Preprint

Hiroaki Nishikawa,
Higher-Order Discretizatoin of Diffusion Terms in Residual Distribution Methods,
in 34th VKI CFD Lecture Series, Very High Order Discretization Methods, VKI Lecture Series, 2005.

Higher-Order Discretization of Diffusion Terms in Residual-Distribution Methods

H. Nishikawa
Department of Aerospace Engineering,
University of Michigan, Ann Arbor, MI 48105 USA

Abstract

We discuss various higher-order discretization methods for diffusion terms in residual-distribution methods. Categorizing methods into two types: node-based (Galerkin methods) and cell-based (residual-distribution methods), we begin with the description of the basic low-order methods. We then consider two different approaches to extend these methods to higher-order: reconstruction and higher-order elements. Applying these to the low-order methods, we derive and discuss various higher-order methods for diffusion, with more emphasis on the cell-based methods, especially the methods based on the first-order system by which we can avoid discretizing second-derivatives. Numerical results are given for a simple test problem to demonstrate the accuracy of the derived schemes. In particular, it is shown that employing the first-order system we can achieve fourth-order accuracy with P_2 elements. Finally, we discuss an issue for integrating these diffusion schemes with advection schemes, comparing two possible ways to construct higher-order advection-diffusion schemes.

1 Introduction

1.1 Motivation

Residual-distribution methods (or fluctuation-splitting methods) are discretization methods based on nodal solutions and cell-residuals. These methods have been developed extensively for problems dominated by advection and wave propagation because of the ability to reflect multidimensional physics of the governing equations. In particular, various multidimensional upwinding schemes have been derived from this residual-distribution methodology. On the other hand, little has been studied regarding its application to diffusion-dominated problems, obviously because diffusion is an isotropic process and does not benefit particularly from such a multidimensional capability. In fact, any naive method such as the standard Galerkin method would work equally well for such problems as long as the discretization is stable and consistent. This is exemplified by the fact that residual-distribution Navier-Stokes codes have been constructed commonly by adding the Galerkin discretization of the viscous term, which is not a residual-distribution method, to an existing residual-distribution Euler code [1, 2, 3]. An issue arises, however, in regions where advection and diffusion effects are equally important: methods for advection and diffusion may not be compatible and the overall accuracy can then be lost significantly. Such a region always exists in the middle of a boundary layer. This was pointed

out in [4] for the advection-diffusion equation, a model equation for the Navier-Stokes equations,

$$u_t + (a, b) \cdot \text{grad}u = \nu \, \text{div} \, (\text{grad}u) \tag{1}$$

showing that schemes obtained simply by adding the Galerkin discretization to an residual-distribution advection scheme indeed lost their formal accuracy. This incompatibility issue is extremely important especially when we construct higher-order methods because any high-order accuracy designed for each individual term may be lost altogether. To improve the compatibility between advection and diffusion discretization methods, one possible solution is proposed in [4]; but there are others [5, 6]. Before we conclude anything about this issue, it would be very important and meaningful to delve into discretization methods for diffusion-dominated problems. Note that the Galerkin method is not the only way to discretize the diffusion term. There are methods of residual-distribution type for diffusion [7, 8, 4], and these methods have several advantages over the Galerkin method. One such is the residual-property that a scheme preserves exact polynomial solutions of certain order on arbitrary grids. This property is partly responsible to the reduced mesh sensitivity of the residual-distribution methods, and therefore it has been a basis of advection schemes but not so much as stressed for diffusion schemes. Therefore, by studying what options are available for diffusion discretization and how they can be extended to higher-order, we enrich our knowledge of diffusion schemes, thus possibly leading to better ideas on how to synthesize them with higher-order advection schemes to develop higher-order advection-diffusion schemes. And this is precisely the purpose of this lecture, i.e. to explore various methods for diffusion terms that can be used in the residual-distribution framework, focusing on residual-distribution and higher-order accuracy.

1.2 Residual-Distribution Methods

We call methods residual-distribution if they can be factored into the two steps, *residual evaluation and distribution*. Consider, for example, solving a pure advection equation

$$u_t + a \, u_x + b u_y = 0 \tag{2}$$

in a certain domain with appropriate boundary conditions. We begin by dividing the domain of interest into a set of triangles $\{T\}$, with a set of nodes $\{J\}$, and store the solution values at nodes. For each triangular cell $T \in \{T\}$, we evaluate a local cell-residual (or fluctuation), ϕ^T,

$$\phi^T = - \iint_T (a \, u_x + b u_y) \, dxdy \tag{3}$$

which becomes, for piecewise linear approximation of u,

$$\phi^T = - \sum_{i \in \{i_T\}} k_i u_i, \quad k_i = \frac{1}{2} (a, b) \cdot \mathbf{n}_i \tag{4}$$

where $\{i_T\}$ denotes a set of nodes that form the cell T and $\mathbf{n}_i$ is the scaled inward normal vector of the edge opposite to node i. This defines a measure of the error in satisfying the equation (2) over the triangular element. And therefore, if the cell-residual is not zero,

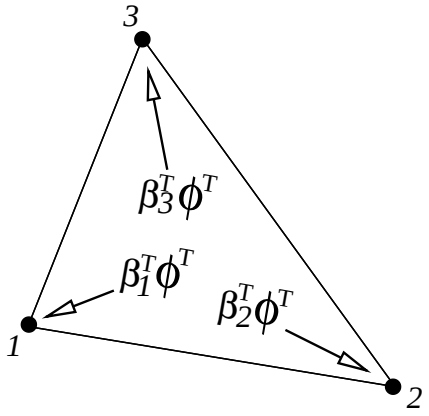


Figure 1: Distribution of non-zero cell-residual. $\{i_T\} = \{1, 2, 3\}$

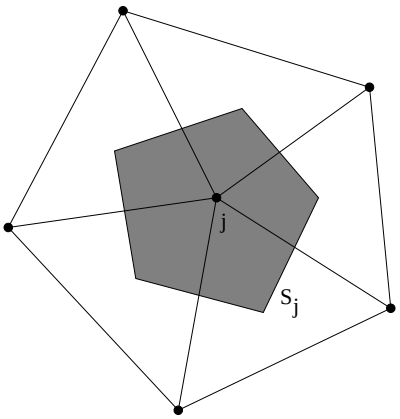


Figure 2: Median dual cell around node j created by connecting the centroids of the triangles $\{T_j\}$. S_j denotes the area.

we must change the nodal solutions to reduce the measure. This brings the distribution step. We determine a fraction of ϕ^T to be distributed to the nodes, ϕ_i^T by

$$\phi_i^T = \beta_i^T \phi^T \quad i \in \{i_T\} \tag{5}$$

(see Figure 1) where β_j^T is a distribution coefficient with the property

$$\sum_{i \in \{i_T\}} \beta_i^T = 1 \tag{6}$$

for conservation. Note that the cell-residual (4) vanishes for exact linear solutions, nothing will be distributed then, and the solution is preserved as a result. This is the residual-property, which is independent of the shape of the cell, thus resulting a superior accuracy on unstructured grids. Now, accumulating the partial residuals distributed, we update the solution at node j by

$$S_j u_j^{n+1} \;=\; S_j u_j^n + \Delta t_j \sum_{T \in \{T_j\}} \phi_j^T \tag{7}$$

where S_j is the median dual cell area, Δt_j is a timestep, and $\{T_j\}$ denotes a set of triangles that share node j (see Figure 2). It is the distribution step that gives the multidimensional capability to the residual-distribution methods: there are three directions for upwinding while there are only two for methods based on one-dimensional flux functions (along or across an edge). In the case of diffusion problems, however, it is natural to perform the distribution isotropically, and so such a capability is not a great advantage. In fact, a popular discretization method for diffusion terms in the residual-distribution methods is the Galerkin method which is not residual-distribution because there exist no cell-residuals. There are residual-distribution methods for diffusion terms [8, 7], i.e. methods based on cell-residuals, but they are very few and have not been extensively explored. The present lecture will place a particular emphasis on these methods.

1.3 Model Equation

We focus on the diffusion part of the advection-diffusion equation (1),

$$u_t = \mathrm{div}\,(\mathrm{grad} u) \tag{8}$$

where we have taken $\nu = 1$. This is a typical heat equation. Obviously, there is no preferred direction along which the solution travels, and therefore any numerical methods for solving this should have a symmetric stencil or isotropic distribution in the case of residual-distribution schemes. As we are mainly interested in the steady solution, this is equivalent to solving the Laplace equation,

$$\mathrm{div}\,(\mathrm{grad}u) = 0 \quad \text{in } \Omega \tag{9}$$

which reads in two-dimensional Cartesian coordinates

$$u_{xx} + u_{yy} = 0 \quad \text{in } \Omega \tag{10}$$

with a Dirichlet condition

$$u = g \quad \text{in } \partial\Omega \tag{11}$$

where g is a given function. It is well known that the solution of the Laplace equation minimizes the so-called energy norm,

$$\mathcal{F} = \frac{1}{2}\iint_{\Omega} \mathrm{grad}u \cdot \mathrm{grad}u \; dV \tag{12}$$

This interpretation is commonly used to prove the existence and the uniqueness of the solution of the Laplace equation [9], but here this is particularly useful in deriving numerical schemes. It is also useful to consider discretizing the model equation (1) in the equivalent first-order form,

$$u_t = p_x + q_y \tag{13}$$
$$p = u_x \tag{14}$$
$$q = u_y \tag{15}$$

This introduces additional variables and equations, but offers several advantages for designing numerical schemes.

1.4 Outline

Just for the sake of convenience, here we split methods for diffusion into two categories: node-based and cell-based methods. Note that the solution values are stored at nodes in both methods and so this categorization is not based on the location of the numerical solution but on the location where the residual is defined. In the node-based methods, which is typically the Galerkin method, the Laplace equation is discretized directly at nodes where the numerical solution resides. This defines residuals at nodes, and there exist no cell-residuals. In this sense, these methods are not residual-distribution methods. On the other hand, in the cell-based methods, the equation is first discretized over cells defining cell-residuals, and then distributed to nodes to drive the evolution of the nodal solutions. Therefore, these are residual-distribution methods. The basic low-order schemes in these two categories are described in Section 2. Section 3 describes two different approaches to higher-order accuracy: reconstruction and high-order elements. Section 4 and 5 discuss higher-order extensions of node-based and cell-based methods respectively. Section 6 presents computational results. Section 7 gives final remarks.

2 Basic Methods

2.1 Node-Based Schemes (Finite-Element Type)

We consider the Galerkin finite-element method. Galerkin-based schemes can be derived in many different ways. An example is the weighted residual formulation. Suppose we have divided the domain into a set of triangles $\{T\}$, store the solution values u_j at nodes $\{J\}$, and introduce the basis functions φ_j by which the numerical solution u_h is represented by

$$u_h \;=\; \sum_{i\in\{J\}} u_i\,\varphi_i \tag{16}$$

Let v be a weight function which vanishes on the domain boundary. Then, multiply the Laplace equation by v and integrate by parts over the domain

$$\iint_\Omega v\,\mathrm{div}(\mathrm{grad}u_h)\,dV \;=\; \oint_{\partial\Omega} v\,\mathrm{grad}u_h\cdot\mathbf{n}\,ds - \iint_\Omega \mathrm{grad}v\cdot\mathrm{grad}u_h\;dV = 0 \tag{17}$$

which becomes, because v vanishes on $\partial\Omega$,

$$\iint_\Omega \mathrm{grad}v\cdot\mathrm{grad}u_h\;dV = 0 \tag{18}$$

This is just one equation. To generate a sufficient number of equations to determine the nodal solutions, we must look for as many distinct functions for v as the number of unknowns. A natural choice would be the basis function φ_j itself, and this is the Galerkin discretization,

$$\iint_\Omega \mathrm{grad}\varphi_j\cdot\mathrm{grad}u_h\;dV = 0 \tag{19}$$

for all interior nodes j. Note that the basis function usually has a compact support, so that the integral above is not over the entire domain but typically over a set of triangles $\{T_j\}$ around node j. Note that the same discretization can be obtained by minimizing the energy norm,

$$\mathcal{F} = \frac{1}{2}\iint_\Omega \mathrm{grad}u_h\cdot\mathrm{grad}u_h\;dV \tag{20}$$

with respect to the nodal unknowns. At a minimum, the derivatives of $\mathcal{F}$ with respect to the nodal solutions must vanish, and therefore we have again

$$\frac{\partial\mathcal{F}}{\partial u_j} = \iint_\Omega \mathrm{grad}\varphi_j\cdot\mathrm{grad}u_h\;dV = 0 \tag{21}$$

for all interior nodes j.

In whatever way the discretization is obtained, it is necessary to solve a set of simultaneous equations for the nodal unknowns. Here, we consider iterative methods in the form,

$$u_j^{n+1} \;=\; u_j^n - \omega_j \iint_\Omega \mathrm{grad}\varphi_j\cdot\mathrm{grad}u_h\;dV \tag{22}$$

where n is the iteration index and ω_j is a constant small enough to ensure the stability. The constant ω_j can be estimated by taking Newton's iteration, but with only the diagonal elements in the Hessian matrix, i.e.

$$u_j^{n+1} = u_j^n - c_j \left(\frac{\partial^2 \mathcal{F}}{\partial u_j^2} \right)^{-1} \frac{\partial \mathcal{F}}{\partial u_j} \tag{23}$$

$$= u_j^n - c_j \left(\frac{\partial^2 \mathcal{F}}{\partial u_j^2} \right)^{-1} \iint_\Omega \text{grad}\varphi_j \cdot \text{grad}u_h \; dV \tag{24}$$

Comparing this with (22), we find

$$\omega_j = c_j \left(\frac{\partial^2 \mathcal{F}}{\partial u_j^2} \right)^{-1} = c_j \left(\iint_\Omega \text{grad}\varphi_j \cdot \text{grad}\varphi_j \; dV \right)^{-1} \tag{25}$$

and therefore c_j is a safety factor that is less than or equal to unity. If we write (22) in the form,

$$S_j u_j^{n+1} = S_j u_j^n - \Delta t_j \iint_\Omega \text{grad}\varphi_j \cdot \text{grad}u_h \; dV \tag{26}$$

then the condition on c_j is translated into, via $\Delta t_j = S_j \omega_j$,

$$\Delta t_j \leq S_j \left(\iint_\Omega \text{grad}\varphi_j \cdot \text{grad}\varphi_j \; dV \right)^{-1} \tag{27}$$

In this form, the scheme looks as if it were a residual-distribution scheme. Even more so, if we assume that the support of φ_j is $\{T_j\}$ and write

$$S_j u_j^{n+1} = S_j u_j^n + \Delta t_j \sum_{T \in \{T_j\}} \phi_j^T \tag{28}$$

where

$$\phi_j^T = - \iint_{T \in \{T_j\}} \text{grad}\varphi_j \cdot \text{grad}u_h \; dV \tag{29}$$

However, this is not a residual-distribution scheme. To see this, consider the total cell-residual over a triangle T, ϕ^T, which is given by

$$\phi^T = \sum_{i \in \{i_T\}} \phi_i^T = - \sum_{i \in \{i_T\}} \iint_T \text{grad}\varphi_i \cdot \text{grad}u_h \; dV \tag{30}$$

$$= - \iint_T \sum_{i \in \{i_T\}} (\text{grad}\varphi_i) \cdot \text{grad}u_h \; dV \tag{31}$$

$$= - \iint_T \text{grad} \left(\sum_{i \in \{i_T\}} \varphi_i \right) \cdot \text{grad}u_h \; dV \tag{32}$$

But we must always have $\sum_{i \in \{i_T\}} \varphi_i = 1$ for consistency. Therefore, the cell-residual is identically zero,

$$\phi^T = 0 \tag{33}$$

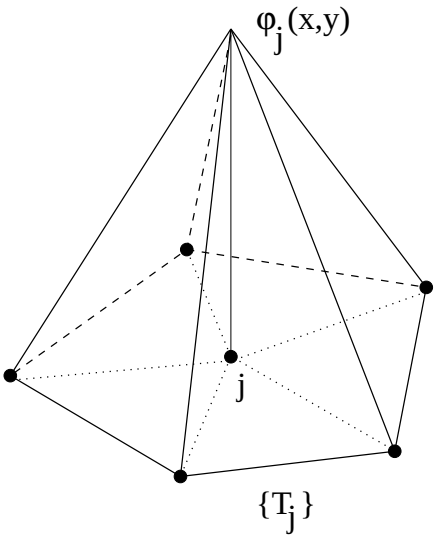


Figure 3: Linear basis function $\varphi_j(x, y)$ for node J. $\{T_j\}$ denotes the set of triangles that share node j.

and hence the Galerkin scheme is not a residual-distribution scheme because the changes in the nodal solutions are not driven by nonzero cell-residuals. This means also that these schemes do not have the residual property.

An example is the standard Galerkin discretization (P_1 Galerkin). This is derived by choosing piecewise linear functions for the basis function φ_j (see Figure 3). In this case, the support of φ_j is $\{T_j\}$, and the gradients are constants within each $T \in \{T_j\}$ and given by

$$(\text{grad}\varphi_i)|_T = \frac{\mathbf{n}_i}{2S_T} \tag{34}$$

$$(\text{grad}u_h)|_T = \frac{1}{2S_T}\sum_{i\in i_T} u_i \mathbf{n}_i \tag{35}$$

where S_T is the area of cell T. From here on, we denote the constant gradients associated with triangle T, such as (34) and (35), simply by using the superscript T, i.e. $(\text{grad}u)^T = (\text{grad}u_h)|_T$ and $u_x^T = \frac{\partial u_h}{\partial x}|_T$. Then, the scheme (26) becomes

$$S_j u_j^{n+1} = S_j u_j^n - \Delta t_j \sum_{T\in\{T_j\}} (\text{grad}u)^T \cdot \mathbf{n}_T \tag{36}$$

where and $\mathbf{n}_T$ is the scaled inward normal of the edge opposite to the node j in the triangle T and

$$\Delta t_j \leq S_j \left(\sum_{T\in\{T_j\}} \frac{\mathbf{n}_T \cdot \mathbf{n}_T}{4S_T} \right)^{-1} \tag{37}$$

This scheme can also be derived by directly integrating the Laplacian over the median dual cell, and also the same condition on the timestep can be obtained from a stability property of the scheme (See [1, 8, 10] for details). Note that we obtain the same scheme

also by minimizing the energy norm (20) which is now constant and can be written as

$$\mathcal{F} = \frac{1}{2} \sum_{T \in \{T\}} \left[(u_x^T)^2 + (u_y^T)^2 \right] S_T \tag{38}$$

This interpretation is useful in deriving higher-order Galerkin schemes as we shall see later. Naturally, because of the assumption of the linear variation of the solution, this scheme is second-order accurate.

2.2 Cell-based Schemes (Residual Distribution Type)

We consider schemes of residual-distribution type, i.e. those that distribute a nonzero cell-residual to nodes. We begin by defining the cell-residual for the Laplace equation, ϕ^T. But in the case of the piecewise linear approximation, we immediately find that the cell-residual is identically zero,

$$\phi^T = \iint_T \mathrm{div}(\mathrm{grad} u_h)\, dxdy \equiv 0 \tag{39}$$

In order to define a nonzero cell-residual, we need to avoid dealing with second derivatives somehow. To this end, we introduce, for the sake of convenience, the gradient variables $(p, q) = (u_x, u_y)$, assume a certain variation of them over a triangle, and then evaluate the cell-residual as

$$\phi^T = \iint_T (p_x + q_y)\, dxdy \tag{40}$$

Note that p and q must be continuous across (at least) two neighboring cells for conservation. Suppose now that solution gradients are available at nodes. Then, they can be linearly interpolated over a cell, and the cell-residual is evaluated exactly and given in any of the following forms,

$$\phi^T = (p_x^T + q_y^T) S_T = \frac{1}{2} \sum_{i \in \{i_T\}} (p_i, q_i) \cdot \mathbf{n}_i = - \sum_{i \in \{i_T\}} (\overline{p}_i, \overline{q}_i) \cdot \mathbf{n}_i \tag{41}$$

where (p_i, q_i) denotes the gradient at node i, and the overbar denotes the arithmetic mean of the nodal gradients along the edge opposite to the node i. Once the cell-residual is evaluated, we distribute it to the nodes with a distribution coefficient β_j^T, resulting the update at the node j,

$$S_j u_j^{n+1} \;=\; S_j u_j^n + \Delta t_j \sum_{T \in \{T_j\}} \phi_j^T \tag{42}$$

where

$$\phi_j^T = \beta_j^T \phi^T \tag{43}$$

The distribution must be central to reflect the nature of diffusion, and we set

$$\beta_j^T = \frac{1}{3} \tag{44}$$

This scheme is a residual-distribution scheme because the changes are driven by nonzero cell-residuals. Now, we consider two different strategies to obtain the nodal gradients.

1. Gradient Reconstruction

2. Solving $p - u_x = 0$ and $q_y - u_y = 0$ for p and q

In the first strategy, we reconstruct gradients at nodes from the numerical solution of u, $\{u_i\}$, and directly evaluate the cell-residual using them. A scheme is then defined by distributing this cell-residual isotropically, which we refer to as nodal gradient scheme. We will discuss this scheme in detail later, including the choice of the gradient reconstruction method. Note that because gradient reconstruction methods typically requires all immediate neighbors of a node of interest, this results in a wide stencil for defining a cell-residual (see Figure 4-(a)). In the second strategy, essentially, we convert the diffusion equation into the first-order system

$$\begin{aligned} u_t &= p_x + q_y && (45)\\ p - u_x &= 0 && (46)\\ q - u_y &= 0 && (47) \end{aligned}$$

and solve for u as well as the new unknowns p and q stored at nodes. A great advantage of this approach is that no reconstruction is required and schemes can be very compact. The cell-residuals for a triangle can be defined within that triangle without any reference to the neighbors (see Figure 4-(b)). The first equation is solved by the distribution scheme (42), and the two slope equations are solved in a similar way, for example by defining, with p and q represented by the same linear basis function,

$$\phi_p^T = \iint_T (p - u_x)\, dxdy = (\overline{p}^T - u_x^T) S_T \tag{48}$$

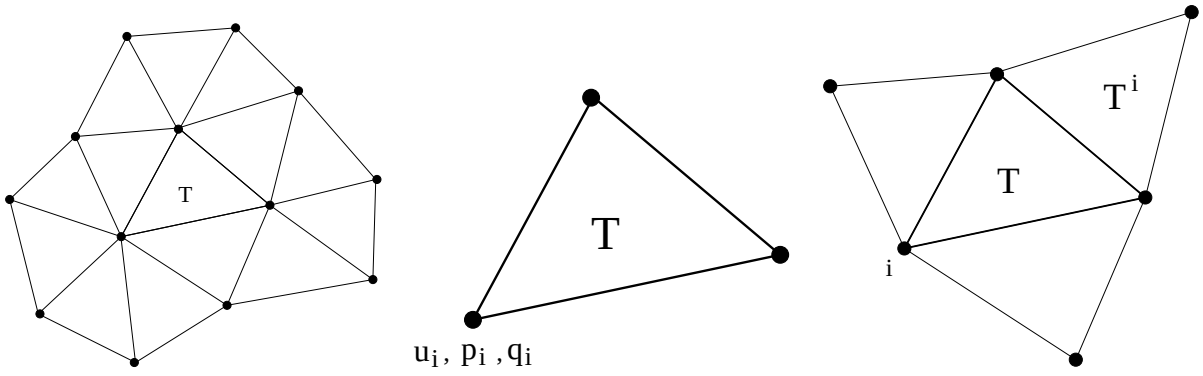
where $\overline{p}^T = (p_1 + p_2 + p_3)/3$, and distributing with the same coefficients (44)

$$S_j p_j^{n+1} = S_j p_j^n - \Delta t_j \sum_{T \in \{T_j\}} \beta_j^T \, \phi_p^T \tag{49}$$

similarly for q; this scheme minimizes the residuals ϕ_p^T in an L_2 norm. Note that the cell-residuals for p and q vanish for exact linear solutions of u and exact constant solutions of p and q, and that the cell-residual (41) also vanish for such exact solutions. Hence, we have the residual-property that the scheme preserves exact linear solutions of u and constant solutions of p and q, implying second-order accuracy for u and first-order accuracy for p and q. We refer to this scheme as P_1 FOS scheme. As will be shown later, this scheme can easily be extended to higher-order. The use of the first-order system for discretization of diffusion terms can be found also in discontinuous Galerkin methods [11]. It is the advantage of discontinuous basis functions that they allow the slope equations to be solved explicitly for p and q in terms of u in each computational cell, so that there is no need to store the variables p and q. In residual-distribution methods, because of the continuous representation of the variables, they are globally coupled and here we suggest to solve for them iteratively as in (49) with the solution u.

Yet another strategy for defining the cell-residual is to recover the gradients along the cell edges and evaluate the cell-residual as a line integral

$$\phi^T = \iint_T (p_x + q_y)\, dxdy = \oint_{\partial T} (p\, dy - q\, dx) \tag{50}$$



(a) Nodal Gradient Reconstruction

(b) First-Order System (no reconstruction needed)

(c) Edgewise Gradient Reconstruction

Figure 4: Stencils required for defining a cell-residual in the three approaches.

This strategy was considered by Paillére *et al.* [8]. They proposed to estimate the gradient by a simple average of the constant gradients of two neighboring triangles that share an edge, i.e.

$$(\tilde{p}_i, \tilde{q}_i) = \frac{1}{2}\left((\mathrm{grad}u)^T + (\mathrm{grad}u)^{T^i}\right) \tag{51}$$

or the area-weighted average

$$(\tilde{p}_i, \tilde{q}_i) = \frac{S_T\,(\mathrm{grad}u)^T + S_{T^i}\,(\mathrm{grad}u)^{T^i}}{S_T + S_{T^i}} \tag{52}$$

where $(\tilde{p}_i, \tilde{q}_i)$ denotes the estimated gradient on the edge opposite to node i and that edge is shared by triangle T and its neighbor T^i (see Figure 4-(c)). Collecting the contributions from all edges, we obtain the cell-residual ϕ^T, written as a sum over the nodes,

$$\phi^T = -\sum_{i\in\{i_T\}} (\tilde{p}_i, \tilde{q}_i)\cdot \mathbf{n}_i \tag{53}$$

Note that the stencil extends but only to the immediate neighbors as shown in Figure 4-(c). This cell-residual is then distributed by (42) to the nodes (upwind schemes are used in [8], which are not appropriate for pure diffusion problems we consider here), resulting a second-order accurate diffusion scheme. In order to extend this method to higher-order, generalizing their method, we introduce the following approach: reconstruct the solution u over the two triangles that share an edge, differentiate it to obtain the gradients along the edge, and then evaluate the line integral by a quadrature rule. In the case of P_1 elements, four vertices are available over two elements, and therefore a bilinear reconstruction of u is possible. As will be shown later, this yields precisely the cell-residual (53) with the area-weighted gradient (52). And this approach extends to higher-order in a natural way as will be discussed later.

3 Approaches to Higher-Order Accuracy

3.1 Higher-Order Cell Gradients

Higher-order accuracy (higher than second-order) requires more accurate representation of the solution (better than linear). Here, we consider a quadratic variation of the solution over a cell, which can be constructed formally by introducing midpoints on the sides of the cell (see Figure 5). Over such a cell, we can evaluate the cell-average of the solution gradient more accurately, which is often needed for higher-order schemes, by Simpson's rule,

$$
\begin{aligned}
(u_x)^T_{high} &= \frac{1}{S_T}\iint_T u_x\, dxdy && (54)\\
&= \frac{1}{S_T}\oint u\, dy && (55)\\
&= \frac{1}{6S_T}\sum_{edges}(u_L + 4u_m + u_R)\Delta y && (56)
\end{aligned}
$$

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, e.g. $\Delta y = y_3 - y_2$ for edge 2-3 of the triangle in Figure 5. A similar expression can be obtained for u_y. Several rearrangements are possible for this formula of which useful ones are

$$
\begin{aligned}
(u_x)^T_{high} &= \frac{1}{S_T}\sum_{edges}\left\{\overline{u} - \frac{2}{3}(\overline{u} - u_m)\right\}\Delta y && (57)\\
&= u_x^T - \frac{2}{3}\left(u_x^T - u_x^{T_{IV}}\right) && (58)\\
&= \frac{1}{6}\left(u_x^{T_I} + u_x^{T_{II}} + u_x^{T_{III}} + 3u_x^{T_{IV}}\right) && (59)\\
&= u_x^T - \frac{1}{3}\left(u_x^{T_I} + u_x^{T_{II}} + u_x^{T_{III}} - 3u_x^{T_{IV}}\right) && (60)
\end{aligned}
$$

where the overbar indicates the arithmetic mean over the edge. It is important to note here that this evaluation is independent of the solution variation inside the triangle because this is obtained from a line integral around the boundary, and that it is *fourth-order accurate* because Simpson's rule integrates cubic polynomials exactly, provided of course the midpoint values are sufficiently accurate. Now, we consider two approaches to define the solution values at the midpoints.

3.2 Reconstruction

Solutions at the midpoints may be obtained by a high-order interpolation. This approach was first proposed by Caraeni and Fuchs to develop a third-order residual-distribution Navier-Stokes code [7]. They showed by least-squares fitting that the midpoint value u_m can be interpolated by

$$
u_m = \overline{u} - \frac{1}{8}\Delta U_{ss} \qquad (61)
$$

where

$$
\Delta U_{ss} = \Delta(u_x)\Delta x + \Delta(u_y)\Delta y \qquad (62)
$$

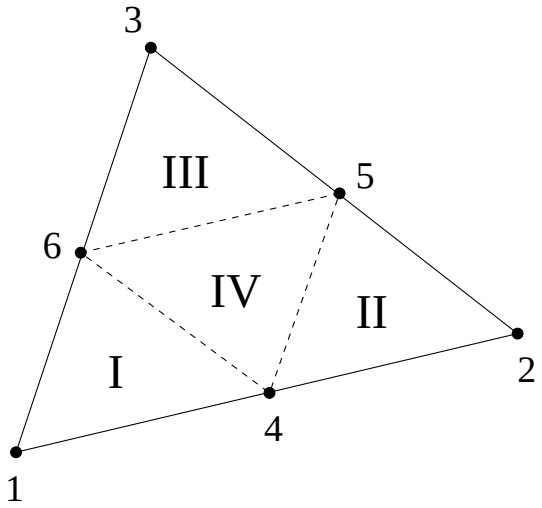


Figure 5: P_2 element

and $\Delta(u_x)$ is the difference of u_x between the end nodes of the edge, similarly for $\Delta(u_y)$, assuming the nodal gradients are available. The same formula can be obtained by using the Hermite cubic interpolation along the edge. This formula requires the nodal gradients, and they are obtained, for example, by the Green-Gauss reconstruction,

$$(u_x)_j = \frac{\sum\limits_{\{T_j\}} S_T u_x^T}{\sum\limits_{\{T_j\}} S_T}, \quad (u_y)_j = \frac{\sum\limits_{\{T_j\}} S_T u_y^T}{\sum\limits_{\{T_j\}} S_T}. \tag{63}$$

Of course, other reconstruction techniques, such as least-squares methods with linear or quadratic functions, can be used, and generally they are more accurate on unstructured grids [12, 13]. Note that the quadratic reconstruction requires 6 nodes at least, i.e. 5 nodes around a node of interest. The number of immediate neighbors is, however, not always 5 on triangular grids. If it happens to be less than 5, we would need to introduce non-immediate neighbors, which is not desirable for compactness. In the case of linear reconstruction, this problem does not happen as we need only three points to construct a linear function. In the linear case, inverse-distance weighting is known to better condition the least-squares matrix[14].

Using the interpolated solution values at the midpoints (61), we can now evaluate the higher-order cell-gradients. This can be done by a quadrature directly applied to the element with 6 nodes (a P_2 element) as in Caraeni and Fuchs [7] by taking a finite-element viewpoint. Or we can substitute (61) into (57) to get the following simple expressions.

$$u_x^T|_{high} = u_x^T - \frac{1}{12S_T} \sum_{edges} \Delta U_{ss} \Delta y \tag{64}$$

$$u_y^T|_{high} = u_y^T + \frac{1}{12S_T} \sum_{edges} \Delta U_{ss} \Delta x \tag{65}$$

which can be interpreted as the low-order gradients (constant gradients from the piecewise linear approximation) with high-order corrections (curvature terms). Nishikawa *et al.* developed a third-order residual-distribution Euler code based on this correction

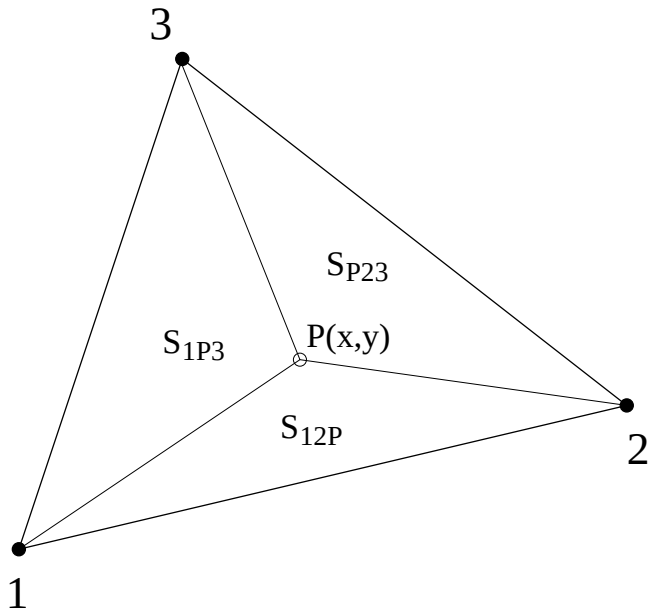


Figure 6: Split of a triangle for defining local area-coordinates

approach [15]. This is particularly useful when we extend existing second-order residual-distribution codes to higher-order because the only change in the code is to add these correction terms to the cell gradients used to evaluate the cell-residuals.

3.3 Higher-Order Elements

Gradient reconstruction may be inaccurate on fully unstructured irregular grids. Also, if we employ quadratic reconstruction, we may need to introduce non-immediate neighbors and we thus lose one of the advantages of the residual-distribution schemes, i.e. compactness. An obvious alternative is to take solution values at midpoints as additional unknowns. In this case, triangular elements with midpoints can be regarded as P_2 elements over which a quadratic variation of the solution can be constructed. The quadratic solution is written, with 6 degrees of freedom $\{u_i\}$ available, as

$$u = \sum_{i=1}^{6} u_i N_i \tag{66}$$

with the quadratic shape functions N_i defined by

$$\begin{array}{rcl} N_1 &=& (2L_1 - 1)L_1 \\ N_2 &=& (2L_2 - 1)L_2 \\ N_3 &=& (2L_3 - 1)L_3 \\ N_4 &=& 4L_1L_2 \\ N_5 &=& 4L_2L_3 \\ N_6 &=& 4L_3L_1 \end{array} \tag{67}$$

where L_1, L_2, and L_3 are the local area-coordinates of the linear triangle 123 defined by

$$L_1 = \frac{S_{p23}}{S_{123}}, \quad L_2 = \frac{S_{1p3}}{S_{123}}, \quad L_3 = \frac{S_{12p}}{S_{123}} \tag{68}$$

(See Figure 6, and see [16] for more details) . Over the P_2 element, we have well-structured self-similar elements within a triangle, which enables us to form a quadratic function with no ambiguity or irregularity. Roe and Abgrall made use of this type of element to develop high-order residual-distribution advection schemes [17]. Also, it is an advantage of this approach that schemes can be extended to even higher-order by introducing more nodes in the element while further extension is not clear in the reconstruction methods. Note that the quadratic variation implies third-order accuracy, but the cell-gradients derived in Section 3.1 are still fourth-order accurate because of the special property of Simpson's rule as mentioned earlier.

4 Higher-Order Node-Based Schemes

4.1 Higher-Order Galerkin by Reconstruction (RC Galerkin)

Recall that the standard Galerkin scheme,

$$u_j^{n+1} = u_j^n - \frac{\omega_j}{2} \sum_{T \in \{T_j\}} (\text{grad}u)^T \cdot \mathbf{n}_T \tag{69}$$

can be derived by minimizing the discrete energy norm

$$\mathcal{F} = \frac{1}{2} \sum_{T \in \{T\}} \left[(u_x^T)^2 + (u_y^T)^2 \right] S_T \tag{70}$$

Then, to derive a higher-order scheme, we define a discrete energy norm with upgraded cell-gradients, $u_x^T|_{high}$ and $u_y^T|_{high}$ given by (64) and (65) in Section 3.2,

$$\mathcal{F} = \frac{1}{2} \sum_{T \in \{T\}} \left[\left(u_x^T|_{high} \right)^2 + \left(u_y^T|_{high} \right)^2 \right] S_T \tag{71}$$

and minimize this to derive the following scheme

$$u_j^{n+1} = u_j^n - \frac{\omega_j}{2} \sum_{T \in \{T_j\}} (\text{grad}u)^T|_{high} \cdot \mathbf{n}_T \tag{72}$$

where the correction terms in (64) and (65) have been taken as purely numerical values, thus resulting the same form as the second-order version, i.e. the only difference between (69) and (72) is the evaluation of the cell-gradients. This scheme has been confirmed to be third-order accurate with the Green-Gauss reconstruction or linear least-squares, and fourth-order accurate with quadratic least-squares for regular triangular grids, but generally one order less for unstructured grids [4].

4.2 P_2 Galerkin Scheme

Another approach is to take the midpoint values as unknowns, i.e. the use of P_2 elements. In this case, we need to derive a scheme to update midpoint solutions. It is tempting to use the discrete energy norm (70) once again with high-order cell-gradients (but not with the gradient reconstruction this time), and derive a scheme by minimizing this with respect to the nodal values including midpoints. However, it turns out that the resulting

scheme is only second-order accurate. In fact, such a method is not really consistent with the quadratic variation of the solution. A correct way to derive a P_2 scheme is to minimize the energy norm consistent with the piecewise quadratic variation. Evaluating the energy norm directly, we obtain

$$\mathcal{F} = \frac{1}{2} \iint \left(u_x^2 + u_y^2\right)\, dxdy = \frac{1}{6} \sum_{T\in\{T\}} \left[\widehat{u_x^2} + \widehat{u_y^2}\right] S_T \tag{73}$$

where

$$\widehat{u_x^2} \;=\; (u_x^{T_{IV}})^2 + (u_x^{T_I})^2 + (u_x^{T_{II}})^2 + (u_x^{T_{III}})^2 - (u_x^T)^2, \tag{74}$$

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

$$\widehat{u_x^2} \;=\; \frac{1}{4}\left\{(u_x^{T_I} - u_x^{T_{II}})^2 + (u_x^{T_{II}} - u_x^{T_{III}})^2 + (u_x^{T_{III}} - u_x^{T_I})^2 \right. \tag{75}$$

$$+ \quad \left. (u_x^{T_I} + u_x^{T_{IV}})^2 + (u_x^{T_{II}} + u_x^{T_{IV}})^2 + (u_x^{T_{III}} + u_x^{T_{IV}})^2\right\} > 0, \tag{76}$$

similarly for $\widehat{u_y^2}$. It follows from (74) that the energy norm can be expanded and written as

$$\mathcal{F} \;=\; \sum_{T\in\{T\}} \left\{ \frac{4}{3} \sum_{T_\xi\in\{T_\xi\}} F_{T_\xi} - \frac{1}{3} F_T \right\} \tag{77}$$

$$\;=\; \frac{4}{3} \sum_{T\in\{T\}} \sum_{T_\xi\in\{T_\xi\}} F_{T_\xi} - \frac{1}{3} \sum_{T\in\{T\}} F_T \tag{78}$$

where $\{T_\xi\} = \{T_I, T_{II}, T_{III}, T_{IV}\}$, and

$$F_T \;=\; \frac{1}{2}\left[(u_x^T)^2 + (u_y^T)^2\right] S_T \tag{79}$$

$$F_{T_\xi} \;=\; \frac{1}{2}\left[(u_x^{T_\xi})^2 + (u_y^{T_\xi})^2\right] S_{T_\xi} \tag{80}$$

which are the energy norms based on piecewise linear approximations within triangle T and the subtriangles $\{T_\xi\}$. Minimizing $\mathcal{F}$, we therefore obtain the following scheme: four-third of the second-order update on each subtriangle,

$$u_i^{n+1} \;=\; u_i^n - \frac{4}{3}\left\{\frac{\omega_i}{2}(\mathrm{grad}u)^{T_\xi}\cdot \mathbf{n}_i^{T_\xi}\right\} \quad i\in\{i_{T_\xi}\} \tag{81}$$

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

$$u_i^{n+1} \;=\; u_i^n + \frac{1}{3}\left\{\frac{\omega_i}{2}(\mathrm{grad}u)^{T}\cdot \mathbf{n}_i^{T}\right\} \quad i\in\{i_T\} \tag{82}$$

where ω_j is a small constant as before. 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 standard P_1 Galerkin method: a leading truncation error term is eliminated by taking a weighted average of two sets of numerical

solutions obtained on two self-similar grids. In particular, in Richardson's extrapolation, the weights $4/3$ and $-1/3$ are known to eliminate the second-order truncation error from a second-order method and yield a third-order method [18]. Therefore, the P_2 Galerkin scheme is third-order accurate in general, but can be fourth-order accurate on regular grids for which the P_1 Galerkin scheme does not have odd order terms in its truncation error and thus it will be automatically upgraded to fourth-order. Also we point out that this scheme can be derived straightforwardly by the standard weak formulation described in 2.1. Hence the scheme is equivalent to the following.

$$u_j^{n+1} = u_j^n - \omega_j \iint_{\{T_j\}} \mathrm{grad}\varphi_j \cdot \mathrm{grad}u_h \, dV \tag{83}$$

where the basis function is quadratic as defined in Section 3.3

$$u_h = \sum_{i\in\{J\}} u_i \, \varphi_i \quad \text{and} \quad \varphi_j|_T = N_i(x,y) \tag{84}$$

In this form, it is clear that the scheme is equivalent to one of the schemes studied for the advection-diffusion equation by Ricchiuto and Villedieu [5] in the diffusion limit.

5 Higher-Order Cell-Based Schemes

5.1 Nodal Gradient Scheme

If we reconstruct the gradients $(p_j, q_j) = ((u_x)_j, (u_y)_j)$ at nodes by using the Green-Gauss or least-squares techniques, we can directly evaluate the cell-residual, assuming the linear variation of p and q,

$$\phi^T = \iint_T (p_x + q_y)\, dxdy = -\sum_{i\in\{i_T\}} (\overline{p}_i, \overline{q}_i) \cdot \mathbf{n}_i \tag{85}$$

Caraeni and Fuchs employed this approach to discretize the viscous term in their Navier-Stokes code[7]; they distributed the viscous residual together with an inviscid residual with an upwind distribution coefficient. It would be reasonable, at least in the diffusion limit, to distribute it isotropically. Employing equal weights $\beta_j^T = 1/3$, we update the solution u_j by

$$S_j u_j^{n+1} = S_j u_j^n + \Delta t_j \sum_{T\in\{T_j\}} \beta_j^T \, \phi^T \tag{86}$$

It can be shown that this scheme is minimizing the energy norm defined by

$$\mathcal{F}' = \frac{1}{2} \sum_{j\in\{J\}} \left[p_j^2 + q_j^2\right] S_j \tag{87}$$

provided the nodal gradients (p_j, q_j) are obtained by the Green-Gauss reconstruction. It is also illustrative that the scheme is equivalent to the following scheme, derived by splitting the cell-residual (85) edgewise,

$$S_j u_j^{n+1} = S_j u_j^n - \Delta t_j \sum_{T\in\{T_j\}} (\overline{p}_T, \overline{q}_T) \cdot \mathbf{n}_T \tag{88}$$

where $\overline{p}_T$ is the arithmetic mean of the nodal derivative over the edge of triangle T opposite to node j, similarly for $\overline{q}_T$. This is simply because the line integral cancels over the edge that is shared by two neighboring elements. In this form, it is clear that the scheme approximates

$$\frac{\partial u}{\partial t} = \int_T \mathrm{div}(\mathrm{grad}u)\, dV = -\oint_{\partial T} \mathrm{grad}u \cdot \mathbf{n}\, dl. \tag{89}$$

at node j. This is similar to the RC Galerkin scheme (72) which employs the constant high-order cell gradients to evaluate the line integral (the same gradients for all edges within a triangle, and this is why the sum of the contributions vanishes and no cell-residual can be defined in the Galerkin scheme).

A truncation error analysis reveals that this scheme is in fact fourth-order accurate. Considering its wide stencil, we may expect such high accuracy. But it is only second-order accurate in practice because the accuracy of the Green-Gauss formula is lost on boundaries. Unfortunately, this scheme does not work with better reconstruction methods such as a quadratic least-squares method (for which minimization property such as (87) is not guaranteed); it goes unstable.

5.2 Reconstruction Scheme for First-Order System (RC FOS)

The scheme based on the first-order system described in Section 2.2 can be upgraded to higher-order easily by the high-order correction method described in Section 3.2. Simply by adding the correction terms in the constant gradients in the low-order cell-residual (41), we obtain a higher-order cell-residual,

$$\phi^T = \left\{ p_x^T|_{high} + q_y^T|_{high} \right\} S_T \tag{90}$$

where $p_x^T|_{high}$ and $q_y^T|_{high}$ are given by

$$p_x^T|_{high} = p_x^T - \frac{1}{12S_T} \sum_{edges} \Delta P_{ss} \Delta y \tag{91}$$

$$q_y^T|_{high} = q_y^T + \frac{1}{12S_T} \sum_{edges} \Delta Q_{ss} \Delta x \tag{92}$$

and

$$\Delta P_{ss} = \Delta(p_x)\Delta x + \Delta(p_y)\Delta y \tag{93}$$

$$\Delta Q_{ss} = \Delta(q_x)\Delta x + \Delta(q_y)\Delta y. \tag{94}$$

We then distribute this by

$$S_j u_j^{n+1} = S_j u_j^n + \Delta t_j \sum_{T \in \{T_j\}} \beta_j^T \phi^T \tag{95}$$

where $\beta_j^T = \frac{1}{3}$. The schemes for p and q can also be upgraded in the same way. The higher-order cell-residual for p is given by

$$\phi_p^T = \left\{ \overline{p}^T|_{high} - u_x^T|_{high} \right\} S_T \tag{96}$$

where $u_x^T|_{high}$ is computed by the formula (64), and $\overline{p}^T|_{high}$ is the cell-average of the quadratic function $p = \sum_{i=1}^{6} p_i N_i$ with the midpoint values cubically interpolated, which results in

$$\overline{p}^T|_{high} = \frac{1}{S_T}\iint_T p\, dxdy = \overline{p}^T - \frac{1}{8}\sum_{edges} \Delta P_{ss}, \quad \Delta P_{ss} = \Delta(p_x)\Delta x + \Delta(p_y)\Delta y \tag{97}$$

The residual is then distributed with the same equal weights

$$S_j p_j^{n+1} = S_j p_j^n - \Delta t_j \sum_{T\in\{T_j\}} \beta_j^T \phi_p^T \tag{98}$$

A scheme for q can be derived in a similar manner. Note that in this approach, we need to reconstruct the gradients not only of u but also of p and q, i.e. $(u_x, u_y, p_x, p_y, q_x, q_y)$. This scheme also has been confirmed to be third-order accurate with the Green-Gauss reconstruction or linear least-squares, and fourth-order accurate with quadratic least-squares in [4], just like the reconstruction Galerkin schemes described in Section 4.1.

5.3 P_2 Scheme for First-Order System (P_2 FOS)

P_2 schemes for the first-order system is highly desirable because the reconstruction of the gradients for all variables u, p, and q is too extensive. With P_2 elements, we store u, p, and q at nodes including the midpoints, and represent the solution by

$$p = \sum_{i=1}^{6} p_i N_i, \quad q = \sum_{i=1}^{6} q_i N_i, \quad u = \sum_{i=1}^{6} u_i N_i \tag{99}$$

The cell-residual is then given by

$$\phi^T = \iint_T (p_x + q_y)\, dxdy = \left\{(p_x)_{high}^T + (q_y)_{high}^T\right\} S_T \tag{100}$$

where $(p_x)_{high}^T$ and $(q_y)_{high}^T$ are given in any of the forms presented in Section 3.1, e.g.

$$(p_x)_{high}^T = p_x^T - \frac{2}{3}\left(p_x^T - p_x^{T_{IV}}\right) \tag{101}$$

$$(q_y)_{high}^T = q_y^T - \frac{2}{3}\left(q_y^T - q_y^{T_{IV}}\right) \tag{102}$$

Note that this residual vanishes exact cubic solutions of p and q because of the special property of Simpson's rule as mentioned earlier. This is distributed to the nodes by

$$S_j u_j^{n+1} = S_j u_j^n + \Delta t_j \sum_{T\in\{T_j\}} \beta_j^T \phi^T \tag{103}$$

where the distribution coefficients are defined by

$$\beta_j^T = \begin{cases} \frac{1}{12} & i = 1, 2, 3 \\ \frac{1}{4} & i = 4, 5, 6 \end{cases} \tag{104}$$

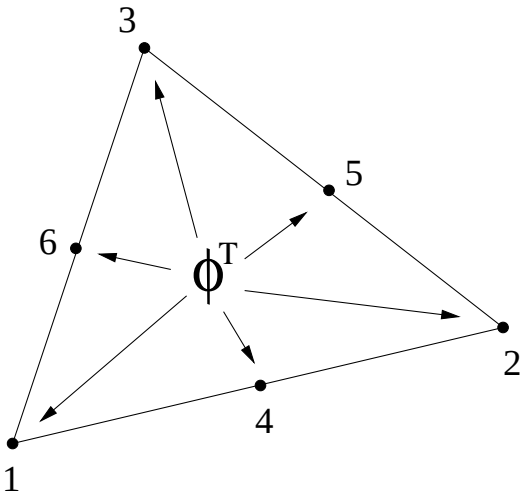


Figure 7: Distribution of the whole cell-residual.

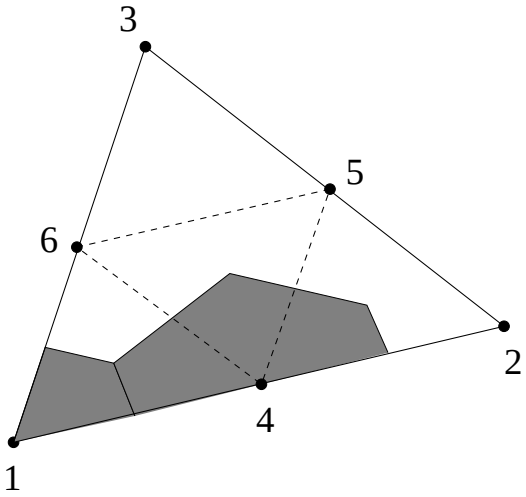


Figure 8: Dual cell area in P_2 element: threefold for midpoints.

based on the ratio of the median dual cell area (See Figures 7 and 8), which of course sum up to unity,

$$\sum_{i=1}^{6} \beta_i^T = 1 \tag{105}$$

so that the scheme is conservative. The slope equations are solved in a similar manner. The higher-order cell-residual for p is given by

$$\phi_p^T = \iint_T (p - u_x)\, dxdy = \left\{ \overline{p}^{T_{IV}} - (u_x)_{high}^T \right\} S_T \tag{106}$$

where $(u_x)_{high}^T$ is given by a formula similar to (101) and

$$\overline{p}^{T_{IV}} = \frac{1}{S_T} \int_T p\, dxdy = \frac{p_4 + p_5 + p_6}{3} \tag{107}$$

The residual is then distributed with the same weights as (104),

$$S_j p_j^{n+1} \;=\; S_j p_j^n - \Delta t_j \sum_{T \in \{T_j\}} \beta_j^T\, \phi_p^T \tag{108}$$

A scheme for q can be derived in the same way. Note that these residuals for p and q as well as the residual for u (100) vanish for exact cubic solutions of u and quadratic solutions of p and q. This means that the scheme preserves exact cubic solutions of u and quadratic solutions of p and q (residual-property). Therefore, the scheme is third-order accurate for p and q and fourth-order accurate for u.

5.4 Another P_2 Scheme for First-Order System (P_2 FOS*)

We can derive another P_2 scheme by applying the concept of residual-distribution to sub-triangles $\{T_I, T_{II}, T_{III}, T_{IV}\}$ rather than the whole P_2 element as we did in the previous section. Hence, the first step is to evaluate a cell-residual for each subtriangle. And the

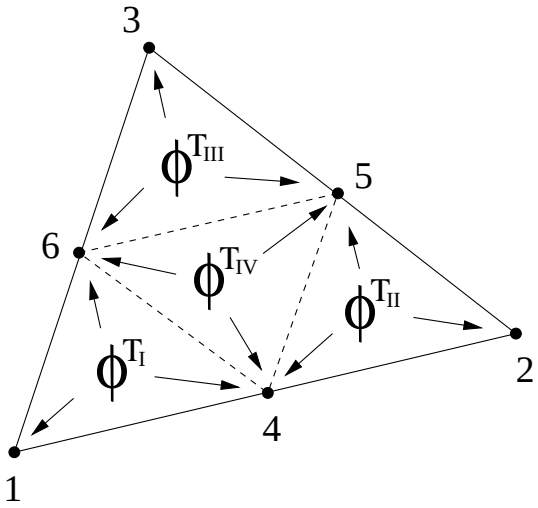


Figure 9: Distribution of cell-residuals of subtriangles.

second step is to distribute this to the nodes of that subtriangle. Now, since each subtriangle has three degrees of freedom, we can employ any familiar distribution coefficients developed for P_1 elements. This is a great advantage of this approach because we do not need to invent new distribution coefficients. To achieve higher-order, of course, the cell-residual must be evaluated based on a higher-order representation of the solution, i.e. quadratic representation on a P_2 element. This idea is due to Roe and Abgrall [17]. They developed higher-order advection schemes with monotonicity-preserving property based on this approach.

Again, over a P_2 element, we have a quadratic representation for each variable.

$$p = \sum_{i=1}^{6} p_i N_i, \quad q = \sum_{i=1}^{6} q_i N_i, \quad u = \sum_{i=1}^{6} u_i N_i \tag{109}$$

Then, we define the cell-residual ϕ^{T_ξ} for subtriangle T_ξ ($\xi = I, II, III, IV$) by

$$\phi^{T_\xi} = \iint_{T_\xi} (p_x + q_y)\, dxdy \tag{110}$$

It can be shown that all the resulting cell-residuals can be written in terms of the constant gradients associated with the subtriangles as follows.

$$\phi^{T_\xi} = \left\{ \left(p_x^{T_\xi} + \sigma \delta p_x \right) + \left(q_y^{T_\xi} + \sigma \delta q_y \right) \right\} S_{T_\xi} \tag{111}$$

where $\sigma = -1$ for $\xi = IV$, $\sigma = 1$ for others, and

$$\delta p_x = \frac{1}{3} \left(p_x^{T_{IV}} - p_x^T \right) \tag{112}$$

$$\delta q_y = \frac{1}{3} \left(q_y^{T_{IV}} - q_y^T \right) \tag{113}$$

Note that this residual preserves exact quadratic solutions of p and q (not cubic because Simpson's rule is not used here). Similarly, we obtain the cell-residuals for the slope

equations.

$$\phi_p^{T_\xi} = \int_{T_\xi} (p - u_x)\, dxdy = \left\{ \left(\overline{p}^{T_\xi} + \delta p\right) - \left(u_x^{T_\xi} + \sigma \delta u_x\right) \right\} S_{T_\xi} \tag{114}$$

where

$$\delta p \;=\; \frac{1}{4}\left(\overline{p}^{T_{IV}} - \overline{p}^{T}\right) \tag{115}$$

and similarly for the equation for q. These cell-residuals are then distributed by within subtriangle T_ξ, thus resulting the following update formulas,

$$S_j u_j^{n+1} \;=\; S_j u_j^n + \Delta t_j \sum_{T \in \{T_j\}} \sum_{T_\xi \in \{T_\xi\}} \beta_j^{T_\xi}\, \phi^{T_\xi} \tag{116}$$

$$S_j p_j^{n+1} \;=\; S_j p_j^n - \Delta t_j \sum_{T \in \{T_j\}} \sum_{T_\xi \in \{T_\xi\}} \beta_j^{T_\xi}\, \phi_p^{T_\xi} \tag{117}$$

and similarly for q (see Figure 9, and also compare this with Figure 7). In the case of diffusion problems, the distribution coefficient is simply $\beta_i^{T_\xi} = 1/3$ for $i \in \{i_{T_\xi}\}$ and $\beta_i^{T_\xi} = 0$ otherwise. Note that the cell-residuals for p and q vanish for exact quadratic solutions of u and linear solutions of p and q. These exact solutions also make the cell-residuals for u, ϕ^{T_ξ}, to vanish. Therefore, this P_2 scheme is third-order accurate for u and second-order accurate for p and q, preserving exact quadratic solutions of u and linear solutions of p and q.

5.5 Edge-Gradient Schemes

Another possibility is to reconstruct the solution over two neighboring elements and evaluate the cell-residual in the form of a line integral

$$\phi^T = \oint_{\partial T} (p\, dy - q\, dx) \tag{118}$$

where p and q are to be computed directly from the reconstruction of the solution u. Consider two triangular elements T and T' that share an edge as shown in Figure 10. Over the quadrilateral formed by the two elements, we can construct a bilinear function that smoothly interpolates the solution values at the four nodes. Then, because the gradient of a bilinear function varies linearly, we may evaluate the line integral by the Trapezoidal rule, using the gradients at points 1 and 2, (p_1, q_1), (p_2, q_2), i.e.

$$\int_1^2 (p\, dy - q\, dx) = -\frac{1}{2}\left\{(p_1, q_1) + (p_2, q_2)\right\} \cdot \mathbf{n}_3 \tag{119}$$

The nodal gradients are obtained directly from the bilinear function. It can be shown that the gradients of the bilinear function at the points 1 and 2 are given simply by the constant gradients over two P_1 elements, denoted by T^R and T^L, formed by the nodes $313'$ and $33'2$ respectively.

$$(p_1, q_1) \;=\; (\mathrm{grad}u)^{T^R} \tag{120}$$

$$(p_2, q_2) \;=\; (\mathrm{grad}u)^{T^L} \tag{121}$$

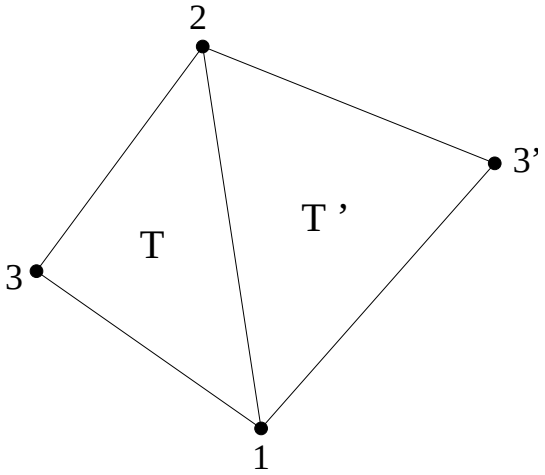


Figure 10: Bilinear Reconstruction over two P_1 elements, T and T'.

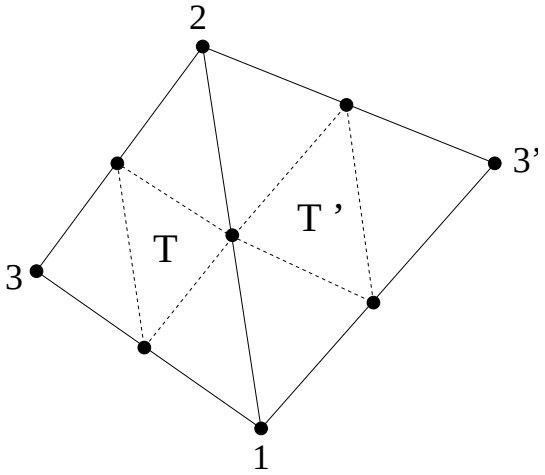


Figure 11: Biquadratic reconstruction over two P_2 elements, T and T'.

Therefore,

$$\int_1^2 (p\,dy - q\,dx) = -\frac{1}{2}\left\{(\text{grad}u)^{T^R} + (\text{grad}u)^{T^L}\right\} \cdot \mathbf{n}_{12} \tag{122}$$

Collecting the contributions from other edges, we obtain the cell-residual for element T,

$$\phi^T = -\sum_{i \in \{i_T\}} (\tilde{p}_i, \tilde{q}_i) \cdot \mathbf{n}_i \tag{123}$$

where

$$(\tilde{p}_i, \tilde{q}_i) = \frac{1}{2}\left\{(\text{grad}u)^{T^R} + (\text{grad}u)^{T^L}\right\} \tag{124}$$

Alternatively, we may employ the Midpoint rule to evaluate the line integral. This requires the gradient only at the midpoint of the edge 1-2. Again, this can be obtained directly from the bilinear interpolant, and it turns out to be identical to the area-weighted average of the constant gradients (120) and (121). Therefore, the cell-residual is still written in the same form as (123) but with a new definition of the edge gradient,

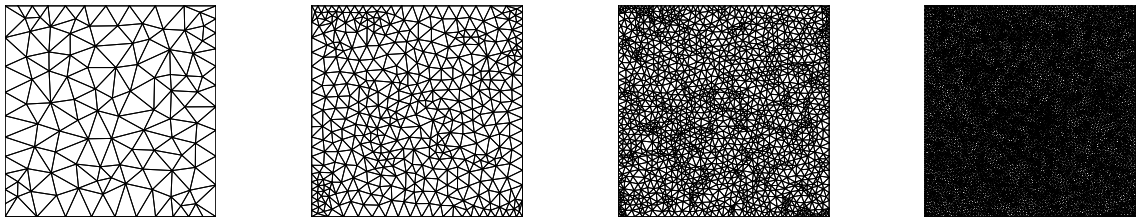
$$(\tilde{p}_i, \tilde{q}_i) = \frac{S_{T^R}\,(\text{grad}u)^{T^R} + S_{T^L}\,(\text{grad}u)^{T^L}}{S_{T^R} + S_{T^L}} \tag{125}$$

which can be easily shown to be equivalent to the area-weighted version of the gradient (51) in Section 2.2. The cell-residual is then distributed by

$$S_j u_j^{n+1} \;=\; S_j u_j^n + \Delta t_j \sum_{T \in \{T_j\}} \beta_j^T \phi^T \tag{126}$$

with $\beta_j^T = 1/3$ or $\beta_j^T = -\partial\phi^T/\partial u_j$. The latter choice corresponds to minimizing the cell-residual in the least-squares norm

$$\mathcal{F}_c = \frac{1}{2}\sum_{T \in \{T\}} \frac{\left(\phi^T\right)^2}{S_T} \tag{127}$$



(a) 200, 121 (b) 800, 441 (c) 3200, 1681 (d) 12800, 6561

Figure 12: Unstructured grids used for convergence study (independently generated). Indicated above are # of triangles (left) and # of nodes (right).

The same approach can be used to derive a P_2 scheme. Over two neighboring P_2 elements, we can construct a biquadratic function that interpolates the solution at 9 points (see Figure 11). From this, we obtain the gradient directly along the edge and evaluate the line integral. The resulting cell-residual can be expressed again in terms of the constant gradients of subtriangles. However, for these schemes further study is necessary, including the issue of possible concave quadrilaterals formed by two triangles which make a mapping to a reference element invalid. Therefore, here, we do not consider this approach any further.

6 Results

6.1 Laplace Equation

We consider a test problem in the square domain $(0 < x < 1, 0 < y < 1)$ with the exact solution,

$$u(x, y) = \frac{\sinh(\pi x)\sin(\pi y) + \sinh(\pi y)\sin(\pi x)}{\sinh(\pi)} \tag{128}$$

The order of convergence is determined through a series of computation using the four different unstructured grids (not necessarily Delaunay) shown in Figure 12. Schemes tested are the P_1 Galerkin scheme, the P_1 FOS scheme, and their higher-order versions upgraded either by a quadratic reconstruction or the use of P_2 elements, and also the nodal gradient scheme. Note that reconstruction schemes with the Green-Gauss and linear least-squares formulas are not included, except for the nodal gradient scheme which works only with the Green-Gauss formula, because only those with a quadratic reconstruction can compete P_2 schemes. For a boundary condition, the exact solution (128) is given on the whole boundary. But no boundary conditions are given for p and q in the case of FOS schemes. All computations were performed in double precision. Results are shown in Tables 1 to 8 and Figure 13. Note that the tables are numbered in the order of the actual error levels, from the least accurate scheme to the most accurate one, and the same is true for the descriptions in Figure 13.

The nodal gradient scheme is second-order accurate as expected. It is disappointing that the actual errors are rather large compared with other second-order schemes. This is partly due to a poor performance of the Green-Gauss formula on irregular grids. As

	L_2 error of u	Order
$Grid(a)$	1.84E-02	
$Grid(b)$	5.25E-03	1.8
$Grid(c)$	1.64E-03	1.7
$Grid(d)$	3.52E-04	2.3

Table 1: Nodal Gradient scheme

	L_2 error of u	Order
$Grid(a)$	3.93E-03	
$Grid(b)$	1.26E-03	1.6
$Grid(c)$	3.60E-04	1.8
$Grid(d)$	9.55E-05	2.0

Table 2: P_1-FOS Scheme

	L_2 error of u	Order
$Grid(a)$	1.68E-03	
$Grid(b)$	5.16E-04	1.6
$Grid(c)$	1.65E-04	1.6
$Grid(d)$	3.52E-05	2.3

Table 3: P_1 Galerkin scheme

	L_2 error of u	Order
$Grid(a)$	8.77E-04	
$Grid(b)$	1.65E-04	2.4
$Grid(c)$	1.92E-05	3.1
$Grid(d)$	1.93E-06	3.4

Table 4: RC FOS scheme

mentioned before, this scheme is in fact fourth-order accurate. We confirmed this by giving the exact solutions at the interior nodes directly connected to the boundary so that the Green-Gauss formula is accurate everywhere (the results are not shown). For the Galerkin and the FOS schemes, the Galerkin schemes are slightly more accurate although the order of convergence is very similar: second-order with linear basis and third-order with reconstruction. Comparing their P_2 versions, the P_2 Galerkin scheme and the P_2 FOS* scheme both of which are third-order accurate, we find again that the P_2 Galerkin scheme is slightly more accurate than the P_2 FOS* scheme. Now, the results for the P_2 FOS scheme are striking: the errors are one or more orders of magnitude smaller than those of other P_2 schemes and this scheme achieved indeed fourth-order accurate with the same P_2 elements. Also, from Table 8, we see that the gradient variables (p, q) were computed with third-order accuracy as expected.

6.2 Advection-Diffusion Equation

In order to show how higher-order schemes can miserably lose their accuracy as well as how the accuracy can be preserved, we consider two different P_2 advection-diffusion schemes. One is a simple sum of the P_2 Galerkin scheme in Section 4.2 and the P_2 LDA advection scheme developed in [19]. The P_2 LDA advection scheme distributes the higher-order P_2 advective residual

$$\phi_{adv}^T = -\iint_T (a\,u_x + b\,u_y)\,dxdy = -\left[a\,(u_x)_{high}^T + b\,(u_y)_{high}^T\right]\,S_T \tag{129}$$

to node j by the coefficient γ_j^T,

$$\gamma_j^T = \begin{cases} \frac{1}{5}\chi_j^T & j = 1,2,3 \\ \frac{4}{5}\chi_j^{T_{IV}} & j = 4,5,6 \end{cases} \tag{130}$$

where χ_j^T is the standard LDA distribution coefficient defined for linear triangle T,

$$\chi_j^T = \frac{(k_j^T)^+}{\sum\limits_{i\in\{i_T\}} (k_i^T)^+}, \quad (k_i^T)^+ = \max\left(0\ ,\ \ \frac{1}{2}(a,b)\cdot\mathbf{n}_i^T\right) \tag{131}$$

	L_2 error of u	Order	L_2 error of p	Order	L_2 error of q	Order
$Grid(a)$	2.52E-04		2.81E-02		2.68E-02	
$Grid(b)$	5.51E-05	2.2	6.98E-03	2.0	6.97E-03	1.9
$Grid(c)$	9.56E-06	2.5	1.78E-03	2.0	1.72E-03	2.0
$Grid(d)$	9.19E-07	3.5	3.04E-04	2.6	3.13E-04	2.5

Table 5: P_2 FOS* scheme

	L_2 error of u	Order
$Grid(a)$	4.04E-04	
$Grid(b)$	4.88E-05	3.0
$Grid(c)$	5.67E-06	3.1
$Grid(d)$	4.82E-07	3.7

Table 6: RC Galerkin scheme

	L_2 error of u	Order
$Grid(a)$	8.83E-05	
$Grid(b)$	1.15E-05	2.9
$Grid(c)$	1.71E-06	2.8
$Grid(d)$	1.92E-07	3.2

Table 7: P_2 Galerkin scheme

which is a simplified form of the original scheme presented in [19]. By adding the P_2 Galerkin scheme to this, we obtain the following P_2 advection-diffusion scheme,

$$S_j u_j^{n+1} = S_j u_j^n + \Delta t_j \left[\sum_{T \in \{T_j\}} \gamma_j^T \phi_{adv}^T - \nu \iint_{\{T_j\}} \text{grad}\varphi_j \cdot \text{grad}u_h \, dxdy \right] \quad (132)$$

with the quadratic basis function φ_j. This scheme is not residual-distribution because there exist no cell-residuals for the advection-diffusion equation. The other advection-diffusion scheme is a combination of the P_2 FOS scheme and the P_2 LDA scheme, which is written as

$$S_j u_j^{n+1} = S_j u_j^n + \Delta t_j \sum_{T \in \{T_j\}} \alpha_j^T \phi^T \quad (133)$$

where the cell-residual ϕ^T is for the whole advection-diffusion equation evaluated as in Section 3.1,

$$\phi^T = \iint_T \left[-(a\, u_x + b\, u_y) + \nu(p_x + q_y) \right] dxdy \quad (134)$$

$$= \left[-a\, (u_x)_{high}^T - b\, (u_y)_{high}^T + \nu \left((p_x)_{high}^T + (q_y)_{high}^T \right) \right] S_T \quad (135)$$

and α_j^T is the combination of the distribution coefficients of the P_2 LDA, γ_j^T, and the P_2 FOS scheme, β_j^T by (104), in the form,

$$\alpha_j^T = \frac{\gamma_j^T + \beta_j^T / Re_h}{1 + 1/Re_h}, \quad Re_h = \sum_{i \in \{i_{T_{IV}}\}} (k_i^{T_{IV}})^+ / \nu \quad (136)$$

This type of distribution coefficient was studied in [4]. We consider also a third-order version of the above scheme which can be derived by computing the cell-residual (134) for each subtriangle as in Section 5.4, distributing it with the standard LDA scheme $\gamma_j^T = \chi_j^T$ and the isotropic coefficient $\beta_j^T = 1/3$ combined as in (136) with Re_h defined locally for that subtriangle. This is equivalent to integrating the third-order LDA scheme in [17], denoted by LDA*, and our P_2 FOS* scheme. Note that these schemes are

	L_2 error of u	Order	L_2 error of p	Order	L_2 error of q	Order
$Grid(a)$	1.68E-05		2.59E-04		3.07E-04	
$Grid(b)$	1.45E-06	3.5	4.00E-05	2.7	4.81E-05	2.6
$Grid(c)$	1.29E-07	3.5	6.74E-06	2.6	6.98E-06	2.8
$Grid(d)$	6.04E-09	4.5	6.17E-07	3.5	6.44E-07	3.5

Table 8: P_2 FOS scheme

residual-distribution because they are based on the cell-residual (over the whole element or subelement) for the entire advection-diffusion equation.

We consider a test problem in a square domain $(0 < x < 1, 0 < y < 1)$ with the exact solution $u = -\cos(2\pi\eta)\exp(0.5\xi(1 - \sqrt{1 + 16\pi^2\nu^2})/\nu)$ with $\nu = 0.1$ and $(a, b) = (5, 1)$ ($\xi = ax + by$, $\eta = bx - ay$), which is taken from [4]. Results are shown in Figure 14. Clearly, simply adding two P_2 schemes does not work. The scheme is not higher-order any more: it is only second-order and it gets worse for finer grids. It is certainly higher-order in the advection and the diffusion limits, but not in between. As pointed out in [4], this is a compatibility problem between the advection discretization and the diffusion discretization: low-order terms in their truncation errors, which vanish for a pure advection and a pure diffusion equations, do not vanish for the advection-diffusion equation. In [4], to overcome this difficulty, a strategy of employing the equivalent first-order system was suggested. The combined P_2 schemes described above are the extension of this approach to P_2 elements. As can be seen in Figure 14, these schemes do not suffer such a convergence problem, and they are still higher-order for the advection-diffusion problem. Note that the scheme based on the cell-residuals for the whole P_2 elements (P_2 LDA and P_2 FOS) is much more accurate here again.

7 Final Remarks

Various methods are available to discretize diffusion terms with higher-order accuracy in residual-distribution methods. They can be categorized into two types: node-based and cell-based. Node-based schemes are based on the Galerkin discretization method directly applied to diffusion terms. Cell-based schemes are based on cell-residuals, and further split into three: nodal gradient scheme, FOS schemes, and edge-gradient schemes. To extend these schemes to higher-order, we considered two approaches: reconstruction and high-order elements. Reconstruction schemes are very easy to implement (just adding high-order correction terms), but gradient reconstruction methods may be grid-dependent and also destroy the compactness of residual-distribution schemes as pointed out earlier. Besides, it can be very expensive for the first-order system for which the gradients of the gradient variables (p, q) must also be reconstructed. For these reasons, one may find P_2 schemes more attractive than reconstruction schemes for practical applications.

For successful higher-order advection-diffusion schemes, we have a choice. Schemes based on the Galerkin discretization are very good candidates. These schemes are based on a solid theoretical background, and can be derived in a highly systematic manner for any type of high-order elements. Although the P_2 Galerkin scheme showed a poor performance in the advection-diffusion testcase in the previous section, it can be made to be uniformly accurate by a careful integration with advection schemes. See the lecture

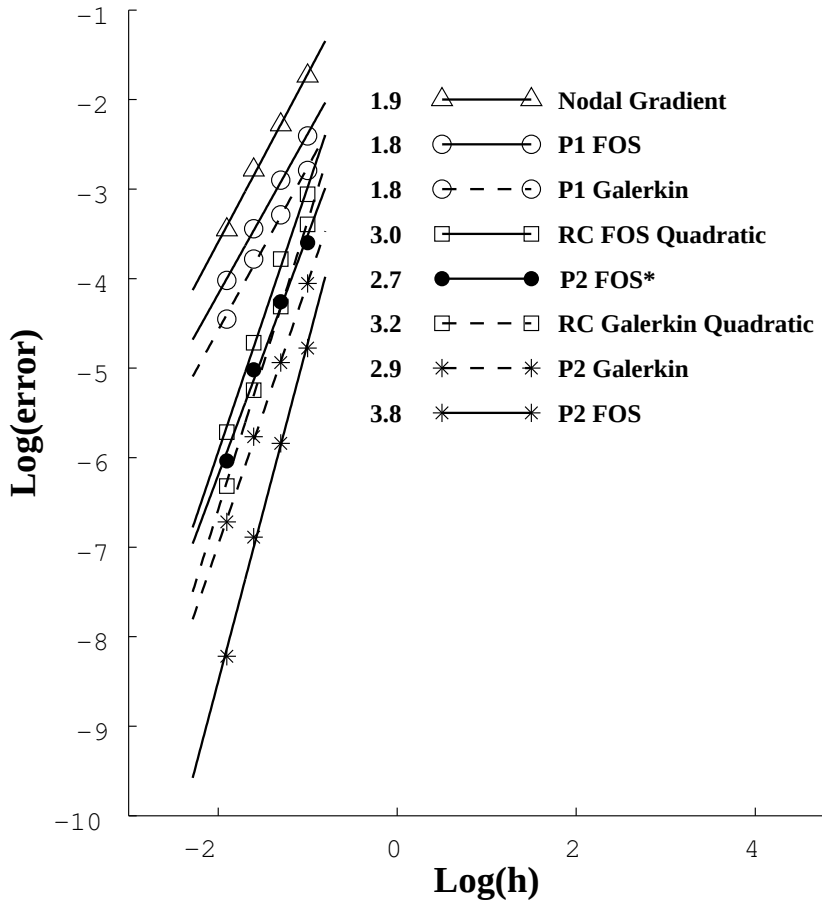


Figure 13: L_2 error for low-order and high-order diffusion schemes. The numbers indicate slopes determined by least-squares fit. h is the average edge length.

notes by Ricchiuto and Villedieu [5] for details. Schemes based on the first-order system are also good candidates. One of the advantages is that the schemes can be easily integrated with higher-order advection schemes, defining the cell-residual for the whole advection-diffusion equation. Moreover they have the residual-property: the P_1 FOS scheme preserves exact linear solutions, the P_2 FOS* scheme preserves exact quadratic solutions, the P_2 FOS scheme preserves exact cubic solutions on arbitrary grids. And because of this, the P_2 FOS scheme achieves fourth-order accuracy with P_2 elements, which is one order higher than the accuracy of other P_2 schemes. Finally, these FOS schemes make it possible to extend the residual-distribution philosophy "*evaluate a cell-residual and distribute it*" to the diffusion and the advection-diffusion equations, and it seems that the P_2 FOS scheme does so in the most natural way.

Acknowledgments

This work has been sponsored by the Space Vehicle Technology Institute, under grant NCC3-989, one of the NASA University Institutes, with joint sponsorship from the Department of Defense. Appreciation is expressed to Claudia Meyer, Mark Klemm and Harry Cikanek of the NASA Glenn Research Center, and to Dr. John Schmisseur and Dr. Walter Jones of the Air Force Office of Scientific Research.

	L_2 error of u	Order
$Grid(a)$	1.65E-02	
$Grid(b)$	3.62E-03	2.2
$Grid(c)$	2.06E-03	0.8
$Grid(d)$	1.59E-03	0.4

Table 9: P_2 LDA + P_2 Galerkin

	L_2 error of u	Order
$Grid(a)$	1.02E-02	
$Grid(b)$	1.30E-03	2.9
$Grid(c)$	4.15E-04	1.7
$Grid(d)$	5.74E-05	2.9

Table 10: P_2 LDA* and P_2 FOS*

	L_2 error of u	Order
$Grid(a)$	1.82E-03	
$Grid(b)$	2.00E-04	3.2
$Grid(c)$	3.68E-05	2.5
$Grid(d)$	2.89E-06	3.8

Table 11: P_2 LDA and P_2 FOS

References

[1] G. T. Tomaich. *A Genuinely Multi-Dimensional Upwinding Algorithm for The Navier-Stokes Equations on Unstructured Grids Using A Compact, Highly-Parallelizable Spatial Discretization*. PhD thesis, University of Michigan, Ann Arbor, Michigan, 1995.

[2] W. A. Wood and W. L. Kleb. 2-d/axisymmetric formulation of multi-dimensional upwind scheme. In *15th AIAA Computational Fluid Dynamics Conference*, AIAA Paper 2001-2630, Anaheim, 2001.

[3] E. van der Weide, H. Deconinck, and G. Degrez. A parallel, implicit, multi-dimensional upwind, residual distribution method for the navier-stokes equations on unstructured grids. *Computational Mechanics*, 23:199–208, 1999.

[4] H. Nishikawa and P. L. Roe. On high-order fluctuation-splitting schemes for navier-stokes equations. In *Third International Conference on Computational Fluid Dynamics*, Toronto, CA, 2004.

[5] M. Ricchiuto and N. Villedieu. Very high order shock capturing residual distribution schemes for advection/diffusion equation. In *34th VKI CFD Lecture Series Very-High Order Discretization Methods*. VKI Lecture Series, 2005.

[6] P. De Palma, G. Pascazio T., Rubino, and M. Napolitano. Residual distribution schemes for advection-diffusion problems on quadrilateral cells. In *17th AIAA Computational Fluid Dynamics Conference*, AIAA Paper 2005-4990, Toronto, 2005.

[7] D. Caraeni and L. Fuchs. Compact third-order multidimensional upwind scheme for navier-stokes simulations. *Theoretical and Computational Fluid Dynamics*, 15:373–401, 2002.

[8] H. Paillére, J. Boxho, G. Degrez, and H. Deconinck. Multidimensional upwind residual distribution schemes for the convection-diffusion equation. *IJNMF*, 15:927–936, 1996.

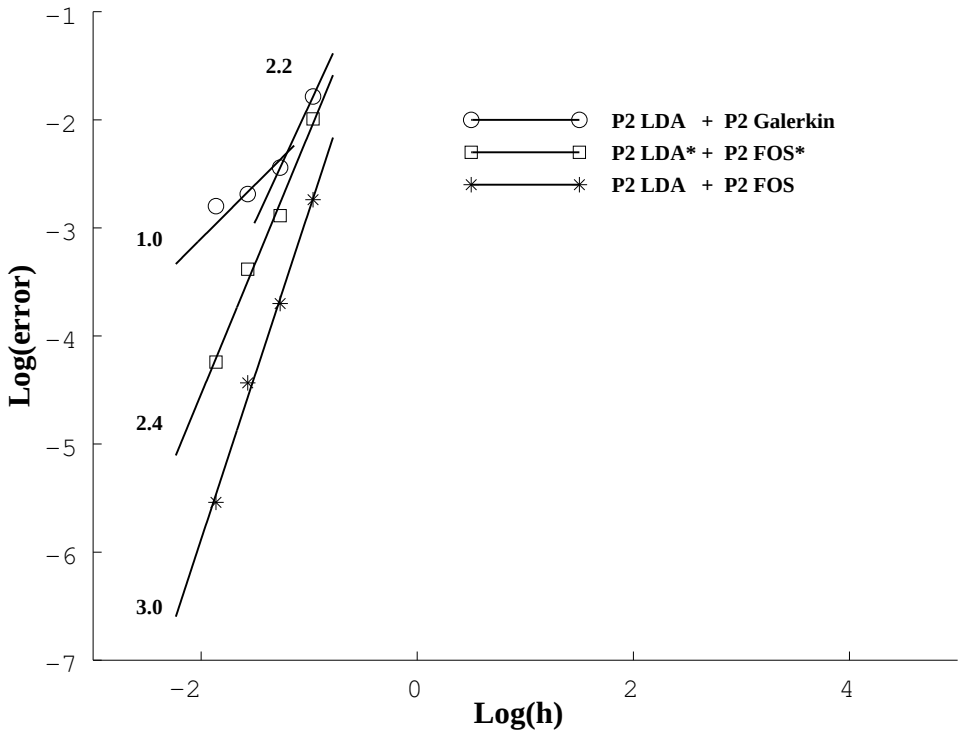


Figure 14: L_2 error for high-order advection-diffusion schemes. The numbers indicate slopes determined by least-squares fit. h is the average edge length.

[9] K. Erikson, D. Estep, P. Hansbo, and C. Johnson. *Computational Differential Equations*. Cambridge University Press, 1996.

[10] T. J. Barth. Numerical methods for gasdynamic systems on unstructured meshes. In D. Kroner, M. Ohlberger, and M. Rohde, editors, *An Introduction to Recent Developments in Theory and Numerics for Conservation Laws*. Springer, 1997.

[11] B. Cockburn and C.-W. Shu. Runge-kutta discontinuous galerkin methods for convection-dominated problems. *Journal of Scientific Computing*, 16(3):173–261, 2001.

[12] T. J. Barth and D. C. Jesperson. The design and application of upwind schemes on unstructured meshes. AIAA Paper 89-0366, 1989.

[13] W. K. Anderson. A grid generation and flow solution method for the euler equations on unstructured grids. *JCP*, 110:23–38, 1994.

[14] D. J. Marvriplis. Revisiting the least-squares procedure for gradient reconstruction on unstructured meshes. AIAA Paper 2003-3986, 2003.

[15] H. Nishikawa, M. Rad, and P. Roe. A third-order fluctuation-splitting scheme that preserves potential flow. In *15th AIAA Computational Fluid Dynamics Conference*, AIAA Paper 01-2595, Anaheim, 2001.

[16] O. C. Zienkiewicz and R. L. Taylor. *The Finite Element Method*. Volume 1. McGraw-Hill Company, 1994.

[17] R. Abgrall and P. L. Roe. High-order fluctuation schemes on triangular meshes. *Journal of Scientific Computing*, 19(1-3):3–36, 2002.

[18] R. L. Burden and J. D. Faires. *Numerical Analysis*. PWS Publishing Company, 1993.

[19] H. Nishikawa and P. L. Roe. Towards high-order fluctuation-splitting schemes for navier-stokes equations. In *17th AIAA Computational Fluid Dynamics Conference*, AIAA Paper 2005-5244, Toronto, 2005.