s} \dot{m}_1 + \tau_2^{\text{out},s} \dot{m}_2.$$

Adding \dot{m}_3 and $\tau_3^{\text{out},s}$ to the equation, and rewriting it as matrix-vector form, we have

$$\tau_1^s \begin{bmatrix} 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} \dot{m}_1 \\ \dot{m}_2 \\ \dot{m}_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} \tau_1^{\text{out},s} & & \\ & \tau_2^{\text{out},s} & \\ & & \tau_3^{\text{out},s} \end{bmatrix} \begin{bmatrix} \dot{m}_1 \\ \dot{m}_2 \\ \dot{m}_3 \end{bmatrix}.$$

Because

$$\begin{bmatrix} 1 & 1 & 0 \end{bmatrix} = \max(V_1, \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}) = V_1^+$$

where $V_1 = \begin{bmatrix} 1 & 1 & -1 \end{bmatrix}$ denotes the row of node incidence matrix related to the intermediate node—Node 1, we have

$$\tau_1^s V_1^+ \dot{m} = V_1^+ \text{diag}(\tau^{\text{out},s}) \dot{m}.$$

Node supply/return temperature equations (19) and (20) for other types of nodes can be obtained in a similar way.

REFERENCES

- [1] Z. Li, W. Wu, M. Shahidehpour *et al.*, “Combined heat and power dispatch considering pipeline energy storage of district heating network,” *IEEE Trans. Sustain. Energy*, vol. 7, no. 1, pp. 12–22, Jan. 2016.
- [2] S. Lu, W. Gu, K. Meng *et al.*, “Thermal inertial aggregation model for integrated energy systems,” *IEEE Trans. Power Syst.*, vol. 35, no. 3, pp. 2374–2387, May 2020.
- [3] M. Zhang, Q. Wu, J. Wen *et al.*, “Optimal operation of integrated electricity and heat system: A review of modeling and solution methods,” *Renew. Sust. Energ. Rev.*, vol. 135, p. 110098, 2021.

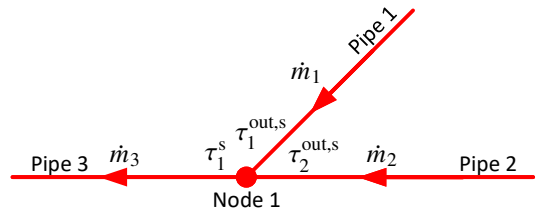


Fig. 21. Temperature mixture case in the supply network.

- [4] Z. Pan, Q. Guo, and H. Sun, “Interactions of district electricity and heating systems considering time-scale characteristics based on quasi-steady multi-energy flow,” *Appl. Energy*, vol. 167, pp. 230–243, 2016.
- [5] Z. Pan, J. Wu, H. Sun *et al.*, “Quasi-dynamic interactions and security control of integrated electricity and heating systems in normal operations,” *CSEE J. Power Energy Syst.*, vol. 5, no. 1, pp. 120–129, Mar. 2019.
- [6] S. Zhang, W. Gu, S. Lu *et al.*, “Dynamic security control in heat and electricity integrated energy system with an equivalent heating network model,” *IEEE Trans. Smart Grid*, vol. 12, no. 6, pp. 4788–4798, Nov. 2021.
- [7] X. Qin, H. Sun, X. Shen *et al.*, “A generalized quasi-dynamic model for electric-heat coupling integrated energy system with distributed energy resources,” *Appl. Energy*, vol. 251, p. 113270, Oct. 2019.
- [8] S. Zhang, W. Gu, H. Lu *et al.*, “Superposition-principle based decoupling method for energy flow calculation in district heating networks,” *Appl. Energy*, vol. 295, p. 117032, 2021.
- [9] J. Yang, N. Zhang, A. Botterud *et al.*, “On an equivalent representation of the dynamics in district heating networks for combined electricity-heat operation,” *IEEE Trans. Power Syst.*, vol. 35, no. 1, pp. 560–570, Jan. 2020.
- [10] Y. Chen, Q. Guo, H. Sun *et al.*, “Integrated heat and electricity dispatch for district heating networks with constant mass flow: A generalized phasor method,” *IEEE Trans. Power Syst.*, vol. 36, no. 1, pp. 426–437, Jan. 2021.
- [11] S. Zhang, W. Gu, H. Lu *et al.*, “Fully analytical model of heating networks for integrated energy systems,” Sep. 2021, preprint. [Online]. Available: <http://dx.doi.org/10.13140/RG.2.2.12808.96002>
- [12] J. Zheng, Z. Zhou, and J. Zhao, “Function method for dynamic temperature simulation of district heating network,” *Appl. Therm. Eng.*, vol. 123, pp. 682–688, 2017.
- [13] Y. Wang, S. You, H. Zhang, X. Zheng *et al.*, “Thermal transient prediction of district heating pipeline: Optimal selection of the time and spatial steps for fast and accurate calculation,” *Appl. Energy*, vol. 206, pp. 900–910, 2017.
- [14] A. Dénarié, M. Aprile, and M. Motta, “Heat transmission over long pipes: New model for fast and accurate district heating simulations,” *Energy*, vol. 166, pp. 267–276, 2019.
- [15] S. Yao, W. Gu, S. Lu *et al.*, “Dynamic optimal energy flow in the heat and electricity integrated energy system,” *IEEE Trans. Sustain. Energy*, vol. 12, no. 1, pp. 179–190, Jan. 2021.
- [16] J. Thomas, *Numerical Partial Differential Equations: Finite Difference Methods*. New York, NY, USA: Springer-Verlag, 1995.
- [17] A. Kurganov and E. Tadmor, “New high-resolution central schemes for nonlinear conservation laws and convection-diffusion equations,” *J. Comput. Phys.*, vol. 160, no. 1, pp. 241–282, 2000.
- [18] S. Zhang, W. Gu, S. Yao *et al.*, “Partitional decoupling method for fast calculation of energy flow in a large-scale heat and electricity integrated energy system,” *IEEE Trans. Sustain. Energy*, vol. 12, no. 1, pp. 501–513, Jan. 2021.
- [19] H. R. Massrur, T. Niknam, J. P. S. Catalão *et al.*, “Fast decomposed energy flow in large-scale integrated electricity–gas–heat energy systems,” *IEEE Trans. Sustain. Energy*, vol. 9, no. 4, pp. 1565–1577, Oct. 2018.
- [20] Y. Liu, K. Sun, R. Yao *et al.*, “Power system time domain simulation using a differential transformation method,” *IEEE Trans. Power Syst.*, vol. 34, no. 5, pp. 3739–3748, Sep. 2019.
- [21] Y. Liu and K. Sun, “Solving power system differential algebraic equations using differential transformation,” *IEEE Trans. Power Syst.*, vol. 35, no. 3, pp. 2289–2299, May 2020.
- [22] Y. Liu, K. Sun, and J. Dong, “A dynamized power flow method based on differential transformation,” *IEEE Access*, vol. 8, pp. 182441–182450, 2020.
- [23] H. Wang and H. Meng, “Improved thermal transient modeling with new 3-order numerical solution for a district heating network with

consideration of the pipe wall's thermal inertia," *Energy*, vol. 160, pp. 171–183, 2018.

- [24] E. Hairer, S. P. Nørsett, and G. Wanner, *Solving Ordinary Differential Equations I*, 2nd ed. Berlin, Heidelberg, Germany: Springer-Verlag, 1993.
- [25] A. A. Medovikov, "High order explicit methods for parabolic equations," *BIT*, vol. 38, no. 2, pp. 372–390, Jun 1998.
- [26] X. Liu, J. Wu, and N. Jenkins, "Combined analysis of electricity and heat networks," *Appl. Energy*, vol. 162, pp. 1238–1250, 2016.
- [27] Supplementary material. [Online]. Available: https://github.com/rzyu45/PaperDT_supplementary_materials
- [28] X. Qin, X. Shen, Y. Guo *et al.*, "Combined Electric and Heat System Testbeds for Power Flow Analysis and Economic Dispatch," *CSEE J. Power Energy Syst.*, vol. 7, no. 1, pp. 34–44, 2021.
- [29] J. Dormand and P. Prince, "A family of embedded Runge-Kutta formulae," *J. Comput. Appl. Math.*, vol. 6, no. 1, pp. 19–26, 1980.
- [30] R. D. Zimmerman, C. E. Murillo-Sánchez, and R. J. Thomas, "Matpower: Steady-state operations, planning, and analysis tools for power systems research and education," *IEEE Trans. Power Syst.*, vol. 26, no. 1, pp. 12–19, Feb. 2011.
- [31] E. Gong, N. Wang, S. You, and Y. Wang, "Optimal operation of novel hybrid district heating system driven by central and distributed variable speed pumps," *Energy Conv. Manag.*, vol. 196, pp. 211–226, 2019.
- [32] D. Zhang, Z. Fu, and L. Zhang, "An improved TS algorithm for loss-minimum reconfiguration in large-scale distribution systems," *Electr. Power Syst. Res.*, vol. 77, pp. 685–694, 2007.



Ruizhi Yu (S'20) received the B.S.E.E. degree in 2020 from the Chien-Shiung Wu honors college of Southeast University, Nanjing, China, where he is currently working toward the Ph.D.E.E. degree. His research interests include simulation and scripting of integrated energy systems.



Wei Gu (M'06-SM'16) received his B.S. and Ph.D. degrees in Electrical Engineering from Southeast University, China, in 2001 and 2006, respectively. From 2009 to 2010, he was a Visiting Scholar in the Department of Electrical Engineering, Arizona State University. He is now a professor in the School of Electrical Engineering, Southeast University the director of the Institute of Distributed Generations and Active Distribution Networks. His research interests include distributed generations and microgrids, integrated energy systems. Dr. Gu is an Editor for the *IEEE Transactions on Power Systems*, the *IET Energy Systems Integration* and the *Automation of Electric Power Systems* (China).



Shuai Yao (S'15) received the B.Eng. degree in electrical engineering from the Honors College of Nanjing Normal University, Nanjing, China, in 2017. He is currently working towards the Ph.D. degree with the School of Electrical Engineering, Southeast University, Nanjing, China. From 2021, he is a visiting ph.D. student at the Cardiff University, Cardiff, UK. His research interests include modeling, simulation and operational optimization of Integrated Energy Systems.



Suhan Zhang (S'19) received his B.S. degree in electrical engineering from North China Electric Power University, China, in 2018. He is currently working toward the Ph.D. degree from the School of Electrical Engineering, Southeast University, Nanjing, Jiangsu, China. His research interests include modeling, simulation and operation of integrated energy systems.



Shuai Lu (S'17-M'21) received his B.S. degree in Smart Grid Information Engineering from Nanjing University of Science and Technology, Nanjing, China, in 2016 and his Ph.D. degree in Electrical Engineering from Southeast University, Nanjing, China, in 2021. From 2018 to 2019, he was a visiting scholar at the University of New South Wales, Sydney, Australia. He is currently a Lecturer at the School of Electrical Engineering, Southeast University. He was selected as an Outstanding Reviewer for *IEEE Transactions on Power Systems* in 2020. His research interests include multi-energy systems, operations research, and data-driven techniques in power systems.



Shixing Ding (S'20) received his M.S. degree in Power Engineering from North China Electric Power University, Beijing, China, in 2019. He is currently pursuing a Ph.D. degree in Cyber Science and Engineering at Southeast University, Nanjing, China. His research interests include modeling, simulation, and optimization of integrated energy systems; CPS security for energy systems.