WATER RESOURCES RESEARCH, VOL. 42, W11502, doi:10.1029/2005WR004365, 2006

Automated Thiessen polygon generation

D. Han[1] and M. Bray[1]

Received 17 June 2005; revised 20 July 2006; accepted 2 August 2006; published 8 November 2006.

[1] Data uncertainty research of rain gauge network requires generation of large numbers of Thiessen polygons. Despite its importance in hydrology, few studies on computational Thiessen polygons have been carried out, and there is little published information in the hydrological literature. This paper describes two automated approaches and the ways for their implementation in hydrological applications: triangulation method and grid method. Triangulation is a lossless method but suffers from complications in coding and slow computational speed with small numbers of gauges. Grid method is easy to implement, but a compromise must be made between the computational grid size, accuracy, and speed. This paper describes a procedure to derive the relationship between the catchment area, grid size, and accuracy indicator based on weighted mean error. The computational speed comparison between the two methods has been found to follow a logarithm curve, and the critical number of gauges could be found from this curve for deciding the method choice if the computational speed is the limiting factor in a project.

Citation: Han, D., and M. Bray (2006), Automated Thiessen polygon generation, *Water Resour. Res.*, *42*, W11502, doi:10.1029/2005WR004365.

1. Introduction

[2] The Thiessen polygon method has played a very important role in hydrology and meteorology for estimating average rainfall (and other hydrometeorological factors, e.g., solar radiation) over a bounded area since its introduction from the Voronoi diagram in mathematics by Thiessen at the beginning of the twentieth century [*Thiessen and Alter*, 1911]. Nowadays, it is still a very important tool in hydrological research and practice [*Grant et al.*, 2004; *Chand et al.*, 2005]. This is particularly analogous to the popularity of the unit hydrograph method which is still being researched [*Yang and Han*, 2006]. Although hydrologists can either derive the Thiessen polygons by hand (pencil and paper) or by using commercial GIS (Geographic Information System) packages (e.g., ARC GIS 9.0), these approaches are not suitable for uncertainty research which would need a huge number of randomly generated Thiessen polygons for Monte Carlo analysis of rain gauge networks. For example, to analyze the uncertainty characteristics of a rain gauge network with various densities, it is necessary to carry out some bootstrap resampling of the existing rain gauges and usually thousands of simulations are needed, all with different Thiessen polygons (Bootstrap is a type of statistical analysis to test the reliability of certain systems to input data variations). In such cases, three important factors would influence the choice made by the hydrological researcher: coding complexity, execution speed and accuracy. Hydrological researchers are not professional mathematicians or programmers, so it is important that the algorithms involved must be simple and easy to implement in a common computational language (FORTRAN, C, MATLAB, etc.).

[3] The principle of the Thiessen polygon is quite simple [*Mumm*, 2005]: for a finite number of distinct sites in a plane (e.g., rain gauges), we wish to partition the plane into disjoint regions called cells, each of which contains exactly one site, so that all other points within a cell are closer to that cell's site than to any other site. Mathematically, suppose $P = \{p_1, p_2, \ldots, p_\mathrm{n}\}$ is a set of distinct points (sites) in the plane. We subdivide the plane into n cells so that each cell contains exactly one site. An arbitrary point (x,y) is in a cell corresponding to a site p_i with coordinates $(x_{pi}, \mathrm{y}_{\mathrm{pi}})$ if and only if $\sqrt{(x - x_{pi})^2 + (y - y_{pi})^2} < \sqrt{(x - x_{pj})^2 + (y - y_{pj})^2}$ for all p_j with $j \neq i, 1 \leq j, i \leq n$. That is, the Euclidean distance from (x,y) to any other site is greater than the distance from (x,y) to p_i. It turns out that the boundaries of the cells defined in this way will be composed of straight lines and segments forming convex polygons and will be defined by the perpendicular bisectors of segments joining each pair of sites. This method of partitioning a plane is called a Voronoi diagram in mathematics.

[4] Despite its importance in hydrology, few studies on computational Thiessen polygons (or automated Thiessen polygons) have been carried out and there is little published information in the hydrological literature. This information is timely needed now since the advancement of computing technology and large rain gauge networks have provide ideal opportunities for hydrological researchers to carry out data uncertainty analysis with a large number of Monte Carlo and Bootstrap simulations, which would require very fast generation of the Thiessen polygons.

[5] In this study, the River Brue catchment, situated in Somerset, England, is used as a test case for automated Thiessen polygon generation (Figure 1). The total catchment area is 136 km^2 with 49 rain gauges. The high-density rain gauge network is only feasible in hydrological experiments and in practice, the operational rain gauge networks are much less dense and hence would create uncertainties

[1] Department of Civil Engineering, University of Bristol, Bristol, UK.

0043-1397/06/2005WR004365

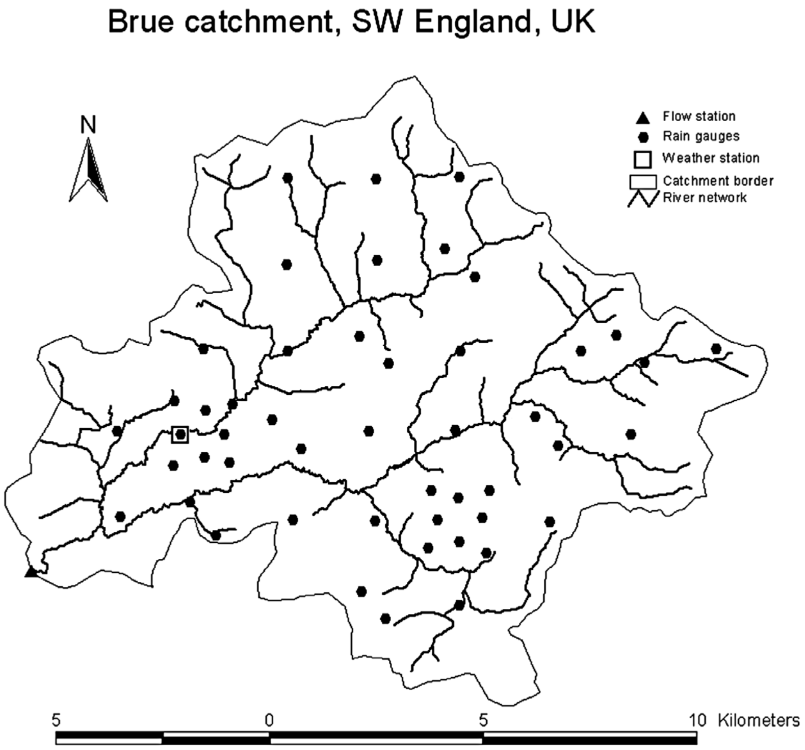

Figure 1. River network, gauges, and boundary of the Brue catchment (location of the river outlet: NGR: ST59003180 or 051°05′01″N, 002°35′07″W).

with the rainfall measured. Nowadays, uncertainty in river flow modeling is very topical [*Han et al.*, 2006] and among the major uncertainty components in the modeling process (data, parameters, and system uncertainties), data uncertainty has been an active research area in hydrology [*Hamlet and Lettenmaier*, 2005]. This 49-gauge network provides an ideal case for assessing the uncertainties in the rain gauge network with different time intervals. For example, to work out the uncertainty in using 10 gauges instead of 49 gauges over the Brue catchment, we would randomly select 10 gauges from the gauge pool by Bootstrap resampling method and the combinational possibilities are very high $(49!/(10!*39!)) = 8 \times 10^9$. Although it is possible to limit the analysis to a smaller quantity (e.g., 10,000 – 25,000 [*Martinez and Martinez*, 2002, p. 214]) of combinations, there would still be a large number of Thiessen polygons to generate, therefore the execution speed of the chosen algorithm is quite important in practice, in addition to its accuracy and ease of coding. In this study, two approaches are included: (1) triangulation method and (2) grid method. The study explores these two methods and their application in automated Thiessen polygon generation based on the aforementioned three factors in hydrological research activities.

2. Triangulation Method

[6] Voronoi diagrams are the foundation for the Thiessen polygon, which have been around for a long time and have undergone a good deal of study. In fact, *Aurenhammer* [1988] claims that about 1 out of 16 papers in computational geometry have been on research concerning Voronoi diagrams! More than 600 papers on the subject are listed by *Okabe et al.* [1992]. Numerous algorithms have been developed by mathematicians in improving the computational efficiency of Voronoi diagrams [*Aggarwal et al.*, 1987; *Klein and Lingas*, 1992], such as linear time randomized algorithm, divide and conquer algorithm and sweep algorithm. It has been found that there is no universally superior algorithm; ultimately one must choose an algorithm based on the particulars of the data and of the application [*Mumm*, 2005].

[7] Mathematically, the triangulation method is based on Delaunay triangulation [*Lee and Lin*, 1986], which has a collection of edges satisfying an “empty circle” property: for each edge we can find a circle containing the edge’s endpoints but not containing any other points. A Voronoi diagram can then be constructed based on Delaunay triangulation (Figure 2). The coding of those processes is very complicated and it is recommended that some third party functions should be used by hydrologists. In this study, MATLAB 7.0 (with mapping toolbox) is used (if other packages are used, e.g., Mathematica, the procedure should be very similar to the one described here). From the Voronoi diagram, there are two problems for us in converting it into a Thiessen polygon diagram (i.e., a bounded Voronoi diagram): (1) many cells are open and do not form proper polygons; (2) some lines are missing between the unclosed Voronoi cells due to their infinite conjunction points (where the points at which the lines meet are at an infinite distance). To solve these problems, all gauges are mirrored in four directions. The rationale for using the mirrored method is based on the symmetrical properties of the Voronoi diagram. The authors have tried other approaches (e.g., use algebraic geometry to close Voronoi cells or to program the closed

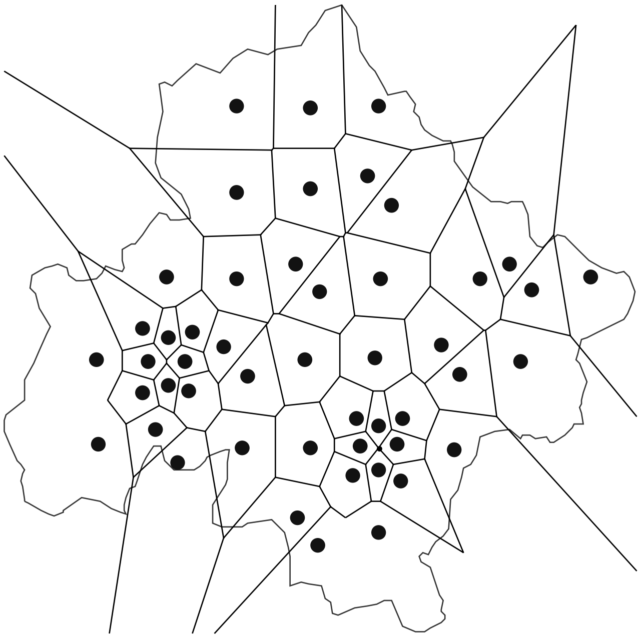

Figure 2. Voronoi diagram for the Brue catchment.

cells from the first principles) and found none of them are practical for hydrologists to code. The new Voronoi diagram based on the mirrored rain gauges can produce fully closed polygons for all the original sites as shown in Figure 3. The final Thiessen polygon scheme can be derived by polygon intersection and separation between the catchment polygon and the ones in Voronoi diagram. In this study, it has been found that the CPU time is closely linked with the number of gauges. For a 49-gauge network, a Monte Carlo simulation of 10,000 Thiessen polygon schemes would require about ten hours of CPU time. If we need to work from two gauges all the way up to 49 gauges, the CPU time required would be much longer, therefore more efficient algorithms are desirable (The computer used in this study has a CPU of Pentium IV with 3GHz speed Hyperthread and one GB RAM).

3. Grid Method

[8] The grid method is a numerical approach, hence it is an approximation to a lossless method (where no accuracy is lost due to the numerical scheme) and some accuracy is sacrificed for the sake of simplicity and speed of the algorithm. Early work in this area uses the paint function (to flood a polygon with the same color) on the screen to count the number of pixels in each cell [*Ge*, 2001]. Their method has a limitation caused by the screen resolution. An improved scheme is proposed in this study to use virtual pixels (or grids) in the computer memory and then use Euclidian distance to allocate each pixel to individual rain gauges. A compromise has to be made about the resolution of the grid and numerical accuracy. A fine grid size would produce high-quality Thiessen polygons, but would demand longer CPU time, and vice verse. A ten gauge Thiessen polygon scheme in the Brue catchment is shown as an illustration of this method in Figure 4. Firstly, a grid of pixels is distributed uniformly covering the rectangular area which encloses the catchment. Those pixels outside of the catchment are then excluded by using azimuth angle counting (calculate cumulative change in azimuth using tan^{-1} from pixel points to all catchment border vertices. If a point is inside the catchment, the cumulative angle would be 360 degrees, otherwise it would be zero). Next, the Euclidian distances between all interior pixels and the rain gauges are calculated. In practice, the grid method can be implemented in two ways: (1) combine the grid generation and allocation in one subroutine and (2) separate those two functions into different subroutines so that the duplicated grid generation part is used once and only a grid allocation function is called in subsequent simulations. The

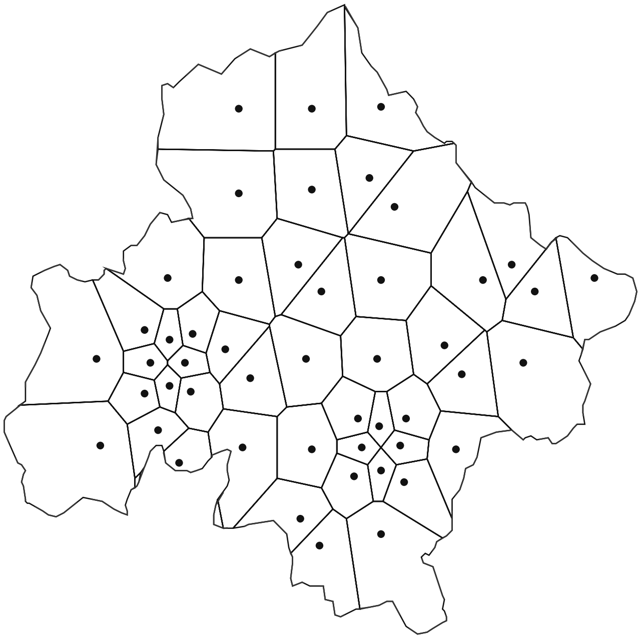

Figure 3. Bounded Voronoi diagram by polygon intersection (the Thiessen polygons).

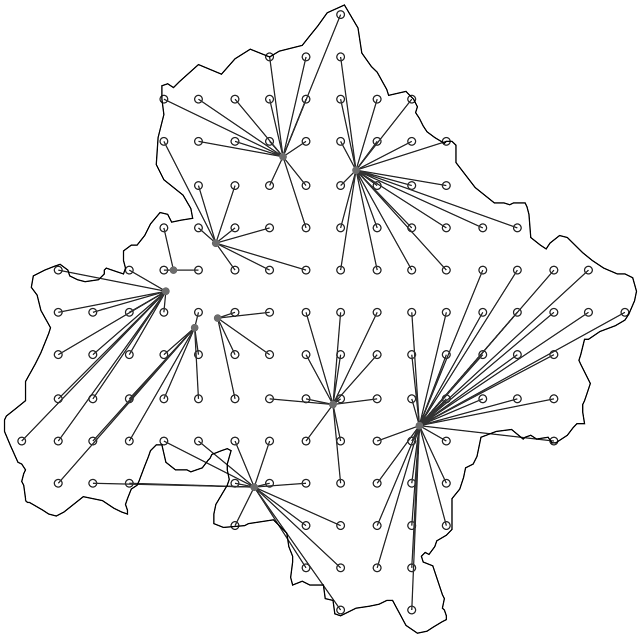

Figure 4. Grid method result for 10 rain gauges with grid size of 1000 m.

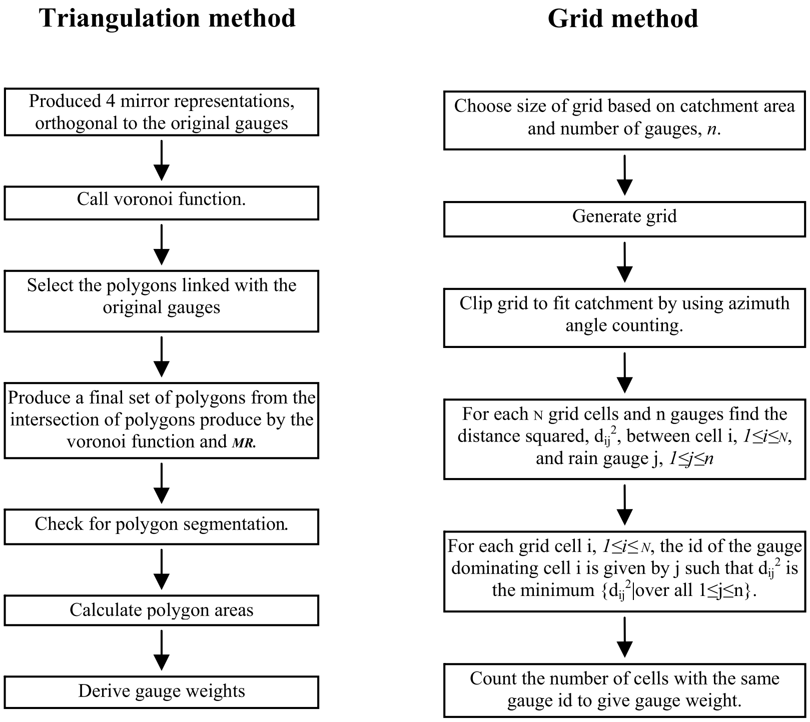
Triangulation method
Produced 4 mirror representations, orthogonal to the original gauges
Call voronoi function.
Select the polygons linked with the original gauges
Produce a final set of polygons from the intersection of polygons produce by the voronoi function and MR.
Check for polygon segmentation.
Calculate polygon areas
Derive gauge weights
Grid method
Choose size of grid based on catchment area and number of gauges, n.
Generate grid
Clip grid to fit catchment by using azimuth angle counting.
For each N grid cells and n gauges find the distance squared, d_{ij}^2, between cell i, $1 \le i \le N$, and rain gauge j, $1 \le j \le n$
For each grid cell i, $1 \le i \le N$, the id of the gauge dominating cell i is given by j such that d_{ij}^2 is the minimum $\{d_{ij}^2 | \text{over all } 1 \le j \le n\}$.
Count the number of cells with the same gauge id to give gauge weight.

Figure 5. Flowcharts of triangulation and grid methods.

study has found that the grid method with separated functions is much faster than the combined method. To aid the implementation of both methods, Figure 5 illustrates the key steps in the relevant procedures.

[9] To improve the computational efficiency further, it is important to use the squared distance as a proxy of Euclidian distance. This is because the inequality relationship based on Euclidian distance $\sqrt{(x - x_{pi})^2 + (y - y_{pi})^2} < \sqrt{(x - x_{pj})^2 + (y - y_{pj})^2}$ is equivalent to $(x - x_{pi})(x - x_{pi}) + (y - y_{pi})(y - y_{pi}) < (x - x_{pj})(x - x_{pj}) + (y - y_{pj})(y - y_{pj})$. The latter is much faster to execute than the former due to the working mechanism of modern computers. For each pixel, a minimum distance (or its proxy) can be found and that will be linked with a specific rain gauge. The Thiessen polygon weights can then be derived by summing up all pixels for every linked rain gauge.

[10] Like any numerical scheme, it is important to understand the relationship between the computation step (or grid size for pixels in this case) and the output accuracy. A large computation step can reduce the CPU time but it will reduce the output accuracy. An optimal scheme should be the one which uses the largest step allowable for the required accuracy. There are many statistics for measuring accuracies (such as RMSE (root mean square error), absolute errors and relative errors, etc.) and there is no single statistic which can satisfy all hydrological applications. In this study, the weighted mean error is (WME) further explored (if other criteria are used, the same procedure could be used as described below).

$$WME = 100 \sum_{i=1}^{N} \left(\frac{W_{Gi} - W_{Ti}}{W_{Ti}} \right) W_{Ti} \qquad (1)$$

where N is the number of rain gauges, W_{Gi} is weight by grid method for the gauge i, and W_{Ti} is weight by triangulation method for the gauge i. WME is basically a mean relative error for the weights in relation to the true weights derived by the triangulation method. To prevent the bias caused by large relative errors from small weights, all relative errors are adjusted by the Thiessen weights so that WME would reflect the overall accuracy of the grid method for the computed grid sizes and gauge numbers. A diagram with different grid sizes and WME could be produced by a large number of simulations as shown in Figure 6. If a threshold of 1% mean error is desired (other thresholds could be chosen depending on the project nature), the maximum allowed grid sizes for different numbers of gauges can then

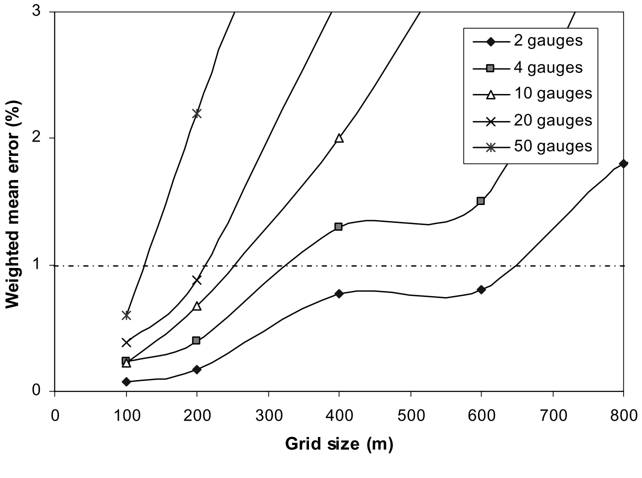

Figure 6. Polygon scheme accuracy based on the grid sizes and number of gauges.

be derived from Figure 6. It is interesting to notice that when the number of grids is divided by the number of gauges, the ratio is roughly around 200, therefore the more gauges in a catchment, the more grid pixels are needed to meet the accuracy demand. In this study, with WME as a measuring indicator, we can derive a formula for the required grid size for 1% error threshold:

$$\text{Grid size} = \sqrt{\frac{\text{catchment area}}{200 \times \text{number of gauges}}} \tag{2}$$

The number of 200 in equation (2) would be different if other thresholds and error statistics are used. On the basis of this formula, the required grid sizes for various numbers of gauges can be derived and the speed comparison between the grid method and triangulation method could be made. In this case study, the computational result shows that the grid method is much faster than the triangulation method for small numbers of gauges. For example, the triangulation method would need almost ten hours for 10,000 simulations with 50 gauges, while the grid method would need just 20 min, which is twenty times faster. However, the speed advantage of the grid method decreases with the increase of gauges, indicating that at a certain level, this speed advantage by the grid method would be lost to the triangulation method. Because of the computer memory problem in this study (out of memory), this critical level has not been simulated. Instead, an extrapolation based on the existing ratio curve could be used to estimate this level. It has been found that a logarithm curve fits very well to the data points and the fitted function is

$$\text{Ratio} = -15.1 \times \ln(\text{number of gauges}) + 80 \tag{3}$$

When the ratio is one, the critical number of gauges would be 187 by equation (3). It should be pointed out that equation (3) has been found based on WME at the Brue catchment and hydrologists with other catchments and measuring indicators should use the methodology described in the paper to find individual critical numbers of gauges. If computation speed is the limiting factor in a project, this critical number of gauges would be very useful for choosing the suitable automated method. For other cases, it would be a compromise to balance the aforementioned three factors.

4. Conclusions

[11] Despite its importance in hydrology and other water resources areas, few studies on computational Thiessen polygons (or automated Thiessen polygons) have been carried out and there is little published information in the hydrological literature. This technical note has proposed two methods that are suitable for hydrologists to implement in water resources projects. The equations derived in the study on the grid size and logarithm ratio curve provide a useful foundation for further work in this area. The two approaches explored have both strengths and weakness in terms of their speed, coding and accuracy. It is important to note that the optimal choice of the computation method should be dependent on the project nature and there is no single method that would suit all application cases. For example, if computation speed is a crucial factor in a project, the adoption of triangulation or grid method would depend on the critical number of gauges. If other factors are used, the decision process would be different. The information from this study is timely needed now since the advancement of computing technology and large rain gauge networks have provide an ideal opportunity for hydrological researchers to carry out data uncertainty analysis of rain gauge networks with a large number of Monte Carlo or bootstrap simulations.

References

Aggarwal, A., L. Guibas, J. Saxe, and P. Shor (1987), A linear time algorithm for computing the Voronoi diagram of a convex polygon, in *Proceedings of the Nineteenth Annual ACM Conference on Theory of Computing*, pp. 39–45, Assoc. for Comput. Mach., New York.

Aurenhammer, F. (1988), Voronoi diagrams—A survey, technical report, Graz Tech. Univ., Graz, Austria.

Chand, R., G. K. Hodlur, M. R. Prakash, N. C. Mondal, and V. S. Singh (2005), Reliable natural recharge estimates in granitic terrain, *Curr. Sci.*, *88*(5), 821–824.

Ge, S. X. (2001), *Modern Flood Forecasting Technology* (in Chinese), Water Publ., Beijing.

Grant, R. H., S. E. Hollinger, K. G. Hubbard, G. Hoogenboom, and R. L. Vanderlip (2004), Ability to predict daily solar radiation values from interpolated climate records for use in crop simulation models, *Agric. For. Meteorol.*, *127*(1–2), 65–75.

Hamlet, A. F., and D. P. Lettenmaier (2005), Production of temporally consistent gridded precipitation and temperature fields for the continental U.S., *J. Hydrometeorol.*, *6*(3), 330–336.

Han, D., T. Kwong, and S. Li (2006), Uncertainties in real time flood forecasting with neural networks, *Hydrol. Processes*, doi:10.1002/hyp.6184, in press.

Klein, R., A. and Lingas (1992), On computing Voronoi diagrams for simple polygons, in *Proceedings of 8th ACM Symposium on Computational Geometry*, pp. 312–319, Assoc. for Comput. Mach., New York.

Lee, D. T., and A. Lin (1986), Generalized Delaunay triangulations for planar graphs, *Discrete Comput. Geom.*, *1*, 201–217.

Martinez, W. L., and A. R. Martinez (2002), *Computational Statistics Handbook With MATLAB*, CRC Press, Boca Raton, Fla.

Mumm, M. (2005), Voronoi diagrams, *Mont. Math. Enthusiast*, *1*(2), 44–55.

Okabe, A., B. Boots, and K. Sugihara (1992), *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*, John Wiley, Hoboken, N. J.

Thiessen, A. H., and J. C. Alter (1911), Climatological data for July, 1911: District No. 10, Great Basin, *Mon. Weather Rev., July*, 1082–1089.

Yang, Z., and D. Han (2006), Derivation of unit hydrograph using a transfer function approach, *Water Resour. Res.*, *42*, W01501, doi:10.1029/2005WR004227.

M. Bray and D. Han, Department of Civil Engineering, University of Bristol, Bristol BS8 1TR, UK. (d.han@bristol.ac.uk)