INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2008; **58**:1101–1133
Published online 27 March 2008 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/fld.1788

A gradient smoothing method (GSM) for fluid dynamics problems

G. R. Liu[1,2] and George X. Xu[3,*,†]

[1]*Centre for Advanced Computations in Engineering Science, Department of Mechanical Engineering, National University of Singapore, 10 Kent Ridge Crescent, Singapore 119260, Singapore*
[2]*Singapore-MIT Alliance (SMA), E4-04-10, 4 Engineering Drive 3, Singapore 117576, Singapore*
[3]*Institute of High Performance Computing, 1 Singapore Science Park Road, #01-01 The Capricorn, Singapore Science Park II, Singapore 117528, Singapore*

SUMMARY

A novel gradient smoothing method (GSM) based on irregular cells and strong form of governing equations is presented for fluid dynamics problems with arbitrary geometries. Upon the analyses about the compactness and the positivity of coefficients of influence of their stencils for approximating a derivative, four favorable schemes (II, VI, VII and VIII) with second-order accuracy are selected among the total eight proposed discretization schemes. These four schemes are successively verified and carefully examined in solving Poisson's equations, subjected to changes in the number of nodes, the shapes of cells and the irregularity of triangular cells, respectively. Numerical results imply us that all the four schemes give very good results: Schemes VI and VIII produce a slightly better accuracy than the other two schemes on irregular cells, but at a higher cost in computation. Schemes VII and VIII that consistently rely on gradient smoothing operations are more accurate than Schemes II and VI in which directional correction is imposed. It is interestingly found that GSM is insensitive to the irregularity of meshes, indicating the robustness of the presented GSM. Among the four schemes of GSM, Scheme VII outperforms the other three schemes, for its outstanding overall performance in terms of numerical accuracy, stability and efficiency. Finally, GSM solutions with Scheme VII to some benchmarked compressible flows including inviscid flow over NACA0012 airfoil, laminar flow over flat plate and turbulent flow over an RAE2822 airfoil are presented, respectively. Copyright © 2008 John Wiley & Sons, Ltd.

Received 5 February 2007; Revised 11 January 2008; Accepted 22 January 2008

KEY WORDS: numerical methods; CFD; FDM; GSM; GSD; stencil

*Correspondence to: George X. Xu, Institute of High Performance Computing, 1 Singapore Science Park Road, #01-01 The Capricorn, Singapore Science Park II, Singapore 117528, Singapore.
†E-mail: xuxg@ihpc.a-star.edu.sg

1. INTRODUCTION

Finite difference method (FDM), finite element method (FEM) and finite volume method (FVM) are the three well-known numerical methods widely used for solving various types of problems in science and engineering. Each of them possesses its own advantages and disadvantages.

FDM is the oldest method among the three, believed to have been introduced by Euler in the 18th century (see [1]). Techniques on FDM were published as early as 1910 by Richardson [2] even before the modern computers were available. FDM is simple and efficient and therefore it is widely adopted in numerical simulations [3–6]. In FDM, a derivative is replaced with an approximate difference formula that can generally be derived from a Taylor series expansion, using single- or multiple-block structured mesh. The strong-form governing equations are discretized onto nodes of a set of structured mesh resulting in a system of algebraic equations with a banded matrix of coefficients. A number of efficient numerical techniques can be used to quickly obtain the solutions for such a system of algebraic equations [6]. However, it is not a trivial process to generate structured mesh for arbitrary geometries, especially in the topological generation of multiple blocks [7]. This limits the FDM being valid only for simple geometries. Although FDM may be used for problems with a slightly complicated geometry, issues related to the mapping from physical domain to computational domain complicate the process in numerical implementation [3, 4].

FEM was earlier used by Courant [8] for solving a torsion problem and was named by Clough [9]. It was studied to great extents and perfected in the 1960s and 1970s, mostly for analyzing structural mechanics problem [10, 11]. The FEM analysis of fluid flow was developed in the mid and late 1970s (see [12, 13]). The classical FEM discretization is based on elements and a piecewise representation of the solution in terms of basis functions and the Galerkin weak form formulation [11, 14]. To improve numerical stability, some more convincing finite element procedures, such as the streamline-upwind Petrov–Galerkin (SUPG) [15], the Galerkin/least-squares (GLS) [16], bubble functions [17] and the variational multiscale method [18], have been developed and widely used nowadays in the finite element community for solving convection-dominated transport problems. The strength of FEM is its ability to deal with arbitrary geometry using different shapes and orders of elements that are usually formed based on a set of unstructured mesh. Generally, FEM can give the highest accuracy on coarse mesh among the three traditional numerical methods. It is very effective for diffusion-dominated problems and free surface problems [13]. However, the coding of FEM is much more complicated than that of FDM, and FEM is relatively costly in computation especially when it is used for solving turbulent fluid flow problems.

As depicted in [1, 19], the FVM is devised to discretize an integral form of the partial differential equation (PDE) for a physical law (e.g. conservation of mass, momentum or energy). The problem domain is divided into a set of finite volumes or cells, and the PDE is expressed in a form that can govern each finite volume (or cell). The resulting system of equations usually involves fluxes of the conserved variable, and thus the vital FVM procedure is how to accurately calculate fluxes [20]. The basic advantage of this method over FDM is that it is applicable for any type of mesh, both structured and unstructured. The resulting approximate solution for field variables can be placed at cell centers or at nodes. The values of field variables at non-storage locations are obtained using interpolation. Further details can be found in [19]. Early well-documented use of FVM was made by Evans and Harlow [21] and Gentry *et al.* [22]. In the late 1970s and the

DOI: 10.1002/fld

early 1980s, it was further explored to be applied onto body-fitted mesh [23, 24]. By the early 1990s, unstructured FVM had been comprehensively developed, e.g. [20, 25–27]. Comparatively, FVM is conservative even on coarse mesh [1]. FVM is now widely adopted for solving fluid flow problems and implemented in the well-known commercial CFD packages [19]. Similar to FDM and FEM, FVM also has some disadvantages. As addressed by Blazek [20], false diffusion often occurs in numerical predictions with FVM, especially when simple numerics are engaged. It is also difficult to develop schemes with higher than second-order accuracy for multidimensional problems [27].

More recently, various meshfree methods have been developed (see, for example, References [28–34]). Comparatively, meshfree methods are implemented on nodes at first place instead of meshes as done in traditional methods described above. A survey paper written by Babuska *et al.* [34] provides the mathematical foundation of various meshfree methods. The overview of theoretical, computational and implementation issues related to various meshfree methods about both weak and strong forms can be found in the monographs by Liu [30] and Liu and Gu [35]. By removing some of the restrictions owing to the use of elements, meshfree methods are more flexible and more suitable for adaptive analyses. They have been successfully applied to problems where element-based methods are often difficult to give satisfactory results. For example [35–37], meshfree methods are very attractive in the simulation of cracks, underwater shock, explosion and free surface problems. In general, many meshfree methods, as mentioned above, are comprehensively studied based on weak-form governing equations. When governing equations in strong form are treated, most of these meshfree methods exhibit instability [38, 39], and thus special techniques are required to resolve it. Gradient smoothing operation is often employed in stabilizing the nodal integrations [40, 41] and installing linear conformability [42] for methods based on Galerkin weak forms.

Enlightened by the attractive merits of gradient smoothing operation in Galerkin weak forms as mentioned above, we here intend to examine the effectiveness of gradient smoothing operation as it is applied for strong-form (differential form) governing equations. We call the method as gradient smoothing method (GSM). In GSM, all the unknowns are stored at nodes and their derivatives at various locations are consistently and directly approximated with gradient smoothing operation based on relevant gradient smoothing domains (GSDs). Both regular and irregular grids are concerned in the development of GSM. A unified and stable scheme regardless of the type of grid is pursued in this study. As a whole, the implementation procedure of GSM is as simple as the traditional FDM. GSM can, however, be applied very effectively with excellent stability for arbitrary geometries, as will be shown in our studies in this work.

In the following sections, the theory of GSM is first introduced. The GSM approximations to the gradients (first-order derivative) and Laplace operator (second-order derivative) of a field variable are presented in detail. Stencil analyses on coefficients of influence corresponding to various GSM schemes are then conducted. Important features of stencils for the discretized Laplace operator are discussed. Numerical solutions to Poisson's equations are obtained using four favorable GSM schemes and investigated in detail to reveal the properties on convergence and stability. The computational efficiency, accuracy in results and the robustness to the cell irregularity for GSM are also comprehensively examined. Finally, the GSM results for some benchmarked compressible flow problems, e.g. inviscid flow over a NACA0012 airfoil, laminar flow over flat plate and turbulent flow over an RAE2822 airfoil, are presented. The convergence and accuracy of the GSM solutions are further demonstrated.

DOI: 10.1002/fld

2. GRADIENT SMOOTHING METHOD

In GSM, derivatives at various locations, including nodes, centroids of cells and midpoints of cell edges, are approximated over relevant GSDs using the gradient smoothing operation. The details about the theory, principle and implementation procedure of GSM are introduced in this section with the focus on the approximation of spatial derivatives.

2.1. Gradient smoothing operation

For simplicity, a two-dimensional problem is considered here to illustrate the gradient smoothing operation. The gradients of a field variable U at a point of interest at $\mathbf{x}_i$ in domain Ω_i can be approximated in the form of [43, 44]

$$\nabla U_i \equiv \nabla U(\mathbf{x}_i) \approx \int_{\Omega_i} \nabla U(\mathbf{x})\, \widehat{w}(\mathbf{x}-\mathbf{x}_i)\, \mathrm{d}\mathbf{x} \tag{1}$$

Integrating Equation (1) by parts gives

$$\nabla U_i \approx \int_{\partial\Omega_i} U(\mathbf{x})\, \widehat{w}(\mathbf{x}-\mathbf{x}_i)\, \vec{n}\, \mathrm{d}s - \int_{\Omega_i} U(\mathbf{x}) \nabla \widehat{w}(\mathbf{x}-\mathbf{x}_i)\, \mathrm{d}V \tag{2}$$

where ∇ is a gradient operator and $\widehat{w}$ is a smoothing function that can be chosen properly based on the requirement on the accuracy of the approximation [43]. $\partial\Omega_i$ represents the external boundary of the GSD, and $\vec{n}$ denotes the unit normal vector on $\partial\Omega_i$, as shown in Figure 1.

In this work, we use the following piecewise constant function as the smoothing function:

$$\widehat{w} = \begin{cases} 1/V_i, & \mathbf{x} \in \Omega_i \\ 0, & \mathbf{x} \notin \Omega_i \end{cases} \tag{3}$$

where V_i is the area of Ω_i.

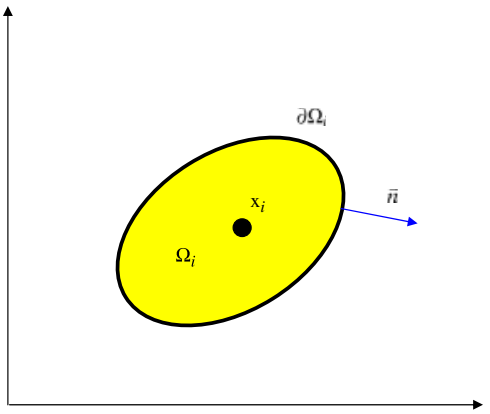

Figure 1. Generic gradient smoothing domain.

DOI: 10.1002/fld

Using Equation (3), the right-hand side second term in Equation (2) vanishes and Equation (2) then becomes

$$\nabla U_i \approx \frac{1}{V_i} \oint_{\partial\Omega_i} U \vec{n} \, \mathrm{d}s \tag{4}$$

Equation (4) gives an approximation of gradients at a point using a local smoothing domain. Such a gradient smoothing technique is often adopted in many meshfree methods, for example, in the widely used smoothed particle method [28, 43], to stabilize the nodal integrations and ensure linear conformability [40, 41, 45, 46]. It has also been used in the development of the smoothed FEM [47, 48].

Analogously, by successively applying the gradient smoothing technique for second-order derivatives [47, 48], the Laplace operator at $\mathbf{x}_i$ can be approximated as

$$\nabla \cdot (\nabla U_i) \approx \frac{1}{V_i} \oint_{\partial\Omega_i} \vec{n} \cdot \nabla U \, \mathrm{d}s \tag{5}$$

Hence, spatial derivatives at any point of interest can be approximated using Equations (4) and (5) together with properly defined smoothing domains that are discussed in the following subsection.

2.2. *Smoothing domains*

In our GSM, the problem domain is first divided into a set of cells (regular or irregular) generated by connecting nodes, and the values of the field functions are stored at nodes and a set of systems of algebraic equations is formed by approximating the derivatives at nodes. Based on primitive cells, a smooth domain can be constructed for any point. Depending on the location of the point of interest, we use different types of smoothing domains based on the compact and conformal principle. As shown in Figure 2, three types of GSDs, which are used for the approximation of spatial derivatives, are constituted on primitive unstructured triangular cells. The first type of smoothing domain is the node-associated GSD (nGSD) for the approximation of the derivatives at a node of interest. It is formed by connecting relevant centroids of triangles with midpoints of relevant cell edges. The second type is formed by a primitive cell, which is used for approximating the derivatives at the centroid of the cell, as in cell-centered FVM. It is called centroid-associated GSD (cGSD) here. The third type is named midpoint-associated GSD (mGSD), which is used for the calculation of the gradients at the midpoint of a cell edge of interest. The preferred mGSD is formed by connecting the end nodes of the cell edge with the centroids on both sides of the cell edge, as shown in Figure 2.

Using Equations (4) and (5), spatial derivatives at any point of interest can be approximated based on the corresponding smoothing domain described above. Different schemes are devised for this purpose, and the details will be elucidated further in the following subsection.

2.3. *Spatial discretization schemes*

We now need to accurately predict the integrals along the boundaries of various types of GSDs. In the current work, both the one-point quadrature (rectangular rule) and the two-point quadrature (trapezoidal rule) are used and examined for the approximation of the derivatives at nodes. Only two-point quadrature is used for the approximation of the gradients at both midpoints and centroids. As listed in Table I, a total of eight discretization schemes for spatial differential terms are developed using different types of quadrature and methods of approximation.

DOI: 10.1002/fld

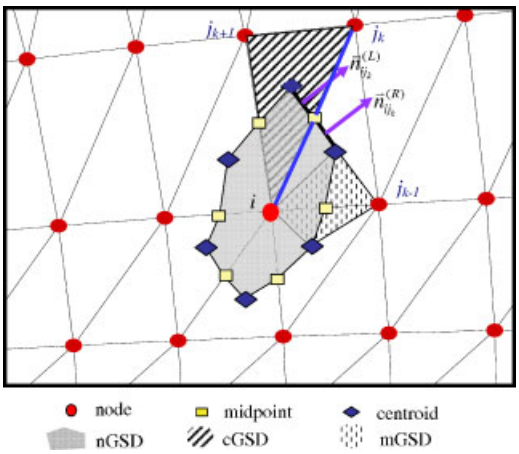

Figure 2. Illustration of gradient smoothing domains and domain-edge vectors adopted in GSM.

Table I. Spatial discretization schemes for the approximation of derivatives.

Scheme	Quadrature	Type of GSD	Approximation of gradients at midpoints	Approximation of gradients at centroids	Directional correction
I	One-point	nGSD	Interpolation	(Not required)	No
II	One-point	nGSD	Interpolation	(Not required)	Yes
III	Two-point	nGSD	Interpolation	Interpolation	No
IV	Two-point	nGSD	Interpolation	Interpolation	Yes
V	Two-point	nGSD, cGSD	Interpolation	Gradient smoothing	No
VI	Two-point	nGSD, cGSD	Interpolation	Gradient smoothing	Yes
VII	One-point	nGSD, mGSD	Gradient smoothing	(Not required)	No
VIII	Two-point	nGSD, mGSD, cGSD	Gradient smoothing	Gradient smoothing	No

In the one-point quadrature schemes (I, II and VII), the integration along a smoothing domain edge is approximated with the rectangular rule, where the integrand is evaluated only at the midpoint of a cell edge of interest. In the two-point quadrature schemes (III–VI and VIII), the integration is evaluated with the trapezoidal rule, where values of the field variable and its gradients at the two endpoints of a smoothing domain edge of interest (the midpoint of the cell edge and the centroid of the cGSD that are connected to the domain edge of interest) are used. In this work, both first- and second-order derivatives at nodes are always approximated with the gradient smoothing operation detailed in Section 2.1. The gradients at the midpoint of a cell edge of interest can be calculated in two ways: either by simple interpolation using the gradients at both end nodes of the cell edge (I–VI) or by gradient smoothing technique using Equation (4) based on mGSD (VII and VIII) for the midpoint. Similarly, the gradients at a centroid can be obtained either by simple interpolation using the gradients at the nodes of the cGSD (III and IV) or by gradient smoothing over the corresponding cGSD (V, VI and VIII).

DOI: 10.1002/fld

Note that when one-point quadrature schemes are used, there is no need to approximate the gradients at centroids, since the integrands in Equations (4) and (5) are evaluated only at the midpoints of cell edges.

2.4. Formulae for spatial derivatives

2.4.1. Two-point quadrature schemes. In GSM, spatial derivatives at any point of interest are approximated using the following formulae based on primitive triangular cells.

• *First-order derivatives at nodes*:

Using Equation (4), at node i, the first-order derivatives of the field variable U are given by

$$\frac{\partial U_i}{\partial x} \approx \frac{1}{V_i} \sum_{k=1}^{n_i} \{\tfrac{1}{2}(\Delta S_x)_{ij_k}^{(L)}[(U_\mathrm{m})_{ij_k} + (U_\mathrm{c})_{\Delta ij_k j_{k+1}}] + \tfrac{1}{2}(\Delta S_x)_{ij_k}^{(R)}[(U_\mathrm{m})_{ij_k} + (U_\mathrm{c})_{\Delta ij_k j_{k-1}}]\} \tag{6}$$

$$\frac{\partial U_i}{\partial y} \approx \frac{1}{V_i} \sum_{k=1}^{n_i} \{\tfrac{1}{2}(\Delta S_y)_{ij_k}^{(L)}[(U_\mathrm{m})_{ij_k} + (U_\mathrm{c})_{\Delta ij_k j_{k+1}}] + \tfrac{1}{2}(\Delta S_y)_{ij_k}^{(R)}[(U_\mathrm{m})_{ij_k} + (U_\mathrm{c})_{\Delta ij_k j_{k-1}}]\} \tag{7}$$

where

$$(\Delta S_x)_{ij_k}^{(L)} = \Delta S_{ij_k}^{(L)} (n_x)_{ij_k}^{(L)} \tag{8}$$

$$(\Delta S_y)_{ij_k}^{(L)} = \Delta S_{ij_k}^{(L)} (n_y)_{ij_k}^{(L)} \tag{9}$$

$$(\Delta S_x)_{ij_k}^{(R)} = \Delta S_{ij_k}^{(R)} (n_x)_{ij_k}^{(R)} \tag{10}$$

$$(\Delta S_y)_{ij_k}^{(R)} = \Delta S_{ij_k}^{(R)} (n_y)_{ij_k}^{(R)} \tag{11}$$

where U, U_m and U_c denote the values of the field variable U at nodes, midpoints of cell edges and centroids of triangular cells, respectively. ΔS_x and ΔS_y are the two components of a domain-edge vector. n_x and n_y represent the two components of the unit normal vector of the domain edge. i denotes the node of interest and j_k is the other end node of the cell edge linked to node i (see Figure 2). Superscripts (L) and (R) are pointers to the two domain edges associated with the cell edge of interest, ij_k. The total number of supporting nodes within the stencil of node i is denoted by n_i. Subscripts $\Delta ij_k j_{k+1}$ and $\Delta ij_k j_{k-1}$ stand for the left-hand side and right-hand side triangles of the cell edge ij_k. n_i denotes the total number of supporting nodes connected to node i. These geometrical parameters are calculated and stored before the intensive calculation starts. The values of the field variable U at non-storage locations, i.e. at midpoints and centroids, are computed by simple interpolation of function values at constitutive nodes, respectively.

• *First-order derivatives at midpoints and centroids*:

Analogous to the discretization at nodes described above, the gradients at the midpoints $((\nabla U_\mathrm{m})_{ij_k})$ of cell edges and centroids $((\nabla U_\mathrm{c})_{\Delta ij_k j_{k+1}}$ and $(\nabla U_\mathrm{c})_{\Delta ij_k j_{k-1}})$ of cells can also be approximated with the gradient smoothing technique using Equation (4) but based on the related mGSDs and cGSDs, respectively. Such a treatment is adopted for the approximation of the gradients at midpoints in Schemes VII and VIII, and gradients at centroids in Schemes V–VIII. Similarly, the geometrical parameters including the areas, domain-edge vectors and normal vectors of domain edges related to mGSDs and cGSDs should be predetermined and stored for use in the iterative process of solving the algebraic equations.

DOI: 10.1002/fld

Alternatively, the gradients at these non-storage locations can be approximated by simple interpolation of the gradients at relevant nodes. In Schemes I–VI, the gradients at the midpoints of the cell edges are approximated in this manner, so are the gradients at centroids in Schemes III and IV.

• *Second-order derivatives at nodes*:

In GSM with the two-point quadrature schemes, the second-order derivatives are obtained using Equation (5) and given in the following form:

$$\begin{aligned}\frac{\partial^2 U_i}{\partial x^2}+\frac{\partial^2 U_i}{\partial y^2} \approx \frac{1}{\Omega_i}\sum_{k=1}^{n_i}\frac{1}{2}\Bigg\{&\left[\frac{\partial}{\partial x}(U_{\mathrm{m}})_{ij_k}+\frac{\partial}{\partial x}(U_{\mathrm{c}})_{\Delta i j_k j_{k+1}}\right](\Delta S_x)_{ij_k}^{(L)}\\ &+\left[\frac{\partial}{\partial y}(U_{\mathrm{m}})_{ij_k}+\frac{\partial}{\partial y}(U_{\mathrm{c}})_{\Delta i j_k j_{k+1}}\right](\Delta S_y)_{ij_k}^{(L)}\\ &+\left[\frac{\partial}{\partial x}(U_{\mathrm{m}})_{ij_k}+\frac{\partial}{\partial x}(U_{\mathrm{c}})_{\Delta i j_k j_{k-1}}\right](\Delta S_x)_{ij_k}^{(R)}\\ &+\left[\frac{\partial}{\partial y}(U_{\mathrm{m}})_{ij_k}+\frac{\partial}{\partial y}(U_{\mathrm{c}})_{\Delta i j_k j_{k-1}}\right](\Delta S_y)_{ij_k}^{(R)}\Bigg\}\end{aligned} \tag{12}$$

All the first-order derivatives used in Equation (12) are approximated as described in previous two subsections.

2.4.2. One-point quadrature schemes. In one-point quadrature schemes (I, II and VII), it is assumed that

$$(U_{\mathrm{c}})_{\Delta i j_k j_{k+1}}=(U_{\mathrm{c}})_{\Delta i j_k j_{k-1}}=(U_{\mathrm{m}})_{ij_k} \tag{13}$$

and

$$(\nabla U_{\mathrm{c}})_{\Delta i j_k j_{k+1}}=(\nabla U_{\mathrm{c}})_{\Delta i j_k j_{k-1}}=(\nabla U_{\mathrm{m}})_{ij_k} \tag{14}$$

Therefore, Equations (6), (7) and (12) can be simplified as

$$\frac{\partial U_i}{\partial x}\approx\frac{1}{\Omega_i}\sum_{k=1}^{n_i}(\Delta S_x)_{ij_k}(U_{\mathrm{m}})_{ij_k} \tag{15}$$

$$\frac{\partial U_i}{\partial y}\approx\frac{1}{\Omega_i}\sum_{k=1}^{n_i}(\Delta S_y)_{ij_k}(U_{\mathrm{m}})_{ij_k} \tag{16}$$

and

$$\nabla\cdot(\nabla U)\approx\frac{1}{\Omega_i}\sum_{k=1}^{n_i}\left[\frac{\partial}{\partial x}(U_{\mathrm{m}})_{ij_k}(\Delta S_x)_{ij_k}+\frac{\partial}{\partial y}(U_{\mathrm{m}})_{ij_k}(\Delta S_y)_{ij_k}\right] \tag{17}$$

where

$$(\Delta S_x)_{ij_k}=(\Delta S_x)_{ij_k}^{(L)}+(\Delta S_x)_{ij_k}^{(R)} \tag{18}$$

$$(\Delta S_y)_{ij_k}=(\Delta S_y)_{ij_k}^{(L)}+(\Delta S_y)_{ij_k}^{(R)} \tag{19}$$

DOI: 10.1002/fld

As shown in Equation (15)–(17), in one-point quadrature schemes, only values for the field variable and its gradients at the midpoints of cell edges are needed in the approximations. Thus, the vectors for a pair of domain edges connected with the cell edge ij_k can be lumped together, which in return reduces the storage space for geometrical parameters.

It should be noted that the gradients at midpoints in Equation (17) are approximated using the gradient smoothing technique based on mGSDs in Scheme VII, whereas they are approximated using the simple interpolation approach in Schemes I and II.

The one-point quadrature schemes are clearly simpler and much more cost-effective in terms of both flops and storage. Schemes based on two-point quadrature impose extra requirements in computation and storage for values of variables at centroids and domain-edge vectors for cGSDs. When mGSDs are used for prediction of gradients at the midpoints of cell edges, such demands become even higher. However, theoretically, schemes based on two-point quadrature can give more accurate results. This will be verified and discussed later in the section on numerical examples.

2.4.3. Directional correction. In our study, we find that as the gradients at the midpoints of cell edges, which are approximated using simple interpolation approach, are used for approximating the second-order derivatives, decoupling solutions (checkerboard problem) are attained. To circumvent such a problem, as proposed by Crumpton *et al.* [49], the approximated gradients at midpoints are required to be remedied with the directional correction technique. The directional correction takes the form of

$$(\nabla \tilde{U}_{\mathrm{m}})_{ij_k} = (\nabla U_{\mathrm{m}})_{ij_k} - \left[(\nabla U_{\mathrm{m}})_{ij_k} \cdot \vec{t}_{ij_k} - \left(\frac{\partial U}{\partial l} \right)_{ij_k} \right] \vec{t}_{ij_k} \tag{20}$$

where

$$\left(\frac{\partial U}{\partial l} \right)_{ij_k} \approx \frac{U_{j_k} - U_i}{\Delta l_{ij_k}}, \quad \vec{t}_{ij_k} = \frac{\vec{r}_{ij_k}}{\Delta l_{ij_k}}, \quad \vec{r}_{ij_k} = \mathbf{x}_{j_k} - \mathbf{x}_i$$

and

$$\Delta l_{ij_k} = |\mathbf{x}_{j_k} - \mathbf{x}_i|$$

Here $\mathbf{x}_i$ and $\mathbf{x}_{j_k}$ denote the spatial locations of nodes i and j_k, respectively. This technique is adopted in Schemes II, IV and VI. Details about the role of directional correction will be addressed in the following section on stencil analyses.

2.5. Discretization of temporal terms

For a transient or pseudo-transient problem, the governing equation can be rewritten in the form of

$$\frac{\partial U}{\partial t} = -R \tag{21}$$

DOI: 10.1002/fld

where R represents the residual dependent on the field variable U and its derivatives. In this study, the temporal term $(\partial U/\partial t)$ is discretized with the following explicit five-stage Runge–Kutta ($RK5$) method [50]:

$$\begin{aligned} U_i^{(0)} &= U_i^n \\ U_i^{(1)} &= U_i^{(0)} - \alpha_1 \Delta t\, R_i^{(0)} \\ U_i^{(2)} &= U_i^{(0)} - \alpha_2 \Delta t\, R_i^{(1)} \\ U_i^{(3)} &= U_i^{(0)} - \alpha_3 \Delta t\, R_i^{(2)} \\ U_i^{(4)} &= U_i^{(0)} - \alpha_4 \Delta t\, R_i^{(3)} \\ U_i^{(5)} &= U_i^{(0)} - \alpha_5 \Delta t\, R_i^{(4)} \end{aligned} \tag{22}$$

where residual $R_i^{(k)}$ is evaluated with the values of the field function and its derivatives approximated with the kth-stage RK solution at node i for every time step. Δt denotes the time step, and the coefficients adopted in this study are $\alpha_1 = 0.0695$, $\alpha_2 = 0.1602$, $\alpha_3 = 0.2898$, $\alpha_4 = 0.5060$ and $\alpha_5 = 1.000$.

With the *RK5* method, only the 0th- and 5th-stage solutions at nodes should be stored in the memory. The *RK5* method has been widely used in the simulations of many transient fluid flow problems, because of its satisfactory efficiency and stability.

2.6. *Edge-based data structure*

The edge-based data structure recommended by Lohner [51] is adopted in our GSM, together with the scatter–gather approach as described by Barth [26, 27]. The linkages among cell edges, nodes, cells, GSDs, as well as domain-edge vectors and normals, are pre-determined. Our results prove that GSM with the edge-based data structure is very efficient.

3. ANALYSES OF DISCRETIZATION STENCIL

Before conducting intensive numeric excises, careful studies of the stencils of supporting nodes for various schemes proposed for GSM are carried out. The coefficients of influence for a node of interest where derivatives are approximated are derived using GSM and analyzed. The objectives for stencil analyses are to select the most suitable schemes that satisfy the basic principles of numerical discretization. For clarity, the stencils for approximating the Laplace operator based on the cells in both square and equilateral triangle shapes are focused in this study.

3.1. *Basic principles for stencil assessment*

In the stencil analyses, the following five basic rules are considered to assess the quality of a stencil resulting from a discretization scheme:

(a) Consistency at each interface of two adjacent GSDs.
(b) Positivity of coefficients of influence.
(c) Sum of the coefficients of influence.

DOI: 10.1002/fld

(d) Negative-slope linearization of the source term.
(e) The compactness of the stencil.

The first four rules are summarized by Patankar [52] with consideration of solutions with physically realistic behavior and overall balance. To satisfy Rule (a), it requires that the same expression of approximation must be used on the interface of two adjacent GSDs. Rule (b) requires that the coefficient for the node of interest and the coefficients of influence must be positive, once the discretization equation is expressed in the form of $a_{ii}U_i = \sum_{k=1}^{n_i} a_{ij_k}U_{j_k} + b_i$. Rule (c) further implies that $a_{ii} = \sum_{k=1}^{n_i} a_{ij_k}$. Rule (d) relates to the treatment of the source terms. As addressed by Patankar [52], it is essential to keep the slope of linearization to be negative, since a positive slope can lead to computational instabilities and physically unrealistic solutions. It is also necessary for a good discretization stencil to satisfy Rule (e) for the concerns about numerical accuracy and efficiency, as commented by Barth [27]. The very first layer of nodes surrounding the node of interest should be included in the discretization stencil. Moreover, as the stencil becomes larger, not only the computational cost increases, but eventually the accuracy decreases as less valid data from further away are brought into approximation.

In addition, Barth [27] has proposed a few lemmas to address the necessity of positivity of coefficients to satisfy a discrete maximum principle that is a key tool in the design and analysis of numerical schemes suitable for non-oscillatory discontinuity (for example, shock). At steady state, non-negativity of the coefficients becomes sufficient to satisfy a discrete maximum principle that can be applied successively to obtain the global maximum principle and stable results. His statements reiterate the importance of Rule (b) as mentioned by Patankar [52].

In GSM, when the gradient smoothing technique is applied to GSDs, Rule (a) is automatically satisfied, meaning that the local conservation of quantities is ensured and therefore for the global conservation once proper boundary conditions are used. In this study, Rules (b), (c) and (e) focus on the assessment of stencils for different discretization schemes.

3.2. Stencils for approximated gradients

The stencils for gradient approximation using the eight types of discretization schemes are derived. The coefficients of influence based on the cells in square and equilateral triangle shapes with unit length are shown in Figures 3 and 4.

3.2.1. On cells in square shape. We find that the three one-point quadrature schemes (I, II and VII) give the same stencil when cells in square shape are used, as shown in Figure 3(a). This stencil is also identical to that of the two-point-based central-differencing scheme in FDM. We also observe that the stencil for all two-point quadrature schemes is also the same, as shown in Figure 3(b). This stencil is also identical to that of the six-point-based central-differencing scheme in FDM. Such findings confirm that when cells in square shape are used, GSM is identical to FDM. However, GSM works for cells in irregular shapes.

3.2.2. On cells in equilateral triangle shape. It is interesting to know that the stencil for approximated gradients based on the cells in equilateral triangle shape is found to be the same regardless of the GSM schemes, as shown in Figure 4. This stencil is identical to that of the interpolation method using six surrounding nodes [30]. Note that for irregular triangular cells, the interpolation method can in general fail as addressed by Liu [30], but our GSM still performs well, as will

DOI: 10.1002/fld

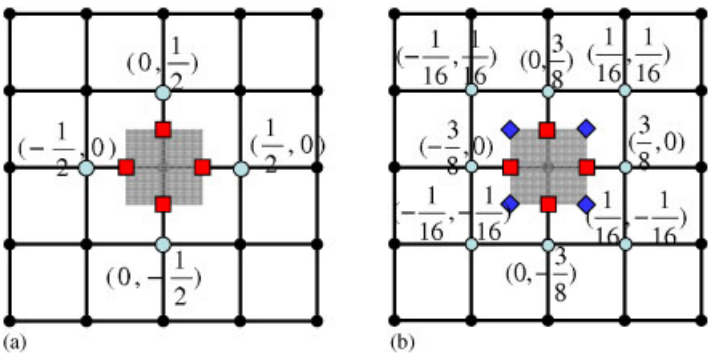

Figure 3. Stencils for approximated gradients based on the cells in square shape: (a) I, II and VII and (b) III–VI and VIII.

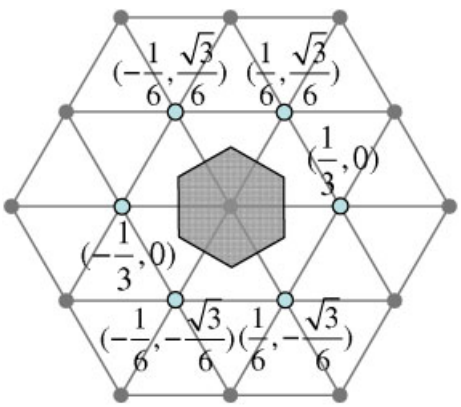

Figure 4. The stencil for approximated gradients based on the cells in equilateral triangle shape (identical for I–VIII).

be demonstrated in the section on numerical examples. This is because of the crucial stability provided by the smoothing operation.

3.3. Stencils for approximated Laplace operator

3.3.1. On cells in square shape. The stencils for the approximated Laplace operator with GSM schemes based on the cells in square shape are derived and listed in Figure 5. Three schemes, I, III and V, as shown in Figures 5(a), (c) and (e), result in wide stencils with unfavorable weighting coefficients (zero and negative) on cells in square shape. With such kinds of stencils, unexpected decoupling solutions may be produced [20]. This is also confirmed in the present analysis and will be illustrated in the following section on numerical examples. As mentioned by Moinier [50], such kinds of stencils cannot damp out high-frequency numerical errors. The situation becomes even

DOI: 10.1002/fld

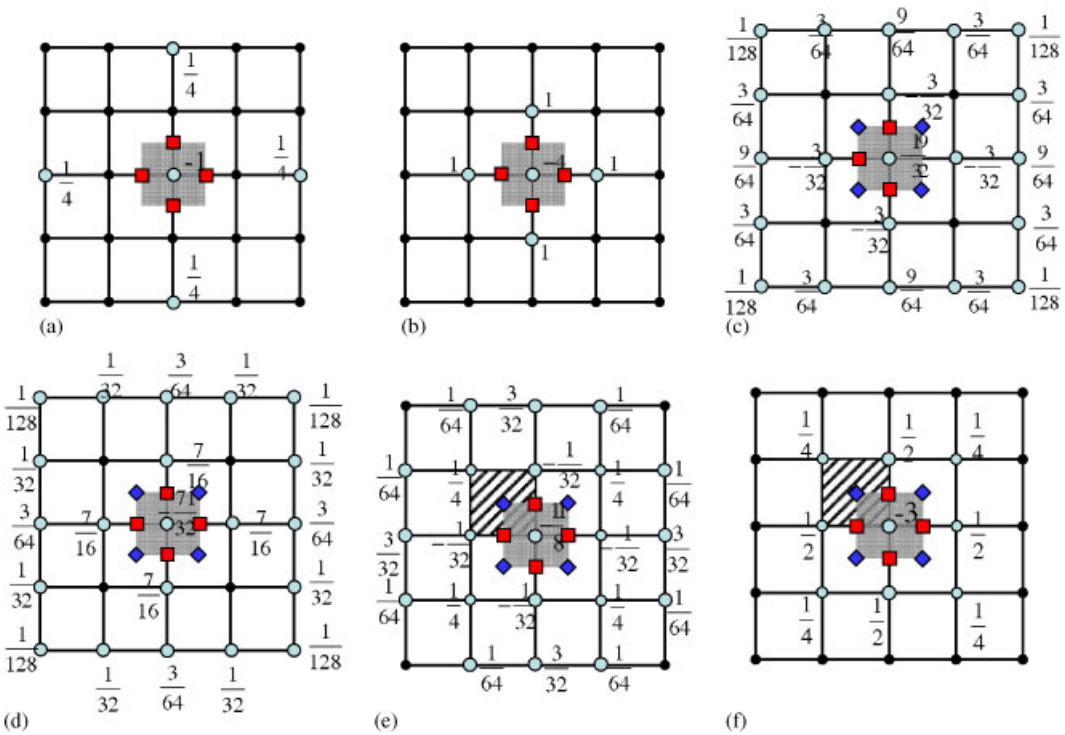

Figure 5. Stencils for the approximated Laplace operator on the cells in square shape: (a) I; (b) II and VII; (c) III; (d) IV; (e) V; and (f) VI and VIII.

worse within the boundary layer region where the viscous effect becomes dominant. Therefore, Schemes I, III and V are now labeled as unfavorable.

It is found that such unfavorable stencils are resulted from the simple interpolation that is used to approximate the gradients at the midpoints of cell edges in the three schemes. In Schemes II and VI where the directional correction to the approximated gradients at midpoints is made, relatively compact stencils with favorable coefficients are obtained, as depicted in Figures 5(b) and (f). Scheme II is a five-point-based stencil and Scheme VI corresponds to a 9-node-based compact stencil, which are the same as those for central-difference scheme in FDM. However, as shown in Figure 5(d), even with directional correction, an unfavorable stencil can still occur in Scheme IV. Therefore, Schemes II and VI are found to be favorable, and Scheme IV is also labeled as unfavorable.

The compact and favorable stencils are also obtained using Schemes VII and VIII where the gradients at the midpoints of cell edges are approximated by applying the gradient smoothing operation to mGSDs, as shown in Figures 5(b) and (f). This implies that for approximating the gradients at the midpoint of a cell edge of interest, the gradient smoothing operation over the corresponding mGSD is a good alternative to the simple interpolation with directional correction technique. From the points of views of the consistency in the approximation of derivatives at different locations and the stability feature of the gradient smoothing technique, Schemes VII and VIII are more preferable than Schemes II and VI.

DOI: 10.1002/fld

3.3.2. On cells in equilateral triangle shape. The analyses are also conducted on cells in equilateral triangle shape and resulted stencils are shown in Figure 6. It is found again that Schemes I and III result in an unfavorable stencil, because the coefficients of influence at the very first layer of neighboring nodes are negative as shown in Figure 6(a), which violates the basic Rule (b). Schemes IV and V, and Schemes II, VI, VII and VIII produce two sets of stencils with favorable coefficients, as shown in Figures 6(b) and (c), respectively. The stencil for Schemes IV and V consists of two layers of surrounding points around the node of interest. The stencil for Schemes II, VI, VII and VIII uses only the first layer of neighboring nodes. According to Rule (e), Schemes II, VI, VII and VIII are more favorable than Schemes IV and V, because of their relatively compact stencils.

It should be mentioned that Rule (c) is satisfied in stencils for all schemes studied here. In summary, based on the stencil analyses, four GSM schemes, i.e. II, VI, VII and VIII, are selected as favorable schemes and are further examined, because they consistently produce compact stencil with favorable coefficients on both shapes of cells of concern (square and equilateral triangle).

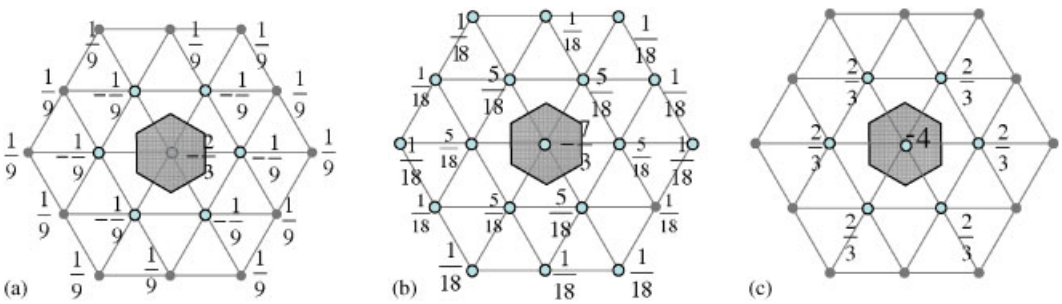

Figure 6. Stencils for the approximated Laplace operator on the cells in equilateral triangle shape: (a) I and III; (b) IV and V; and (c) II, VI, VII and VIII.

Table II. Truncation errors in the approximation of first derivatives in GSM.

Scheme	Shape of cells	Truncation error
II and VII	Square	$O_x(h^2)=-\frac{h^2}{6}\frac{\partial^3 U_{ij}}{\partial x^3}+O(h^3)$
		$O_y(h^2)=-\frac{h^2}{6}\frac{\partial^3 U_{ij}}{\partial y^3}+O(h^3)$
VI and VIII	Square	$O_x(h^2)=-h^2\left(\frac{5}{24}\frac{\partial^3 U_{ij}}{\partial x^3}+\frac{1}{2}\frac{\partial^3 U_{ij}}{\partial x\partial y^2}\right)+O(h^3)$
		$O_y(h^2)=-h^2\left(\frac{5}{24}\frac{\partial^3 U_{ij}}{\partial y^3}+\frac{1}{2}\frac{\partial^3 U_{ij}}{\partial x^2\partial y}\right)+O(h^3)$
II, VI, VII and VIII	Equilateral triangle	$O_x(h^2)=-h^2\left(\frac{1}{24}\frac{\partial^3 U_i}{\partial x^3}+\frac{1}{8}\frac{\partial^3 U_i}{\partial x\partial y^2}\right)+O(h^3)$
		$O_y(h^2)=-h^2\left(\frac{1}{24}\frac{\partial^3 U_i}{\partial y^3}+\frac{1}{8}\frac{\partial^3 U_i}{\partial x^2\partial y}\right)+O(h^3)$

DOI: 10.1002/fld

Table III. Truncation errors in the approximation of the Laplace operator in GSM.

Scheme	Shape of cells	Truncation error
II and VII	Square	$O(h^2)=-\frac{h^2}{12}\left(\frac{\partial^4 U_{ij}}{\partial x^4}+\frac{\partial^4 U_{ij}}{\partial y^4}\right)+O(h^3)$
VI and VIII	Square	$O(h^2)=-\frac{h^2}{12}\left(\frac{\partial^4 U_{ij}}{\partial x^4}+3\frac{\partial^4 U_{ij}}{\partial x^2\partial y^2}+\frac{\partial^4 U_{ij}}{\partial y^4}\right)+O(h^3)$
II, VI, VII and VIII	Equilateral triangle	$O(h^2)=-\frac{h^2}{16}\left(\frac{\partial^4 U_i}{\partial x^4}+2\frac{\partial^4 U_i}{\partial x^2\partial y^2}+\frac{\partial^4 U_i}{\partial y^4}\right)+O(h^3)$

3.4. Truncation errors

We next conduct analyses on truncation errors in the approximation of first- and second-order derivatives with the four recommended schemes in GSM, and the results are summarized in Tables II and III. They are derived based on the cells in square and equilateral triangle shapes, respectively. It is clear that all these schemes are of second-order accuracy. The truncation errors for Scheme VII are identical to those for Scheme II, and Schemes VIII and VI have the same truncation errors. All these theoretical findings will be further conformed when these schemes are used to solve Poisson's equations.

4. APPLICATION AND VALIDATION OF GSM

Through the stencil analyses for all the schemes, four schemes (II, VI, VII and VIII) are selected to use in the GSM, because of their compact stencils with favorable coefficients. Numerical tests for solution to two-dimensional Poisson's equations are consequently conducted using our GSM code. Different spatial discretization schemes are tested and compared with one another in terms of numerical accuracy and computational efficiency. The roles of directional correction and gradient smoothing technique used for the approximation of the gradients at the midpoints of cell edges are numerically verified. In addition, the effects of the shape, the density and the irregularity of cells on the accuracy and stability are intensively investigated. Finally, GSM is successfully used to predict compressible flow over an RAE2822 airfoil.

4.1. Solutions to Poisson's equations

4.1.1. Governing equations. Poisson's equations for a square computational domain are first solved with our GSM code. Poisson's equations govern many physical problems, such as the heat conduction problems with sources. Two problems with variations in source and boundary conditions are studied. In this study, the Dirichlet conditions are applied on the boundaries, i.e. the values of the field functions at the boundaries are prescribed. The pseudo-transient approach is adopted for pursuing steady-state solutions.

The governing equations under investigation take the following form:

$$\frac{\partial U}{\partial t}=\frac{\partial^2 U}{\partial x^2}+\frac{\partial^2 U}{\partial y^2}-f(x,y)\quad(0\leqslant x\leqslant 1, 0\leqslant y\leqslant 1) \tag{23}$$

DOI: 10.1002/fld

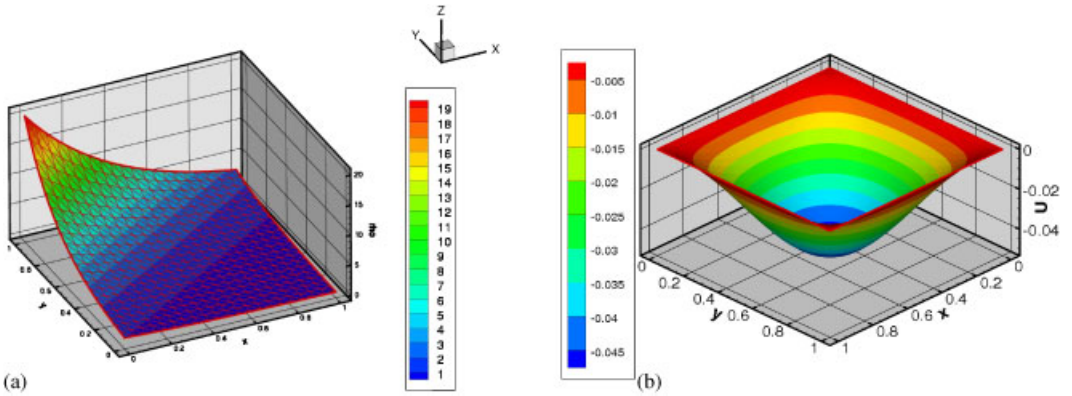

Figure 7. Contour plots of exact solutions to the two Poisson problems: (a) the first Poisson problem and (b) the second Poisson problem.

For the first problem, the source and initial conditions are prescribed as

$$\left.\begin{aligned} f(x,y,t) &= 13\exp(-2x+3y) \\ U(x,y,0) &= 0 \end{aligned}\right\}, \quad (0\leqslant x\leqslant 1, 0\leqslant y\leqslant 1) \tag{24}$$

As plotted in Figure 7(a), the analytical solution to this problem is known as

$$\hat{U}(x,y) = \mathrm{e}^{(-2x+3y)} \quad (0\leqslant x\leqslant 1, 0\leqslant y\leqslant 1) \tag{25}$$

For the second Poisson problem, the source, initial conditions and analytical solution are given as follows:

$$\left.\begin{aligned} f(x,y,t) &= \sin(\pi x)\sin(\pi y) \\ U(x,y,0) &= 0 \end{aligned}\right\}, \quad (0\leqslant x\leqslant 1, 0\leqslant y\leqslant 1) \tag{26}$$

$$\hat{U}(x,y) = -\frac{1}{2\pi^2}\sin(\pi x)\sin(\pi y) \quad (0\leqslant x\leqslant 1, 0\leqslant y\leqslant 1) \tag{27}$$

The contour plot of the analytical solution to the second problem is shown in Figure 7(b). Boundary conditions adopted in simulations for both problems are consistent with the corresponding analytical solutions. These analytical solutions are used for the evaluation of numerical errors in the GSM solutions.

4.1.2. Evaluation of numerical errors. Three types of numerical errors are evaluated in the study. The convergence error index, $\varepsilon_{\mathrm{con}}$, takes the form of

$$\varepsilon_{\mathrm{con}} = \sqrt{\sum_{i=1}^{n_{\mathrm{node}}} (U_i^{(n+1)} - U_i^{(n)})^2} \Bigg/ \sqrt{\sum_{i=1}^{n_{\mathrm{node}}} (U_i^{(1)} - U_i^{(0)})^2} \tag{28}$$

DOI: 10.1002/fld

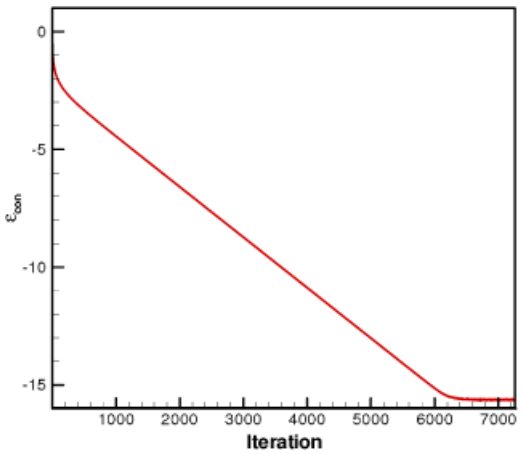

Figure 8. Plot of the profile on convergence history.

where $U_i^{(n)}$ denotes the predicted value of the field variable at node i at the nth iteration, and n_{node} is the total number of nodes in the domain. The value of ε_{con} is monitored during iterations and used to terminate the iterative process. In most simulations, in order to exclude the effect owing to the temporal discretization, computations are not stopped until ε_{con} becomes stabilized, as indicated in Figure 8.

The numerical error in a GSM solution for the overall field is defined using the L_2-norm of error as

$$\text{error} = \sqrt{\sum_{i=1}^{n_{\text{node}}} (U_i - \hat{U}_i)^2} \Bigg/ \sqrt{\sum_{i=1}^{n_{\text{node}}} \hat{U}_i^2} \tag{29}$$

where U_i and $\hat{U}_i$ are predicted and analytical solutions at node i, respectively. This type of error is used to compare the accuracy among different schemes.

The third type of error is the node-wise relative error, which is estimated in the form of

$$\text{rerror}_i = |U_i - \hat{U}_i| / |\hat{U}_i| \tag{30}$$

The node-wise relative errors distributed over the computational domain are used to identify problematic regions in simulations.

4.1.3. Cell types used in the study. Four types of cells varying in shapes, i.e. square, right triangle, regular triangle and irregular triangle, as shown in Figure 9, are used in GSM and investigated in this study. The irregular triangles are designed for the study of robustness of GSM to the irregularity of triangular cells. The numerical results for various types of cells with different spatial discretization schemes are discussed in detail below.

4.1.4. The role of directional correction. As shown in Figure 10(a), the decoupled solution is predicted using Scheme I when it is applied onto cells in square shape in solving the first Poisson

DOI: 10.1002/fld

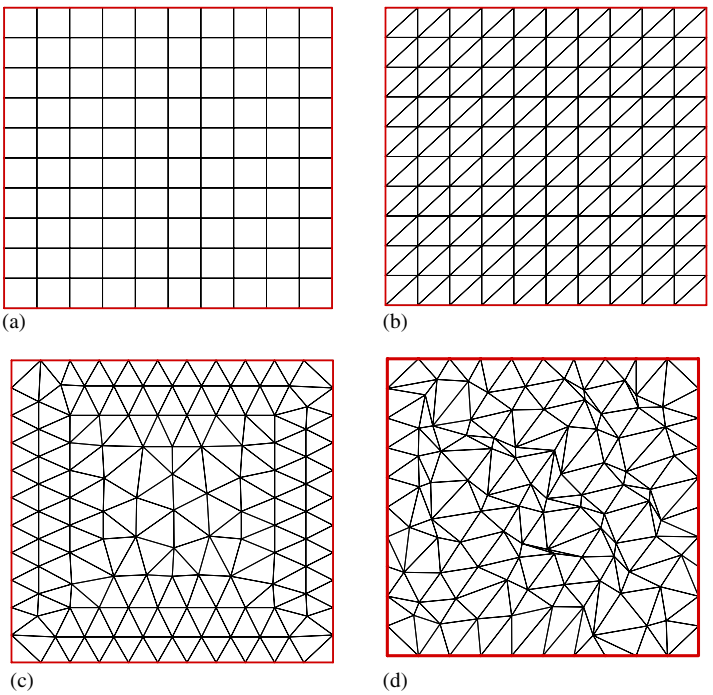

Figure 9. Representative cells under investigation: (a) square; (b) right triangle; (c) regular triangle; and (d) irregular triangle.

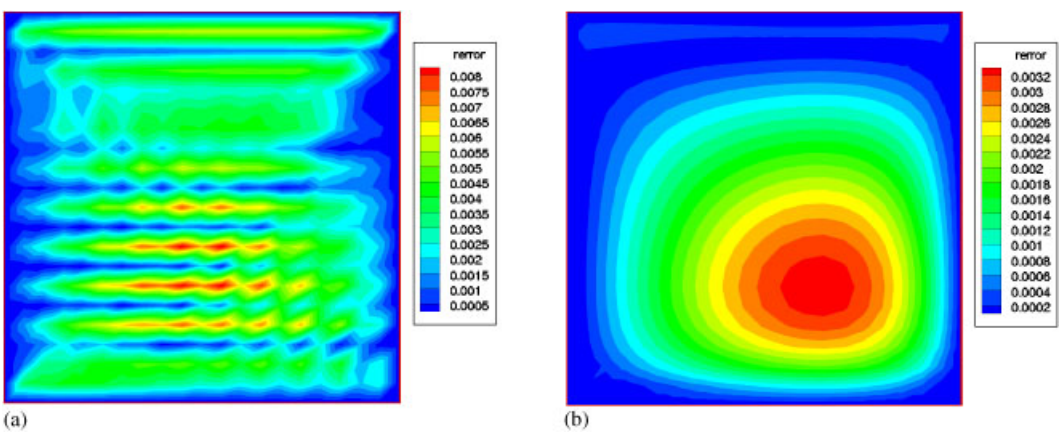

Figure 10. Contour plots of relative errors on cells in square shape in the first Poisson problem: (a) Scheme I and (b) Scheme II.

problem. The saw-toothed numerical errors (checkerboard problem) are generated and cannot be dampened out. With the directional correction in Scheme II, such a problem is overcome, as shown in Figure 10(b). This is consistent with the findings in stencil analyses.

DOI: 10.1002/fld

Note that with directional corrections, the overall numerical error is significantly reduced, as shown in Table IV. Comparing to Scheme I, the magnitudes of overall errors with Scheme II decrease by 5 times or so. However, with Scheme II, subjected to CFL condition, smaller time step is needed for stability requirement, resulting in more computations, compared with Scheme I. This becomes more obvious when finer meshes are used in simulation.

4.1.5. Comparison among four favorable schemes. The four favorable schemes are comprehensively tested on different types of cells with various node resolutions for solutions to Poisson's equations. Evolution of numerical error with averaged node spacing for the first Poisson problem is plotted in Figures 11–13, together with fitted lines. Note that the errors in these plots are evaluated using Equation (29), and the averaged node spacing, h, is evaluated using the formula given by Liu [30] in the form of

$$h = \frac{V}{\sqrt{n_{\text{node}}} - 1} \tag{31}$$

where V and n_{node} denote the area of the whole computational domain and the total number of nodes in the domain, respectively. It is apparent that the averaged node spacing decreases as the number of nodes increases.

• *On cells in square shape*: As shown in Figures 11(a) and (b), on cells in square shape, Scheme VII is as accurate as Scheme II, because the two schemes result in the same discretization stencils, as already highlighted in the stencil analyses. This is also true for Schemes VI and VIII.

In addition, the two-point quadrature schemes (VI and VIII) give relatively lower accuracy than the one-point quadrature schemes (II and VII). They also result in higher computational costs than the one-point quadrature schemes. Therefore, based on the cells in square shape, Schemes II and VII are equivalent and they are superior to Schemes VI and VIII.

The slope coefficients of trendlines confirm that these four schemes are of second-order accuracy, which is consistent with our finding in the analyses of truncation errors.

• *On cells in right triangle shape*: Profiles of numerical error against averaged node spacing based on the cells in right triangle shape are shown in Figures 12(a) and (b). Results reveal that Scheme VI gives a slightly more accurate prediction than Scheme II. In addition, it is interesting to find that based on the right triangular cells, Scheme VII is as accurate as Scheme VIII.

• *On cells in regular triangle shape*: When regular triangular cells are used, both two-point quadrature schemes (VI and VIII) produce a slightly more accurate prediction than one-point quadrature schemes, as shown in Figures 13(a) and (b). Such a finding is consistent with what is observed when right triangular cells are used.

Table IV. Comparison of numerical errors with Schemes I and II for the first Poisson problem.

	Scheme I		Scheme II	
Number of nodes	Error	Iteration	Error	Iteration
36	1.96e−2	20	5.07e−3	24
121	8.58e−3	89	1.66e−3	82
441	2.63e−3	202	4.87e−4	303
1681	7.16e−4	728	1.33e−4	1379
6561	1.86e−4	2679	3.38e−5	4299

DOI: 10.1002/fld

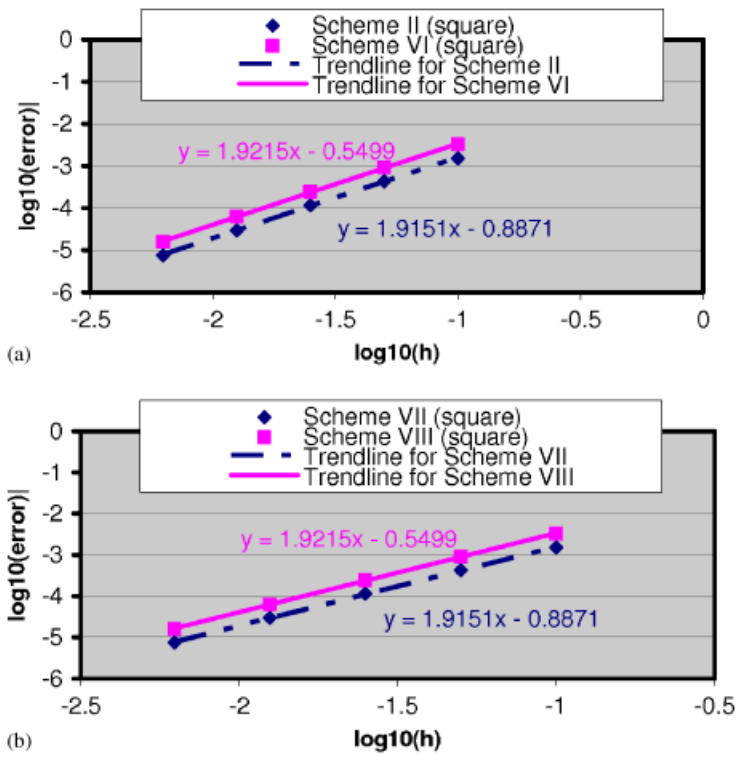

Figure 11. Evolution of numerical error with averaged node spacing based on the cells in square shape.

Table V summarizes the numerical errors in number for different discretization schemes when regular triangular cells are used. It is clear that Scheme VII is slightly more accurate than Scheme II, and Scheme VIII is more accurate than Scheme VI. This is also true when right triangular cells are used in simulations. Such discrepancy in accuracy is related to the approximation of gradient at boundary nodes. In Schemes II and VI, the gradients at boundary nodes are needed to be approximated, as they are desired by the interpolation approach for the approximation of gradients at midpoints of internal cell edges that are linked with boundary nodes. Thus, additional errors are introduced due to the approximation. Comparatively, in Schemes VII and VIII, subjected to Dirichlet boundary conditions, approximations to gradients at the boundary nodes are entirely avoided, thanks to the gradient smoothing techniques applied to the mGSDs.

In general, in terms of computational cost, Schemes VII and II are almost the same, and Schemes VIII and VI are very close to each other. Schemes VI and VIII require roughly twice the computational time as Schemes II and VII. Schemes VI and VIII are more accurate than Schemes II and VII, but the improvement in accuracy is not significant. Therefore, to balance the numerical accuracy and computational efficiency, the two one-point quadrature schemes (II and VII) are preferred to be used in practice.

4.1.6. Robustness to irregularity of the cells. The consistent findings described above are also observed in the solutions to the second Poisson problem. For the second Poisson problem, additional

DOI: 10.1002/fld

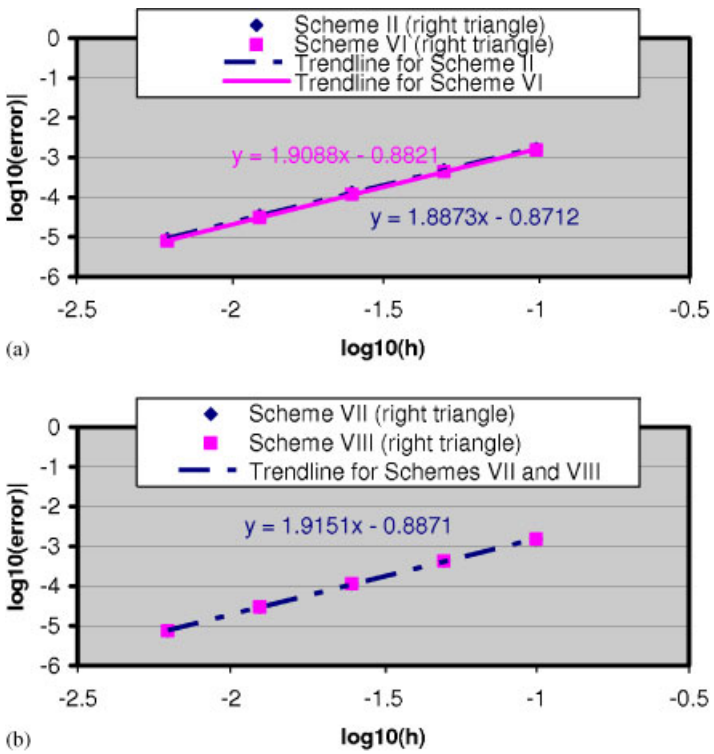

Figure 12. Evolution of numerical error against averaged node spacing based on the cells in right triangle shape.

studies on the effects of irregularity of the triangular cells are carried out. It is well known that triangular cells have best adaptivity to complex geometries and can be generated automatically in very efficient ways. Therefore, it is desirable to use triangular cells in GSM so that GSM can become a very robust method for engineering problems with complicated geometries. Our objective is thus to further identify the sensitivity of GSM to the cell quality. Following the findings as described previously, the two one-point quadrature schemes (II and VII) are used in this study (Table VI).

To study this in a systematic manner, we first define the irregularity for all triangular cells in the computational domain, γ, using the following formula:

$$\gamma = \frac{\sum_{i=1}^{n_e} \dfrac{(a_i - b_i)^2 + (b_i - c_i)^2 + (c_i - a_i)^2}{a_i^2 + b_i^2 + c_i^2}}{n_e} \tag{32}$$

where a_i, b_i and c_i, respectively, denote the lengths of cell edges of a triangular cell, and n_e stands for the total number of cells in the overall domain. Equation (32) is derived from the formula proposed by Stillinger *et al.* [53] for a single triangle. Using Equation (32), the irregularity vanishes for equilateral triangles and is positive for all other shapes including isosceles triangles.

DOI: 10.1002/fld

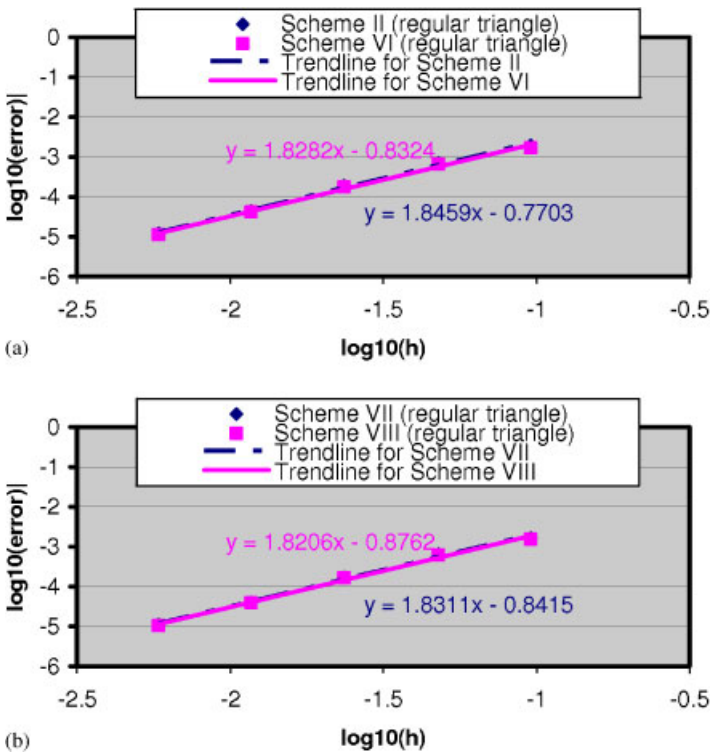

Figure 13. Evolution of numerical error against averaged node spacing based on the cells in regular triangle shape.

Table V. Comparison of numerical errors predicted on regular triangular cells for the first Poisson problem with favorable schemes.

		Error			
Number of nodes	Δt	Scheme II	Scheme VII	Scheme VI	Scheme VIII
131	0.008	1.95e−3	1.65e−3	1.68e−3	1.55e−3
478	0.001	7.02e−4	6.53e−4	6.46e−4	6.20e−4
1887	0.0005	1.89e−4	1.70e−4	1.74e−4	1.65e−4
7457	0.0001	4.38e−5	3.98e−5	4.12e−5	3.94e−5
29 629	0.00003	1.22e−5	1.10e−5	1.11e−5	1.05e−5

Figure 14 shows six sets of triangular cells with various irregularities but the same number of nodes. It is obvious that as the irregularity increases, the mesh is distorted further.

Note that when the irregularity is larger than 0.152, overlapped cells are found in the domain, as shown in Figure 15, which is an extreme that will not normally happen in practice. Such an extreme case cannot be accurately reflected in the irregularity using Equation (32). Nevertheless, we still use such extremely distorted cells to test the robustness of GSM.

DOI: 10.1002/fld

Table VI. Comparison of allowable maximum time step and numerical error for irregular triangular cells.

Mesh	Irregularity	Maximum time step (Δt)	Error	
			Scheme VII	Scheme II
(a)	0.021	0.01	0.0163	0.0172
(b)	0.028	0.01	0.0169	0.0177
(c)	0.048	0.009	0.0179	0.0188
(d)	0.079	0.0075	0.0194	0.0206
(e)	0.118	0.004	0.0214	0.0231
(f)	0.152	0.0005	0.0234	0.0259

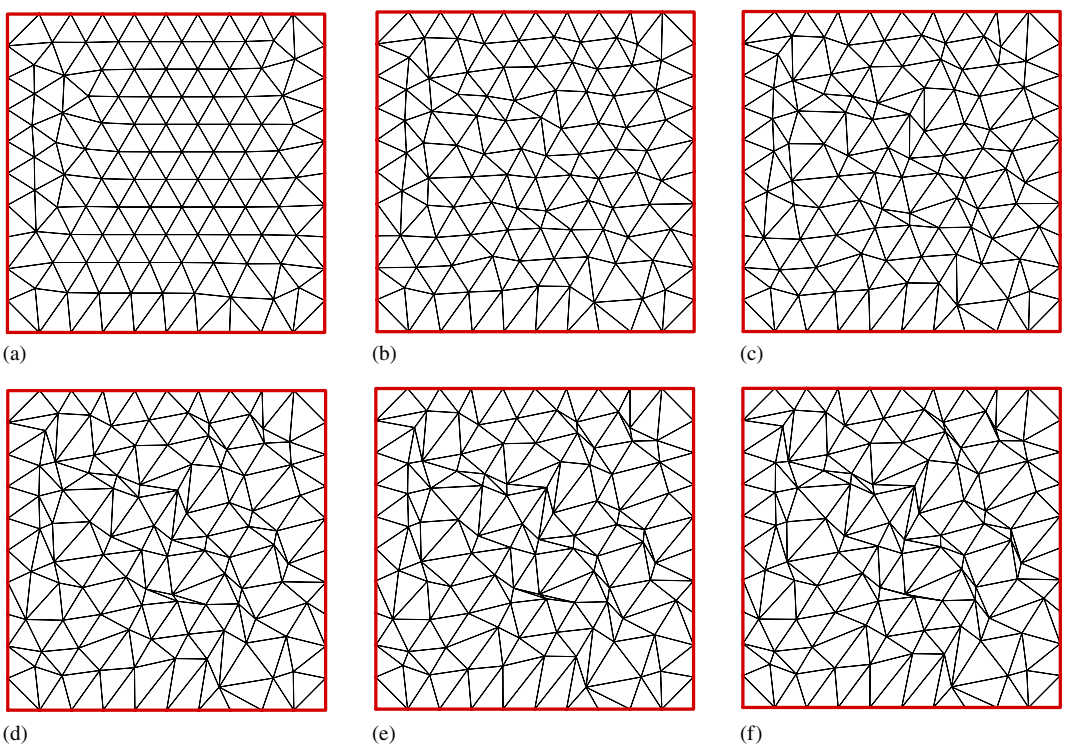

Figure 14. Triangular cells with various irregularities: (a) $\gamma=0.021$; (b) $\gamma=0.028$; (c) $\gamma=0.048$; (d) $\gamma=0.079$; (e) $\gamma=0.118$; and (f) $\gamma=0.152$.

Convergent results for all irregular meshes are obtained using GSM with Schemes II and VII. Contour plots of the predicted solutions with Scheme VII are selectively shown in Figure 16. It is clear that the predicted results are reasonably accurate on all sets of irregular cells. Note that as the irregularity of the cells increases, the time step (Δt) has to be reduced so as to guarantee stability and convergence of the results, as shown in Table VI.

DOI: 10.1002/fld

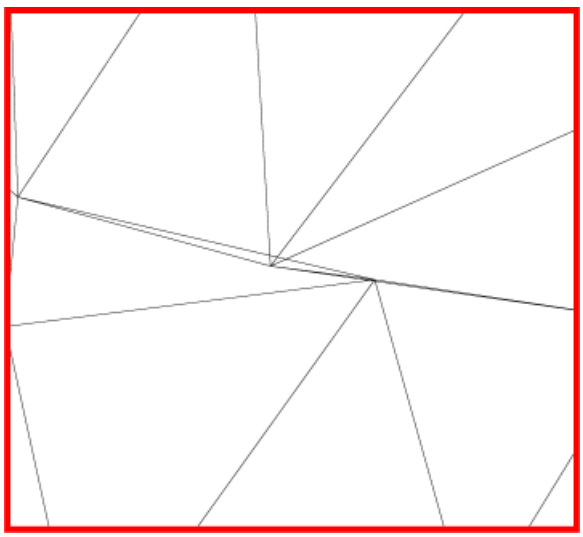

Figure 15. Overlapped cells in the computational domain ($\gamma = 0.16$).

It is interesting to find that for cases without non-overlapped cells, the numerical errors predicted with GSM do not vary so much as the irregularity of cells increases (see Figure 17). Once overlapped cells occur in the domain, sudden jumps in numerical errors are observed. However, for Schemes II and VII, stable results are still obtained. Besides, Scheme VII shows much better stability and accuracy than Scheme II amongst all irregular cells examined here. This means that GSM with Scheme VII is remarkably robust and insensitive to cell irregularity. In other words, with the proposed GSM, stable and accurate results can be obtained even with highly distorted triangular cells. Such an attractive feature can be attributed to the consistent use of smoothing techniques in Scheme VII, which provides the crucial stability and robustness to our GSM.

4.2. Solutions to Navier–Stokes equations

The GSM procedure with Scheme VII has also been successfully applied to solutions to some benchmarked compressible flows. The full Navier–Stokes equations for conservative variables are solved accordingly. The well-known second-order Roe flux-different slitting scheme [54] is adopted to evaluate the convective fluxes, because of its high accuracy for boundary layers and good resolution of shocks [20]. The left and right states are predicted with the Barth and Jespersen method [25]. To avoid oscillation and non-physical solutions, the Venkatakrishnan's limiter [55] is used. The flow turbulence is simulated with the Spalart–Allmaras one-equation turbulence model [56] in our GSM code. In this section, the results for inviscid flow over a NACA0012 airfoil, laminar flow over a flat plate and turbulent flow over an RAE2822 airfoil are presented, respectively.

4.2.1. Inviscid flow over an NACA0012 airfoil. Euler solver with the proposed GSM scheme dependent on multiple GSDs has been developed and it is applied to the solutions to an inviscid flow

DOI: 10.1002/fld

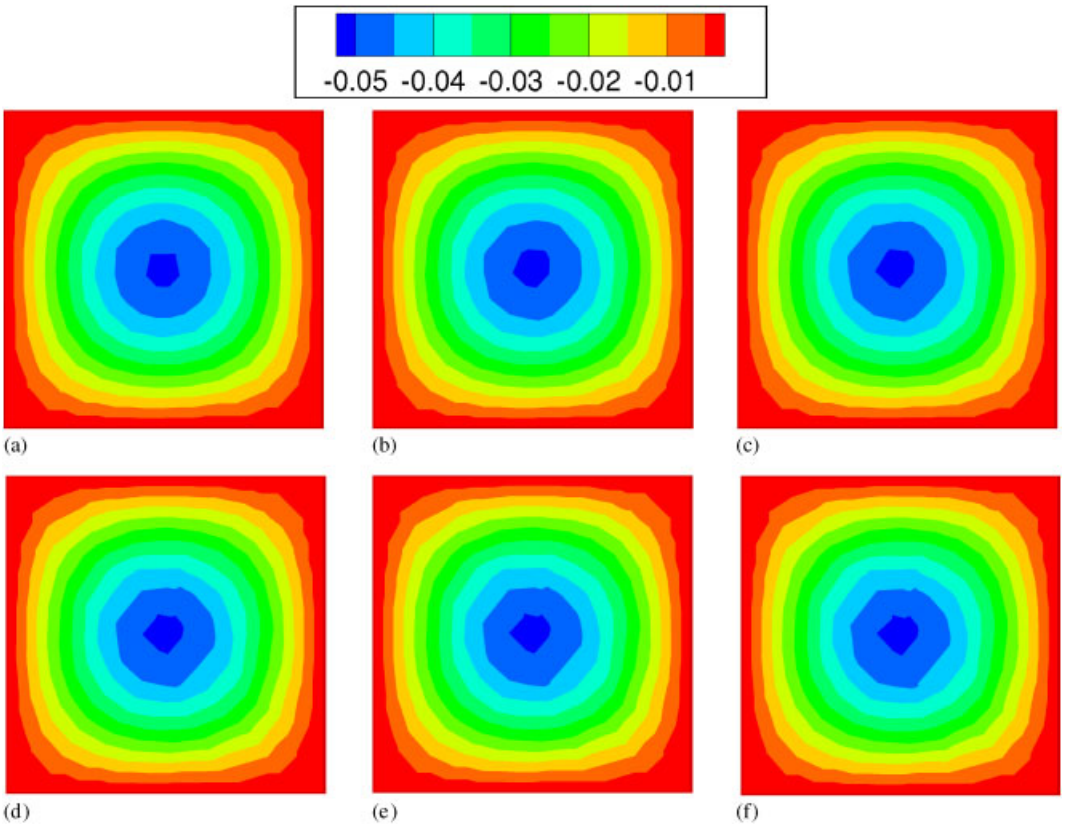

Figure 16. Contour plots of solutions to the second Poisson equation discretized onto irregular triangular meshes: (a) $\gamma = 0.021$; (b) $\gamma = 0.028$; (c) $\gamma = 0.048$; (d) $\gamma = 0.079$; (e) $\gamma = 0.118$; and (f) $\gamma = 0.152$.

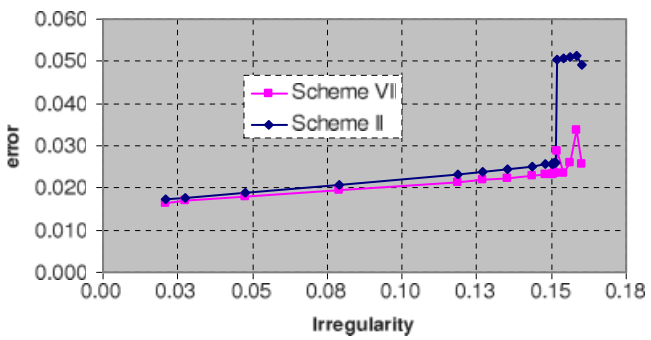

Figure 17. Numerical errors in GSM solutions (Schemes II and VII) to the second Poisson problem with respect to irregularity of cells.

DOI: 10.1002/fld

over an NACA0012 airfoil. In the testing case, for the freestream, $T_\infty = 288\,\mathrm{K}$, $p_\infty = 1.0 \times 10^5\,\mathrm{Pa}$, $Ma = 0.8$ and the angle of attack $\alpha = 1.25°$. Numerical results are selectively presented below.

Figure 18 shows the unstructured grid used in the computation. Plots of contours about the density, the static pressure and Mach number are displayed in Figures 19–21, respectively. It is

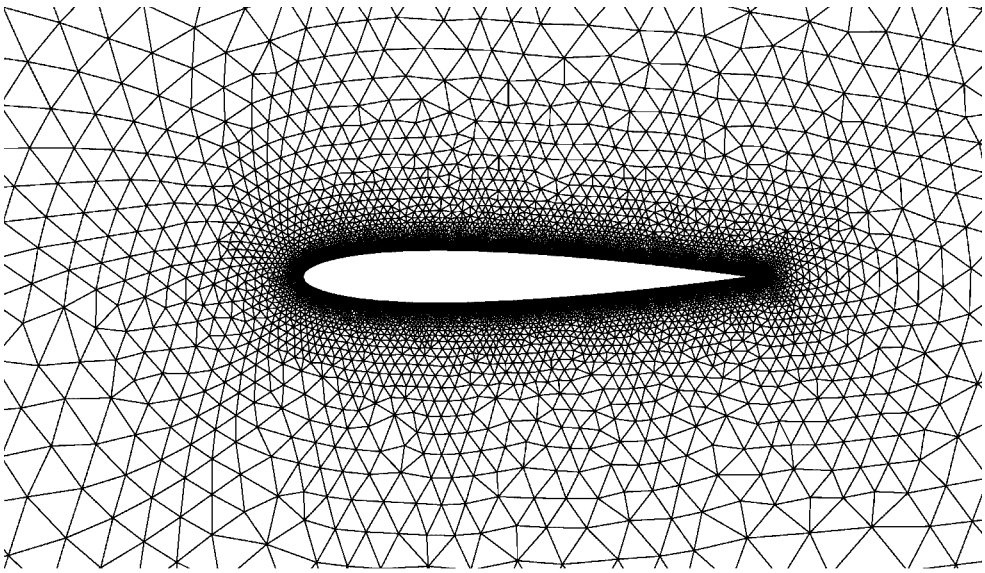

Figure 18. Irregular grid near the NACA0012 airfoil.

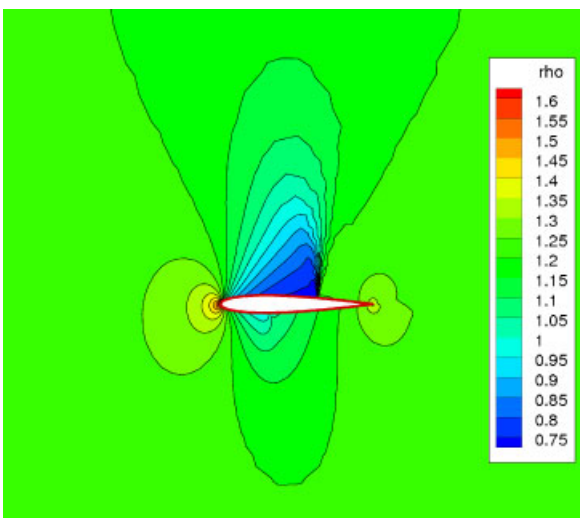

Figure 19. Spatial distribution of the predicted density.

DOI: 10.1002/fld

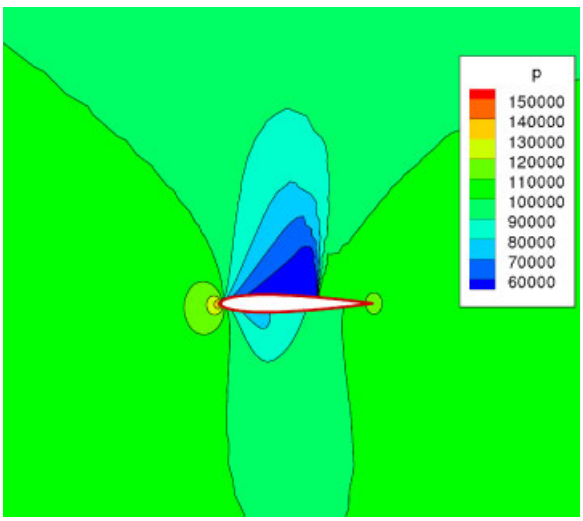

Figure 20. Plot of contours of the predicted static pressure.

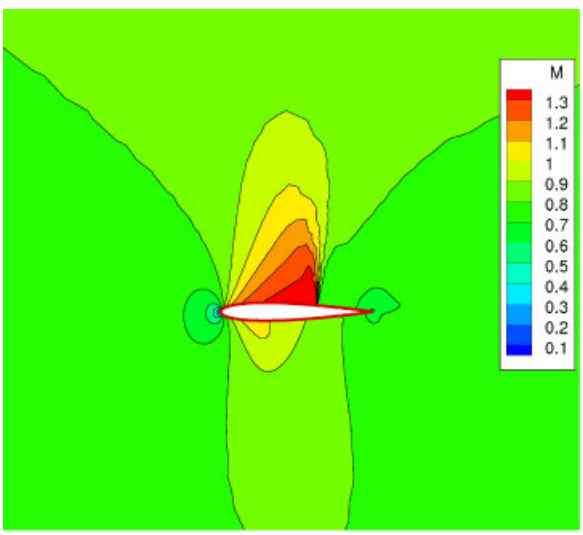

Figure 21. Plot of contours of the predicted Mach number.

obvious that the strong shock occurring on the upper surface of the airfoil is nicely captured, as well as the weak shock taking place on the lower surface. The results are qualitatively agreeable with the results published by Barth [26].

DOI: 10.1002/fld

4.2.2. Laminar flow over a flat plate. The third testing case is with regard to the laminar flow over a flat plate. The freestream is subject to the conditions $Re=5000$ and $Ma=0.5$. The angle of attack is zero for this case. A rectangular computational domain was generated and gridded by right triangles. Non-slip conditions are imposed onto the wall and symmetry conditions are applied to the region ahead of the plate along the x-axis. Farfield conditions (see [20]) are imposed at the external boundaries.

Figure 22 shows the convergence history of the iterative process. The boundary layer effect is clearly indicated in Figure 23 about the vector plot near the boundary. The profile about the

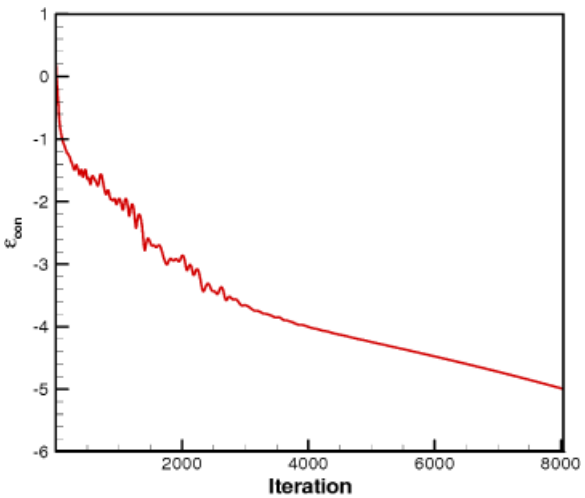

Figure 22. Profile of the convergence history.

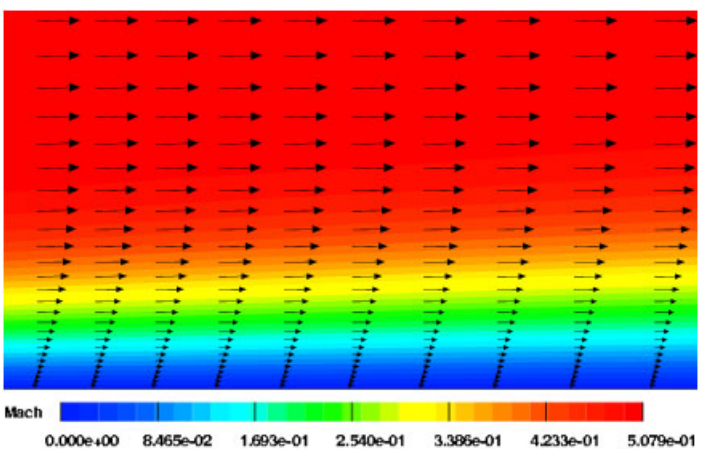

Figure 23. Velocity profiles near the boundary and contour plot of Mach number.

DOI: 10.1002/fld

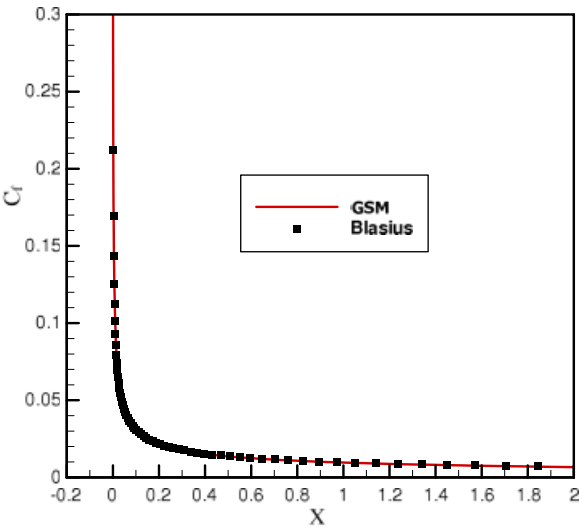

Figure 24. Comparison of predicted and analytical wall friction coefficients.

predicted skin friction coefficient C_f varied with the distance x is compared with analytical Blasius solution, as shown in Figure 24. It is apparent that our numerical results show wonderful agreement with the analytical solution.

4.2.3. Turbulence flow over an RAE2822 airfoil. In this testing case, the freestream corresponds to the following conditions:

$T_\infty = 255.556\,\mathrm{K}$, $p_\infty = 1.0756256 \times 10^5\,\mathrm{Pa}$, $Re = 6.5 \times 10^6$, $Ma = 0.729$ and $\alpha = 2.31$. Here, T_∞, p_∞, Re and Ma denote, respectively, the temperature, static pressure, Reynolds number and Mach number of the freestream, and α stands for the angle of attack of the RAE2822 airfoil.

The unstructured triangular cells shown in Figure 25 are used in the simulation. As seen in Figure 26 on the contour plot of the Mach number, a shock occurring on the upper side of the airfoil is captured. The predicted pressure coefficient (C_p) distributed on the airfoil surface is plotted in Figure 27, with comparison to the results from the commercial CFD package—FLUENT (FVM-based) [57], and experimental data [58]. Apparently, our GSM result is more accurate than the FLUENT result, especially near the leading edge. The noticeable difference between our GSM solution and experimental data, occurring around the shock region, may be due to the insufficiently fine cells generated across the shock. It should be improved once GSM is embarked with solution-based adaptive techniques, which will be explored in the future.

5. CONCLUSION

In this study, a conservative and efficient GSM formulated for the strong form of governing equations is developed. GSM is valid for both regular and irregular cells so that it can be readily

DOI: 10.1002/fld

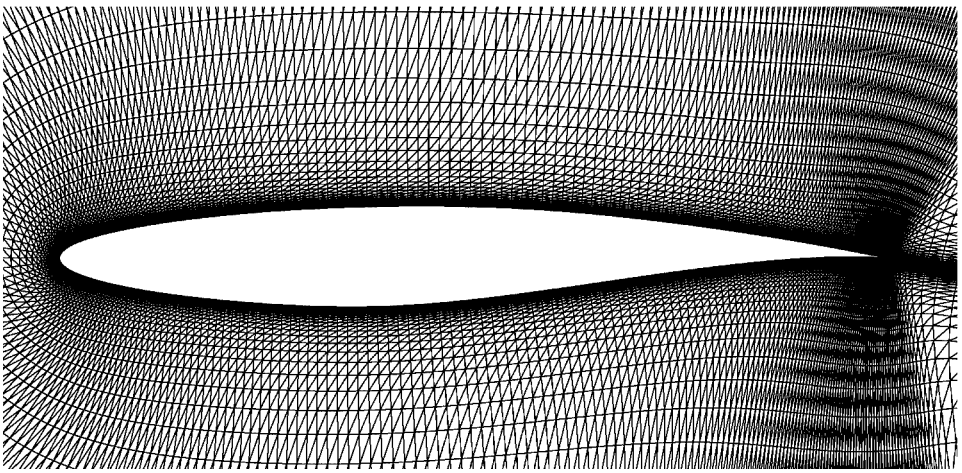

Figure 25. Unstructured triangular cells around the RAE2822 airfoil.

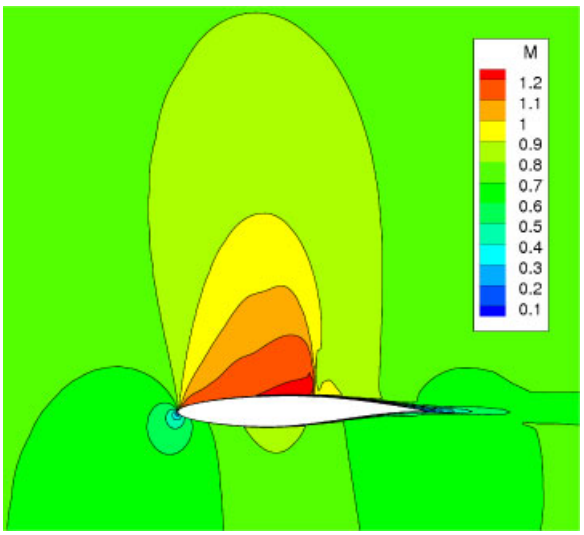

Figure 26. Contour plot of Mach number for the flow over the RAE2822 airfoil.

applied to fluid flow problems with arbitrary geometry. With efforts in studying eight proposed GSM schemes, it is found that

- Schemes II, IV, VI and VIII are favorable, because of their compact stencils with positive coefficients of influence.
- Schemes VII and VIII that use the gradient smoothing technique on mGSDs outperform Schemes II and VI in terms of robustness, stability and accuracy.

DOI: 10.1002/fld

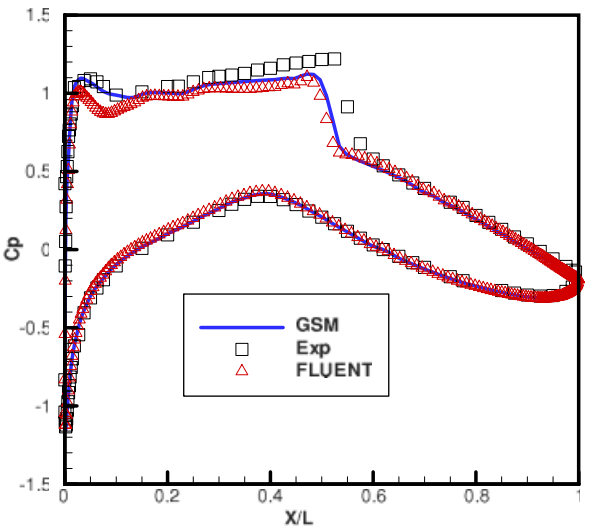

Figure 27. Profiles of pressure coefficients on the surface of an RAE2822 airfoil.

- The one-point quadrature-based schemes (Schemes II and VII) have well-balanced performance in terms of both efficiency and accuracy.
- Scheme VII is superior to Scheme II from the points of view of consistency in derivative approximation at various locations and robustness against irregularity of cells. Therefore, Scheme VII is most preferable in practice, especially for large-scale problems.

REFERENCES

1. Ferziger JH, Peric M. *Computational Methods for Fluid Dynamics* (3rd edn). Springer: Berlin, 2002.
2. Richardson LF. The approximate arithmetical solution by finite differences of physical problems involving differential equations, with an application to the stresses in a masonry dam. *Philosophical Transactions of the Royal Society of London*, *Series A* 1910; **210**:307–357.
3. Fletcher CAJ. *Computational Techniques for Fluid Dynamics*, vols I and II (2nd edn). Springer: Berlin, 1991.
4. Hoffmann KA, Chiang ST. *Computational Fluid Dynamics for Engineers*, vols I and II. Engineering Education System^{TM}: Austin, TX, 1993.
5. Knabner P, Angermann L. *Numerical Methods for Elliptic and Parabolic Partial Differential Equations*. Springer: New York, 2003.
6. Tannehill JC, Anderson DA, Pletcher RH. *Computational Fluid Mechanics and Heat Transfer* (2nd edn). Taylor & Francis: London, 1997.
7. Thompson JF, Soni BK, Weatherill NP. *Handbook of Grid Generation*. CRC Press: Boca Raton, FL, 1999.
8. Courant R. Variational methods for the solution of problems of equilibrium and vibration. *Bulletin of the American Mathematical Society* 1943; **49**:1–23.
9. Clough RW. The finite element method in plane stress analysis. *Proceedings of the 2nd ASCE Conference on Electronic Computation*, Pittsburgh, 1960.
10. Liu GR, Quek SS. *The Finite Element Method*: *A Practical Course*. Butterworth Heinemann: Oxford, 2003.
11. Zienkiewicz OC, Taylor RL. *The Finite Element Method* (5th edn). Butterworth Heinemann: Oxford, 2000.
12. Donea J, Huerta A. *Finite Element Methods for Flow Problems*. Wiley: New York, 2003.
13. Gresho PM, Sani RL. *Incompressible Flow and the Finite Element Method* (*Volume 1*: *Advection–Diffusion*). Wiley: New York, 2000.

DOI: 10.1002/fld

14. Reddy JN, Gartling DK. *The Finite Element Method in Heat Transfer and Fluid Dynamics* (2nd edn). CRC Press: London, 2001.
15. Hughes TJR, Franca LP, Mallet M. A new finite element formulation for computational fluid dynamics. VI. Convergence analysis of the generalized SUPG formulation for linear time-dependent multidimensional advective–diffusive systems. *Computer Methods in Applied Mechanics and Engineering* 1987; **65**(1):97–112.
16. Hughes TJR, Franca LP, Hulbert GM. A new finite element formulation for computational fluid dynamics. VIII. The Galerkin/least-squares method for advective–diffusive equations. *Computer Methods in Applied Mechanics and Engineering* 1989; **73**(2):173–189.
17. Baiocchi C, Brezzi F, Franca L. Virtual bubbles and the Galerkin/least-squares method. *Computer Methods in Applied Mechanics and Engineering* 1993; **105**(1):125–141.
18. Hughes TJR. Multiscale phenomena: Green's functions, the Dirichlet-to-Neumann formulation, subgrid scale models, bubbles and the origins of stabilized methods. *Computer Methods in Applied Mechanics and Engineering* 1995; **127**(1–4):387–401.
19. Versteeg HK, Malalasekera W. *An Introduction to Computational Fluid Dynamics—The Finite Volume Method*. Longman: New York, 1995.
20. Blazek J. *Computational Fluid Dynamics*: *Principles and Application* (1st edn). Elsevier: Oxford, 2001.
21. Evans MW, Harlow FH. The particle-in-cell method for hydrodynamic calculations. *Los Alamos Scientific Laboratory Report LA-2139*, 1957.
22. Gentry R, Martin RE, Daly BJ. An Eulerian differencing method for unsteady compressible flow problems. *Journal of Computational Physics* 1966; **1**:87–118.
23. Jameson A, Yoon S. Multigrid solution of the Euler equations using implicit schemes (with S. Yoon). *AIAA Journal* 1986; **24**:1737–1743.
24. Jameson A, Schmidt W, Turkel E. Numerical solutions of the Euler equations by finite volume methods using Runge–Kutta time-stepping schemes. *AIAA Paper 1259*, 1981.
25. Barth TJ, Jespersen DC. The design and application of upwind schemes on unstructured grids. *AIAA Paper 0366*, 1989.
26. Barth TJ. *Aspects of Unstructured Grids and Finite-volume Solvers for the Euler and Navier–Stokes Equations*. VKI Lecture Series in CFD course 1994, Von Karman Institute, 1994.
27. Barth TJ. *Numerical Methods for Conservation Laws on Structured and Unstructured Grids*. VKI Lecture Series in CFD course 2003, Von Karman Institute, 2003.
28. Monaghan JJ. An introduction to SPH. *Computer Physics Communications* 1988; **48**:89–96.
29. Li S, Liu WK. Moving least-square reproducing kernel method. Part II: Fourier analysis. *Computer Methods in Applied Mechanics and Engineering* 1996; **139**:159–193.
30. Liu GR. *Mesh Free Methods*: *Moving Beyond the Finite Element Method*. CRC Press: Boca Raton, FL, 2002.
31. Belytschko T, Krongauz Y, Organ D, Fleming M, Krysl P. Meshless methods: an overview and recent developments. *Computer Methods in Applied Mechanics and Engineering* 1996; **139**:3–47.
32. Atluri SN, Zhu T. A new meshless local Petrov–Galerkin (MLPG) approach in computational mechanics. *Computational Mechanics* 1998; **22**:117–127.
33. Babuska I, Melenk M. The partition of unity method. *International Journal for Numerical Methods in Engineering* 1997; **40**:727–758.
34. Babuska I, Banerjee U, Osbor JE. Survey of meshless and generalized finite element methods: a unified approach, *TICAM Report 02-03*, University of Texas, Austin, 2002.
35. Liu GR, Gu YT. *An Introduction to Meshfree Methods and their Programming*. Springer: Berlin, 2005.
36. Idelsohn SR, Onate E. To mesh or not to mesh. That is the question. *Computer Methods in Applied Mechanics and Engineering* 2006; **195**:4681–4696.
37. Onate E, Idelsohn S, Zienkiewicz OC, Taylor RL. A finite point method in computational mechanics. Applications to convective transport and fluid flow. *International Journal for Numerical Methods in Engineering* 1996; **39**(22):3839–3866.
38. Kee BTB, Liu GR, Lu C. A regularized least-square radial point collocation method (RLS-RPCM) for adaptive analysis. *Computational Mechanics* 2007; **40**:837–853.
39. Liu GR, Kee BTB, Lu C. A stabilized least-squares radial point collocation method (LS-RPCM) for adaptive analysis. *Computer Methods in Applied Mechanics and Engineering* 2006: **195**:4843–4861.
40. Chen JS, Wu CT, Yoon S, You Y. A stabilized conforming nodal integration for Galerkin mesh-free methods. *International Journal for Numerical Methods in Engineering* 2001; **50**:435–466.
41. Chen JS, Yoon S, Wu CT. Non-linear version of stabilized conforming nodal integration for Galerkin mesh-free methods. *International Journal for Numerical Methods in Engineering* 2002; **53**:2587–2615.

DOI: 10.1002/fld

42. Zhang GY, Liu GR, Wang YY, Huang HT, Zhong ZH, Li GY, Han X. A linearly conforming point interpolation method (LC-PIM) for three-dimensional elasticity problems. *International Journal for Numerical Methods in Engineering* 2007; **72**:1524–1543.
43. Liu GR, Liu MB. *Smoothing Particle Hydrodynamics—A Meshfree Particle Method*. World Scientific: Singapore, 2003.
44. Lucy L. A numerical approach to testing the fission hypothesis. *The Astronomical Journal* 1977; **82**:1013–1024.
45. Liu GR, Li Y, Dai KY, Luan MT, Xue W. A linearly conforming radial point interpolation method for solid mechanics. *International Journal of Computational Methods* 2006; **3**:401–428.
46. Liu GR, Zhang GY, Dai KY, Wang YY, Zhong ZH, Li GY, Han X. A linearly conforming point interpolation method (LC-PIM) for 2D solid mechanics problems. *International Journal of Computational Methods* 2005; **2**(4):645–665.
47. Liu GR, Dai KY, Nguyen TT. A smoothed finite element method for mechanics problems. *Computational Mechanics* 2007; **39**:859–877.
48. Liu GR, Nguyen TT, Dai KY, Lam KY. Theoretical aspects of the smoothed finite element method (SFEM). *International Journal for Numerical Methods in Engineering* 2007; **71**:902–930.
49. Crumpton PI, Moinier P, Giles MB. An unstructured algorithm for high Reynolds number flows on highly-stretched grids. *Proceedings of the 10th International Conference on Numerical Methods in Laminar and Turbulent Flows*, Swansea, England, 1997.
50. Moinier P. Algorithm developments for an unstructured viscous flow solver. *Ph.D. Thesis*, University of Oxford, Oxford, 1999.
51. Lohner R. *Applied Computational Fluid Dynamics Techniques*: *An Introduction Based on Finite Element Methods*. Wiley: New York, 2001.
52. Patankar SV. *Numerical Heat Transfer and Fluid Flow*. McGraw-Hill: New York, 1980.
53. Stillinger DK, Stillinger FH, Torquato S, Truskett TM, Debenedetti PG. Triangle distribution and equation of state for classical rigid disks. *Journal of Statistical Physics* 2000; **100**(1–2):49–71.
54. Roe PL. Approximate Riemann solvers, parameter vectors, and difference schemes. *Journal of Computational Physics* 1981; **43**:357–382.
55. Venkatakrishnan V. Convergence to steady-state solutions of the Euler equations on unstructured grids with limiters. *Journal of Computational Physics* 1995; **118**:120–130.
56. Spalart PR, Allmaras SA. One-equation turbulence model for aerodynamic flows. *La Recherche Aerospatiale* 1994; **1**:5–21.
57. *FLUENT 6.3 User's Guide*. Ansys, Inc., 2006.
58. NPARC Alliance CFD Verification and Validation Web Site. http://www.grc.nasa.gov/WWW/wind/valid/validcases.html.

DOI: 10.1002/fld