A significantly expanded version has been published in VKI Lecture Series:

CFD- High-Order Discretization Methods: VKI Lecture Notes 2006

High-Order Viscous Discretization on Unstructured Grids

H. Nishikawa and P. L. Roe

July 2004

1 Model Equation

We consider high-order spatial discretization of the following model equation over a triangular grid.

$$\partial_t u = \mu \, \mathrm{div}(\mathrm{grad} u). \tag{1}$$

2 Standard Galerkin (STD Galerkin)

The standard Galerkin discretization of the Laplacian is derived by integrating over the median dual cell around the vertex j.

$$\begin{aligned} \int_{\Omega_j} \partial_t u \, dV &= \int_{\Omega_j} \mathrm{div}(\mathrm{grad} u) \, dV & (2) \\ &= -\oint_{\partial\Omega_j} \mathrm{grad} u \cdot \mathbf{n} \, dS. & (3) \end{aligned}$$

For piecewise linear approximation, the gradient is constant over each triangle, $\mathrm{grad} u = (\mathrm{grad} u)^T = (u_x^T, u_y^T)$,

$$u_x^T = \frac{1}{2S_T} \sum_{i \in \{i_T\}} u_i \Delta y_i, \quad u_y^T = -\frac{1}{2S_T} \sum_{i \in \{i_T\}} u_i \Delta x_i \tag{4}$$

where the difference $\Delta()$ is taken counterclockwise along the edge opposite to the node i and S_T is the area of the triangle T. We therefore obtain

$$S_j \frac{du_j}{dt} = -\frac{\mu}{2} \sum_{T \in \{T_j\}} (\mathrm{grad} u)^T \cdot \mathbf{n}_T \tag{5}$$

where S_j is the area of the median dual cell and $\mathbf{n}_T$ is the scaled inward normal for the triangle T of the edge opposite to the node j. This dicretization is second-order accurate since it is exact for linear solutions. It is important to note that this scheme can be derived from minimizing the energy norm associated with the Laplacian, defined by

$$\mathcal{F} \;=\; \frac{\mu}{2}\sum_{T\in\{T\}}\left[(u_x^T)^2+(u_y^T)^2\right]S_T. \tag{6}$$

The steepest descent minimization gives the update scheme,

$$\begin{aligned} u_j^{n+1} &= u_j^n-\omega\,\frac{\partial\mathcal{F}}{\partial u_j} && (7)\\ &= u_j^n-\omega\,\frac{\mu}{2}\sum_{T\in\{T_j\}}\left(\Delta y_T u_x^T-\Delta x_T u_y^T\right) && (8)\\ &= u_j^n-\omega\,\frac{\mu}{2}\sum_{T\in\{T_j\}}(\mathrm{grad}u)^T\cdot\mathbf{n}_T && (9)\end{aligned}$$

where ω is a small constant. Note that this is equivalent to (5). Hence the Galerkin discretization possesses the minimum energy property.

On a regular triangular grid, we expand a solution and insert it into the update formula to find the truncation error. At convergence, we have

$$\begin{aligned} 0 &= (u_{xx}+u_{yy})+\frac{h^2}{12}\left[u_{xxxx}+u_{yyyy}\right]\\ &\quad+\frac{h^4}{360}\left[u_{xxxxxx}+u_{yyyyyy}\right]. && (10)\end{aligned}$$

3 High-Order Discretization: Gradient Reconstruction (GR)

The standard Galerkin discretization can be upgraded to higher order. A simple way to achieve this is to estimate the derivatives in the energy norm to higher order and then minimize it. For example, ignoring any change in the gradient for simplicity, we may use the same update formula (i.e. distribution coefficient)

$$u_j^{n+1} \;=\; u_j^n-\omega\,\frac{\mu}{2}\sum_{T\in\{T_j\}}(\mathrm{grad}u)^T_{high}\cdot\mathbf{n}_T \tag{11}$$

except that now the gradient is to be approximated more accurately within each triangle. For more accurate gradient evaluation, we introduce additional nodes at the midpoints of the edge. Then, we can estimate u_x, for example, by Simpson's rule,

$$\begin{aligned} (u_x)^T_{high} &= \frac{1}{S_T}\int_T\!\!\int \partial_x u\,dxdy && (12)\\ &= \frac{1}{S_T}\oint u\,dy && (13)\\ &= \frac{1}{6S_T}\sum_{edges}(u_L+4u_m+u_R)\Delta y && (14)\end{aligned}$$

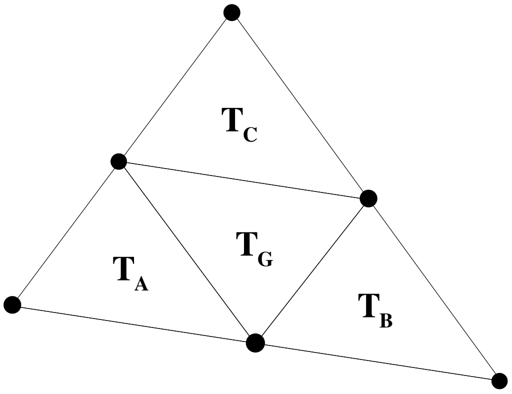

Figure 1: Triangle with midpoints

where u_L and u_R are the solution values at the end points, u_m is the midpoint value, and the difference $\Delta()$ is taken clockwise along the edge. Similarly for u_y. Note that this is 4th order accurate discretization (provided the midpoint values are reasonably accurate). Several rearrangements are possible for this formula of which useful ones are

$$\begin{aligned}
(u_x)^T_{high} &= \frac{1}{S_T}\sum_{edges}\left\{\overline{u}-\frac{2}{3}(\overline{u}-u_m)\right\}\Delta y && (15)\\
&= u_x^T-\frac{2}{3}\left(u_x^T-u_x^{T_G}\right) && (16)\\
&= \frac{1}{6}\left(u_x^{T_A}+u_x^{T_B}+u_x^{T_C}+3u_x^{T_G}\right) && (17)\\
&= u_x^T-\frac{1}{3}\left(u_x^{T_A}+u_x^{T_B}+u_x^{T_C}-3u_x^{T_G}\right) && (18)
\end{aligned}$$

where the overbar indicates the arithmetic mean over the edge.

Instead of computing, the midpoint value may be interpolated from nodal quantities. To achieve higher-order, it must be better than a linear interpolation. To this end, we reconstruct the solution gradient at the nodes, for example, by the area-weighted average of the first-order cell gradient (4),

$$(u_x)_j=\frac{\sum_{\{T_j\}}S_T u_x^T}{\sum_{\{T_j\}}S_T},\quad (u_y)_j=\frac{\sum_{\{T_j\}}S_T u_y^T}{\sum_{\{T_j\}}S_T}. \qquad (19)$$

This formula is 2nd order accurate for a regular grid. Various ways are available for gradient which we leave open for a moment. Using the recovered nodal gradient, we can construct the Hermite polynomial over each edge to interpolate the midpoint value. The result is

$$u_m \quad = \quad \overline{u} - \frac{1}{8}\Delta P \tag{20}$$

where

$$\Delta P \quad = \quad \Delta(u_x)\Delta x + \Delta(u_y)\Delta y. \tag{21}$$

Substitute this into (15) gives

$$u_x^T|_{high} \quad = \quad u_x^T - \frac{1}{12S_T}\sum_{edges}\Delta P \Delta y \tag{22}$$

$$u_y^T|_{high} \quad = \quad u_y^T + \frac{1}{12S_T}\sum_{edges}\Delta P \Delta x. \tag{23}$$

These can be inserted into (11) to give a higher-order scheme.

On a regular grid, the resulting scheme has the following discretization error.

$$\begin{aligned} 0 \quad = \quad & (\Delta u) + \frac{h^2}{12}\left[\partial_{xx} + \partial_{xy} + \partial_{yy}\right](\Delta u) \\ & + h^4\left[\frac{1}{360}\partial_{xxxx} + \frac{1}{48}\partial_{xxxy} + \frac{43}{720}\partial_{xxyy} + \frac{1}{48}\partial_{xyyy} + \frac{1}{360}\partial_{yyyy}\right](\Delta u) \\ & + \frac{h^4}{24}u_{xxxyyy} \end{aligned} \tag{24}$$

where $\Delta u = u_{xx} + u_{yy}$. If the solution is the exact solution for the Laplace equation, we have

$$\frac{h^4}{24}u_{xxxyyy}. \tag{25}$$

The scheme described above is not very accurate on unstructured grids because of the inaccuracy of the gradient (19) on such grids. To retain the high accuracy on unstructured grids. we need more accurate approximation to the nodal gradient. Leaset-squares quadratic reconstruction is a good alternative. The simplest strategy is to reconstruct a linear function around a node of interest in the least-squares sense and then evaluate the gradient, which yields the exact gradient for linear solutions. Or we may reconstruct a quadratic function to gain more accuracy, i.e. reconstruct

$$u = ax^2 + 2bxy + cy^2 + dx + ey + f \tag{26}$$

around each node, and then recover the gradient

$$u_x = 2ax + 2by + d, \quad u_y = 2bx + 2cy + e. \tag{27}$$

The computed gradient in this way is exact for quadratic on any grids. Note that the quadratic reconstruction requires 6 nodes, i.e. 5 nodes around the node of interest. The number of immediate neighbors is, however, not always 5 on triangular grids. If it is more than 5, we can compute the coefficients in the least-squares sense. On the other hand, if it is less than 5, we would need to introduce non-immediate neighbors. This can be done by choosing the sufficient number (≥ 5) of nodes from the nearest one. In the case of linear reconstruction, this problem does not happen as we need only three points to construct a liear function.

In the linear case, inverse-distance weighting is known to better condition the least-squares matrix. In the quadratic case, such a weight is not known, and therefore we overcome singular situation by introducing more nodes in the least-squares formulation.

4 High-Order Discretization: Additional Unknowns (P2)

Another approach is to take the midpoint values as unknowns. However it turns out that the same approach as in the previous section, i.e. minimizing the energy norm (6) with high-order gradient, does not work at all. As a matter of fact, it is not really a Galerkin method because such a discrete energy norm is not consistent with the piecewise quadratic element. The consistent energy norm is derived by performing the integration of the energy with the quadratic variation of the solution . The result is

$$
\begin{aligned}
\mathcal{F} &= \frac{\mu}{2} \int \left(u_x^2 + u_y^2\right) dxdy \qquad (28)\\
&= \frac{\mu}{6} \sum_{T \in \{T\}} \left[(u_x^q)^2 + (u_y^q)^2\right] S_T \qquad (29)
\end{aligned}
$$

where

$$
(u_x^q)^2 = (u_x^{T_G})^2 + (u_x^{T_A})^2 + (u_x^{T_B})^2 + (u_x^{T_C})^2 - (u_x^{T})^2, \qquad (30)
$$

similarly for $(u_y^q)^2$. We remark that these quantities are always positive (as they should be). In fact, the terms can be rearranged as

$$
\begin{aligned}
(u_x^q)^2 &= \frac{1}{4}\left\{(u_x^{T_A} - u_x^{T_B})^2 + (u_x^{T_B} - u_x^{T_C})^2 + (u_x^{T_C} - u_x^{T_A})^2\right. \qquad (31)\\
&+ \left.(u_x^{T_A} + u_x^{T_G})^2 + (u_x^{T_B} + u_x^{T_G})^2 + (u_x^{T_C} + u_x^{T_G})^2\right\} > 0, \qquad (32)
\end{aligned}
$$

similarly for $(u_y^q)^2$. For the purpose of deriving a scheme of distribution-type, it is convenient to rewrite the energy norm as

$$\begin{aligned} \mathcal{F} &= \sum_{T\in\{T\}} \left\{ \frac{4}{3} \sum_{T_\xi\in\{T_\xi\}} F_{T_\xi} - \frac{1}{3} F_T \right\} && (33) \\ &= \frac{4}{3} \sum_{T\in\{T\}} \sum_{T_\xi\in\{T_\xi\}} F_{T_\xi} - \frac{1}{3} \sum_{T\in\{T\}} F_T && (34) \end{aligned}$$

where $\{T_\xi\} = \{T_A, T_B, T_C, T_G\}$, and

$$\begin{aligned} F_T &= \frac{\mu}{2} \left[(u_x^T)^2 + (u_y^T)^2 \right] S_T && (35) \\ F_{T_\xi} &= \frac{\mu}{2} \left[(u_x^{T_\xi})^2 + (u_y^{T_\xi})^2 \right] S_{T_\xi}. && (36) \end{aligned}$$

Minimizing this consistnet energy norm, we obtain the following distribution scheme: four-third of the second-order update on each subtriangle,

$$u_i^{n+1} = u_i^n - \frac{4}{3} \left\{ \frac{\omega\mu}{2} (\mathrm{grad} u)^{T_\xi} \cdot \mathbf{n}_i^{T_\xi} \right\} \quad i \in i_{T_\xi} \tag{37}$$

followed by subtracting one-third of the second-order update on the original triangle,

$$u_i^{n+1} = u_i^n + \frac{1}{3} \left\{ \frac{\omega\mu}{2} (\mathrm{grad} u)^{T} \cdot \mathbf{n}_i^{T} \right\} \quad i \in i_T. \tag{38}$$

The resulting scheme is in the form of a collection of the standard second-order Galerkin scheme with appropriate weights, which is a convenient form in upgrading an existing second-order code.

Note that this is equivalent to the Richardson extrapolation applied to the 2nd-order Galerkin method. The weights, 4/3 and −1/3 could have been found readily from this viewpoint.

5 Gradient-Based Discretization (GBD) of Laplacian

If we have solution gradient at each node, we can devise an alternative discretization. We go back to the integral form of the diffusion term, but now *over a triangle.*

$$\int_T \mathrm{div}(\mathrm{grad} u)\, dV = -\oint_{\partial T} \mathrm{grad} u \cdot \mathbf{n}\, dS. \tag{39}$$

We can integrate the line integral on the right using the nodal gradient, assuming that the solution is quadratic. This gives

$$-\oint_{\partial T} \mathrm{grad} u \cdot \mathbf{n}\, dS = -\sum_{edges} (\overline{u_x}, \overline{u_y}) \cdot \mathbf{n} \tag{40}$$

where the overbar indicates the arithmetic mean over the edge. Employing the fluctuation-splitting approach, this is then split edge-wise and distributed to the node opposite to the edge, resulting the following update formula for node j.

$$S_j u_j^{n+1} \;=\; S_j u_j^n - \mu \sum_{T\in\{T_j\}} \left(\overline{u_x}_T, \overline{u_y}_T\right)\cdot \mathbf{n}_T \tag{41}$$

where $\overline{u_x}_T$ is the arithmetic mean of the nodal derivative over the edge (of triangle T) opposite to the node j, similarly for $\overline{u_y}_T$. This scheme can be obtained also by distributing the fluctuation with the equal weight 1/3. It can be shown that this scheme is also minimizing the energy norm defined by

$$\mathcal{F}' \;=\; \frac{\mu}{2}\sum_{Nodes}\left[(u_x)_j^2 + (u_y)_j^2\right] S_j \tag{42}$$

where the nodal gradient is given by (19).

On a regular grid, inserting the expanded solution into the update, we obtain

$$\begin{aligned} du \;=\; & (\Delta u) + \frac{h^2}{3}\left[\partial_{xx} + \partial_{xy} + \partial_{yy}\right](\Delta u) \\ & + h^4\left[\frac{2}{45}\partial_{xxxx} + \frac{1}{12}\partial_{xxxy} + \frac{3}{20}\partial_{xxyy} + \frac{1}{12}\partial_{xyyy} + \frac{2}{45}\partial_{yyyy}\right](\Delta u) \\ & + \frac{h^4}{18}u_{xxxyyy}. \end{aligned} \tag{43}$$

Suppose the solution is the exact solution for the Laplace equation. Then we have the truncation error,

$$\frac{h^4}{18}u_{xxxyyy}. \tag{44}$$

However, this scheme is only 2nd-order accurate (and not very accurate) in practice. The truncation error at a node adjacent to the boundary (an interior node directly connected to boundary nodes) is not $\mathcal{O}(h^4)$ but $\mathcal{O}(h^0)$. This implies that this scheme is inconsistent near boundaries. We could use better gradient reconstruction methods, but we found that the method is unstable with other reconstruction methods (least-squares linear and quadratic) for which minimization property such as (42) is not clear.

6 Results

All the methods have been tested for a Dirichlet problem with the exact solution

$$u \;=\; \frac{\sinh(\pi x)\sin(\pi y) + \sinh(\pi y)\sin(\pi x)}{\sinh(\pi)} \tag{45}$$

in a square domain. Numerical tests were performed on a series of uniform grids ($10\times 10, 20\times 20$ and 40×40; See Fig. 2) and also on a series of unstructured

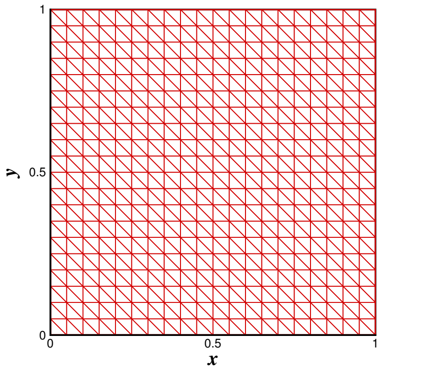

Figure 2: Structured grid. 20x20: 441 Nodes and 800 Triangles.

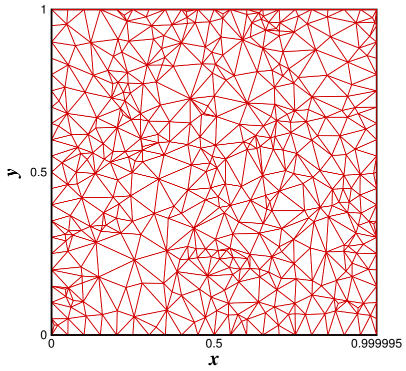

Figure 3: Unstructured grid. 20x20: 441 Nodes and 800 Triangles.

grids with the same boundary node distribution and the same number of (randomly distributed) interior nodes (See Fig. 3). The unstructured grids were generated by a Delaunay triangulation method with a few iterations of Laplacian smoothing. The method was taken to be converged when the L_2 norm of the nodal residual is reduced below 10E-13.

Table 1 shows the L_2 errors for all the methods tested on structured grids (See also 4), where the methods are ordered according to the error level: from the least accurate to the most accurate in the descending order. "GR-EXACT" refers to the method of gradient-reconstruction type with the exact gradient values computed from the exact solution. This is by no means a practical scheme, but included as the ultimate GR-type scheme only just for comparison purpose. As can be seen from the table, the most accurate scheme is the GR-EXACT. Next to that is the P2 method, which is therefore the most accurate among the realistic schemes. Somewhat puzzling is that the GBD scheme is only 2nd order accurate and much less accurate than the standard Galerkin.

Table 2 shows the L_2 errors for all the methods tested on unstructured grids (See also 4). All the scheme have lost their design accuracy on random grids. The best practical scheme is either GR-Q-LS or P2. In terms of ease of implementation, P2 would be a better choice.

7 Remarks

We have tested several high-order schemes for a diffusion operator. In terms of the accuracy, we may conclude that the P2 method be the best. Also, the P2 method can be extended to even higher-order by breaking the triangle into more sub-triangles while such an extension is not clear for other GR methods. Moreover, one can easily modify an existing 2nd-order code to produce the P2 code.

	10×10	20×20	40×40	Order
GBD	2.03E-02	5.14E-03	1.29E-03	1.99
STD Galerkin	2.97E-03	7.08E-04	1.73E-04	2.05
GR-AVE	1.10E-03	1.41E-04	1.79E-05	2.97
GR-L-LS	8.57E-04	9.50E-05	1.15E-05	3.11
GR-Q-LS	2.47E-04	1.57E-05	9.92E-07	3.98
P2	2.28E-05	1.43E-06	8.87E-08	3.99
GR-EXACT	7.39E-06	4.46E-07	2.23E-08	4.04

Table 1: L_2 errors and the order of convergence on structured grids. The order of convergence has been determined by the linear least-squares fit.

	10×10	20×20	40×40	Order
GBD	1.98E-02	7.29E-03	2.37E-03	1.53
STD Galerkin	4.18E-03	1.23E-03	4.32E-04	1.66
GR-AVE	1.88E-03	5.15E-04	1.85E-04	1.67
GR-L-LS	1.29E-03	3.37E-04	1.29E-04	1.66
GR-Q-LS	7.31E-04	4.62E-05	6.62E-06	3.39
P2	1.60E-04	2.34E-05	4.62E-06	2.56
GR-EXACT	5.23E-05	3.43E-06	3.22E-07	3.67

Table 2: L_2 errors and the order of convergence on unstructured grids. The order of convergence has been determined by the linear least-squares fit.

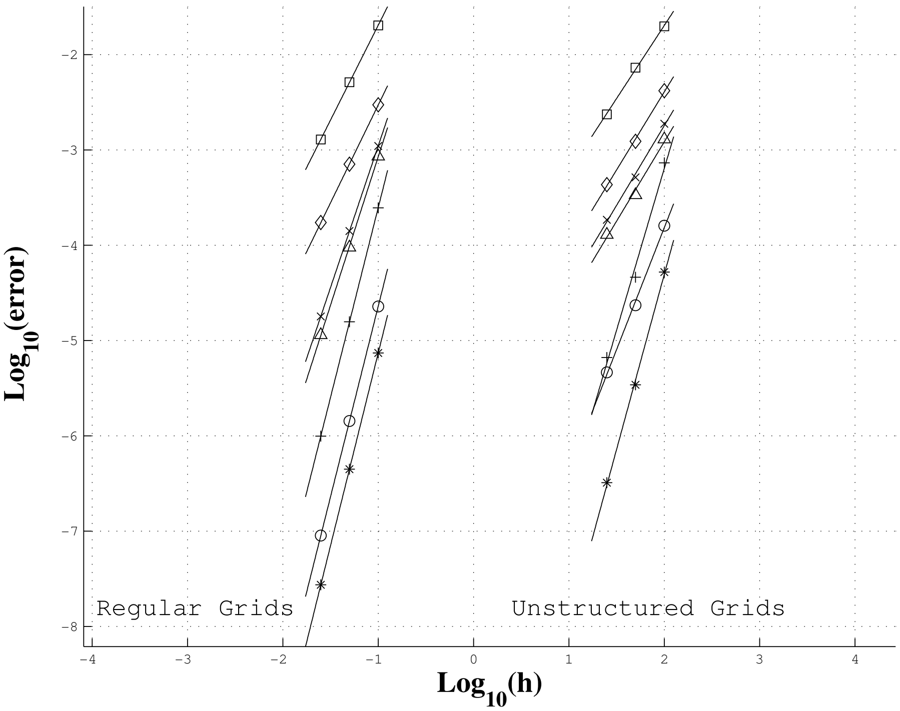

Figure 4: L_2 Error convergence. Left: Results on uniform grids; Right: Results on unstructured grids (plots are shifted). □: GBD; ◇: STD Galerkin; ×: GR-AVE; △: GR-L-LS; +: GR-Q-LS; ∘: P2; ∗: GR-EXACT.