Preprint: Computers and Fluids, Volume 49, Issue 1, October 2011, Pages 62-86
http://dx.doi.org/10.1016/j.compfluid.2011.04.014

Robust and Accurate Viscous Discretization via Upwind Scheme - I: Basic Principle

Hiroaki Nishikawa*

National Institute of Aerospace,
100 Exploration Way, Hampton VA 23666, USA

Abstract

In this paper, we introduce a general principle for constructing robust and accurate viscous discretization, which is applicable to various discretization methods, including finite-volume, residual-distribution, discontinuous-Galerkin, and spectral-volume methods. The principle is based on a hyperbolic model for the viscous term. It is to discretize the hyperbolic system by an advection scheme, and then derive a viscous discretization from the result. A distinguished feature of the proposed principle is that it automatically introduces a damping term into the resulting viscous scheme, which is essential for effective high-frequency error damping and, in some cases, for consistency also. In this paper, we demonstrate the general principle for the diffusion equation on uniform grids in one dimension and unstructured grids in two dimensions, for node/cell-centered finite-volume, residual-distribution, discontinuous-Galerkin, and spectral-volume methods. Numerical results are presented to verify the accuracy of the derived diffusion schemes and to illustrate the importance of the damping term for highly-skewed typical viscous grids.

1 Introduction

Towards highly efficient and accurate viscous simulations by Navier-Stokes codes, a great deal of effort has recently been devoted to the development of diffusion schemes with particular emphases on high-order methods [1–9] and unstructured grid methods [10–17]. A background approach of constructing diffusion schemes common to many methods is to evaluate the solution gradient on a control volume boundary (e.g., by reconstruction) and compute the diffusive flux directly with them. The evaluation of the gradient is performed typically by using every nearby solution value around, reflecting, in a way, the isotropic nature of diffusion. Although seemingly flawless, this approach is known to result in unsatisfactory schemes: for example, non-convergent schemes [18, 19] and poor h-elliptic (poor high-frequency damping) schemes [20–22]. Hence, this approach by itself is not complete. In many methods, it is, therefore, augmented in one form or another in a manner specific to each discretization method. An example is the so-called penalty term in the discontinuous-Galerkin method [23, 24], and another is the edge-term in the finite-volume method [12, 25, 26]. These terms, playing a role of high-frequency damping, are known to improve the h-ellipticity and also to resolve the inconsistency problem that arises in high-order methods. However, these techniques are highly specialized. It is not straightforward to extend them to other methods, e.g., to the residual-distribution method for which a similar practical technique has not yet been developed. Even if a satisfactory diffusion scheme is devised, it may still encounter a compatibility problem (e.g., lost accuracy) when combined with an advection scheme for advection-diffusion problems [27,28]. Although it may be possible to continue to devise another specialized fix for each problematic situation, what is really needed or highly desired is to improve the approach that is incomplete, so that practical diffusion schemes can be easily constructed for all discretization methods.

In high-order methods, several improved approaches have recently been proposed [2, 5–8, 29, 30]. These approaches have been shown to generate some satisfactory schemes in the method they are proposed for. But it is, again, not immediately clear how to extend them to other discretization methods, particularly to the residual-distribution method. What is still missing, or highly sought, is a guiding principle that is independent of the

*Senior Research Scientist (hiro@nianet.org), National Institute of Aerospace, 100 Exploration Way, Hampton, VA 23666 USA

discretization method, such as 'upwinding' for advection schemes. The failures of the common approach seem to be suggesting that the diffusion equation by itself is not enough to devise practical numerical schemes. Then, a possible avenue that we may take would be to explore other physical models for diffusion. A better model to be solved could then be found, but it is also possible to use such an alternative model just to derive a numerical scheme for the diffusion equation. The present paper demonstrates that the latter is indeed possible and it moreover gives birth to a general principle for deriving practical diffusion schemes for various discretization methods.

A model that we employ for deriving a diffusion scheme is a first-order hyperbolic relaxation system for diffusion [31–33], which we call the first-order hyperbolic diffusion system, or simply the hyperbolic diffusion system. The system being hyperbolic, a well-established principle for advection such as the upwinding principle is directly applicable. We propose to discretize this system (instead of the diffusion equation) by an upwind advection scheme, and derive a diffusion scheme from the result by discarding extra variables and their associated equations. As will be shown by many examples, the diffusion scheme thus derived contains a term responsible for high-frequency damping which is inherited from the dissipation term of the generating advection scheme. It may be pointed out at this point that the proposed principle is independent of the discretization method. In this paper, we demonstrate its general applicability by deriving diffusion schemes for node/cell-centered finite-volume, residual-distribution, discontinuous-Galerkin, and spectral-volume methods.

There exist other methods that utilize a first-order representation of the diffusion equation [5,34], but their system is fundamentally different from the system we employ here. Our first-order system is hyperbolic in time while their system has no such characters. In deriving a diffusion scheme, we thus fully exploit the hyperbolic structure of the first-order system, enabling a straightforward design of a numerical scheme by a well-established principle for hyperbolic systems. There is also a method based on a relaxation model for diffusion [35]. Their relaxation system is also a first-order system, but it has a form of a system of advection equations with a relaxation term for the diffusive flux [35]. On the other hand, the system we employ here has no such advective terms; a part of the relaxation term is used to form a hyperbolic system. There is also a kinetic method based on the BGK kinetic model of the Boltzmann equation [36]. Our construction is different from the kinetic method in that it does not require any kinetic model and directly discretizes the macroscopic equation. The first-order system we employ is similar to the one used in Ref.[32] for a study of relaxation methods. But the same system is used here for a different purpose: we utilize the system to derive a scheme for the diffusion equation whereas the method in Ref.[32] attempts to solve the first-order system instead of the diffusion equation. The difference between the proposed approach and others will be made clearer in a subsequent paper presenting extensions to the Navier-Stokes equations.

Main contributions of the present paper are three-fold. Firstly, we propose a simple universal recipe for constructing robust and accurate diffusion schemes in various discretization methods. Its general applicability will be demonstrated by many examples: node/cell-centered finite-volume, residual-distribution, discontinuous-Galerkin, and spectral-volume methods. Secondly, we show that a diffusion scheme generally consists of two essential terms: consistent and damping terms. The former is responsible for approximating the diffusive flux consistently while the latter is for providing a high-frequency error damping property. This particular perspective is obvious in interior-penalty schemes, but has not been well perceived in other methods. Thirdly, we present a practical guide for defining the damping coefficient with a single parameter that controls the damping property independently of the grid size. In particular, we introduce the grid skewness measure into the damping coefficient and demonstrate that it is essential to robust and accurate computations on highly-skewed grids typical in practical viscous simulations. In this paper, we consider only second-order accurate schemes for the diffusion equation which are sufficient to illustrate these contributions.

The paper is organized as follows. Section 2 describes the first-order hyperbolic diffusion system employed for deriving diffusion schemes. Section 3 presents the principle for constructing diffusion schemes and the definition of the relaxation time suitable for deriving diffusion schemes. Section 4 describes applications of the proposed principle to the one-dimensional diffusion equation, deriving various finite-volume, discontinuous-Galerkin, and spectral-volume schemes. Section 5 presents various diffusion schemes for the two-dimensional diffusion equation on unstructured grids: node/cell-centered finite-volume, residual-distribution, discontinuous-Galerkin, and spectral-volume schemes. Section 6 presents numerical results for time-dependent diffusion problems on uniform grids in one dimension and on highly-skewed anisotropic irregular triangular grids in two dimensions. Finally, Section 7 contains conclusions and discussions on future developments.

2 First-Order Hyperbolic Diffusion System

Consider the diffusion equation:

$$u_t = \nu\, u_{xx}, \tag{2.1}$$

where $\nu > 0$. To derive a numerical scheme for the diffusion equation, we propose to discretize the first-order *hyperbolic diffusion system*:

$$\begin{aligned} u_t &= \nu\, p_x, \\ p_t &= (u_x - p)/T_r, \end{aligned} \tag{2.2}$$

that is asymptotically equivalent to the diffusion equation at large time, $t \gg T_r$ [31,32] when the extra variable p, called here the gradient variable, relaxes to u_x. This relaxation process is characterized by the time scale, T_r, which is called the relaxation time. In the vector form, the system (2.2) is written as

$$\mathbf{U}_t + \mathbf{F}_x = \mathbf{Q}, \tag{2.3}$$

where

$$\mathbf{U} = \begin{bmatrix} u \\ p \end{bmatrix}, \quad \mathbf{F} = \begin{bmatrix} -\nu p \\ -u/T_r \end{bmatrix}, \quad \mathbf{A} = \frac{\partial \mathbf{F}}{\partial \mathbf{U}} = \begin{bmatrix} 0 & -\nu \\ -1/T_r & 0 \end{bmatrix}, \quad \mathbf{Q} = \begin{bmatrix} 0 \\ -p/T_r \end{bmatrix}. \tag{2.4}$$

The Jacobian matrix, $\mathbf{A}$, has a pair of real eigenvalues,

$$\pm\sqrt{\frac{\nu}{T_r}}, \tag{2.5}$$

and linearly independent eigenvectors [37]. Hence, the first-order system (2.2) is a hyperbolic system; various well-established techniques are available for discretization [38–41]. The system describes two waves traveling in opposite directions at the same speed. Note, however, that as shown in Ref. [32], at large t compared to T_r, the waves will be damped out by a source term effect, recovering a smooth feature of the diffusion equation (2.1). This particular behavior is best illustrated by a Riemann problem for the hyperbolic diffusion system (2.2): two different states brought in contact begin to interact instantaneously. Typically, the initial discontinuity breaks up into two pieces carried by the two waves with a smooth Gaussian-type solution developed in between as shown in Figure 2.1. These waves eventually disappear by a damping effect of the source term, and the smooth solution spreads over the domain [32]. The initial stage is called the relaxation stage or frozen limit (the wave speeds (2.5) are called frozen speeds); thus, p is not equal to u_x yet. The last stage is called the equilibrium limit at which the waves have disappeared and we have $p = u_x$, thus satisfying the original diffusion equation (2.1). Further details can be found in Ref. [32]. Here, we are interested not to solve the hyperbolic diffusion system (2.2) but to derive time-accurate numerical schemes for the diffusion equation (2.1) by discretizing the hyperbolic diffusion system: derive an equilibrium scheme from a frozen scheme. Yet in other words, we derive a diffusion scheme from an advection scheme.

3 Principle

3.1 Diffusion Scheme from Advection Scheme

For simplicity, but without loss of generality, consider a one-dimensional grid of uniform spacing, Δx. The numerical solution is stored at each data point j (a node or a cell center). The solution vector at a time level n is denoted by $\mathbf{U}_j^n$. Suppose we discretized the hyperbolic diffusion system (2.3) in the form:

$$\frac{\mathbf{U}_j^{n+1} - \mathbf{U}_j^n}{\Delta t} = -\mathbf{Res}_j^n + \mathbf{Q}_j^n, \tag{3.1}$$

where Δt is a time step, and $\mathbf{Res}_j^n = [Res_{j,1}^n, Res_{j,2}^n]^t$, where the superscript t denotes the transpose, is a discretized version of $\mathbf{F}_x$. We emphasize here that the system is hyperbolic and hence the type of scheme we need to define $\mathbf{Res}_j^n$ is of advection, not diffusion. Hence, we assume that the spatial discretization, $\mathbf{Res}_j^n$, is

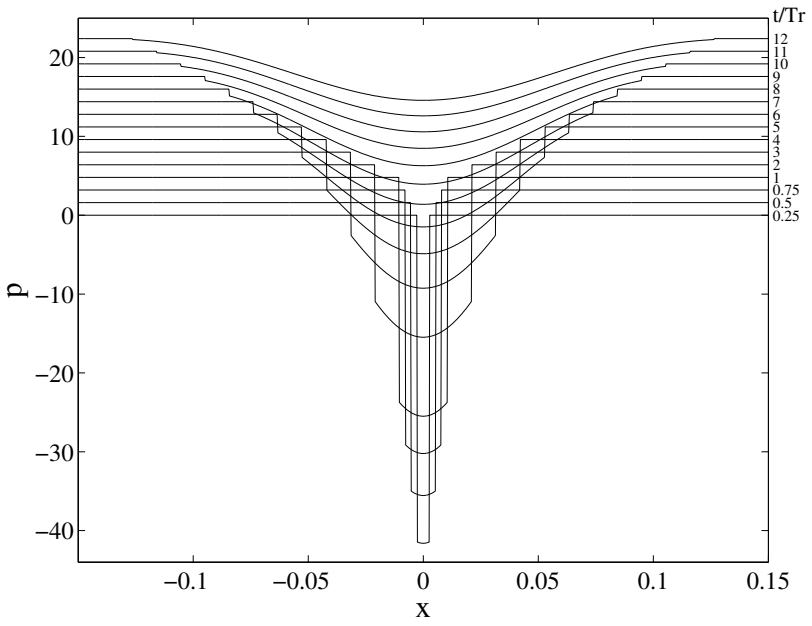


Figure 2.1: Exact solution to the Riemann problem for the gradient variable, p, computed by an analytical formula given in Ref. [32]. Plots are shown for various times: $t/T_r = 0.25, 0.5, 0.75, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12$, vertically shifted for better visibility. The initial condition is a jump in u and $p = 0$.

obtained by an advection scheme. The specific form of $\mathbf{Res}_j^n$ depends on the method; it is left open here. The following argument is applicable to any discretization method. Write out the scheme (3.1) by components:

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} = -Res_{j,1}^n, \tag{3.2}$$

$$\frac{p_j^{n+1} - p_j^n}{\Delta t} = -Res_{j,2}^n - \frac{1}{T_r} p_j^n. \tag{3.3}$$

This scheme is not time-accurate for the diffusion equation (2.1) in general because p_j^n may not be an accurate approximation to the solution gradient. To make this scheme time-accurate, we must instantaneously drive the gradient variable to the equilibrium limit: $p \to u_x$. We propose to achieve this by an explicit evaluation of p_j^n from the approximate solution of u (e.g., gradient reconstruction), so that the extra variable, p_j^n, no longer needs to be stored. Then, Equation (3.3) is also not needed any more, and we are left with

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} = -Res_{j,1}^n, \tag{3.4}$$

which is now time-accurate for the diffusion equation. We have thus derived a time-accurate diffusion scheme from an advection scheme applied to the hyperbolic diffusion system.

Some remarks are in order. It is expected that well-designed advection schemes incorporate some form of dissipation; it then enters the right hand side of Equation (3.4) as a damping term. This is how a damping term is introduced into the derived diffusion scheme. This key feature will be illustrated later by many examples. It should be noted also that we have great flexibility in evaluating the gradient variable, p_j^n. Gradient reconstruction often requires an extended stencil, but a compact diffusion scheme can be constructed by evaluating the gradient with a compact stencil. Such examples will be shown later for a finite-volume method. Finally, we emphasize that although we begin with a system scheme for the hyperbolic diffusion system, the derived diffusion scheme is a scalar scheme for the diffusion equation. That is, the extra variable, p, is not stored; instead it is explicitly evaluated by the approximate solution of u.

3.2 Relaxation Time

To complete the construction of a diffusion scheme, it is necessary to define the relaxation time, T_r, which would enter the right hand side of Equation (3.4). Recall that we first discretize the hyperbolic diffusion system by an advection scheme. Hence, we define T_r such that the hyperbolic diffusion system remains strongly hyperbolic. For explicit time-stepping schemes, it suffices to take T_r comparable to the maximum time step, so that the hyperbolic behavior is well kept during every time step. The maximum time step is defined based on the CFL condition:

$$\Delta t \leq \frac{\Delta x}{\sqrt{\nu/T_r}}, \tag{3.5}$$

where Δx is the minimum mesh spacing of a given grid. Then,

$$\Delta t_{\max} \equiv \frac{\Delta x}{\sqrt{\nu/T_r}} = \alpha T_r, \tag{3.6}$$

where α is a positive constant of $O(1)$ representing the ratio, $\Delta t_{\max}/T_r$. Solving Equation (3.6) for T_r, we obtain

$$T_r = \frac{\Delta x^2}{\alpha^2 \nu}. \tag{3.7}$$

We typically set $\alpha = 1$ for the physical reason mentioned above, but there are some other values having special properties as we will show later. Substituting Equation (3.7) into the CFL condition (3.5), we find

$$\Delta t \leq \frac{1}{\alpha}\frac{\Delta x^2}{\nu}. \tag{3.8}$$

This is the CFL condition for the derived diffusion scheme. Observe that a typical mesh-dependence of the time step for diffusion schemes has emerged: $\Delta t = O(\Delta x^2)$.

3.3 Principle

A principle for constructing diffusion schemes can be stated as follows:

> *Discretize the hyperbolic diffusion system by an advection scheme, then ignore the discrete equation for p, and instead approximate $p = u_x$ directly. The result is a time-accurate scheme for the diffusion equation, having a damping term inherited from the dissipation term of the advection scheme.*

As discussed earlier and will be shown by many examples, the diffusion scheme derived from the principle will inherit the dissipation term of the advection scheme from which it is derived, which then acts as a high-frequency damping term. The damping term, such as the edge-term in finite-volume methods or the penalty-term in discontinuous-Galerkin methods, is then automatically incorporated into the diffusion scheme by an appropriate choice of the advection scheme, e.g., the upwind scheme. This is a very useful feature. Simply following the principle, we obtain a diffusion scheme with a term responsible for high-frequency damping automatically built in. No special techniques nor extra considerations are necessary.

In the rest of the paper, we describe in details how the proposed principle can be applied to derive diffusion schemes in node/cell-centered finite-volume, residual-distribution, discontinuous-Galerkin, and spectral-volume methods. Each section is self-contained for a particular method, so that readers may skip sections dealing with methods of no interest to them.

4 One Dimension

4.1 Finite-Volume Diffusion Schemes

4.1.1 Formulation

Consider a one-dimensional grid of N cells with uniform spacing, Δx. The solution data are considered as cell-averages and stored at the cell center denoted by x_j, $j = 1, 2, 3, \ldots, N$; thus, the data are discontinuous

across cells. Integrating the hyperbolic diffusion system (2.3) over a cell, $I_j = [x_{j-1/2}, x_{j+1/2}]$, we obtain a standard semi-discrete finite-volume discretization:

$$\frac{d\mathbf{U}_j}{dt} = -\frac{1}{\Delta x}\left[\mathbf{F}_{j+1/2} - \mathbf{F}_{j-1/2}\right] + \frac{1}{\Delta x}\int_{I_j} \mathbf{Q}\, dx, \tag{4.1}$$

where $\mathbf{U}_j = [u_j, p_j]^t$ is the cell-averaged solution vector, and $\mathbf{F}_{j+1/2}$ is the interface flux to be defined. Note that the source term discretization is not important since it has a nonzero term only in the second equation which we will ignore. For time integration, we employ the forward-Euler time-stepping scheme for all schemes in this paper unless otherwise stated. The forward-Euler scheme is formally first-order accurate in time, but it gives second-order accuracy in space because the time step is proportional to Δx^2 for diffusion schemes as shown in Section 3.2. The interface flux, $\mathbf{F}_{j+1/2}$, can be determined, from two states, $\mathbf{U}_L$ and $\mathbf{U}_R$, extrapolated from the left and right cells to the interface by any flux function suitable for the hyperbolic system. In this paper, we employ the upwind flux:

$$\mathbf{F}_{j+1/2} = \frac{1}{2}\left[\mathbf{F}_R + \mathbf{F}_L\right] - \frac{1}{2}\left|\mathbf{A}\right|\left(\mathbf{U}_R - \mathbf{U}_L\right) = \frac{1}{2}\left[\mathbf{F}_R + \mathbf{F}_L\right] - \frac{1}{2}\sqrt{\frac{\nu}{T_r}}(\mathbf{U}_R - \mathbf{U}_L), \tag{4.2}$$

which can be written by Equation (3.7) as

$$\mathbf{F}_{j+1/2} \quad = \quad \frac{1}{2}\left[\mathbf{F}_R + \mathbf{F}_L\right] - \frac{\nu\alpha}{2\Delta x}(\mathbf{U}_R - \mathbf{U}_L), \tag{4.3}$$

where $\mathbf{F}_L$ and $\mathbf{F}_R$ denote the physical flux evaluated by the left and right states respectively.

The discretization of the hyperbolic diffusion system (2.3) is now completed. At this point, the resulting finite-volume scheme (4.1) may not be time-accurate for the diffusion equation because p_j may not be an accurate approximation of u_x. To derive a time-accurate diffusion scheme, we discard the second component (i.e., the equation for dp_j/dt) and instead directly reconstruct p_j from u_j. Then, we are left with the first component:

$$\frac{du_j}{dt} = -\frac{1}{\Delta x}\left[f_{j+1/2} - f_{j-1/2}\right], \tag{4.4}$$

where

$$f_{j+1/2} = -\frac{\nu}{2}\left[p_R + p_L\right] - \frac{\nu\alpha}{2\Delta x}(u_R - u_L). \tag{4.5}$$

This is a time-accurate diffusion scheme. Note that the second term in the interface flux (4.5) is the damping term which comes directly from the dissipation term of the upwind scheme for the hyperbolic diffusion system. It is this term, as we will show later, that provides sufficient damping and can also improve the order of accuracy of the diffusion scheme. On the other hand, the first term approximates the physical flux; it is thus called the consistent term or part of the numerical flux. To complete the spatial discretization, we need to define the left and right states: u_L, u_R, p_L, and p_R. In particular, we need to evaluate the gradient variables, p_R and p_L, by using a set of discrete solutions of u (we thus do not store p_j). In doing so, the consistent term should always be retained and evaluated accurately as it is the term that makes the numerical flux consistent, thus resulting in a consistent scheme for the diffusion equation (2.1). It should be noted also that the derived diffusion scheme is implemented in the same way as the advection scheme: interface flux evaluated by two (possibly discontinuous) states. Examples that follow differ only in the definition of the left and right states.

4.1.2 Piecewise Constant Data

This example is intended to emphasize the importance of the consistent term. Assuming that the solution is piecewise constant, we immediately notice that the solution gradient is not available and thus the consistent part of the flux cannot be evaluated. Nevertheless, there is one special case where the scheme can be made consistent. We have at the interface,

$$p_L = p_R = 0, \quad u_L = u_j, \quad u_R = u_{j+1}. \tag{4.6}$$

Inserting these into the interface flux (4.5), we obtain

$$f_{j+1/2} = -\frac{\nu\alpha}{2\Delta x}(u_{j+1} - u_j). \tag{4.7}$$

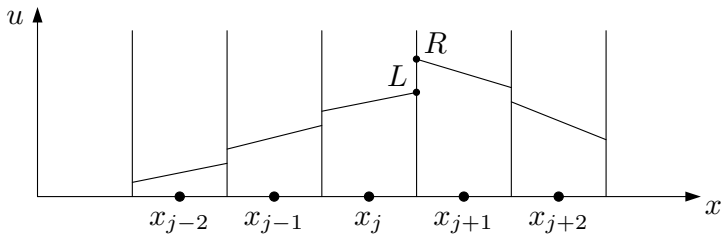


Figure 4.1: Discontinuous piecewise linear data in one dimension.

Note that this flux has only the damping term since the consistent part has vanished identically. But this flux can be made to approximate the physical flux consistently, by setting

$$\alpha = 2. \tag{4.8}$$

Then, the resulting scheme is a standard three-point finite-difference scheme for diffusion:

$$\frac{du_j}{dt} = \frac{\nu}{\Delta x^2}(u_{j+1} - 2u_j + u_{j-1}). \tag{4.9}$$

In this case, there are no other choices for α to make the scheme consistent. As we will see in the next section, if we retain the consistent part of the flux, α will be a free parameter.

4.1.3 Reconstructed Piecewise Linear Data

We consider reconstructing the gradient in order to evaluate the consistent part of the flux function. Note that this would be the most natural option in our construction since it is a standard way to upgrade a first-order advection scheme to second-order [42]. Assuming that the solution is smooth for diffusion problems, we reconstruct the solution gradient within each cell (not at the interface) simply by the central-difference formula:

$$p_j = \frac{u_{j+1} - u_{j-1}}{2\Delta x}, \quad p_{j+1} = \frac{u_{j+2} - u_j}{2\Delta x}. \tag{4.10}$$

We now have a piecewise linear variation within each cell (see Figure 4.1), and the interface quantities can be evaluated as follows:

$$\begin{aligned} u_L &= u_j + \tfrac{1}{2}\, p_j \Delta x, \qquad & u_R &= u_{j+1} - \tfrac{1}{2}\, p_{j+1} \Delta x, \\ p_L &= p_j, & p_R &= p_{j+1}. \end{aligned} \tag{4.11}$$

These values are used to compute the interface flux (4.5), and the diffusion scheme is completely defined. Note that the resulting diffusion scheme is implemented in the same way as a second-order finite-volume advection scheme: interface flux computed by two discontinuous interface values. If such an advection scheme is implemented in a code already, it can be readily extended to an advection-diffusion scheme simply by adding the diffusive flux to the advective flux.

To investigate the accuracy and the damping property of the derived diffusion scheme, substitute (4.11) into the interface flux (4.5) to get

$$f_{j+1/2} = -\frac{\nu}{2}\left[p_j + p_{j+1}\right] - \frac{\nu\alpha}{2\Delta x}\left[u_{j+1} - u_j - \frac{1}{2}\left(p_j + p_{j+1}\right)\Delta x\right]. \tag{4.12}$$

We point out in passing that the damping term (the second term) is a quantity of $O(\Delta x^2)$ since $u_R - u_L = O(\Delta x^3)$ for smooth solutions and it is the consistent term (the first term) that approximates the solution gradient at the interface. Inserting this flux into Equation (4.4), we can write the derived diffusion scheme as

$$\frac{du_j}{dt} = \nu\frac{u_{j+2} - 2u_j + u_{j-2}}{4\Delta x^2} + \frac{\nu\alpha}{2\Delta x^2}\left[u_{j+1} - 2u_j + u_{j-1} - \frac{1}{4}(u_{j+2} - 2u_j + u_{j-2})\right], \tag{4.13}$$

where the gradient variables, p_j and p_{j+1}, have been replaced by the central-difference formula (4.10). Note that this scheme has an extended five-point stencil (unless $\alpha = 2$ which leads to the three-point scheme). The

scheme is consistent, not just for one but for many choices of α. To see this, insert a smooth function into the scheme and expand it to obtain

$$\frac{du_j}{dt} = \nu u_{xx} + \nu u_{xxxx}\left(\frac{1}{3} - \frac{\alpha}{8}\right)\Delta x^2 + O(\Delta x^4), \tag{4.14}$$

This shows that the scheme is consistent as well as second-order accurate for arbitrary α. It also follows that a particular choice,

$$\alpha = \frac{8}{3}, \tag{4.15}$$

makes the scheme fourth-order accurate. Now, insert a Fourier mode of phase angle $\beta \in [0, \pi]$, $u_0 \exp(i\beta x/\Delta x)$, where u_0 is a constant and $i = \sqrt{-1}$, into the scheme (4.13) to get

$$\frac{du_0}{dt} = \lambda u_0, \tag{4.16}$$

where λ is the damping factor:

$$\lambda = \frac{1}{\Delta x^2}\left(-\nu \sin^2\beta - 2\nu\alpha \sin^4\frac{\beta}{2}\right). \tag{4.17}$$

It follows that the highest frequency error mode, $\beta = \pi$, cannot be damped if $\alpha = 0$, meaning that the consistent part does not provide sufficient damping. This is a typical odd-even decoupling problem which can be observed also in Equation (4.13). Then, the second term, which corresponds to the damping term, is the one that gives damping for high frequency modes. Figure 4.2 shows a plot of the damping factor for $\alpha = 0$, 1, 2, 8/3, and 5, compared with the damping factor of the exact diffusion operator. It confirms that the scheme with $\alpha = 0$ has poor high-frequency damping, and that increasing α leads to larger damping. In terms of the h-ellipticity, which is a measure of high-frequency damping representing the minimum damping factor over the high-frequency domain, $\beta = [\pi/2, \pi]$, (see Ref. [20] for details), the scheme with $\alpha = 0$ has zero h-ellipticity (it is therefore not h-elliptic), nonzero α makes the scheme h-elliptic, and increasing α improves the h-ellipticity of the scheme.

We remark that the derived diffusion scheme (4.13) is similar to a three-parameter family of five-point diffusion schemes derived in Ref.[35]. No discussions on the h-ellipticity are given in Ref.[35]; the above discussion on the choice of α may be helpful in choosing their parameters to single out h-elliptic schemes from the family. Also, the existence of a fourth-order scheme is not mentioned in Ref.[35]; a set of parameters may be discovered from $\alpha = 8/3$ to reproduce the fourth-order accurate scheme given above. We point out also that the same five-point scheme can be derived by applying the principle to the residual-distribution method. The derivation is given in Ref.[43].

To estimate the maximum time step allowable for explicit time-stepping, we apply the forward-Euler time integration to obtain the stability condition:

$$\left|1 + \frac{\Delta t}{\Delta x^2}\left(-\nu \sin^2\beta - 2\nu\alpha \sin^4\frac{\beta}{2}\right)\right| \leq 1. \tag{4.18}$$

This leads to the following time-step restriction:

$$\Delta t \leq \begin{cases} (2-\alpha)\dfrac{\Delta x^2}{\nu}, & 0 \leq \alpha < 1, \\ \dfrac{1}{\alpha}\dfrac{\Delta x^2}{\nu}, & \alpha \geq 1. \end{cases} \tag{4.19}$$

Another stability condition can be obtained in terms of positivity: the discrete solution at each data point is a convex combination of neighbor values. It can be easily shown that the derived diffusion scheme (4.13) positive if

$$0 \leq \alpha \leq 2, \quad \Delta t \leq \frac{4}{3\alpha + 2}\frac{\Delta x^2}{\nu}. \tag{4.20}$$

It follows that the fourth-order scheme ($\alpha = 8/3$) is not positive. Note that the scheme is positive even for $\alpha = 0$, i.e., zero damping. Therefore, the positivity property does not guarantee the h-ellipticity. The positivity is not even necessary for the h-ellipticity. The fourth-order scheme ($\alpha = 8/3$) is not positive but strongly h-elliptic as shown in Figure 4.2. A further discussion on positivity is beyond the scope of the present paper.

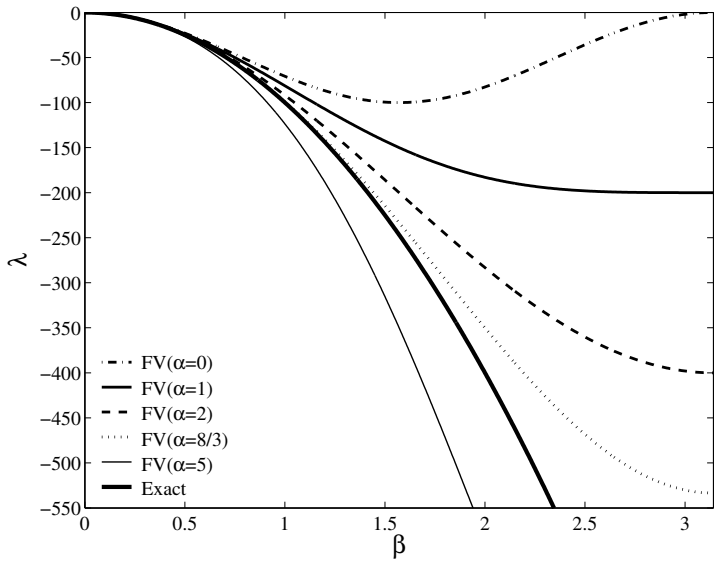


Figure 4.2: Damping factor λ for the exact operator, νu_{xx} and the finite-volume scheme with $\alpha = 0$, 1, 2, 8/3, and 5. $\nu = 1$ and $\Delta x = 0.1$.

4.1.4 Reconstructed Interfacial Linear Data

Yet another possibility is to reconstruct the gradient at the interface,

$$p_{j+1/2} = \frac{u_{j+1} - u_j}{\Delta x}. \tag{4.21}$$

Using this common gradient in both adjacent cells to construct piecewise linear data, we obtain the interface quantities as

$$\begin{aligned} u_L &= u_j + \frac{1}{2}\, p_{j+1/2}\Delta x = \frac{u_j + u_{j+1}}{2}, \qquad & u_R &= u_{j+1} - \frac{1}{2}\, p_{j+1/2}\Delta x = \frac{u_j + u_{j+1}}{2}, \\ p_L &= p_{j+1/2}, & p_R &= p_{j+1/2}, \end{aligned} \tag{4.22}$$

i.e., the reconstructed data are continuous at the interface. Then, the interface flux (4.5) becomes

$$f_{j+1/2} = \frac{u_{j+1} - u_j}{\Delta x}. \tag{4.23}$$

Note that this flux consists of the consistent part only since the damping term has vanished identically. Using this flux, we will obtain the three-point scheme (4.9) again. Although we arrive at the same result, this time, we have consistency automatically (i.e., no need to specifically choose $\alpha = 2$). This is because we have accurately evaluated the consistent part of the flux. In effect, the above procedure corresponds to the lowest-order version of the recovery method [29] where the diffusive flux is evaluated directly by a continuous reconstruction across the interface. In higher dimensions, however, the damping term does not necessarily vanish even when a common gradient is used between two cells. It can still provide damping and improve the h-ellipticity as we will show in Section 5.3.3.

4.2 Discontinuous-Galerkin Diffusion Schemes

Discontinuous Galerkin (DG) method is based on a weak formulation with discontinuous basis functions [18]. Multiplying the hyperbolic diffusion system (2.3) by a test function, v, and integrating over a computational cell, $I_j = [x_{j-1/2}, x_{j+1/2}]$, we obtain

$$\int_{I_j} v\mathbf{U}_t\, dx + \int_{I_j} v\mathbf{F}_x\, dx = \int_{I_j} v\mathbf{Q}\, dx. \tag{4.24}$$

Integration by parts gives

$$\int_{I_j} v\mathbf{U}_t\,dx = -\left[(v\mathbf{F})|_{j+1/2} - (v\mathbf{F})|_{j-1/2}\right] + \int_{I_j} v_x\mathbf{F}\,dx + \int_{I_j} v\mathbf{Q}\,dx. \tag{4.25}$$

To represent the solution, $\mathbf{U}$, discontinuous basis functions are employed that are defined independently within each cell. It consequently creates a gap at cell interfaces; the flux at each interface will be determined from the left and right values, $\mathbf{U}_R$ and $\mathbf{U}_L$, by the upwind flux (4.3).

The P_0 method corresponds to piecewise constant data; it has been considered in 4.1. Here, we consider the P_1 method which corresponds to piecewise linear data:

$$\mathbf{U}(x) = \mathbf{U}_j + (x - x_j)\frac{\overline{\Delta\mathbf{U}}_j}{\Delta x}, \tag{4.26}$$

where $\mathbf{U}_j = [u_j, p_j]^t$ is the cell-averaged solution vector and $\overline{\Delta\mathbf{U}}_j = [\overline{\Delta u}_j, \overline{\Delta p}_j]^t$ is the undivided gradient vector. Choosing the basis functions, 1 and $(x - x_j)/\Delta x$, as test functions, we obtain the following evolution equations for $\mathbf{U}_j$ and $\overline{\Delta\mathbf{U}}_j$:

$$\frac{d\mathbf{U}_j}{dt} = -\frac{1}{\Delta x}\left[\mathbf{F}_{j+1/2} - \mathbf{F}_{j-1/2}\right] + \frac{1}{\Delta x}\int_{I_j}\mathbf{Q}\,dx, \tag{4.27}$$

$$\frac{d(\overline{\Delta\mathbf{U}}_j)}{dt} = -\frac{6}{\Delta x}\left[\mathbf{F}_{j+1/2} + \mathbf{F}_{j-1/2}\right] + \frac{12}{\Delta x^2}\int_{I_j}\mathbf{F}\,dx + \frac{12}{\Delta x^2}\int_{I_j}\frac{x - x_j}{\Delta x}\mathbf{Q}\,dx, \tag{4.28}$$

where the interface flux is given by Equation (4.3). The volume integrals involving $\mathbf{Q}$ are not important because $\mathbf{Q}$ has a nonzero entry only in the second component which we will ignore. On the other hand, the volume integral of the flux in the second equation is important. An efficient formula, which reuses the interface fluxes, is

$$\int_{I_j}\mathbf{F}\,dx = \frac{\Delta x}{6}\left(\mathbf{F}_{j+1/2} + 4\mathbf{F}_j + \mathbf{F}_{j-1/2}\right), \tag{4.29}$$

where $\mathbf{F}_j$ is evaluated directly by the solution at the cell-center. Now, discarding the equations for p_j and $\overline{\Delta p}_j$ from Equations (4.27) and (4.28), respectively, we obtain the following DG diffusion scheme:

$$\frac{du_j}{dt} = -\frac{1}{\Delta x}\left[f_{j+1/2} - f_{j-1/2}\right], \tag{4.30}$$

$$\frac{d(\overline{\Delta u}_j)}{dt} = -\frac{4}{\Delta x}\left[f_{j+1/2} - 2f_j + f_{j-1/2}\right], \tag{4.31}$$

where the interface flux is given by Equation (4.5). Note that the right hand side of Equation (4.31) is a second-order accurate approximation of $-f_{xx} = -\nu u_{xxx}$. To evaluate the interface flux, e.g., at $j+1/2$, we take the following values:

$$\begin{aligned} u_L &= u_j + \overline{\Delta u}_j/2, \qquad & u_R &= u_{j+1} - \overline{\Delta u}_{j+1}/2, \\ p_L &= \overline{\Delta u}_j/\Delta x, & p_R &= \overline{\Delta u}_{j+1}/\Delta x. \end{aligned} \tag{4.32}$$

We emphasize that at this point we have discarded both p_j and $\overline{\Delta p}_j$, and thus store only u_j and $\overline{\Delta u}_j$. Necessary quantities such as p_L and p_R are explicitly evaluated by the approximate solution of u as in Equation (4.32). The construction of a DG diffusion scheme is now completed. Note that the derived DG scheme is compact since the numerical flux involves only the left and right cells, and no extra variables nor equations are required. Some well-known DG diffusion schemes, e.g., the Bassi-Rebay scheme [34] or the Local DG (LDG) scheme [5], are also based on a first-order system representation of the diffusion equation, but their system involves neither time derivatives of auxiliary variables nor the relaxation time. Consequently, their system does not possess a hyperbolic character; the discretization is not trivial [5]. On the other hand, our first-order system is hyperbolic in time; the discretization is straightforward since well-established techniques for hyperbolic systems are directly applicable. The DG diffusion scheme derived here is much similar to the interior-penalty schemes [23, 44–47].

We can actually rediscover some well-known interior-penalty schemes as variants of the diffusion scheme derived above [43]. Note that the parameter α in the derived scheme is not mesh-dependent: the scheme is stable and second-order accurate with $\alpha = 1$ or any positive constant for arbitrary grids, and can be fourth-order accurate for a special value as shown below. It is of different nature from a kind of empiricism known for the classical interior penalty schemes [23].

It is instructive, although not necessary for implementation, to write the derived diffusion scheme in an expanded form. Inserting Equation (4.32), we can write the diffusive flux (4.5) as

$$f_{j+1/2} = -\frac{\nu}{2\Delta x}\left[\overline{\Delta u}_{j+1} + \overline{\Delta u}_j\right] - \frac{\nu\alpha}{2\Delta x}\left[u_{j+1} - u_j - \frac{1}{2}\left(\overline{\Delta u}_{j+1} + \overline{\Delta u}_j\right)\right], \tag{4.33}$$

which is then substituted into Equations (4.30) and (4.31) to get

$$\frac{du_j}{dt} = \frac{\nu}{\Delta x^2}\left[\frac{1}{2}\left(1 - \frac{\alpha}{2}\right)\left(\overline{\Delta u}_{j+1} - \overline{\Delta u}_{j-1}\right) + \frac{\alpha}{2}\left(u_{j+1} - 2u_j + u_{j-1}\right)\right], \tag{4.34}$$

$$\frac{d(\overline{\Delta u}_j)}{dt} = \frac{\nu}{\Delta x^2}\left[2\left(\overline{\Delta u}_{j+1} - 2\overline{\Delta u}_j + \overline{\Delta u}_{j-1}\right) + 4\alpha\left\{\frac{u_{j+1} - u_{j-1}}{2} - \frac{\overline{\Delta u}_{j+1} + 2\overline{\Delta u}_j + \overline{\Delta u}_{j-1}}{4}\right\}\right]. \tag{4.35}$$

It is interesting that the two variables decouple if $\alpha = 2$, and the scheme reduces to the standard three-point scheme for u_j. To investigate stability and accuracy of the diffusion scheme, insert a Fourier mode,

$$\exp(i\beta x/\Delta x)\begin{pmatrix} u_0 \\ \overline{\Delta u}_0 \end{pmatrix} \tag{4.36}$$

into the scheme to get the evolution equation of the Fourier mode:

$$\frac{d}{dt}\begin{pmatrix} u_0 \\ \overline{\Delta u}_0 \end{pmatrix} = M(\beta)\begin{pmatrix} u_0 \\ \overline{\Delta u}_0 \end{pmatrix}, \tag{4.37}$$

where

$$M(\beta) = \frac{\nu e^{-i\beta}}{\Delta x^2}\begin{bmatrix} \frac{\alpha}{2} & \frac{\alpha-2}{4} \\ -2\alpha & 2-\alpha \end{bmatrix} + \frac{\nu}{\Delta x^2}\begin{bmatrix} -\alpha & 0 \\ 0 & -2(\alpha+2) \end{bmatrix} + \frac{\nu e^{i\beta}}{\Delta x^2}\begin{bmatrix} \frac{\alpha}{2} & -\frac{\alpha-2}{4} \\ 2\alpha & 2-\alpha \end{bmatrix}. \tag{4.38}$$

This matrix has the following eigenvalues:

$$\lambda_{1,2} = \frac{\nu}{\Delta x^2}\left[\frac{\alpha-4}{2}\cos\beta + \frac{3\alpha+4}{2} \pm \frac{1}{2}\sqrt{(\alpha-4)^2(\cos^2\beta+1) + 2(\alpha+4)(3\alpha-4)\cos\beta + 8\alpha^2}\right], \tag{4.39}$$

where λ_1 and λ_2 are associated with the negative and positive signs, respectively. When $\alpha = 0$, they reduce to

$$\lambda_1 = 0, \quad \lambda_2 = \frac{4\nu}{\Delta x^2}(\cos\beta - 1), \tag{4.40}$$

Hence, the scheme has no damping for the error mode associated with λ_1. Note also that the second eigenvalue is not consistent:

$$\lambda_2 = -\frac{2\nu\beta^2}{\Delta x^2} + O(\beta^4), \tag{4.41}$$

which differs from the exact operator, $-\nu\beta^2/\Delta x^2$, by a factor of 2. This inconsistency problem is well known [48]. An intuitive explanation for this failure would be that without the damping term, the diffusive flux (4.33) completely vanishes for a piecewise constant solution for which $\overline{\Delta u}_j = 0$; thus there is no way to remove a piecewise constant error mode. For nonzero α, expanding the eigenvalues for small β, we obtain

$$\lambda_1 = -\frac{\nu\beta^2}{\Delta x^2} + \frac{6-\alpha}{24\alpha}\frac{\nu\beta^4}{\Delta x^2} + O(\beta^6), \quad \lambda_2 = -\frac{4\alpha\nu}{\Delta x^2} + \left(\frac{\alpha}{2} - 1\right)\frac{\nu\beta^2}{\Delta x^2} - \frac{(\alpha-2)(\alpha-3)}{24\alpha}\frac{\nu\beta^4}{\Delta x^2} + O(\beta^6), \tag{4.42}$$

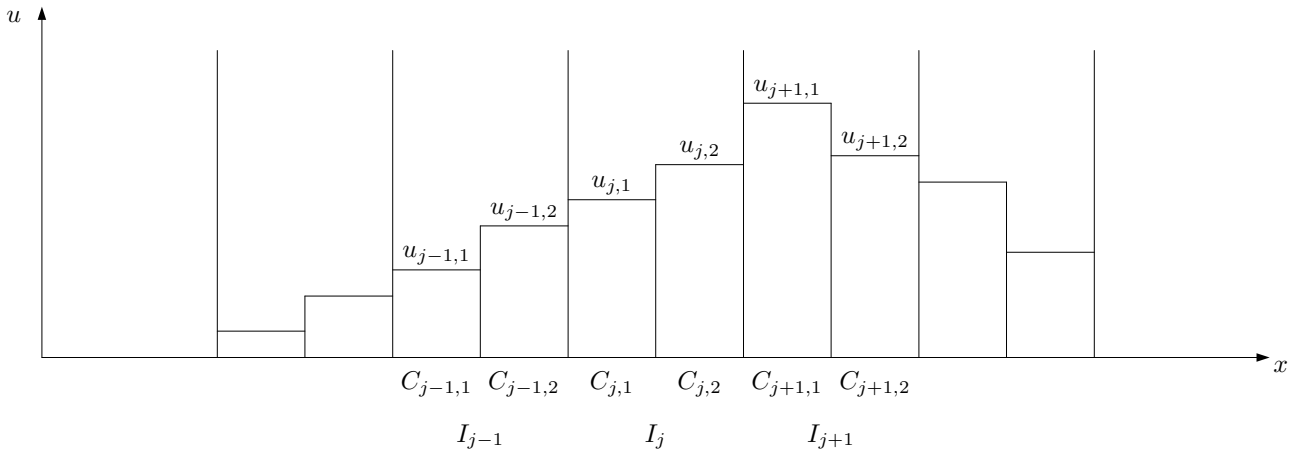


Figure 4.3: One-dimensional data for the second-order SV method.

The first mode approximates the exact diffusion operator with second-order accuracy. It can be fourth-order accurate if

$$\alpha = 6. \tag{4.43}$$

Increasing α, we can expect a large damping effect from the second eigenvalue λ_2, but it would impose a severe restriction on the time step. For the forward-Euler time integration, we have the following stability conditions:

$$|1 + \lambda_1 \Delta t| \le 1 \quad \text{and} \quad |1 + \lambda_2 \Delta t| \le 1. \tag{4.44}$$

It can be shown that these conditions lead to the following restriction on the time step:

$$\Delta t \le \begin{cases} \dfrac{1}{4}\dfrac{\Delta x^2}{\nu}, & 0 \le \alpha < 2, \\ \dfrac{1}{2\alpha}\dfrac{\Delta x^2}{\nu}, & \alpha \ge 2. \end{cases} \tag{4.45}$$

Again, we take $\alpha = 1$ for the same physical reason as before. We may take $\alpha = 6$ if fourth-order accuracy is desired; this however severely limits the time step.

Various other DG diffusion schemes can be derived in the same framework, e.g., by different interface fluxes or quadrature formulas. Examples can be found in Ref.[43], including the well-known (σ, μ)-schemes [24, 49] which includes the so-called symmetric scheme [23] ($\sigma = 1$) and the Baumann scheme [44, 45] ($\sigma = -1$). Also, a scheme that is consistent even with $\alpha = 0$ can be derived by a different quadrature formula [43].

4.3 Spectral-Volume Diffusion Scheme

In the spectral-volume method [19, 50], a cell, $I_j = [x_{j-1/2}, x_{j+1/2}]$, is systematically subdivided into a set of control volumes, $\{C_{j,i}\}$, $i = 1, 2, \ldots, m$, where m depends on the accuracy sought. The cell thus divided is called the spectral volume (SV). The solution value is stored within each control volume as a volume-averaged value, $\mathbf{U}_{j,i} = [u_{j,i}, p_{j,i}]^t$. Given the volume-averaged solutions, a higher-order polynomial of order $m-1$ is then constructed within each spectral volume such that its volume average over a control volume is equal to $\mathbf{U}_{j,i}$. For second-order accuracy, a cell is divided into two control volumes ($m = 2$) defined by $C_{j,1} = [x_{j-1/2}, x_j]$ and $C_{j,2} = [x_j, x_{j+1/2}]$ (see Figure 4.3), and the solution values are defined by

$$\mathbf{U}_{j,1} = \frac{1}{\Delta x/2}\int_{x_{j-1/2}}^{x_j} \mathbf{U}\, dx, \quad \mathbf{U}_{j,2} = \frac{1}{\Delta x/2}\int_{x_j}^{x_{j+1/2}} \mathbf{U}\, dx. \tag{4.46}$$

We can then construct a polynomial of degree 1:

$$\mathbf{U}_j(x) = \mathbf{U}_{j,1} L_1(x) + \mathbf{U}_{j,2} L_2(x), \tag{4.47}$$

where

$$L_1(x) = -\frac{x - x_{j,2}}{\Delta x/2}, \quad L_2(x) = \frac{x - x_{j,1}}{\Delta x/2}. \tag{4.48}$$

The coordinates, $x_{j,1} = x_j - \Delta x/4$ and $x_{j,2} = x_j + \Delta x/4$, are the center coordinates of the control volumes. It can be shown by straightforward integration that the polynomial (4.47) satisfies

$$\frac{1}{\Delta x/2} \int_{C_{j,i}} \mathbf{U}_j(x)\, dx = \mathbf{U}_{j,i}, \quad i = 1, 2. \tag{4.49}$$

To derive a diffusion scheme, we follow the principle and begin by discretizing the hyperbolic diffusion system (2.3). Evolution equation for the volume-averaged values are obtained by integrating the system (2.3) over the control volumes:

$$\frac{d\mathbf{U}_{j,1}}{dt} = -\frac{1}{\Delta x/2}\left[\mathbf{F}_j - \mathbf{F}_{j-1/2}\right] + \frac{1}{\Delta x/2}\int_{C_{j,1}} \mathbf{Q}\, dx, \tag{4.50}$$

$$\frac{d\mathbf{U}_{j,2}}{dt} = -\frac{1}{\Delta x/2}\left[\mathbf{F}_{j+1/2} - \mathbf{F}_j\right] + \frac{1}{\Delta x/2}\int_{C_{j,2}} \mathbf{Q}\, dx, \tag{4.51}$$

where $\mathbf{F}_{j+1/2}$ and $\mathbf{F}_{j-1/2}$ are the SV-interface fluxes, and $\mathbf{F}_j$ is the interior flux to be suitably defined. At the SV-interface where two independently defined polynomials meet, the solution can be discontinuous, and thus we employ the upwind flux:

$$\mathbf{F}_{j+1/2} = \frac{1}{2}\left[\mathbf{F}_R + \mathbf{F}_L\right] - \frac{\nu\alpha}{\Delta x}(\mathbf{U}_R - \mathbf{U}_L), \tag{4.52}$$

where the subscripts L and R denote the interior state and the neighboring state. Note that the dissipation coefficient of the above flux differs from that of the upwind flux (4.3) by factor 2 because each control volume here has a measure $\Delta x/2$, not Δx. On the other hand, the solution is continuous over the spectral volume, and therefore the interior flux is given by the physical flux at x_j: $\mathbf{F}_j = \mathbf{A}\mathbf{U}_j(x_j)$. Now, we derive a diffusion scheme by discarding the second component in each equation:

$$\frac{du_{j,1}}{dt} = -\frac{1}{\Delta x/2}\left[f_j - f_{j-1/2}\right], \quad \frac{du_{j,2}}{dt} = -\frac{1}{\Delta x/2}\left[f_{j+1/2} - f_j\right], \tag{4.53}$$

where

$$f_{j+1/2} = -\frac{\nu}{2}\left[p_R + p_L\right] - \frac{\nu\alpha}{\Delta x}(u_R - u_L), \quad f_j = -\nu\frac{du_j(x)}{dx}. \tag{4.54}$$

We complete the derivation by defining the left and right states:

$$\begin{aligned} u_L &= u_j(x_{j+1/2}), \qquad & u_R &= u_{j+1}(x_{j+1/2}), \\ p_L &= \frac{du_j(x)}{dx}, & p_R &= \frac{du_{j+1}(x)}{dx}. \end{aligned} \tag{4.55}$$

Again, we set $\alpha = 1$ but there is a special value that makes the scheme fourth-order accurate as shown below. Note that we have discarded both $p_{j,1}$ and $p_{j,2}$, and thus store only $u_{j,1}$ and $u_{j,2}$. Necessary quantities, such as p_L and p_R, are evaluated by differentiating the polynomial representation of the solution in the corresponding spectral volume. The derived SV diffusion scheme is compact: it involves the neighboring spectral-volumes only. The scheme is very similar to the penalty scheme considered in Ref. [19].

To examine accuracy and stability of the derived SV diffusion scheme, insert into the system (4.53) a Fourier mode,

$$\exp(i\beta x/\Delta x)\begin{pmatrix} u_{0,1} \\ u_{0,2} \end{pmatrix}, \tag{4.56}$$

where $u_{0,1}$ and $u_{0,2}$ are constant amplitudes, to get

$$\frac{d}{dt}\begin{pmatrix} u_{0,1} \\ u_{0,2} \end{pmatrix} = M_{sv}(\beta)\begin{pmatrix} u_{0,1} \\ u_{0,2} \end{pmatrix}. \tag{4.57}$$

The matrix, $M_{sv}(\beta)$, is given by

$$M_{sv}(\beta) = \frac{\nu e^{-i\beta}}{\Delta x^2}\begin{bmatrix} 2-\alpha & 3\alpha-2 \\ 0 & 0 \end{bmatrix} + \frac{\nu}{\Delta x^2}\begin{bmatrix} -(3\alpha+2) & \alpha+2 \\ \alpha+2 & -(3\alpha+2) \end{bmatrix} + \frac{\nu e^{i\beta}}{\Delta x^2}\begin{bmatrix} 0 & 0 \\ 3\alpha-2 & 2-\alpha \end{bmatrix}. \quad (4.58)$$

This matrix has the following eigenvalues:

$$\lambda_{1,2} = \frac{\nu}{\Delta x^2}\left[(2-\alpha)\cos\beta - (3\alpha+2) \pm \sqrt{(\alpha-2)^2\cos^2\beta + 2(3\alpha-2)(\alpha+2)\cos\beta + (3\alpha+2)^2 - 16\alpha}\right], \quad (4.59)$$

where λ_1 and λ_2 are associated with the positive and negative signs, respectively. When $\alpha = 0$, they reduce to

$$0, \quad \frac{4\nu}{\Delta x^2}(\cos\beta - 1). \quad (4.60)$$

These are exactly the same as those of the DG scheme with $\alpha = 0$, and thus the scheme is inconsistent. For nonzero α, we can expand them for small β to get

$$\lambda_1 = -\frac{\nu\beta^2}{\Delta x^2} + \frac{3-\alpha}{24\alpha}\frac{\nu\beta^4}{\Delta x^2} + O(\beta^6), \quad (4.61)$$

$$\lambda_2 = -\frac{8\alpha\nu}{\Delta x^2} + \frac{(\alpha-1)\,\nu\beta^2}{\Delta x^2} - \frac{(\alpha-1)(2\alpha-3)}{24\alpha}\frac{\nu\beta^4}{\Delta x^2} + O(\beta^6). \quad (4.62)$$

It follows that the scheme is stable and second-order accurate for any non-negative α except for $\alpha = 3$ which makes the scheme fourth-order accurate. Note, as pointed out in Ref. [50], that the predicted accuracy is obtained in $u_j(x)$, not necessarily in the control-volume averages, $u_{j,1}$ and $u_{j,2}$, individually. Numerical experiments show that the control-volume averages are generally second-order accurate but only third-order accurate for $\alpha = 3$. For the forward-Euler time-stepping scheme, we obtain the following stability condition:

$$\Delta t \le \begin{cases} \dfrac{1}{4}\dfrac{\Delta x^2}{\nu}, & 0 \le \alpha < 1, \\ \dfrac{1}{4\alpha}\dfrac{\Delta x^2}{\nu}, & \alpha \ge 1. \end{cases} \quad (4.63)$$

This condition is essentially the same as the condition (4.45) for the DG scheme if the factor 2 in the control volume is taken into account. The same is true, in fact, for the eigenvalues. Note, however, that the discrete equations are different, and thus the numerical solutions will be different. See Ref. [50] for more details on the similarities and differences of the SV and DG methods.

5 Two Dimensions

5.1 First-Order Hyperbolic Diffusion System

We consider constructing numerical schemes for the diffusion equation in two dimensions:

$$u_t = \nu\,(u_{xx} + u_{yy}). \quad (5.1)$$

The first-order hyperbolic diffusion system is given by

$$\begin{aligned} u_t &= \nu\,(p_x + q_y), \\ p_t &= (u_x - p)/T_r, \\ q_t &= (u_y - q)/T_r, \end{aligned} \quad (5.2)$$

where p and q are the gradient variables which relax to the solution derivatives, u_x and u_y, respectively. Write the system in the vector form,

$$\mathbf{U}_t + \mathbf{F}_x + \mathbf{G}_y = \mathbf{Q}, \quad (5.3)$$

where

$$\mathbf{U} = \begin{bmatrix} u \\ p \\ q \end{bmatrix}, \quad \mathbf{F} = \begin{bmatrix} -\nu p \\ -u/T_r \\ 0 \end{bmatrix}, \quad \mathbf{G} = \begin{bmatrix} -\nu q \\ 0 \\ -u/T_r \end{bmatrix}, \quad \mathbf{Q} = \begin{bmatrix} 0 \\ -p/T_r \\ -q/T_r \end{bmatrix}. \tag{5.4}$$

The flux Jacobians are given by

$$\mathbf{A} = \frac{\partial \mathbf{F}}{\partial \mathbf{U}} = \begin{bmatrix} 0 & -\nu & 0 \\ -1/T_r & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \mathbf{B} = \frac{\partial \mathbf{G}}{\partial \mathbf{U}} = \begin{bmatrix} 0 & 0 & -\nu \\ 0 & 0 & 0 \\ -1/T_r & 0 & 0 \end{bmatrix}. \tag{5.5}$$

Consider a projected Jacobian along an arbitrary vector, $\mathbf{n} = (n_x, n_y)$:

$$\mathbf{A}_n = \mathbf{A}n_x + \mathbf{B}n_y. \tag{5.6}$$

The projected Jacobian has the following eigenvalues:

$$\lambda_1 = -\sqrt{\frac{\nu}{T_r}}, \quad \lambda_2 = \sqrt{\frac{\nu}{T_r}}, \quad \lambda_3 = 0. \tag{5.7}$$

The first two eigenvalues are identical to those in one dimension. Note also that these eigenvalues are independent of $\mathbf{n}$, and therefore the system describes a wave propagating isotropically. The third eigenvalue corresponds to the inconsistency damping mode [33], acting on the components of p and q such that $q_x - p_y \neq 0$; it is irrelevant in the present framework. The absolute Jacobian, $|\mathbf{A}_n|$, which is needed to define the upwind flux, is constructed by the right-eigenvector matrix, $\mathbf{R}_n$, and the diagonal eigenvalue-matrix, $\mathbf{\Lambda}_n$,

$$\mathbf{R}_n = \begin{bmatrix} \sqrt{\nu T_r} & -\sqrt{\nu T_r} & 0 \\ n_x & n_x & -n_y \\ n_y & n_y & n_x \end{bmatrix}, \quad \mathbf{\Lambda}_n = \begin{bmatrix} -\sqrt{\nu/T_r} & 0 & 0 \\ 0 & \sqrt{\nu/T_r} & 0 \\ 0 & 0 & 0 \end{bmatrix}, \tag{5.8}$$

as follows:

$$|\mathbf{A}_n| = \mathbf{R}_n |\mathbf{\Lambda}_n| \mathbf{R}_n^{-1} = \sqrt{\frac{\nu}{T_r}} \begin{bmatrix} 1 & 0 & 0 \\ 0 & n_x^2 & n_x n_y \\ 0 & n_x n_y & n_y^2 \end{bmatrix}. \tag{5.9}$$

It is the first row that is relevant to diffusion schemes.

5.2 Relaxation Time

The relaxation time, T_r, can be derived as in Section 3.2 based on the maximum explicit time-step. The precise form of T_r may depend on the mesh type and/or the discretization method, but it can be expressed generally in the form,

$$T_r = \frac{L_r^2}{\alpha^2 \nu}, \tag{5.10}$$

where L_r is a length scale that must be suitably defined for a given discretization, and α is again a positive constant representing the ratio of the maximum explicit time step to the relaxation time. As in one dimension, we typically set $\alpha = 1$ to keep the system strongly hyperbolic over every time step. It is important to note in two dimensions, however, that a length scale arising in the CFL condition is typically not a mesh size, h, but $h/2$. This is well known to be the case for donor-cell-type schemes, i.e., those involving only the face-adjacent neighbors in the stencil [38,41]. For example, if we apply the upwind difference scheme defined in each coordinate direction on a uniform mesh with spacing h, the time step is restricted by

$$\Delta t \leq \frac{h/2}{\sqrt{\nu/T_r}}. \tag{5.11}$$

Then, requiring the maximum time-step to be comparable to the relaxation time as in Section 3.2, we obtain

$$T_r = \frac{h^2}{4\alpha^2\nu}, \tag{5.12}$$

which corresponds to the formula (5.10) with

$$L_r = \frac{h}{2}. \tag{5.13}$$

Based on this observation and for simplicity, we define L_r as half a typical mesh spacing for two-dimensional schemes considered in this paper. In three dimensions, for similar reasons, L_r may be defined as the third of a typical mesh spacing.

It should be noted also that the length scale should be defined in the direction relevant to the wave propagation. For example, in finite-volume methods, L_r is taken to be a length measured in the direction of the face normal along which a Riemann problem is considered. As will be shown, this definition brings a significant distinction of two-dimensional (as well as three-dimensional) schemes from one-dimensional schemes. It introduces a measure of grid-skewness into the damping coefficient in such a way that it increases damping for highly-skewed grids, enabling robust and accurate computations on typical viscous grids.

5.3 Node-Centered Finite-Volume Diffusion Schemes

5.3.1 Formulation

We begin by dividing a domain of interest into a set of elements, $\{E\}$, including both triangles and quadrilaterals (see Figure 5.1). We denote the associated set of nodes by $\{J\}$. In node-centered schemes, we store the solution values at nodes, and approximate the integral of the hyperbolic diffusion system over a dual control volume, Ω_j, around a node $j \in \{J\}$:

$$\frac{d\mathbf{U}_j}{dt} = \frac{1}{V_j}\int_{\Omega_j} \left(-\mathbf{F}_x - \mathbf{G}_y + \mathbf{Q}\right) dV = -\frac{1}{V_j}\oint_{\partial\Omega_j} \mathbf{H}\, dA + \int_{\Omega_j} \mathbf{Q}\; dV, \tag{5.14}$$

where V_j denotes the volume of the dual control volume, dV denotes the infinitesimal volume, $\partial\Omega_j$ denotes the boundary of the dual control volume, dA is the infinitesimal area of the boundary, and $\mathbf{H}$ is the flux projected along the unit outward normal vector of the boundary, $\hat{\mathbf{n}}$:

$$\mathbf{H} = [\mathbf{F}, \mathbf{G}] \cdot \hat{\mathbf{n}}. \tag{5.15}$$

Focusing on the boundary integral in Equation (5.14), we consider two different discretization methods: edge-based method and node-centered compact method.

5.3.2 Edge-Based Diffusion Scheme

A common edge-based finite-volume scheme is based on the following edge-based quadrature formula, approximating Equation (5.14):

$$\frac{d\mathbf{U}_j}{dt} = -\frac{1}{V_j}\sum_{k\in\{K_j\}} \mathbf{\Phi}_{jk} A_{jk} + \frac{1}{V_j}\int_{\Omega_j} \mathbf{Q}\, dV, \tag{5.16}$$

where $\{K_j\}$ is a set of neighbors of j, $\mathbf{\Phi}_{jk}$ is a numerical flux along the directed area vector (see Figure 5.1),

$$\mathbf{n}_{jk} = \mathbf{n}^{\ell}_{jk} + \mathbf{n}^{r}_{jk}, \tag{5.17}$$

and A_{jk} is the magnitude of the directed area vector, i.e., $A_{jk} = |\mathbf{n}_{jk}|$. We remark that this edge-based quadrature formula is exact for linear fluxes only for triangles or parallelograms; there may be errors committed along an edge shared by a triangle and a quadrilateral [21,43,51]; yet another care must be taken along boundary edges [43, 52]. See Ref. [43] for a comprehensive list of quadrature formulas that are exact for linear fluxes in both two and three dimensions. For the numerical flux, we employ the upwind flux:

$$\mathbf{\Phi}_{jk} = \frac{1}{2}\left[\mathbf{H}_{jk}\left(\mathbf{U}_R\right) + \mathbf{H}_{jk}\left(\mathbf{U}_L\right)\right] - \frac{1}{2}\left|\mathbf{A}_n\right|\left(\mathbf{U}_R - \mathbf{U}_L\right), \tag{5.18}$$

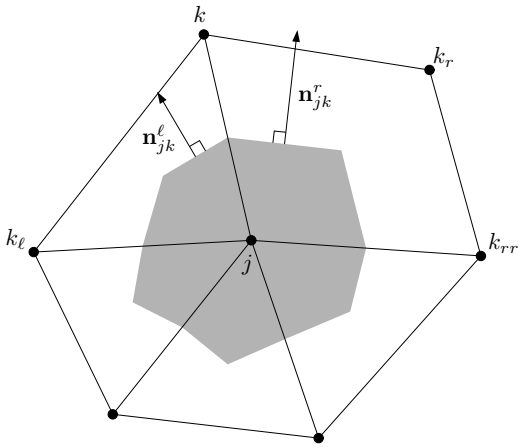


Figure 5.1: Dual control volume for node-centered finite-volume schemes with unit normals associated with an edge, $\{j, k\}$.

where $\mathbf{U}_R$ and $\mathbf{U}_L$ are extrapolated solution vectors at the midpoint of the edge, $\{j, k\}$, the absolute Jacobian, $|\mathbf{A}_n|$, is defined based on the directed area vector, and $\mathbf{H}_{jk}$ is the physical flux projected along the directed area vector:

$$\mathbf{H}_{jk} = [\mathbf{F}, \mathbf{G}] \cdot \hat{\mathbf{n}}_{jk}, \quad \hat{\mathbf{n}}_{jk} = \frac{\mathbf{n}_{jk}}{A_{jk}}. \tag{5.19}$$

The semi-discrete equation (5.16) is then integrated in time by the forward-Euler time-stepping scheme. The global time step, Δt, is defined as the minimum of the local time step, Δt_j, restricted by the local CFL condition:

$$\Delta t_j \leq \frac{2V_j}{\sum\limits_{k \in \{K_j\}} \sqrt{\dfrac{\nu}{T_r}} A_{jk}}. \tag{5.20}$$

Having completed the discretization of the hyperbolic diffusion system, we now ignore the second and third equations in Equation (5.16) and obtain the following edge-based diffusion scheme:

$$\frac{du_j}{dt} = -\frac{1}{V_j} \sum_{k \in \{K_j\}} \phi_{jk} A_{jk}, \tag{5.21}$$

where

$$\phi_{jk} \quad = \quad -\frac{\nu}{2} \left[(p, q)_R + (p, q)_L \right] \cdot \hat{\mathbf{n}}_{jk} - \frac{1}{2} \sqrt{\frac{\nu}{T_r}} \left(u_R - u_L \right). \tag{5.22}$$

Note that the second term is the damping term inherited from the dissipation term of the upwind scheme. The relaxation time, T_r, is defined as in Equation (5.10), thus giving

$$\phi_{jk} \quad = \quad -\frac{\nu}{2} \left[(p, q)_R + (p, q)_L \right] \cdot \hat{\mathbf{n}}_{jk} - \frac{\nu\alpha}{2L_r} \left(u_R - u_L \right), \tag{5.23}$$

where, again, we set $\alpha = 1$, but there are special values as will be discussed later. For unstructured grids, the length scale L_r must be defined at each edge, so that the same interface flux is used per edge and the conservation is not violated. Based on the fact that L_r defines the wave speed of the hyperbolic diffusion system in a Riemann problem in the direction $\hat{\mathbf{n}}_{jk}$, it would be reasonable to take the edge length projected along $\hat{\mathbf{n}}_{jk}$:

$$L_r = \frac{1}{2} \left| \Delta \mathbf{l}_{jk} \cdot \hat{\mathbf{n}}_{jk} \right|, \tag{5.24}$$

where $\Delta \mathbf{l}_{jk} = (x_k - x_j, y_k - y_j)$. To evaluate the interface flux, the left and right states must be defined. One way is to reconstruct the solution gradient at every node and extrapolate the solution to the interface from

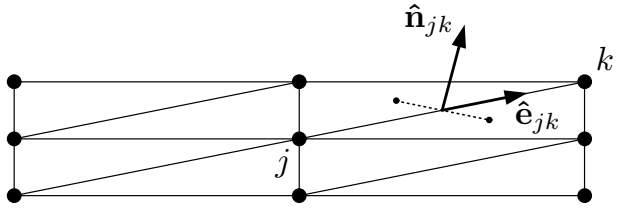


Figure 5.2: Example of skewed grids. $\hat{\mathbf{e}}_{jk} \cdot \hat{\mathbf{n}}_{jk} \to 0$ as the vertical spacing decreases.

each end node, just like a reconstruction-based advection scheme. The gradient at a node j, denoted by $(\nabla u)_j$, can be computed by the Green-Gauss formula or by a least-squares reconstruction method. Once the solution gradients are made available at all nodes, the interface values can be obtained as follows:

$$\begin{aligned} u_L &= u_j + \frac{1}{2}(\nabla u)_j \cdot \Delta\mathbf{l}_{jk}, \qquad & u_R &= u_k - \frac{1}{2}(\nabla u)_k \cdot \Delta\mathbf{l}_{jk}, \\ (p,q)_L &= (\nabla u)_j, & (p,q)_R &= (\nabla u)_k. \end{aligned} \tag{5.25}$$

The diffusion scheme is now completely defined. We point out that the derived diffusion scheme, Equation (5.21) with the flux (5.23), is a general edge-based diffusion scheme directly applicable to arbitrary grids. Extension to three dimensions is straightforward. We remark also that this edge-based diffusion scheme can be implemented in the same framework of advection schemes: reconstruction, extrapolation, and interface flux evaluation. If such mechanisms are already in place, the diffusion scheme can be directly implemented simply by adding the interface flux. For solving advection-diffusion type equations, this will make the code very simple and efficient. Note also that the scheme reduces to a standard five-point finite-difference formula on a uniform quadrilateral mesh. In that case, the scheme is essentially equivalent to the one-dimensional scheme derived in Section 4.1.3 in each coordinate direction, and thus can be made fourth-order accurate by $\alpha = 4/3$ as shown Section 4.1.3 (not $\alpha = 8/3$ because of the difference in the factor $1/2$ in the definition of L_r). As will be shown later, even on irregular grids, the scheme with $\alpha = 4/3$ produces, although not fourth-order accurate, much more accurate solutions than other node-centered schemes.

To gain more insight, we insert the left and right states into the diffusive flux (5.23) to get

$$\phi_{jk} = -\frac{\nu}{2}\left[(\nabla u)_k + (\nabla u)_j\right] \cdot \hat{\mathbf{n}}_{jk} - \frac{\nu\alpha}{|\hat{\mathbf{e}}_{jk} \cdot \hat{\mathbf{n}}_{jk}|}\left[\frac{u_k - u_j}{\Delta l_{jk}} - \frac{1}{2}\left\{(\nabla u)_k + (\nabla u)_j\right\} \cdot \hat{\mathbf{e}}_{jk}\right], \tag{5.26}$$

where $\Delta l_{jk} = |\Delta\mathbf{l}_{jk}|$ and $\hat{\mathbf{e}}_{jk} = \Delta\mathbf{l}_{jk}/|\Delta\mathbf{l}_{jk}|$. Note that as in one dimension, it is the consistent part (the first term) that approximates the solution gradient at the interface; the damping term (the second term) is a quantity of $O(h^m)$ for m-th order accurate gradients. Now, there is a special choice for α:

$$\alpha = (\hat{\mathbf{e}}_{jk} \cdot \hat{\mathbf{n}}_{jk})|\hat{\mathbf{e}}_{jk} \cdot \hat{\mathbf{n}}_{jk}|. \tag{5.27}$$

The diffusive flux (5.26) then becomes

$$\phi_{jk} = -\nu\left[\frac{u_k - u_j}{\Delta l_{jk}}\hat{\mathbf{e}}_{jk} + \overline{\nabla u} - \left(\overline{\nabla u} \cdot \hat{\mathbf{e}}_{jk}\right)\hat{\mathbf{e}}_{jk}\right] \cdot \hat{\mathbf{n}}_{jk}, \tag{5.28}$$

where

$$\overline{\nabla u} = \frac{1}{2}\left[(\nabla u)_k + (\nabla u)_j\right]. \tag{5.29}$$

This flux corresponds to one of the widely-used average-least-squares schemes [12, 26, 53], often called the edge-normal average-least-squares (Avg-LSQ-EN) scheme. The Avg-LSQ-EN scheme is constructed based on the common approach such that the gradient in the diffusive flux (the term in the square bracket in Equation (5.28)) recovers the edge-term along $\hat{\mathbf{e}}_{jk}$ and the average-least-squares gradient, $\overline{\nabla u}$, along the direction normal to $\hat{\mathbf{e}}_{jk}$ (see Ref.[43]). Note that the edge-term, $\frac{u_k - u_j}{\Delta l_{jk}}$, which is considered as important for high-frequency damping, is therefore deliberately incorporated to improve the h-ellipticity. In contrast, in the derived diffusion scheme

(5.26), the edge-term has been incorporated into the scheme automatically by going through the advection scheme. There is another special choice [54]:

$$\alpha = \frac{|\hat{\mathbf{e}}_{jk} \cdot \hat{\mathbf{n}}_{jk}|}{\hat{\mathbf{e}}_{jk} \cdot \hat{\mathbf{n}}_{jk}}. \tag{5.30}$$

This results in the so-called face-tangent average-least-squares (Avg-LSQ-FT) scheme [55]. The Avg-LSQ-FT scheme is constructed such that the gradient in the diffusive flux recovers the edge-term along $\hat{\mathbf{e}}_{jk}$ and the average-least-squares gradient along the direction tangential to the face (see Refs. [43, 55]). Comparing these schemes in the form (5.26), we see that the damping coefficient generally depends on the inner-product, $\hat{\mathbf{e}}_{jk} \cdot \hat{\mathbf{n}}_{jk}$, which may be considered as a skewness measure. For the Avg-LSQ-EN scheme, the damping coefficient is proportional to $\hat{\mathbf{e}}_{jk} \cdot \hat{\mathbf{n}}_{jk}$. For the Avg-LSQ-FT scheme and the derived scheme (5.26) with any positive real value of α, it is inversely proportional to $\hat{\mathbf{e}}_{jk} \cdot \hat{\mathbf{n}}_{jk}$. Consequently, the Avg-LSQ-EN scheme will suffer from a lack of damping for highly-skewed grids (typical viscous grids such as shown in Figure 5.2) for which $\hat{\mathbf{e}}_{jk} \cdot \hat{\mathbf{n}}_{jk} \to 0$. On the other hand, the Avg-LSQ-FT scheme and the derived scheme will gain a large damping effect. On such grids, the Avg-LSQ-FT scheme is in fact very similar to the derived scheme with $\alpha = 1$; they differ only in the sign of the damping coefficient. If the skewness angle exceeds 90°, the damping coefficient of the Avg-LSQ-FT scheme goes negative (damping turns into amplifying), while that of the derived diffusion scheme with positive α remains positive. For a grid with no skewness ($\hat{\mathbf{e}}_{jk} \cdot \hat{\mathbf{n}}_{jk} = 1$), such as a structured quadrilateral grid, both average-least-squares schemes will become identical to the derived scheme with $\alpha = 1$, and all reduce to a standard five-point finite-difference scheme. As mentioned earlier, then, the derived scheme can be made to be fourth-order accurate by taking $\alpha = \frac{4}{3}$.

5.3.3 Node-Centered Compact Diffusion Scheme

Reconstruction methods typically result in an extended stencil with not just neighbors but neighbors in the next level. Compact stencil may be desired for an exact linearization in implicit formulations and/or ease with parallelization. To devise a compact diffusion scheme involving only the nearest neighbors, we must evaluate the gradient within a compact stencil. The following is a good example for illustrating a flexibility of the proposed principle in designing a diffusion scheme.

We now redesign the previous edge-based scheme into a compact scheme by replacing the nodal gradients by a common gradient reconstructed for each edge using immediate neighbors only. Note that it is thus only the definition of the left and right states that will be modified. The interface flux to be evaluated is the same as before, i.e., the flux (5.23). In Figure 5.1, for the edge $\{j, k\}$, we reconstruct a linear solution centered at the edge-midpoint by a least-squares method involving five nodes: j, k, k_ℓ, k_r, and k_{rr}. The least-squares problem is to find the solution value, $\tilde{u}_{jk}$, and the gradient (two components), $\nabla\tilde{u}_{jk}$, at the edge-midpoint: 5 equations for 3 unknowns, which can be solved straightforwardly. For our purpose, the solution value is not needed; only the gradient is used to define the left and right states in Equation (5.25). Alternatively, we may employ the Green-Gauss gradient computed over the two elements that share the edge $\{j, k\}$. Using the reconstructed gradient at both end nodes of the edge, we thus define the left and right states as

$$\begin{aligned} u_L &= u_j + \frac{1}{2}\nabla\tilde{u}_{jk} \cdot \Delta\mathbf{l}_{jk}, \qquad & u_R &= u_k - \frac{1}{2}\nabla\tilde{u}_{jk} \cdot \Delta\mathbf{l}_{jk}, \\ (p, q)_L &= \nabla\tilde{u}_{jk}, & (p, q)_R &= \nabla\tilde{u}_{jk}. \end{aligned} \tag{5.31}$$

Using these values to evaluate the diffusive flux (5.23), we obtain a compact finite-volume diffusion scheme. Note that we have $u_L \neq u_R$ in general, unless the reconstructed linear solution happens to pass through the solutions at j and k. Therefore, the damping term is nonzero and plays its role. We emphasize that although the gradient is reconstructed at the edge-midpoint, it is not used here to compute the physical diffusive flux directly as commonly done. The gradient is used here to define the left and right states, i.e., to extrapolate the solution value from the left and right nodes to the edge-midpoint as above, and then compute the numerical flux (5.23).

For triangular grids, if we employ the Green-Gauss gradient over the two elements sharing the edge, the damping term will vanish identically since the Green-Gauss gradient projected along the edge reduces to the edge-derivative. The resulting scheme corresponds to the one proposed in Ref. [56] by Braaten and Connell for tetrahedral grids (see Ref.[43] for more details).

It may be instructive, although not necessary for implementation, to derive the diffusive flux resulted from inserting the left and right states defined above into Equation (5.23):

$$\phi_{jk} = -\nu \nabla \tilde{u}_{jk} \cdot \hat{\mathbf{n}}_{jk} - \frac{\nu\alpha}{|\hat{\mathbf{e}}_{jk} \cdot \hat{\mathbf{n}}_{jk}|} \left[\frac{u_k - u_j}{\Delta l_{jk}} - \nabla \tilde{u}_{jk} \cdot \hat{\mathbf{e}}_{jk} \right]. \tag{5.32}$$

The second term is the damping term. Clearly, it will vanish if the reconstructed gradient projected along the edge matches the edge derivative. Note that if one takes the common approach and uses the least-squares gradient to evaluate the diffusive flux directly as $-\nu \nabla \tilde{u}_{jk} \cdot \hat{\mathbf{n}}_{jk}$, then the resulting scheme corresponds to the above scheme with $\alpha = 0$, i.e., without the damping term. It can suffer from oscillations as will be shown later. Braaten and Connell [56] take the common approach as well, but their scheme does not suffer. Their success appears due to the use of the Green-Gauss gradient (instead of the least-squares gradient) which cancels the damping term even if formulated so as to ensure high-frequency damping as above. It must be noted, however, that the Braaten-Connell scheme may suffer from checkerboard errors if naively extended to quadrilateral or mixed grids since the Green-Gauss gradient is, then, not guaranteed to match the edge-derivative. In that case, a damping term must be incorporated somehow. In contrast, following the proposed principle, we derived a general (more flexible) formula (5.32) that works with both the least-squares and Green-Gauss gradients on *arbitrary* grids.

Element-based compact schemes, which utilize the gradient constructed over an element, can also be constructed by following the principle. For triangular grids, it leads to the Galerkin scheme while for quadrilateral grids, it leads to the Green-Gauss-type scheme [12, 53] with a built-in damping term that damps the spurious checkerboard error modes associated with quadrilateral elements. See Ref. [43] for details.

5.4 Residual-Distribution Diffusion Scheme

The residual-distribution method is a good example for illustrating the general applicability of the proposed principle. This example shows that we are not merely proposing a new diffusive flux.

We consider constructing a residual-distribution scheme for diffusion on unstructured triangular grids. Following the principle, we begin by discretizing the hyperbolic diffusion system. In the residual-distribution method, we first define the cell-residual over a cell T (see Figure 5.3):

$$\Phi^T = \int_T \left(-\mathbf{A}\mathbf{U}_x - \mathbf{B}\mathbf{U}_y + \mathbf{Q} \right) dV. \tag{5.33}$$

Assuming a piecewise linear variation of $\mathbf{U}$ over the cell, we obtain

$$\Phi^T = -\sum_{i=1}^{3} \mathbf{K}_i \mathbf{U}_i + \overline{\mathbf{Q}}_T V_T, \tag{5.34}$$

where

$$\mathbf{K}_i = \frac{1}{2} \left(\mathbf{A}, \mathbf{B} \right) \cdot \mathbf{n}_i, \;\; \overline{\mathbf{Q}}_T = \frac{\mathbf{Q}_1 + \mathbf{Q}_2 + \mathbf{Q}_3}{3}, \tag{5.35}$$

$\mathbf{n}_i = (n_{i_x}, n_{i_y})$ is the scaled inward normal (see Figure 5.4), and V_T is the cell area. The derivatives, $\mathbf{U}_x$ and $\mathbf{U}_y$, have been evaluated by the Green-Gauss integration over the cell which is exact for linear functions. The source term has been discretized to be also exact for linear functions, but it is not important since we will ignore it. The cell-residual is now distributed to the nodes of the cell by a distribution matrix, $\mathcal{B}_j$. In choosing the distribution matrix, we must remember that the system being discretized is hyperbolic. Hence, we employ an upwind distribution matrix. The LDA scheme [57] is a widely-used upwind residual-distribution scheme, which is defined by the following distribution matrix:

$$\mathcal{B}_i^T = \mathbf{K}_i^+ \left(\sum_{i=1}^{3} \mathbf{K}_i^+ \right)^{-1}, \tag{5.36}$$

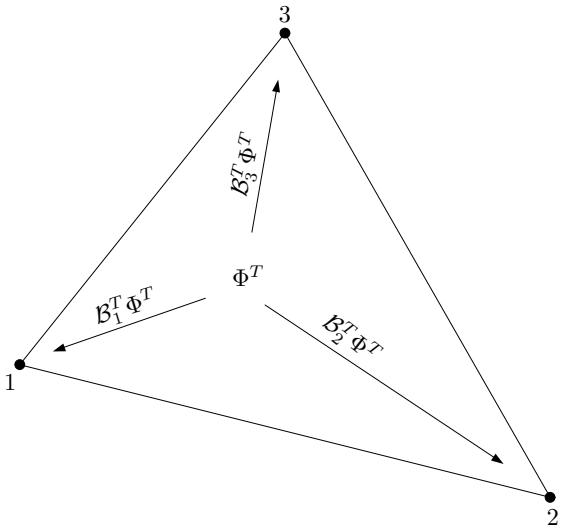


Figure 5.3: Distribution of a cell-residual to the set of vertices $\{i_T\} = \{1, 2, 3\}$. Each contribution is determined by multiplying the cell-residual by the distribution matrix, $\mathcal{B}_i^T$, where $i \in \{i_T\}$.

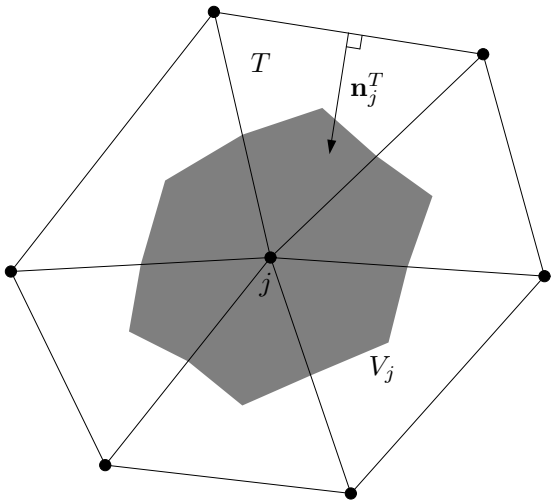


Figure 5.4: Median dual cell around a node j over the set of surrounding triangles, $\{T_j\}$. V_j is the dual cell area. $\mathbf{n}_j^T$ is the scaled inward normal (not drawn to scale) associated with a triangle $T \in \{T_j\}$.

where

$$\mathbf{K}_i^+ = \frac{1}{2}\mathbf{A}_{n_i}^+ = \frac{1}{2}\mathbf{R}_{n_i}\mathbf{\Lambda}_{n_i}^+\mathbf{R}_{n_i}^{-1} = \frac{\alpha\nu}{L_r}\frac{|\mathbf{n}_i|}{4}\begin{bmatrix} 1 & -\frac{L_r}{\alpha}\hat{n}_{x_i} & -\frac{L_r}{\alpha}\hat{n}_{y_i} \\ -\frac{\alpha}{L_r}\hat{n}_{x_i} & \hat{n}_{x_i}^2 & \hat{n}_{x_i}\hat{n}_{y_i} \\ -\frac{\alpha}{L_r}\hat{n}_{y_i} & \hat{n}_{x_i}\hat{n}_{y_i} & \hat{n}_{y_i}^2 \end{bmatrix}, \tag{5.37}$$

$\hat{\mathbf{n}}_i = (\hat{n}_{x_i}, \hat{n}_{y_i})$ is the unit inward normal vector, $\mathbf{R}_{n_i}$ is the right-eigenvector matrix of $\mathbf{A}_{n_i}$, and $\mathbf{\Lambda}_{n_i}^+$ is the corresponding diagonal matrix with positive eigenvalues only. This distribution matrix is often computed numerically, but it can be obtained analytically for the hyperbolic diffusion system:

$$\mathcal{B}_i^T = \begin{bmatrix} \beta_i^T & \frac{L_r}{\alpha}D_{x_i} & -\frac{L_r}{\alpha}D_{y_i} \\ -\frac{\alpha n_{x_i}}{n_T L_r} & -\hat{n}_{x_i}D_{x_i} & \hat{n}_{x_i}D_{y_i} \\ -\frac{\alpha n_{y_i}}{n_T L_r} & -\hat{n}_{y_i}D_{x_i} & \hat{n}_{y_i}D_{y_i} \end{bmatrix}, \tag{5.38}$$

where

$$\beta_i^T = \frac{|\mathbf{n}_i|}{n_T}, \quad n_T = |\mathbf{n}_1| + |\mathbf{n}_2| + |\mathbf{n}_3|, \tag{5.39}$$

$$D_{x_i} = \frac{\hat{n}_{y_k} - \hat{n}_{y_j}}{\hat{n}_{x_1}(\hat{n}_{y_2} - \hat{n}_{y_3}) + \hat{n}_{x_2}(\hat{n}_{y_3} - \hat{n}_{y_1}) + \hat{n}_{x_3}(\hat{n}_{y_1} - \hat{n}_{y_2})}, \tag{5.40}$$

$$D_{y_i} = \frac{\hat{n}_{x_k} - \hat{n}_{x_j}}{\hat{n}_{x_1}(\hat{n}_{y_2} - \hat{n}_{y_3}) + \hat{n}_{x_2}(\hat{n}_{y_3} - \hat{n}_{y_1}) + \hat{n}_{x_3}(\hat{n}_{y_1} - \hat{n}_{y_2})}, \tag{5.41}$$

and $(i, j, k) = (1, 2, 3)$, $(2, 3, 1)$, or $(3, 1, 2)$. After performing the distribution step all over the cells, we have the following semi-discrete equation at each node:

$$\frac{d\mathbf{U}_j}{dt} = \frac{1}{V_j}\sum_{T\in\{T_j\}} \mathcal{B}_j^T\Phi^T, \tag{5.42}$$

where $\{T_j\}$ denotes a set of triangles that share the node j and V_j is the median dual cell area (see Figure 5.4). Note that the distribution matrix sums up to the identity matrix over a cell, so that the sum of the distributed partial cell-residuals reduces to the total cell-residual:

$$\sum_{i=1}^{3} \mathcal{B}_i^T \Phi^T = \Phi^T. \tag{5.43}$$

This ensures a discrete conservation: a global sum of the solution changes are solely due to events on the domain boundary. We must point out here that the above semi-discrete formulation may not be time-accurate because the right hand side may be inconsistent by a constant factor with the spatial part of the time-dependent hyperbolic diffusion system (5.3). This is not important for steady computations, but it is crucial for unsteady computations. A consistent semi-discrete form is given by

$$\frac{d\mathbf{U}_j}{dt} = \left(\sum_{T \in \{T_j\}} \mathcal{B}_j^T V_T \right)^{-1} \sum_{T \in \{T_j\}} \mathcal{B}_j^T \Phi^T. \tag{5.44}$$

This is a semi-discrete form at least first-order accurate in time. See Refs. [58–60] for more sophisticated discussions on time-accurate residual-distribution schemes. Time-accurate computation is now possible by integrating Equation (5.44) by the forward-Euler time-stepping. The time step may be chosen as the minimum of the local time step, Δt_j, defined at each node by

$$\Delta t_j \leq \frac{V_j}{\displaystyle\sum_{T \in \{T_j\}} \frac{1}{2} \sqrt{\frac{\nu}{T_r}} \left| \mathbf{n}_j^T \right|}, \tag{5.45}$$

where $\mathbf{n}_j^T$ is the scaled inward normal of the edge opposite to the node j. Having completed the discretization of the hyperbolic diffusion system, we now derive a diffusion scheme by discarding the second and third components:

$$\frac{du_j}{dt} = \frac{1}{\sum_{T \in \{T_j\}} \beta_j^T V_T} \sum_{T \in \{T_j\}} \left[\beta_j^T \phi^T - \frac{\alpha \nu V_T}{L_r} \left\{ \nabla u^T - (\overline{p}_T, \overline{q}_T) \right\} \cdot (D_{x_j}, -D_{y_j}) \right], \tag{5.46}$$

where we take $\alpha = 1$ as before, and ϕ^T is the cell-residual for the diffusion equation,

$$\phi^T = \nu \left[(p_x)^T + (q_y)^T \right] V_T. \tag{5.47}$$

Here, $(p_x)^T$ and $(q_y)^T$ are the Green-Gauss gradients of p and q over the cell computed with nodal gradients reconstructed, for example, by the Green-Gauss formula. The gradient reconstruction is necessary because we have lost the update equations for the gradient variables, and the gradient variables are not stored any more at nodes as unknowns. Note that we have retained only the first diagonal entry of $\left(\sum_{T \in \{T_j\}} \mathcal{B}_j^T V_T \right)^{-1}$. The scheme is still time-consistent because only the term proportional to the cell-residual, ϕ^T, is responsible for the consistency. The second term vanishes when summed over the cell, and thus does not contribute the total cell-residual. This implies that it does not affect the conservation and the consistency of the scheme. In fact, the second term acts as damping. Without this term (e.g., if $\alpha = 0$), the scheme will fail to damp out high-frequency errors. As will be demonstrated by numerical experiments, a lack of damping leads to large solution errors. The failure of an isotropic scheme ($\beta_j^T = 1/3$) without any damping term has already been pointed out in Ref. [22]: insufficient high-frequency damping and extremely slow convergence to a steady state.

The time step for the derived diffusion scheme may be chosen as in Equation (5.45) with the relaxation time, T_r, is defined as in Equation (5.10). The length scale L_r may be taken simply a half of the minimum height of the triangle:

$$L_r = \frac{1}{2} h_{min} = \frac{V_T}{\max(|\mathbf{n}_1|, |\mathbf{n}_2|, |\mathbf{n}_3|)}. \tag{5.48}$$

An alternative definition can be found in Ref.[43], which is derived from a cell-wise CFL condition. For numerical experiments in this paper, we use the above formula.

The derived diffusion scheme, which is a new scheme to author's knowledge, may be called the LDA diffusion scheme. Other diffusion schemes can be derived by other advection schemes. For example, a diffusion scheme can be derived from the Lax-Wendroff scheme as shown in Ref.[43].

5.5 Cell-Centered Finite-Volume Diffusion Scheme

We consider constructing cell-centered diffusion schemes. Without loss of generality, we consider only triangular grids. Resulting diffusion schemes can be easily applied to other types of grids.

A domain of interest is divided into a set of non-overlapping triangular cells. The solution values are stored at centroids of triangular cells and considered as cell-averaged values. Evolution of a cell-averaged value, $\mathbf{U}_j$, is determined by approximating the integral of the hyperbolic diffusion system over the corresponding triangular cell, T_j:

$$\frac{d\mathbf{U}_j}{dt} = \frac{1}{V_j}\int_{T_j} \left(-\mathbf{F}_x - \mathbf{G}_y + \mathbf{Q}\right) dV = -\frac{1}{V_j}\oint_{\partial T_j} \mathbf{H}\; dA + \int_{T_j} \mathbf{Q}\; dV, \tag{5.49}$$

where V_j here denotes the volume of the cell T_j, ∂T_j denotes the cell boundary, dA is the infinitesimal area of the boundary, and $\mathbf{H}$ is the flux dotted with the unit outward normal vector of the boundary, $\hat{\mathbf{n}}$. We evaluate the flux integral by the midpoint rule over each face of the triangle; a face is associated with a pair of adjacent data points, j and k (see Figure 5.5). Replacing $\mathbf{H}$ by a numerical flux, we thus obtain the following semi-discrete form:

$$\frac{d\mathbf{U}_j}{dt} = -\frac{1}{V_j}\sum_{k\in\{K_j\}} \mathbf{\Phi}_{jk} A_{jk} + \frac{1}{V_j}\int_{T_j} \mathbf{Q}\, dV, \tag{5.50}$$

where $\{K_j\}$ is a set of faces surrounding the data point j, and A_{jk} is the face area. The numerical flux $\mathbf{\Phi}_{jk}$ is the upwind flux which is essentially the same as the one employed in the node-centered scheme:

$$\mathbf{\Phi}_{jk} = \frac{1}{2}\left[\mathbf{H}_{jk}\left(\mathbf{U}_R\right) + \mathbf{H}_{jk}\left(\mathbf{U}_L\right)\right] - \frac{1}{2}\left|\mathbf{A}_n\right|\left(\mathbf{U}_R - \mathbf{U}_L\right), \tag{5.51}$$

where $\mathbf{H}_{jk}$ is the physical flux projected along the outward face-normal direction,

$$\mathbf{H}_{jk} = [\mathbf{F}, \mathbf{G}] \cdot \hat{\mathbf{n}}_{jk}, \tag{5.52}$$

Note that $\hat{\mathbf{n}}_{jk}$ is the unit face-normal vector. For second-order accuracy, the solution gradients are reconstructed in each cell, and the left and right states, $\mathbf{U}_L$ and $\mathbf{U}_R$, are extrapolated from each cell to the face midpoint (see Figure 5.5). Time integration is performed by the forward-Euler time-stepping scheme with a time step restricted by a CFL condition similar to the node-centered finite-volume scheme discussed in Section 5.3.2.

Having completed the discretization of the hyperbolic diffusion system, we now discard the second and third components and obtain a diffusion scheme:

$$\frac{du_j}{dt} = -\frac{1}{V_j}\sum_{k\in\{K_j\}} \phi_{jk} A_{jk}, \tag{5.53}$$

where

$$\phi_{jk} = -\frac{\nu}{2}\left[(p,q)_R + (p,q)_L\right] \cdot \hat{\mathbf{n}}_{jk} - \frac{1}{2}\sqrt{\frac{\nu}{T_r}}\left(u_R - u_L\right). \tag{5.54}$$

The dissipation term of the upwind flux has now entered into the diffusive flux as a damping term. It has been automatically introduced with no specific technique to incorporate it, and its coefficient is determined unambiguously. The relaxation time, T_r, is defined as in Equation (5.10), thus yielding

$$\phi_{jk} \;=\; -\frac{\nu}{2}\left[(p,q)_R + (p,q)_L\right] \cdot \hat{\mathbf{n}}_{jk} - \frac{\nu\alpha}{2L_r}\left(u_R - u_L\right). \tag{5.55}$$

The length scale, L_r, is defined based on the distance between the two data points across the face projected along the face-normal direction:

$$L_r = \frac{1}{2}\left|\Delta\mathbf{l}_{jk} \cdot \hat{\mathbf{n}}_{jk}\right|, \tag{5.56}$$

where $\Delta\mathbf{l}_{jk} = (x_k - x_j, y_k - y_j)$. As in the edge-based schemes in Section 5.3.2, the above definition of L_r brings the skewness measure in the denominator of the damping coefficient. For the parameter α, we again set $\alpha = 1$,

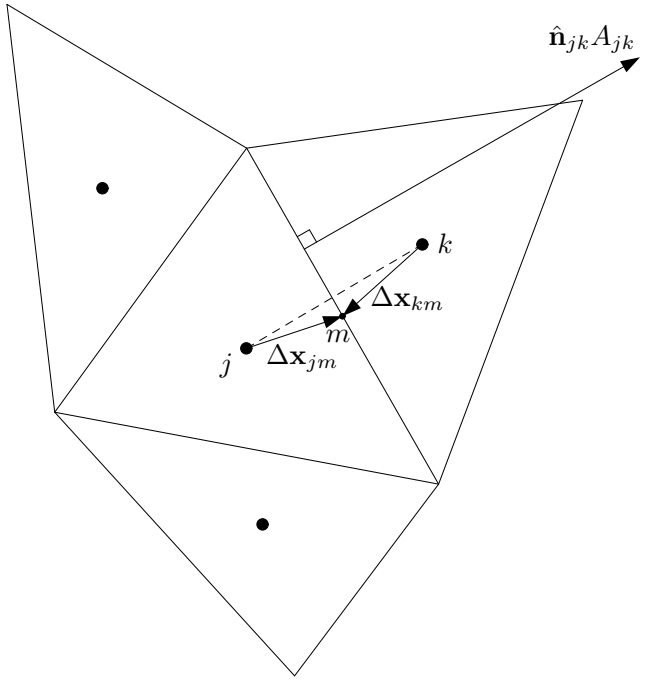


Figure 5.5: Stencil for cell-centered schemes.

but also consider using $\alpha = 4/3$ which makes the scheme fourth-order accurate on structured quadrilateral grids. To define the left and right states, we evaluate the gradient variables by reconstructed solution gradients in cells, ∇u_k and ∇u_j. Then, linearly extrapolating the solution to the face-midpoint, we obtain

$$\begin{aligned} u_L &= u_j + \nabla u_j \cdot \Delta\mathbf{x}_{jm}, \qquad & u_R &= u_k + \nabla u_k \cdot \Delta\mathbf{x}_{km}, \\ (p,q)_L &= \nabla u_j, & (p,q)_R &= \nabla u_k. \end{aligned} \tag{5.57}$$

The diffusion scheme has thus been completely defined. Note that there is no need to store p and q as they are explicitly evaluated as above. To reconstruct the gradient in each cell, a least-squares gradient reconstruction method can be employed. It should be noted again that the derived cell-centered diffusion scheme can be implemented in the same framework of advection schemes: reconstruction, extrapolation, and interface flux evaluation.

Finally, we point out that the above scheme is different from widely-used cell-centered diffusion schemes, Avg-LSQ-EN and Avg-LSQ-FT schemes, [25, 26, 55]. As discussed in Ref.[43], widely-used schemes correspond to evaluating the interface flux halfway between the two adjacent data points, i.e., potentially off the control volume boundary, with α defined similarly to those discussed for the node-centered schemes. In particular, the damping coefficient can get vanishingly small or even go negative for widely-used schemes, especially on agglomerated (arbitrary polygonal) grids typically used in multigrid methods [55, 61–63], while the derived diffusion scheme has a positive damping coefficient for any positive real value of α. See Ref.[43] for details.

5.6 Discontinuous-Galerkin Diffusion Scheme

We consider constructing a P_1 DG diffusion scheme on triangular grids. Following the principle, we begin by discretizing the first-order hyperbolic diffusion system. Within a cell, T_j, we express the solution as a polynomial of degree 1 centered at the centroid $\mathbf{x}_c = (x_j, y_j)$:

$$\mathbf{U}(x,y) = \psi_0 \mathbf{U}_j + \psi_1 \overline{\partial_x \mathbf{U}}_j + \psi_2 \overline{\partial_y \mathbf{U}}_j, \tag{5.58}$$

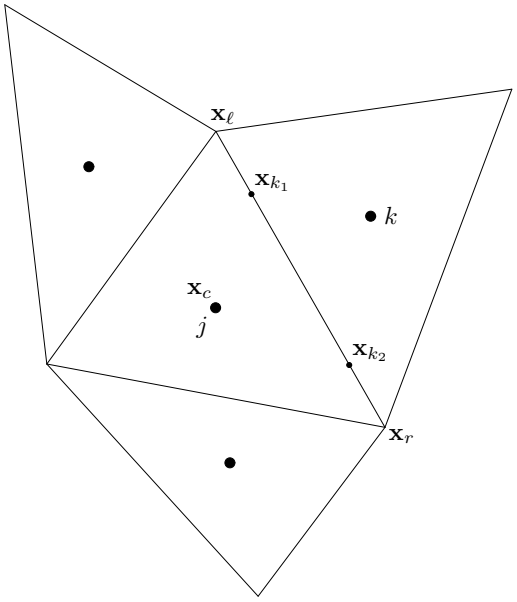


Figure 5.6: Stencil for discontinuous-Galerkin schemes. Quadrature points are shown for the k-th face.

where $\mathbf{U}_j$ is a vector of the cell-averaged solutions, and $\overline{\partial_x \mathbf{U}}_j$ and $\overline{\partial_y \mathbf{U}}_j$ are vectors of the x- and y-derivatives, respectively:

$$\mathbf{U}_j = \begin{bmatrix} u_j \\ p_j \\ q_j \end{bmatrix}, \quad \overline{\partial_x \mathbf{U}}_j = \begin{bmatrix} \overline{\partial_x u_j} \\ \overline{\partial_x p_j} \\ \overline{\partial_x q_j} \end{bmatrix}, \quad \overline{\partial_y \mathbf{U}}_j = \begin{bmatrix} \overline{\partial_y u_j} \\ \overline{\partial_y p_j} \\ \overline{\partial_y q_j} \end{bmatrix}. \tag{5.59}$$

It should be noted that the degrees of freedom for p and q have been introduced here just for the sake of convenience; they will be abandoned once the discretization of the hyperbolic diffusion system is completed. The functions, ψ_m, $m = 0, 1, 2$, are the basis functions,

$$\psi_0 = 1, \quad \psi_1 = x - x_j, \quad \psi_2 = y - y_j, \tag{5.60}$$

specifically chosen such that ψ_0 is orthogonal to the other two:

$$\int_{T_j} \psi_0 \psi_m \, dV = 0, \quad m = 1, 2. \tag{5.61}$$

Evolution equations for the unknowns, $\mathbf{U}_j$, $\overline{\partial_x \mathbf{U}}_j$, and $\overline{\partial_y \mathbf{U}}_j$, are derived from the weak formulation involving a test function v:

$$\int_{T_j} v \frac{\partial \mathbf{U}}{\partial t} \, dV = \int_{T_j} v \left(-\mathbf{F}_x - \mathbf{G}_y + \mathbf{Q}\right) \, dV = - \sum_{k \in \{K_j\}} \int_{\partial T_j^k} v \, \mathbf{H} \, dA_k + \int_{T_j} \nabla v \cdot (\mathbf{F}, \mathbf{G}) \, dV + \int_{T_j} v \, \mathbf{Q} \, dV, \tag{5.62}$$

where ∂T_j^k denotes the boundary corresponding to the k-th face, and $\mathbf{H}$ is the flux normal to the boundary. Setting $v = \psi_0$, we obtain a decoupled evolution equation for the cell-average:

$$V_j \frac{d\mathbf{U}_j}{dt} \quad = \quad - \sum_{k \in \{K_j\}} \int_{\partial T_j^k} \mathbf{H} \, dA_k + \int_{T_j} \mathbf{Q} \, dV. \tag{5.63}$$

On the other hand, by setting $v = \psi_1$ and $v = \psi_2$, we obtain a coupled system for the gradients:

$$\begin{bmatrix} \dfrac{d\left(\overline{\partial_x \mathbf{U}_j}\right)}{dt} \\ \dfrac{d\left(\overline{\partial_y \mathbf{U}_j}\right)}{dt} \end{bmatrix} \quad = \quad M_j^{-1} \begin{bmatrix} - \displaystyle\sum_{k \in \{K_j\}} \int_{\partial T_j^k} \psi_1 \, \mathbf{H} \, dA_k + \int_{T_j} \nabla \psi_1 \cdot (\mathbf{F}, \mathbf{G}) \, dV + \int_{T_j} \psi_1 \, \mathbf{Q} \, dV \\ - \displaystyle\sum_{k \in \{K_j\}} \int_{\partial T_j^k} \psi_2 \, \mathbf{H} \, dA_k + \int_{T_j} \nabla \psi_2 \cdot (\mathbf{F}, \mathbf{G}) \, dV + \int_{T_j} \psi_2 \, \mathbf{Q} \, dV \end{bmatrix}. \tag{5.64}$$

The mass matrix, M_j, which has been formally inverted above to decouple the time derivatives, is given by

$$M_j = \begin{bmatrix} \int_{T_j} \psi_1^2 \, dV & \int_{T_j} \psi_1 \psi_2 \, dV \\ \int_{T_j} \psi_2 \psi_1 \, dV & \int_{T_j} \psi_2^2 \, dV \end{bmatrix} = \frac{V_j}{12} \sum_{i=1}^{3} \begin{bmatrix} (x_i - x_j)^2 & (x_i - x_j)(y_i - y_j) \\ (x_i - x_j)(y_i - y_j) & (y_i - y_j)^2 \end{bmatrix}, \tag{5.65}$$

where (x_i, y_i), $i = 1, 2, 3$, are the vertex coordinates of the triangle. This matrix depends only on the geometry of the grid. Therefore, it can be precomputed and inverted just once, and reused during computations. For the integral that appears on the right hand side of each evolution equation, it is required that the boundary and volume integrals be exact for polynomials of degree three and of degree two (i.e., fourth-order and third-order accurate), respectively [64]. Following Ref. [64], we evaluate the boundary integral with two quadrature points per face (see Figure 5.6):

$$\int_{\partial T_j^k} \psi_s(\mathbf{x}) \, \mathbf{H} \, dA_k = \sum_{n=1}^{2} w_{k_n} \psi_s(\mathbf{x}_{k_n}) \, \mathbf{\Phi}_{k_n}(\mathbf{x}_{k_n}) A_k, \quad k \in \{K_j\}, \; s = 0, 1, 2, \tag{5.66}$$

where the normal flux has been replaced by the upwind numerical flux in the face normal direction:

$$\mathbf{\Phi}_{k_n}(\mathbf{x}_{k_n}) = \frac{1}{2} \left[\mathbf{H}\left(\mathbf{U}_{k_n}^+\right) + \mathbf{H}\left(\mathbf{U}_{k_n}^-\right) \right] - \frac{1}{2} \left| \mathbf{A}_n \right| \left(\mathbf{U}_{k_n}^+ - \mathbf{U}_{k_n}^- \right). \tag{5.67}$$

at each quadrature point, where $\mathbf{U}_{k_n}^-$ and $\mathbf{U}_{k_n}^+$ denote the interface states in the internal element and the neighboring element, respectively. The quadrature parameters are given by

$$w_{k_1} = w_{k_2} = \frac{1}{2}, \tag{5.68}$$

$$\mathbf{x}_{k_1} = \frac{1}{2}\left(1 + \frac{1}{\sqrt{3}}\right)\mathbf{x}_\ell + \frac{1}{2}\left(1 - \frac{1}{\sqrt{3}}\right)\mathbf{x}_r, \quad \mathbf{x}_{k_2} = \frac{1}{2}\left(1 - \frac{1}{\sqrt{3}}\right)\mathbf{x}_\ell + \frac{1}{2}\left(1 + \frac{1}{\sqrt{3}}\right)\mathbf{x}_r. \tag{5.69}$$

Now, we discard the second and third components from the evolution equations, and obtain the following diffusion scheme:

$$V_j \frac{du_j}{dt} = - \sum_{k \in \{K_j\}} \sum_{n=1}^{2} w_{k_n} \, \phi_{k_n}(\mathbf{x}_{k_n}) A_k, \tag{5.70}$$

$$\begin{bmatrix} \dfrac{d\left(\overline{\partial_x u_j}\right)}{dt} \\ \dfrac{d\left(\overline{\partial_y u_j}\right)}{dt} \end{bmatrix} = M_j^{-1} \begin{bmatrix} - \displaystyle\sum_{k \in \{K_j\}} \sum_{n=1}^{2} w_{k_n} \psi_1(\mathbf{x}_{k_n}) \, \phi_{k_n}(\mathbf{x}_{k_n}) A_k + \int_{T_j} \nabla \psi_1 \cdot (f, g) \, dV, \\ - \displaystyle\sum_{k \in \{K_j\}} \sum_{n=1}^{2} w_{k_n} \psi_2(\mathbf{x}_{k_n}) \, \phi_{k_n}(\mathbf{x}_{k_n}) A_k + \int_{T_j} \nabla \psi_2 \cdot (f, g) \, dV \end{bmatrix}, \tag{5.71}$$

where $\phi_{k_n}(\mathbf{x}_{k_n})$ is the first component of the upwind flux, $\mathbf{\Phi}_{k_n}(\mathbf{x}_{k_n})$,

$$\phi_{k_n}(\mathbf{x}_{k_n}) = -\frac{\nu}{2} \left[(p, q)_{k_n}^+ + (p, q)_{k_n}^- \right] \cdot \hat{\mathbf{n}}_{jk} - \frac{1}{2} \sqrt{\frac{\nu}{T_r}} \left(u_{k_n}^+ - u_{k_n}^- \right). \tag{5.72}$$

The vector $\hat{\mathbf{n}}_{jk}$ is the unit outward normal of the face k. Note that the dissipation term of the upwind flux has entered into the diffusive flux as a damping (or penalty) term, and that its coefficient is determined unambiguously. The relaxation time is defined as in Equation (5.10) based on the distance between the centroids of the two adjacent cells (via a quadrature point) projected along the face-normal vector as in Equation (5.56):

$$L_r = \frac{1}{2} \left| \Delta \mathbf{l}_{ik} \cdot \hat{\mathbf{n}}_{ik} \right|, \tag{5.73}$$

where $\Delta \mathbf{l}_{jk} = (x_k - x_j, y_k - y_j)$. For the parameter α, we take $\alpha = 1$ or $\alpha = 3$; the latter corresponds to the fourth-order scheme in one dimension. To evaluate the diffusive flux, we need to define the interface

states. In doing so, we need to evaluate the gradient variables, p and q, whose degrees of freedom have been discarded. In the DG method, these gradients may be evaluated simply by differentiating the approximation of the solution. For the P_1 element considered here, it leads to the constant gradients, $\overline{\nabla u_j} = \left(\overline{\partial_x u_j}, \overline{\partial_y u_j}\right)$ in T_j, and $\overline{\nabla u_k} = \left(\overline{\partial_x u_k}, \overline{\partial_y u_k}\right)$ in T_k. The interface states at the quadrature point, $\mathbf{x}_{k_n}$, are then given by

$$\begin{aligned} u^-_{k_n} &= u_j + \overline{\nabla u_j} \cdot (\mathbf{x}_{kn} - \mathbf{x}_j), \qquad & u^+_{k_n} &= u_k + \overline{\nabla u_k} \cdot (\mathbf{x}_{kn} - \mathbf{x}_k), \\ (p,q)^-_{k_n} &= \overline{\nabla u_j}, & (p,q)^+_{k_n} &= \overline{\nabla u_k}. \end{aligned} \tag{5.74}$$

The volume integral can be evaluated by integration by parts (see Ref.[43] for a method without using integration by parts). The physical flux being the gradient of the solution, i.e., $(f,g) = -\nu\nabla u$, we can perform integration by parts to get

$$\int_{T_j} \nabla\psi_1 \cdot (f,g)\, dV = -\nu \oint_{\partial T_j} u\nabla\psi_1 \cdot \hat{\mathbf{n}}_k dA_k - \int_{T_j} u\nabla^2\psi_1\, dV, \tag{5.75}$$

$$\int_{T_j} \nabla\psi_2 \cdot (f,g)\, dV = -\nu \oint_{\partial T_j} u\nabla\psi_2 \cdot \hat{\mathbf{n}}_k dA_k - \int_{T_j} u\nabla^2\psi_2\, dV. \tag{5.76}$$

For the P_1 DG method, the second terms vanish, and we are left with the boundary integrals only. Since they are now boundary integrals, we can use the same two-point quadrature formula as before with the solution values (the internal state) already prepared for evaluating the interface flux:

$$\int_{T_j} \nabla\psi_1 \cdot (f,g)\, dV = -\nu \sum_{k\in\{K_j\}} \sum_{n=1}^{2} w_{k_n} u(\mathbf{x}_{k_n}) \nabla\psi_1(\mathbf{x}_{k_n}) \cdot \hat{\mathbf{n}}_k A_k, \tag{5.77}$$

$$\int_{T_j} \nabla\psi_2 \cdot (f,g)\, dV = -\nu \sum_{k\in\{K_j\}} \sum_{n=1}^{2} w_{k_n} u(\mathbf{x}_{k_n}) \nabla\psi_2(\mathbf{x}_{k_n}) \cdot \hat{\mathbf{n}}_k A_k. \tag{5.78}$$

This procedure is genuinely third-order accurate, thus satisfying the accuracy requirement for the volume integral. We remark that here, integration by parts was applied to the volume integral with a solid objective: to evaluate the volume integral with a required accuracy.

The diffusion scheme thus constructed is compact, involving only the immediate neighbors. No gradient reconstruction is necessary since we carry the solution gradient in each cell. Also, unlike other DG schemes utilizing auxiliary variables, such as the Bassi-Rebay scheme [34] and the LDG scheme [5], the derived diffusion scheme requires no extra variables and careful discretizations of their equations [65]. Numerical experiments show that the derived diffusion scheme are, despite being remarkably simpler, comparably or more accurate than these well-known DG schemes (see Section 6.2). As in one dimension, the derived diffusion scheme is similar to the interior-penalty schemes [23, 44–47]; but it has distinguished features that it involves no mesh-dependent parameters and even more importantly it involves the skewness measure in the denominator of the damping coefficient through L_r. The latter has an effect of increasing the damping on highly-skewed grids, enabling robust computations on such grids. Again, it should be noted that the derived diffusion scheme has the same implementation structure as a corresponding advection scheme: interface flux evaluated by two discontinuous states meeting at a quadrature point.

5.7 Spectral-Volume Diffusion Scheme

Here, we consider constructing a spectral-volume diffusion scheme on unstructured triangular grids [66]. Similarly to the one-dimensional case, a triangular cell, T, is subdivided systematically into a set of control volumes, $\{C_T\}$, and the solution values are stored within the control volumes as volume-averaged values:

$$\mathbf{U}_i = \frac{1}{V_i} \int_{C_i} \mathbf{U}\, dV, \quad i \in \{C_T\}, \tag{5.79}$$

where C_i denotes the i-th control volume and V_i is the volume of C_i. The triangular cell thus partitioned is called the spectral volume. Given volume-averaged solutions, we construct a polynomial, $\mathbf{U}_T$, of degree m:

$$\mathbf{U}_T(x,y) = \sum_{i\in\{C_T\}} \mathbf{U}_i L_i(x,y), \tag{5.80}$$

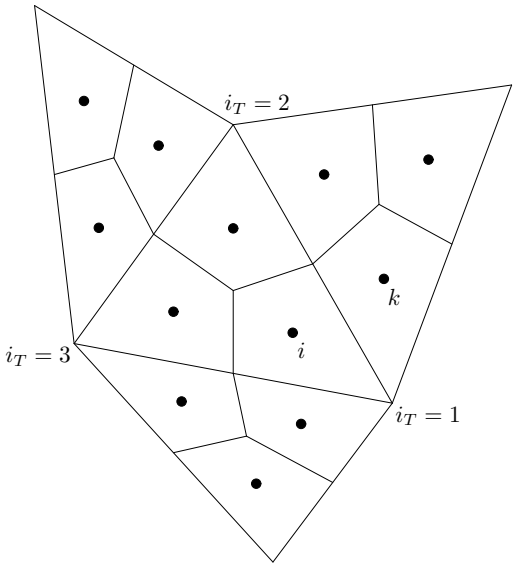


Figure 5.7: Stencil for spectral-volume schemes. Dots indicate the data points defined by the averaged coordinates of the control volumes.

where $L_i(x, y)$ is a shape function that satisfies

$$\frac{1}{V_i}\int_{C_i} \mathbf{U}_T \, dV = \mathbf{U}_i, \quad i \in \{C_T\}. \tag{5.81}$$

The number of control volumes, N, within a spectral volume is determined by the polynomial degree [66]: $N = (m+1)(m+2)/2$. Evolution equation for the volume-averaged value is obtained by integrating the system over the control volume:

$$\frac{d\mathbf{U}_i}{dt} = \frac{1}{V_i}\int_{C_i} (-\mathbf{F}_x - \mathbf{G}_y + \mathbf{Q}) \, dV = -\frac{1}{V_i}\sum_{k\in\{K_i\}} \int_{\partial C_i^k} \mathbf{H} \, dA + \frac{1}{V_i}\int_{C_i} \mathbf{Q} \, dV, \tag{5.82}$$

where $\{K_i\}$ is a set of faces of C_i. The boundary integral over each face is discretized by an $(m+1)^{th}$ order accurate quadrature:

$$\int_{\partial C_i^k} \mathbf{H} \, dA = \sum_{n=1}^{N_p} w_n \mathbf{\Phi}_{ik}(\mathbf{x}_n) A_k, \tag{5.83}$$

where N_p is the number of quadrature points, w_n is the weight at the n-th point, $\mathbf{x}_n$, and A_k is the area of the k-th face. Note also that the boundary flux, $\mathbf{H}$, has been replaced by the upwind numerical flux (5.67) at each quadrature point. Now, by ignoring the second and third components, we derive a diffusion scheme:

$$\frac{du_i}{dt} = -\frac{1}{V_i}\sum_{k\in\{K_i\}} \sum_{n=1}^{N_p} w_n \phi_{ik}(\mathbf{x}_n) A_k, \tag{5.84}$$

where ϕ_{ik} is a diffusive flux of the form (5.72). The interface states for the numerical flux are evaluated at the quadrature point directly by the polynomial representation of the solution over the spectral volume,

$$u_T(x, y) = \sum_{i\in\{C_T\}} u_i L_i(x, y), \tag{5.85}$$

and the neighbors. Note that we now store only the solution values for u. The gradient variables at the interface, which are needed to compute the consistent part of the numerical flux, are evaluated by differentiating the

corresponding polynomial representations at each quadrature point. The two states are continuous over the faces inside the spectral volume and discontinuous over the SV faces. This means that the damping term of the numerical flux vanishes at quadrature points on the interior face, and plays its role at quadrature points on the SV faces. The length scale, L_r, in the numerical flux is determined similarly to the cell-centered schemes:

$$L_r = \frac{1}{2} |\Delta \mathbf{l}_{ik} \cdot \hat{\mathbf{n}}_{ik}|, \tag{5.86}$$

where $\Delta \mathbf{l}_{ik} = (x_k - x_i, y_k - y_i)$ and $\hat{\mathbf{n}}_{ik}$ is the unit outward normal vector of the k-th face. Again, it should be noted that the derived diffusion scheme has exactly the same implementation structure as a corresponding advection scheme: interface flux evaluated by two states meeting at a quadrature point.

For second-order accuracy, we divide a triangular cell into three control volumes ($N = 3$) by connecting the face midpoints and the centroid (see Figure 5.7). Given three volume-averaged solutions, we can construct a polynomial of degree one ($m = 1$) in the form (5.85) over the spectral volume, T. Let $\{i_T\}$ denote a set of vertices of T. The shape function, $L_i(x, y)$, can be expressed in terms of the area-coordinates of T [67]:

$$L_i(x, y) = \frac{36}{5} Z_{i_T}(x, y) - \frac{7}{5}, \tag{5.87}$$

where $i_T \in \{i_T\}$ denotes the vertex of T belonging to the control volume i, and $Z_{i_T}(x, y)$ is the area-coordinate representing the area of a triangle defined by an arbitrary point (x, y) and the two vertices of T other than i_T, divided by the area of the triangle T. For example, $i_T = 1$ for the control volume i in Figure 5.7, and thus

$$Z_1(x, y) = \frac{x(y_2 - y_3) + x_2(y_3 - y) + x_3(y - y_2)}{x_1(y_2 - y_3) + x_2(y_3 - y_1) + x_3(y_1 - y_2)}, \tag{5.88}$$

where (x_1, y_1), (x_2, y_2), (x_3, y_3) are the vertex coordinates of T. It can be verified by direct integration that the polynomial thus defined satisfies the condition (5.81).

As in the discontinuous-Galerkin method, the damping term is crucial in the spectral-volume method also. That is, diffusion schemes may not be consistent without the damping term. The inconsistency problem has been pointed out in Ref. [19] for the one-dimensional diffusion equation.

6 Numerical Results

6.1 One-Dimensional Problem

We consider the following time-dependent diffusion problem:

$$u_t = \nu \, u_{xx}, \quad x \in (0, 1), \tag{6.1}$$

where $\nu = 1$, with the initial solution,

$$u(x, 0) \quad = \quad \sin(14\pi x), \tag{6.2}$$

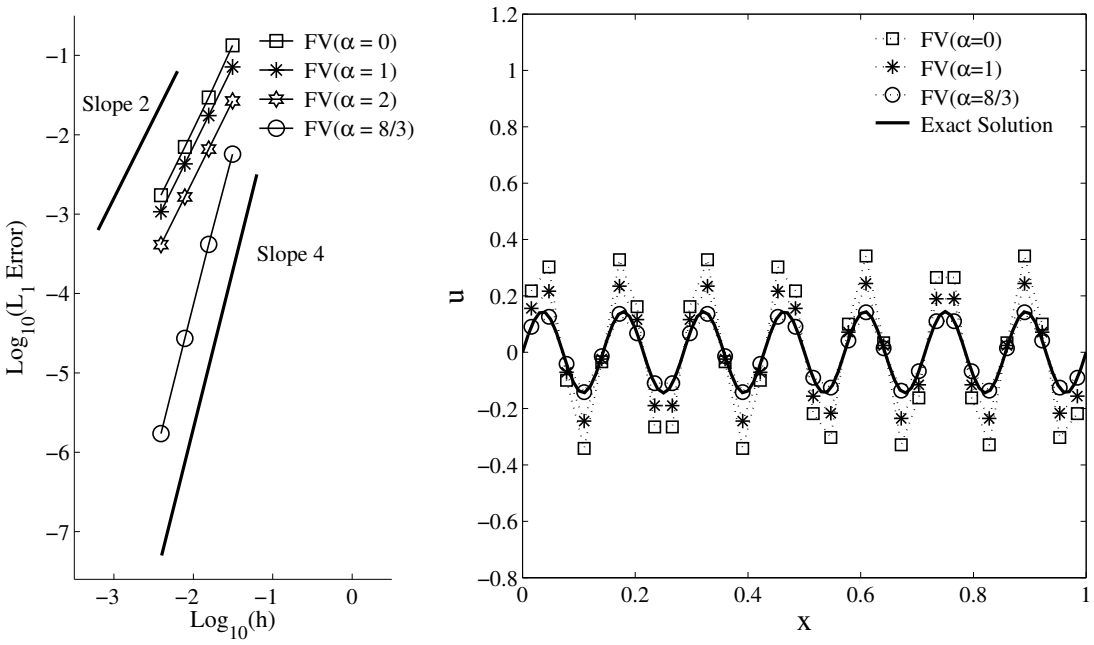
and the boundary conditions, $u(0, t) = 0$ and $u(1, t) = 0$. The exact solution is given by

$$u(x, t) = \exp(-196\pi^2 \nu t) \sin(14\pi x). \tag{6.3}$$

We compute the solution at time $t = 0.001$ on a series of uniform meshes: 32, 64, 128, and 256 cells. The initial solution was assigned in each cell by the exact cell-averaged value of Equation (6.2). The error at the finial time is defined as a difference between the numerical solution and the exact cell-averaged value of Equation (6.3). For the SV schemes, the error is defined for the SV-averaged solution in each SV, not the CV-averaged solutions. The boundary conditions are specified by setting the solution in a ghost cell to be the negative of the solution in the adjacent interior cell, and enforcing equal gradients over the ghost and the adjacent interior cells. For all diffusion schemes, the time step is defined by

$$\Delta t = 0.01 \Delta x^2. \tag{6.4}$$

Hence, all schemes take the same number of time steps to reach the final time: 104 steps for 32 cells, 411 steps for 64 cells, 1640 steps for 128 cells, and 6555 steps for 256 cells. For second-order schemes, we employ the forward-Euler time-stepping scheme to integrate them in time. For fourth-order schemes, we employ the classical fourth-order Runge-Kutta time integration scheme.



(a) L_1 error convergence.

(b) Numerical solutions (32 cells).

Figure 6.1: Numerical results for one-dimensional FV schemes.

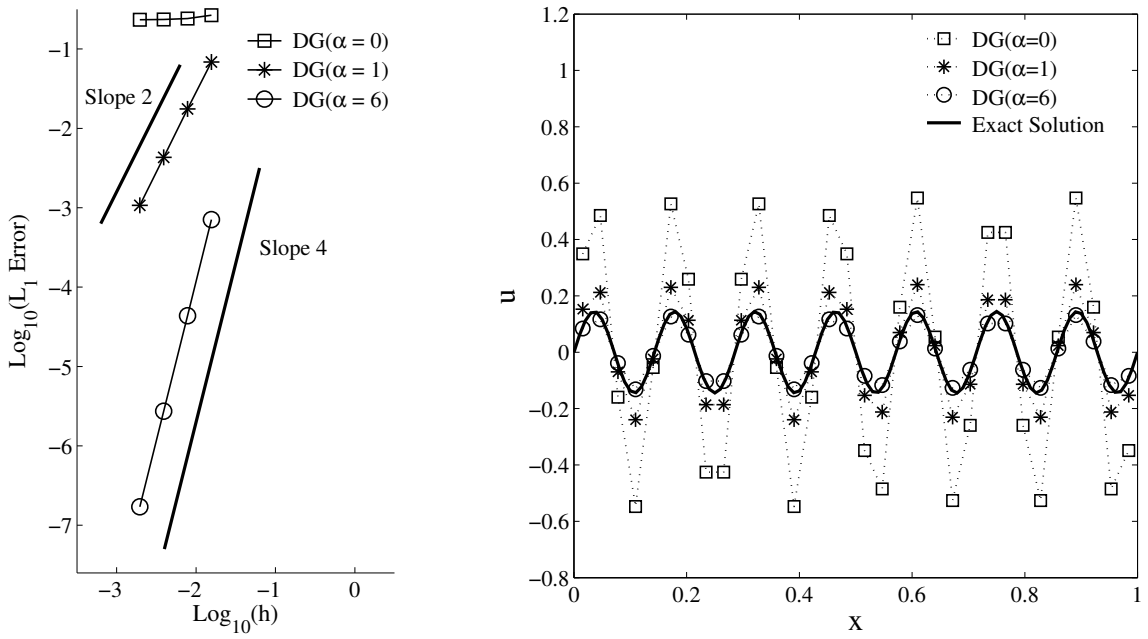
FV($\alpha = 1$)	FV($\alpha = 2$)	FV($\alpha = 8/3$)	DG($\alpha = 1$)	DG($\alpha = 6$)	SV($\alpha = 1$)	SV($\alpha = 3$)
$\frac{\Delta x^2}{\nu}$	$\frac{1}{2}\frac{\Delta x^2}{\nu}$	$\frac{3}{8}\frac{\Delta x^2}{\nu}$	$\frac{1}{4}\frac{\Delta x^2}{\nu}$	$\frac{1}{12}\frac{\Delta x^2}{\nu}$	$\frac{1}{4}\frac{\Delta x^2}{\nu}$	$\frac{1}{12}\frac{\Delta x^2}{\nu}$

Table 6.1: Maximum explicit time step for the one-dimensional FV, DG, and SV diffusion schemes.

In Figure 6.1, the L_1 error convergence results and the solution plots (32 cells) are shown for the one-dimensional finite-volume (FV) diffusion schemes. The mesh size h in Figure 1(a) is a cell volume. As predicted, the FV diffusion scheme is second-order accurate with $\alpha = 1$, and fourth-order accurate with $\alpha = 8/3$. The FV scheme with $\alpha = 0$ is also second-order accurate, but slightly less accurate than the scheme with $\alpha = 1$. This is considered due to the lack of damping, which can be clearly seen in the solution plot. The scheme with $\alpha = 2$ corresponds to the standard three-point finite-difference scheme (the Galerkin scheme). The corresponding solution plot is not shown for clarity, but it lies between those of $\alpha = 1$ and $\alpha = 8/3$, as can be expected. The three-point scheme produces more accurate solutions than the other second-order schemes, but its maximum allowable time step is as small as the one for the fourth-order scheme ($\alpha = 8/3$).

In Figure 6.2, results are shown for the DG schemes constructed in Section 4.2. The mesh size h here is defined as half a cell volume to take into account the fact that the DG schemes carry two degrees of freedom per cell. The DG diffusion scheme is confirmed to be second-order accurate with $\alpha = 1$, fourth-order accurate with $\alpha = 6$, and inconsistent with $\alpha = 0$. As can be seen from the solution plot, the inconsistent scheme is highly inaccurate. It never approaches the exact solution in the grid refinement.

In Figure 6.3, results are shown for the SV schemes. Figure 3(a) confirms that the SV scheme is second-order accurate with $\alpha = 1$, fourth-order accurate with $\alpha = 3$, and inconsistent with $\alpha = 0$. Note that similarly to the DG case, the mesh size h is defined as half the cell volume. Again, we see that the inconsistent scheme gives highly inaccurate solution, and it does not converge to the exact solution with grid refinement. We note also that the SV scheme ($\alpha = 1$) gives more accurate solutions than the FV and DG schemes with $\alpha = 1$. We observed also from experiments that the SV scheme produced very similar second-order results for various other values of α such as 0.6, 1.5, 10, 100, etc.



(a) L_1 error convergence.

(b) Numerical solutions (32 cells).

Figure 6.2: Numerical results for one-dimensional DG schemes.

Finally, the maximum allowable explicit (forward-Euler) time-steps are compared in Table 6.1, which are derived based on the Fourier analysis in Section 4. It shows that the FV schemes generally allow larger time steps than the DG and SV schemes considered here. Also, it can be seen that the fourth-order FV scheme allows as a large time step as the Galerkin scheme does, which is almost five times larger than the maximum time step for the fourth-order DG and SV schemes.

6.2 Two-Dimensional Problem: Anisotropic Irregular Grids

We consider the following diffusion problem:

$$u_t = \nu(u_{xx} + u_{yy}), \quad (x, y) \in (0, 1) \times (0, 0.005), \tag{6.5}$$

where $\nu = 1$, with the strongly anisotropic initial solution,

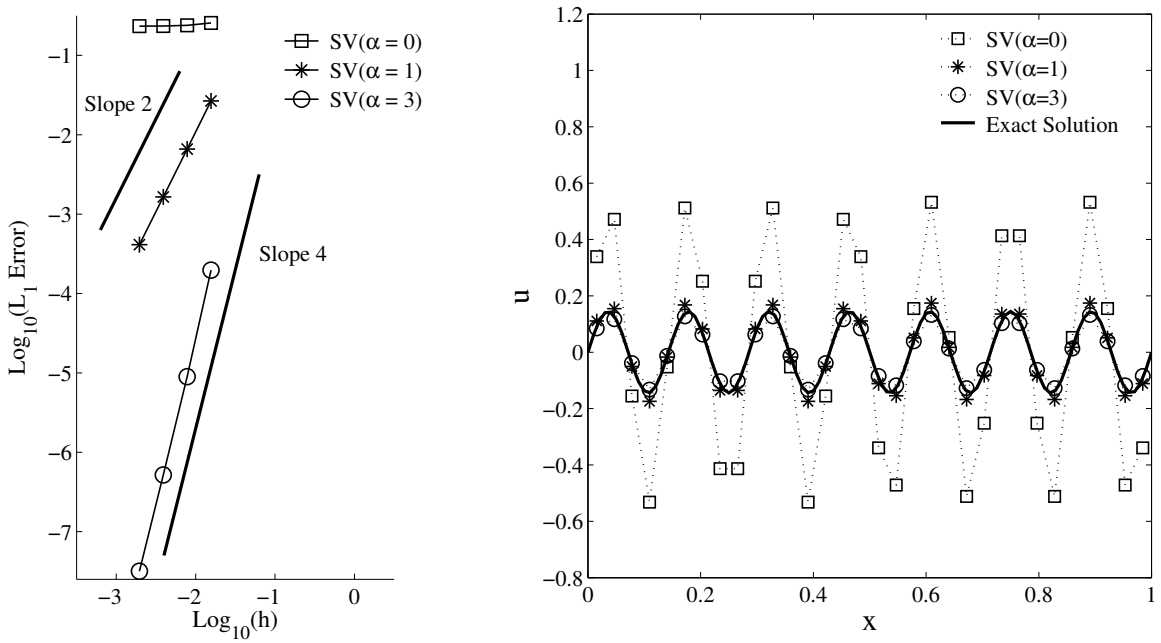
$$u(x, y, 0) = 5 \sin(\pi x) \sin(4000\pi y), \tag{6.6}$$

and the boundary condition, $u = 0$ on the boundary. The exact solution to this problem is given by

$$u(x, y, t) = 5 \exp(-16000001\pi^2 \nu t) \sin(\pi x) \sin(4000\pi y). \tag{6.7}$$

We compute the solution at time $t = 10^{-8}$ on a series of 15 irregular triangular grids. The irregular grids were generated by random diagonal splittings and nodal perturbations from uniform Cartesian grids of $n \times n$ nodes, where $n = 25, 33, 41, 49, 57, 65, 73, 81, 89, 97, 105, 113, 121, 129, 137$, resulting in the cell aspect-ratio of nearly 200 (see Figure 6.4). These grids are highly-skewed (possibly adapted) viscous-type grids. The L_1-norm of the skewness measure, $\hat{\mathbf{e}}_{jk} \cdot \hat{\mathbf{n}}_{jk}$, is 0.1 for all grids with the minimum value of around 0.007. Diffusion schemes should be able to produce accurate results on such grids to be successfully applied to viscous simulations.

For gradient reconstruction, we employ the Green-Gauss formula [43] for node-based finite-volume and residual-distribution schemes, except for the compact edge-based finite-volume scheme for which we employ the least-squares gradient as discussed in Section 5.3.3. For cell-centered finite-volume schemes, we apply the same least-squares method as above to cell-averages.



(a) L_1 error convergence.

(b) Numerical solutions (32 cells).

Figure 6.3: Numerical results for one-dimensional SV schemes.

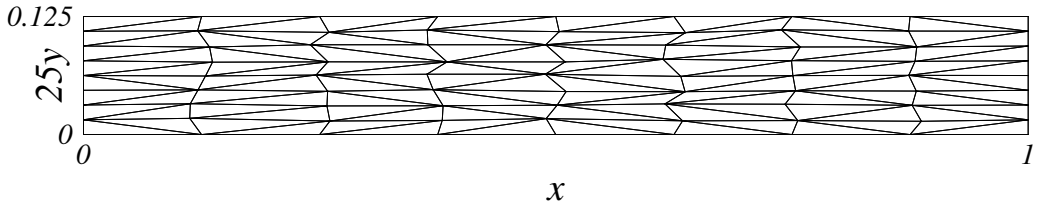


Figure 6.4: A typical high aspect-ratio irregular triangular grid (9x9 grid). The y-axis is stretched by a factor of 25 to show the details of the grid. Interior nodes are randomly perturbed in both coordinate directions.

In comparing the error convergence, we define the mesh size h as the square root of a control volume, except that it is defined as the square root of the 1/3 of the control volume for DG and SV schemes to take into account the fact that they carry three degrees of freedom per control volume. Boundary conditions are specified at the boundary nodes for node-based schemes, and at the cells having boundary faces for cell-based schemes (at control volumes for the SV schemes). To reach the final time, all schemes are integrated in time by the forward-Euler time-stepping scheme with a common global time step:

$$\Delta t = 0.003\, h^2. \tag{6.8}$$

All schemes, therefore, take exactly the same number of time steps to reach the final time. For all schemes, the error is computed based on the difference between the exact solution and the numerical solution at data points: nodes for node-based schemes, centroids of triangles for cell-based schemes including DG and SV schemes.

Figure 6.5 shows the error convergence for the node-centered finite-volume schemes derived in Section 5.3.2 and the Galerkin scheme. It is observed that the edge-based finite-volume (EBFV) and node-centered compact finite-volume (NCFV-Cmp) schemes with $\alpha = 1$ or $\alpha = 4/3$ produce much more accurate solutions than the Galerkin scheme. The edge-based scheme gives the most accurate solutions when $\alpha = 4/3$. The Avg-LSQ-EN scheme, as expected, lacks damping for these highly-skewed grids, and resulted in significantly large errors. Observe that the Avg-LSQ-EN scheme and the EBFV scheme with $\alpha = 0$ give very similar results, clearly

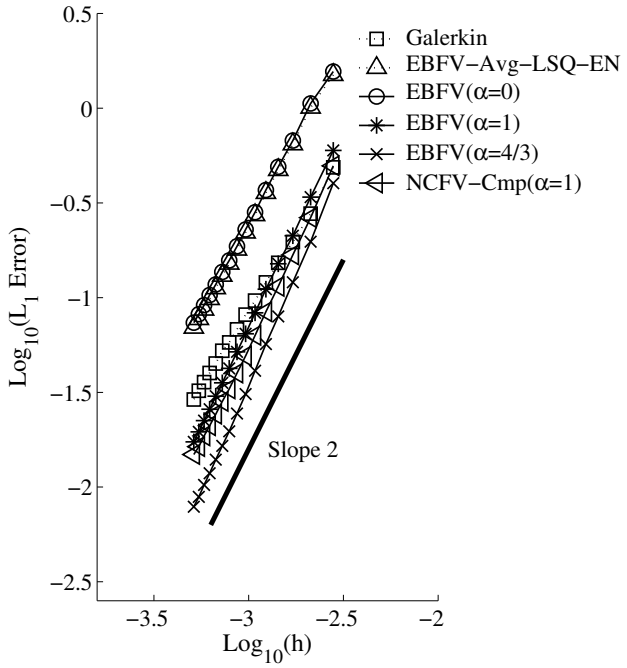


Figure 6.5: L_1 error convergence for NCFV schemes.

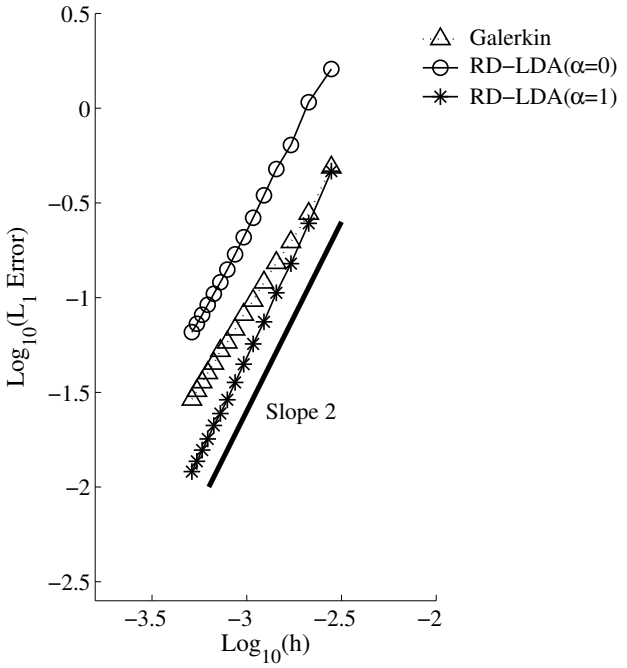


Figure 6.6: L_1 error convergence for RD schemes.

confirming the lack of damping. We remark also that the NCFV-Cmp scheme with the Green-Gauss gradients (the Braaten-Connell scheme) gives very similar results to those obtained by using the least-squares gradients with $\alpha = 1$ (therefore, not shown).

Figure 6.6 shows the error convergence for the residual-distribution (RD) schemes. Observe that the LDA (RD-LDA) diffusion scheme with $\alpha = 1$ yields significantly most accurate solutions with a slightly higher convergence than second-order. The LDA diffusion scheme is second-order accurate with $\alpha = 0$, but significantly less accurate due to the lack of damping.

Figure 6.7 shows the error convergence for the cell-centered finite-volume (CCFV) schemes and the widely-used cell-centered AvgLSQ-EN scheme [25,26,55]. As can be seen in Figure 6.7, the CCFV-AvgLSQ-EN scheme is significantly inaccurate, and gets unstable for fine grids. The lack of damping is evident by a similar behavior of the derived CCFV with $\alpha = 0$. On the other hand, the derived CCFV scheme is second-order accurate with $\alpha = 1$ and $\alpha = 4/3$; the latter gives significantly more accurate solutions.

Figure 6.8 shows the error convergence for the DG schemes. All schemes are nearly second-order accurate. The Bassi-Rebay, LDG, and DG($\alpha = 1$) schemes are equally accurate. The DG ($\alpha = 3$) scheme produces significantly more accurate solutions. Observe that the DG ($\alpha = 0$) scheme is inconsistent and unstable.

Figure 6.9 shows the results for the SV schemes: Bassi-Rebay and LDG methods applied to SV schemes, and the derived SV scheme. Again, all schemes show nearly second-order behavior, and give comparably accurate solutions. The derived SV scheme with $\alpha = 3/2$ yields similar solutions to those with $\alpha = 1$ (therefore, not shown).

Figures 6.16-6.37 show section plots of the numerical solutions at $y = 0.078125$ for various schemes on the 137x137 grid. The dash-dot curve is the section plot of the exact solution. Note that the plot of the exact solution is not perfectly smooth. This is because the section plot was created from the exact solution projected onto the computational grid that is irregular. Figure 6.16 shows that the Galerkin scheme produces highly oscillatory solution. This failure is considered due to the lack of positivity of the Galerkin scheme on grids that are not Delaunay [68]. A popular technique of ignoring all negative contributions does not help: the solution is still oscillatory [43]), and moreover the Galerkin scheme loses consistency with the positivity enforcement [43]. Accurate solution is obtained by the RD-LDA scheme as in Figure 6.17. For comparison, the solution by the RD-LDA with $\alpha = 0$ is shown in Figure 6.18. Without the damping ($\alpha = 0$), the RD-LDA scheme gives highly oscillatory and inaccurate solution. Figures 6.19 and 6.20 show the results for the EBFV-AvgLSQ-EN scheme and the EBFV scheme with no damping ($\alpha = 0$). Their solutions are similarly oscillatory and

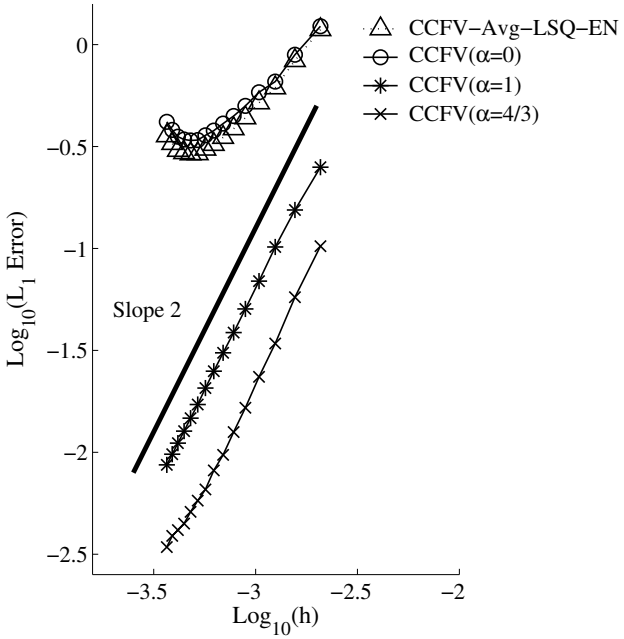


Figure 6.7: L_1 error convergence for CCFV schemes.

inaccurate, which demonstrates the lack of damping of the EBFV-AvgLSQ-EN scheme on highly-skewed grids. Very accurate solutions are obtained by the EBFV schemes as shown in Figures 6.21 and 6.22; the EBFV scheme with $\alpha = 4/3$ gives the best solution among all node-based schemes. Results for the NCFV-Cmp schemes are shown in Figures 6.23-6.26. As can be seen, the NCFV-Cmp schemes with the least-squares and Green-Gauss gradients, and with $\alpha = 1$ and $\alpha = 4/3$ give very similar results. These results indicate that the NCFV-Cmp scheme is relatively insensitive to the gradient reconstruction methods as well as the choice of α. But the NCFV-Cmp scheme with no damping ($\alpha = 0$) yields a highly oscillatory solution (see Figure 6.26).

The rest of the results are for cell-based schemes. To create section plots, we first compute nodal solutions by averaging the solution values extrapolated to the nodes from the surrounding triangles (using the gradients reconstructed or computed as unknowns in each cell), and then plot the solution in the same way as those of the node-based schemes. Figures 6.27 and 6.28 show the results for the CCFV-Avg-LSQ-EN scheme and the derived CCFV scheme with no damping ($\alpha = 0$). Both solutions are oscillatory and inaccurate as expected, again, demonstrating the lack of damping of the CCFV-Avg-LSQ-EN scheme on highly-skewed grids. On the other hand, the CCFV schemes with the least-squares gradients are much more accurate, especially with $\alpha = 4/3$, as shown in Figures 6.29 and 6.30.

Shown in Figures 6.31-6.35 are the results for DG schemes. The Bassi-Rebay scheme and the LDG schemes give somewhat oscillatory solutions but not too far from the exact solution as shown in Figures 6.31 and 6.32. The solution obtained by the derived DG scheme with $\alpha = 1$ is slightly more oscillatory (see Figures 6.33), but gives a much more accurate solution with $\alpha = 3$ (see Figures 6.34). Again, the derived DG scheme with $\alpha = 0$ (i.e., no damping) gives a catastrophically inaccurate solution as seen in Figure 6.35.

For SV schemes, the SV-Bassi-Rebay scheme produces a slightly irregular solution although accurate on average as shown in Figures 6.36 while the SV-LDG scheme produces a much smoother solution to this problem as shown in Figure 6.37. The derived SV scheme with $\alpha = 1$ gives slightly oscillatory but, on average, accurate solutions, as shown in Figure 6.38. Finally, the derived SV scheme with $\alpha = 0$ lead to, due to the lack of damping, extremely inaccurate solutions as shown in Figure 6.39.

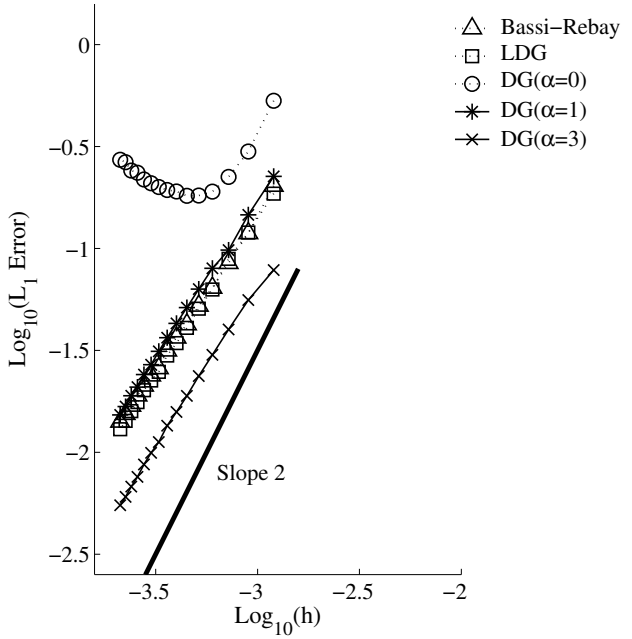


Figure 6.8: L_1 error convergence for DG schemes.

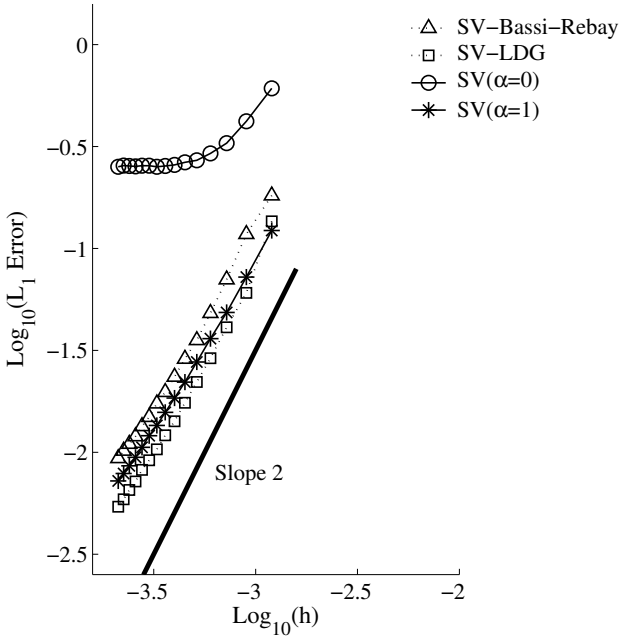


Figure 6.9: L_1 error convergence for SV schemes.

7 Concluding Remarks

7.1 General Remarks

In this paper, we have introduced a general principle for constructing diffusion schemes: discretize the first-order hyperbolic diffusion system by an advection scheme and then derive a diffusion scheme from it by discarding extra equations and explicitly evaluating the solution gradient. We have demonstrated the principle by deriving a one-parameter-family of second-order diffusion schemes for node/cell-centered finite-volume, residual-distribution, discontinuous-Galerkin, and spectral-volume methods for uniform grids in one dimension and for unstructured grids in two dimensions. It has been shown that a damping term, which is important for high-frequency error damping, is automatically incorporated into the derived diffusion scheme through the dissipation term of the generating advection scheme for all methods considered. A parameter α, which represents the ratio of the maximum explicit time step to the relaxation time of the hyperbolic diffusion system, has been shown to play a role of controlling the damping effect of the derived diffusion scheme. Special values of α have been discovered for one-dimensional schemes that make them fourth-order accurate, and they have been shown to give significantly accurate solutions even in two dimensions. For two-dimensional schemes, a skewness parameter has been shown to be incorporated naturally into the damping coefficient, enabling robust and accurate computations on highly-skewed grids. Numerical results have been shown to demonstrate the design accuracy of the derived diffusion schemes and the importance of the damping term (i.e., a lack of damping leads to unstable and/or inconsistent results). Comparison with widely-used schemes have been made to demonstrate that these derived diffusion schemes give comparable or more accurate solutions for time-dependent diffusion problems on a uniform grid in one dimension, and on highly-skewed anisotropic irregular triangular grids in two dimensions.

7.2 Derived Diffusion Schemes

Diffusion schemes derived from the proposed principle are summarized below.

1. Finite-Volume Schemes
   (a) In one dimension, a one-parameter family of second-order diffusion schemes has been derived. It becomes fourth-order accurate with $\alpha = 8/3$.
   (b) Edge-based and compact two-dimensional node-centered diffusion schemes have been constructed. The edge-based scheme has been shown to produce significantly accurate solutions with $\alpha = 4/3$.

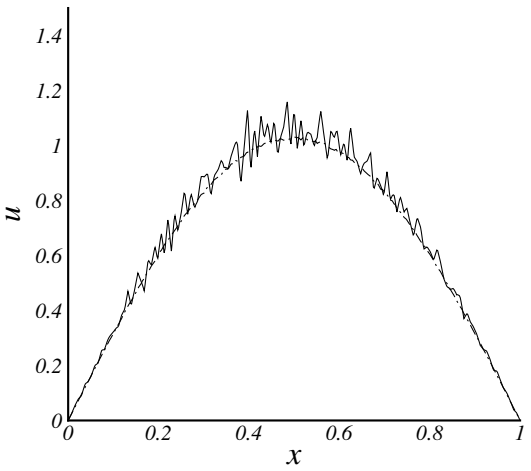


Figure 6.16: Galerkin scheme.

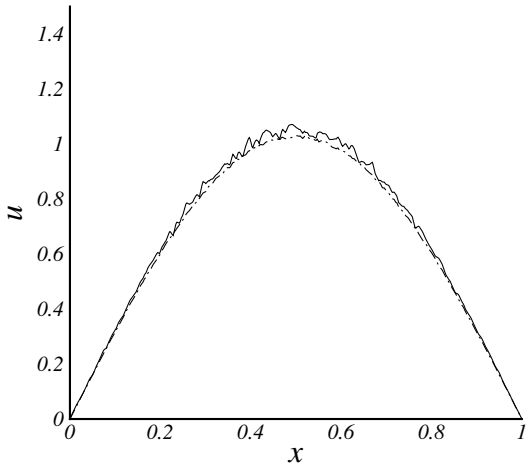


Figure 6.17: RD-LDA scheme ($\alpha = 1$).

(c) Two-dimensional cell-centered diffusion schemes have been constructed. The choice $\alpha = 4/3$ gives significantly accurate solutions on highly-skewed grids.

(d) A widely-used edge-normal average-least-squares scheme has been shown to break down for highly-skewed grids due to a lack of damping, while the derived diffusion schemes produce very accurate results on such grids due to the damping term that is automatically amplified on such grids.

2. Residual-Distribution Schemes

   (a) A two-dimensional diffusion scheme has been constructed from the LDA scheme. The form of a damping term is identified.

   (b) The derived scheme has been shown to produce very accurate solutions on highly-skewed grids.

3. Discontinuous Galerkin Schemes

   (a) A one-parameter family of one-dimensional second-order diffusion schemes has been derived, which has been shown to be fourth-order accurate with $\alpha = 6$.

   (b) A two-dimensional diffusive flux has been constructed from the upwind flux. It has been shown to give comparably or more accurate results than the Bassi-Rebay and LDG schemes. The choice $\alpha = 3$ gives more accurate results than $\alpha = 1$ in two dimensions.

4. Spectral-Volume Schemes

   (a) A one-parameter family of one-dimensional second-order diffusion schemes has been derived, which has been shown to be fourth-order accurate with $\alpha = 3$.

   (b) A two-dimensional diffusion scheme has been constructed from the upwind flux. The derived scheme has been shown to give comparably accurate results with those with the Bassi-Rebay and LDG schemes. The derived scheme is insensitive to the value of α. The choice $\alpha = 3/2$, which corresponds to the value that achieves fourth-order accuracy in one dimension, gives similar results for $\alpha = 1$.

7.3 Future Work

Future work includes extensions to yet other discretization methods. The principle introduced in this paper is independent of the discretization method, and it should be easily applied to other methods. Extensions to three dimensions are straightforward. Many diffusion schemes derived in this paper are directly applicable to three-dimensional grids. If not, a diffusion scheme can always be derived, provided an appropriate three-dimensional

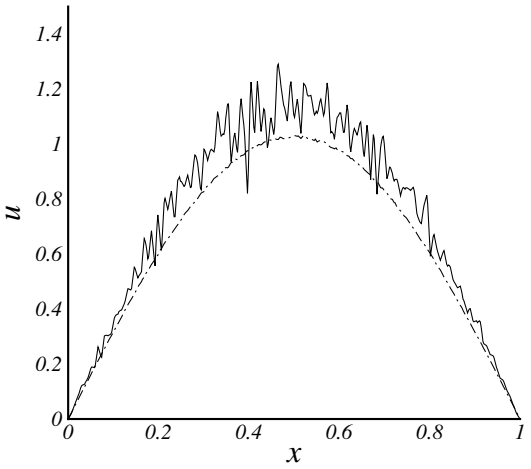


Figure 6.18: RD-LDA scheme ($\alpha = 0$).

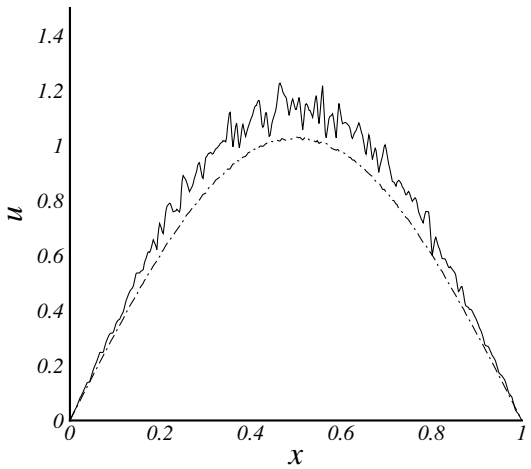


Figure 6.19: Avg-LSQ-EN scheme.

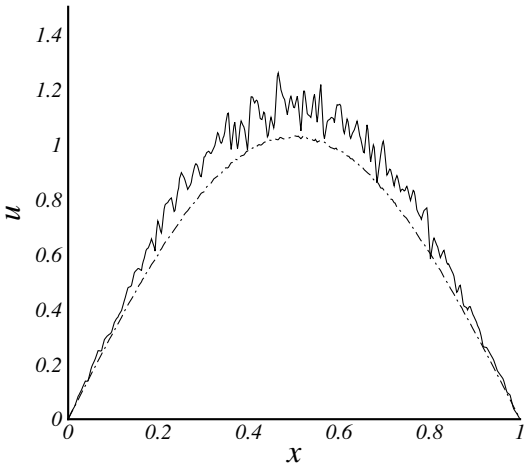


Figure 6.20: EBFV scheme ($\alpha = 0$).

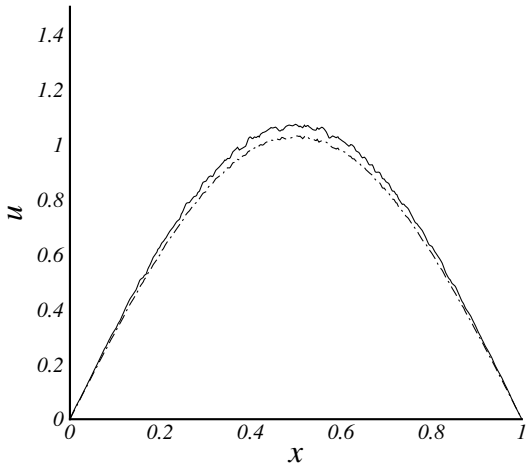


Figure 6.21: EBFV scheme ($\alpha = 1$).

advection scheme is available For practical applications, implicit formulations of the schemes derived in this paper should be explored. The positivity of the damping coefficient is expected to improve diagonal dominance of the resulting linear system for highly-skewed grids. The performance of the diffusion schemes for steady state computations should be studied, not just for accuracy but also for convergence to the steady state, perhaps in multigrid methods where high-frequency damping is essential for grid-independent convergence.

Among various future work, we first focus on the application of the proposed principle and derived schemes to nonlinear systems. In a subsequent paper, we discuss strategies for extending the proposed approach to the compressible Navier-Stokes equations.

References

[1] R. Kannan, Z. J. Wang, A study of viscous flux formulations for a p-multigrid spectral volume navier stokes solver, Journal of Scientific Computing 41 (2009) 165–199.

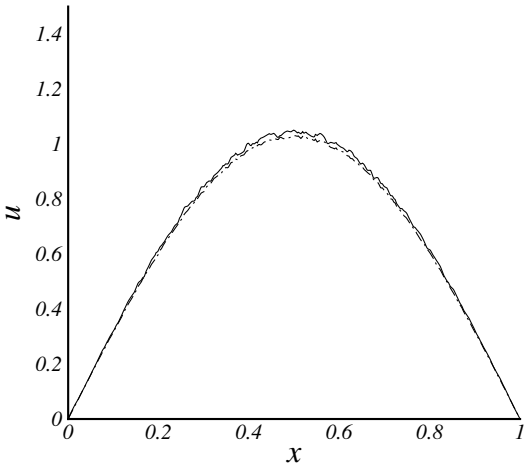


Figure 6.22: EBFV scheme ($\alpha = \frac{4}{3}$).

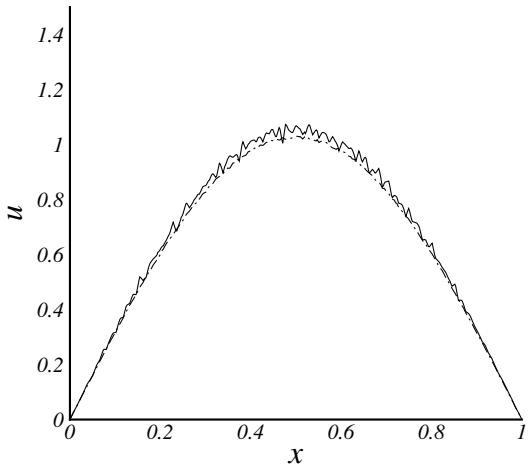


Figure 6.23: NCFV-Cmp scheme ($\alpha = 1$).

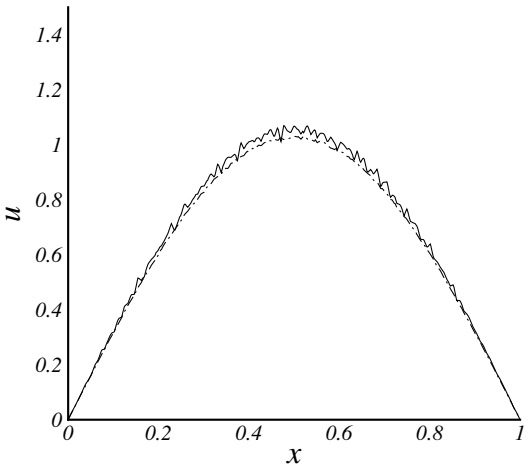


Figure 6.24: NCFV-Cmp scheme with the Green-Gauss gradients (the Braaten-Connell scheme).

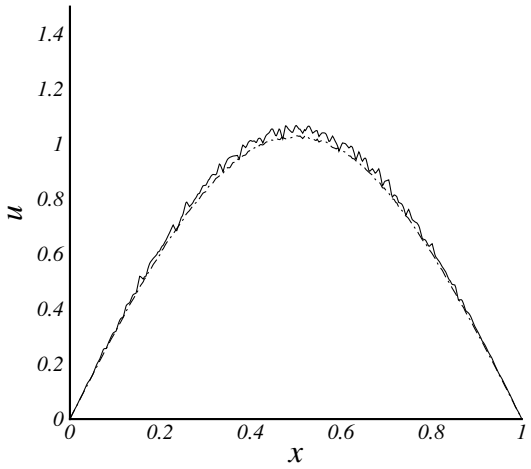


Figure 6.25: NCFV-Cmp scheme ($\alpha = \frac{4}{3}$).

[2] G. Gassner, F. Lörcher, C. D. Munz, A contribution to the construction of diffusion fluxes for finite volume and discontinuous Galerkin schemes, Journal of Computational Physics 224 (2007) 1049–1063.

[3] B. van Leer, M. Lo, Analysis and implementation of recovery-based discontinuous Galerkin for diffusion, in: 19th AIAA Computational Fluid Dynamics Conference, AIAA Paper 2009-3786, San Antonio, 2009.

[4] R. Kannan, Y. Sun, Z. J. Wang, A study of viscous flux formulations for an implicit P-multigrid spectral volume Navier Stokes solver, in: 46th AIAA Aerospace Sciences Meeting, AIAA Paper 2008-783, 2008.

[5] Y. Xu, C.-W. Shu, Local discontinuous Galerkin methods for high-order time-dependent partial differential equations, Communications in Computational Physics 7 (1) (2010) 1–46.

[6] H. Liu, J. Yan, The direct discontinuous Galerkin (DDG) methods for diffusion problems, SIAM Journal of Numerical Analysis 47 (1) (2009) 675–698.

[7] J. Peraire, P.-O. Persson, The compact discontinuous Galerkin (CDG) method for elliptic problems, SIAM Journal of Scientific Computing 30 (2) (2008) 1806–1824.

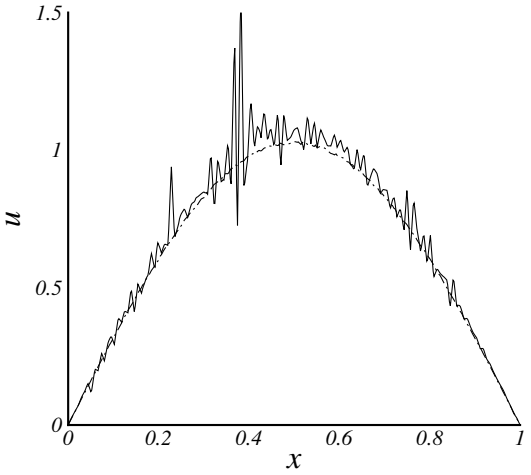


Figure 6.26: NCFV-Cmp scheme ($\alpha = 0$).

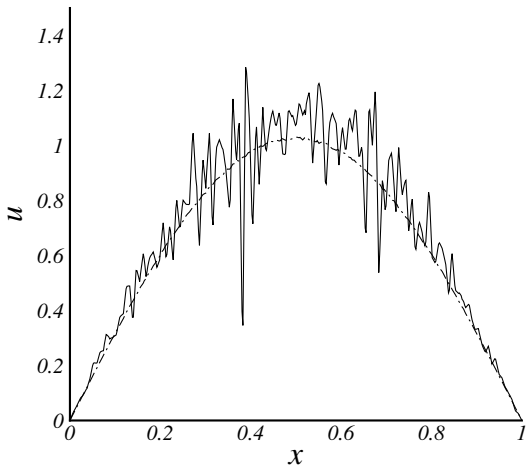


Figure 6.27: CCFV-Avg-LSQ-EN.

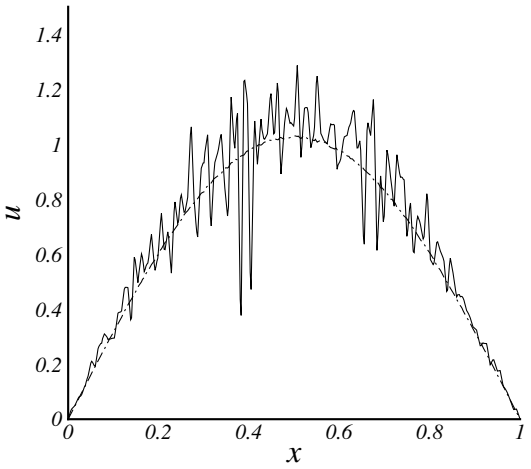


Figure 6.28: CCFV scheme ($\alpha = 0$).

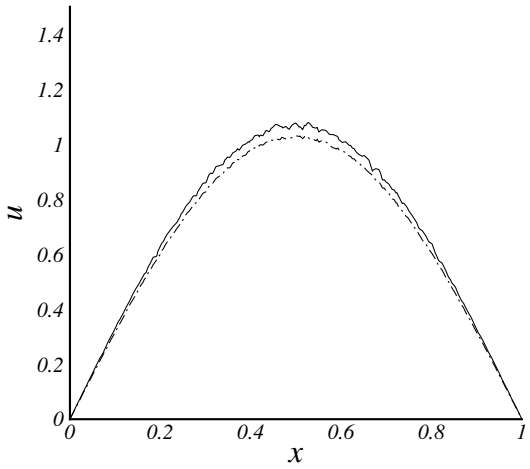


Figure 6.29: CCFV scheme ($\alpha = 1$).

[8] H. T. Huynh, A reconstruction approach to high-order schemes including discontinuous Galerkin for diffusion, in: 47th AIAA Aerospace Sciences Meeting, AIAA Paper 2009-403, Orlando, 2009.

[9] G. Puigt, V. Auffray, J.-D. Müller, Discretization of diffusive fluxes on hybrid grids, Journal of Computational Physics 229 (2010) 1425–1447.

[10] H. Nishikawa, Towards future Navier-Stokes schemes: Uniform accuracy, $O(h)$ time step, and accurate viscous/heat fluxes, in: 19th AIAA Computational Fluid Dynamics Conference, AIAA Paper 2009-3648, San Antonio, 2009.

[11] S. P. Veluri, C. J. Roy, A. Choudhary, E. A. Luke, Finite volume diffusion operators for compressible CFD on unstructured grids, in: 19th AIAA Computational Fluid Dynamics Conference, AIAA Paper 2009-4141, San Antonio, 2009.

[12] B. Diskin, J. L. Thomas, E. J. Nielsen, H. Nishikawa, J. A. White, Comparison of node-centered and cell-centered unstructured finite-volume discretizations: Viscous fluxes, AIAA Journal 48 (7) (2010) 1326–1338.

[13] K. Lipnikov, D. Svyatskiy, Y. Vassilevski, Interpolation-free monotone finite volume method for diffusion equations on polygonal meshes, Journal of Computational Physics 228 (2009) 703–716.

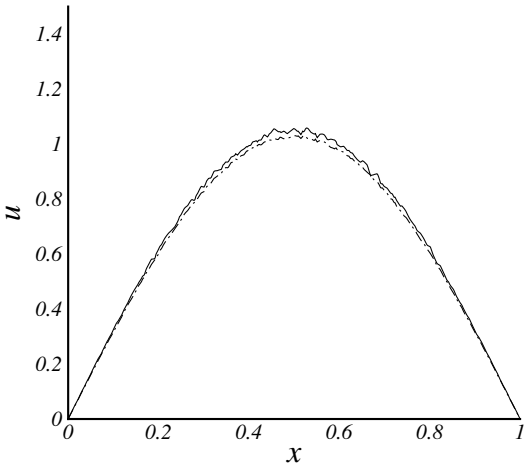


Figure 6.30: CCFV scheme ($\alpha = \frac{4}{3}$).

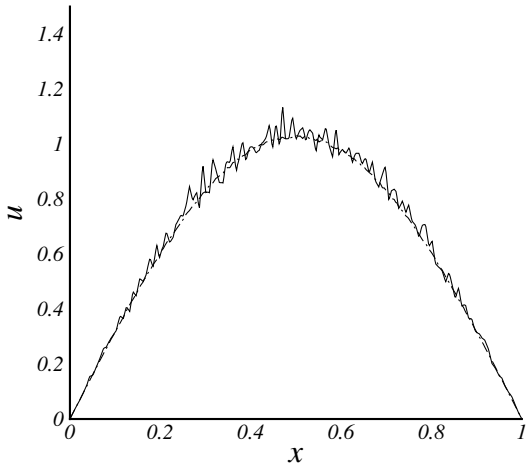


Figure 6.31: Bassi-Rebay scheme.

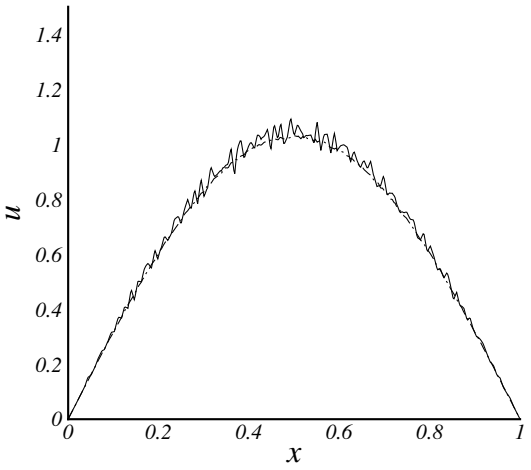


Figure 6.32: LDG scheme.

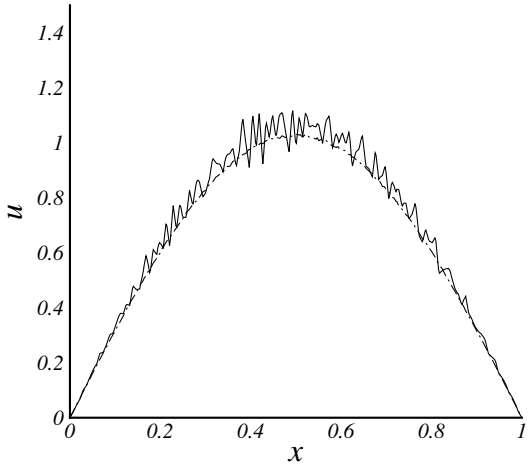


Figure 6.33: DG scheme ($\alpha = 1$).

[14] F. Hermeline, A finite volume method for approximating 3D diffusion operators on general meshes, Journal of Computational Physics 228 (2009) 5763–5786.

[15] P. Traore, Y. M. Ahipo, C. Louste, A robust and efficient finite volume scheme for the discretization of diffusive flux on extremely skewed meshes in complex geometries, Journal of Computational Physics 228 (2009) 5148–5159.

[16] F. Hermeline, Monotone finite volume schemes for diffusion equations on polygonal meshes, Journal of Computational Physics 227 (2008) 6288–6312.

[17] J. Breil, P.-H. Maire, A cell-centered diffusion scheme on two-dimensional unstructured meshes, Journal of Computational Physics 224 (2007) 785–823.

[18] B. Cockburn, C.-W. Shu, Runge-Kutta discontinuous Galerkin methods for convection-dominated problems, Journal of Scientific Computing 16 (3) (2001) 173–261.

[19] Y. Sun, Z. J. Wang, Formulations and analysis of the spectral volume method for the diffusion equation, Communications in Numerical Methods in Engineering 20 (2004) 927–937.

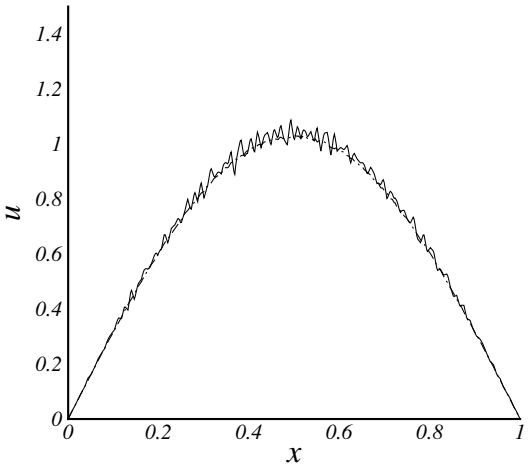


Figure 6.34: DG scheme ($\alpha = 3$).

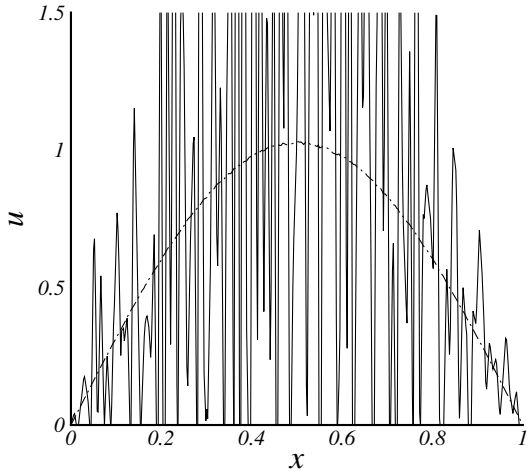


Figure 6.35: DG scheme ($\alpha = 0$).

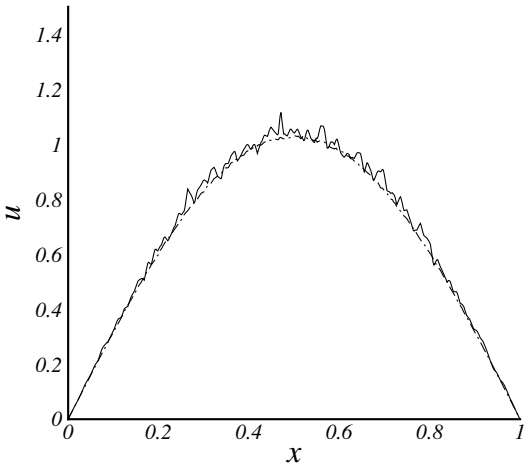


Figure 6.36: SV-Bassi-Rebay scheme.

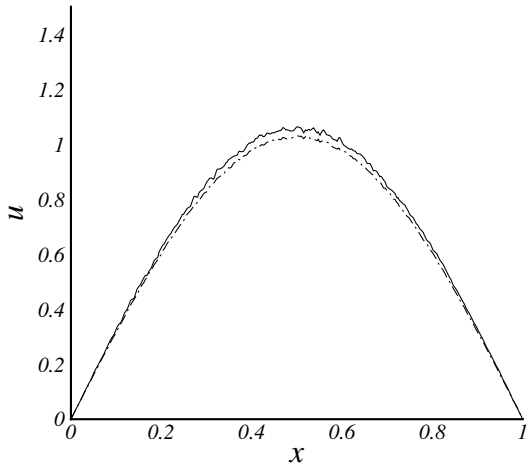


Figure 6.37: SV-LDG scheme.

[20] U. Trottenberg, C. W. Oosterlee, A. Schüller, Multigrid, Academic Press, 2000.
[21] B. Diskin, J. L. Thomas, Accuracy analysis for mixed-element finite-volume discretization schemes, NIA Report No. 2007-08.
[22] P. De Palma, G. Pascazio, D. T. Rubino, M. Napolitano, Residual distribution schemes for advection and advection-diffusion problems on quadrilateral cells, Journal of Computational Physics 218 (2006) 159–199.
[23] D. N. Arnold, An interior penalty finite element method with discontinuous elements, SIAM Journal on Numerical Analysis 19 (4) (1982) 742–760.
[24] D. N. Arnold, F. Brezzi, B. Cockburn, L. D. Marini, Unified analysis of discontinuous Galerkin methods for elliptic problems, SIAM Journal on Numerical Analysis 39 (5) (2002) 1749–1779.
[25] A. Haselbacher, J. J. McGuirk, G. J. Page, Finite volume discretization aspects for viscous flows on mixed unstructured grids, AIAA Journal 37 (2) (1999) 177–184.
[26] J. M. Weiss, J. P. Maruszeski, W. A. Smith, Implicit solution of preconditioned Navier-Stokes equations using algebraic multigrid, AIAA Journal 37 (1) (1999) 29–36.
[27] H. Nishikawa, P. L. Roe, On high-order fluctuation-splitting schemes for Navier-Stokes equations, in:

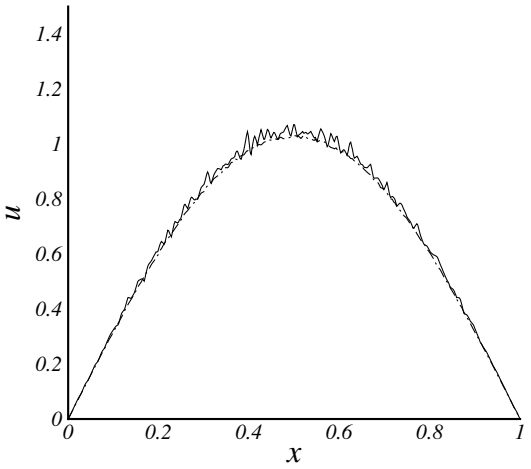


Figure 6.38: SV scheme ($\alpha = 1$).

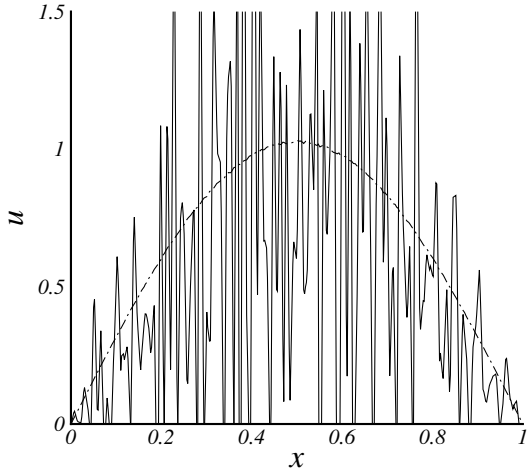


Figure 6.39: SV scheme ($\alpha = 0$).

C. Groth, D. W. Zingg (Eds.), Computational Fluid Dynamics 2004, Springer-Verlag, 2004, pp. 799–804.

[28] M. Ricchiuto, N. Villedieu, R. Abgrall, H. Deconinck, On uniformly high-order accurate residual distribution schemes for advection-diffusion, Journal of Computational and Applied Mathematics 215 (2008) 547–556.

[29] B. van Leer, S. Nomura, Discontinuous Galerkin for diffusion, in: 17th AIAA Computational Fluid Dynamics Conference, AIAA Paper 2005-5108, Toronto, 2005.

[30] H. Luo, L. Luo, R. Nourgaliev, V. A. Mousseau, A reconstructed discontinuous Galerkin method for the compressible Navier-Stokes equations on arbitrary grids, in: 47th AIAA Aerospace Sciences Meeting, AIAA Paper 2010-364, Orlando, 2010.

[31] C. Cattaneo, A form of heat-conduction equations which eliminates the paradox of instantaneous propagation, Ct. R. Acad. Sci., Paris 247 (1958) 431–433.

[32] P. L. Roe, M. Arora, Characteristic-based schemes for dispersive waves I. the method of characteristics for smooth solutions, Numerical Methods for Partial Differential Equations 9 (1993) 459–505.

[33] H. Nishikawa, A first-order system approach for diffusion equation. I: Second-order residual-distribution schemes, Journal of Computational Physics 227 (2007) 315–352.

[34] F. Bassi, S. Rebay, A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier-stokes equations, Journal of Computational Physics 131 (1997) 267–279.

[35] D. Aregba-Driollet, R. Natalini, S. Tang, Explicit diffusive kinetic schemes for nonlinear degenerate parabolic systems, Mathematics of Computation 73 (245) (2003) 63–94.

[36] M. Torrihon, K. Xu, Stability and consistency of kinetic upwinding for advection.diffusion equations, IMA Journal of Numerical Analysis 26 (2006) 686–722.

[37] H. Nishikawa, A first-order system approach for diffusion equation. II: Unification of advection and diffusion, Journal of Computational Physics 229 (2010) 3989–4016.

[38] R. J. LeVeque, Finite Volume Methods for Hyperbolic Problems, Cambridge University Press, 2002.

[39] J. C. Tannehill, D. A. Anderson, R. H. Pletcher, Computational Fluid Mechanics and Heat Transfer, 2nd Edition, Taylor & Francis, 1997.

[40] E. Godlewski, P.-A. Raviart, Numerical Approximation of Hyperbolic Systems of Conservation Laws, Springer, 1996.

[41] C. Hirsch, Numerical Computation of Internal and External Flows, Vol. 2, A Wiley - Interscience Publications, 1990.

[42] B. van Leer, Towards the ultimate conservative difference scheme. V. a second-order sequel to Godunov's method, Journal of Computational Physics 32 (1979) 101–136.

[43] H. Nishikawa, Beyond interface gradient: A general principle for constructing diffusion schemes, in: 40th AIAA Fluid Dynamics Conference and Exhibit, AIAA Paper 2010-5093, Chicago, 2010.

[44] C. E. Baumann, An hp-adaptive discontinuous Finite Element Method for Computational Fluid Dynamics,

Ph.D. thesis, University of Texas at Austin (1997).
[45] J. T. Oden, I. Babušuka, C. E. Baumann, A discontinuous hp finite element method for diffusion problems, Journal of Computational Physics 146 (1998) 491–519.
[46] K. Shahbazi, An explicit expression for the penalty parameter of the interior penalty method, Journal of Computational Physics 205 (2005) 401–407.
[47] R. Hartmann, P. Houston, An optimal order interior penalty discontinuous Galerkin discretization of the compressible Navier-Stokes equations, Journal of Computational Physics 227 (2008) 9670–9685.
[48] M. Zhang, C.-W. Shu, An analysis of three different formulations of the discontinuous Galerkin method for diffusion equations, Mathematical Models and Methods in Applied Sciences 13 (2003) 395–413.
[49] M. H. van Raalte, Multigrid analysis and embedded boundary conditions for discontinuous Galerkin discretization, Ph.D. thesis, University of Amsterdam (2004).
[50] Z. J. Wang, Spectral (finite) volume method for conservation laws on unstructured grids, Journal of Computational Physics 178 (2002) 210–251.
[51] M. Aftosmis, D. Gaitonde, T. S. Tavares, Behavior of linear reconstruction techniques on unstructured meshes, AIAA Journal 33 (11) (1995) 2038–2049.
[52] H. Luo, J. D. Baum, R. Löhner, An improved finite volume scheme for compressible flows on unstructured grids, in: 33rd Aerospace Sciences Meeting and Exhibit, AIAA Paper 95-0348, 1995.
[53] A. Haselbacher, J. Blazek, Accurate and efficient discretization of Navier-Stokes equations on mixed grids, AIAA Journal 38 (11) (2000) 2094–2102.
[54] J. L. Thomas, Private Communication (2009).
[55] J. L. Thomas, B. Diskin, H. Nishikawa, A critical study of agglomerated multigrid methods for diffusion on highly-stretched grids, Computers and Fluids 41 (1) (2011) 82–93.
[56] M. E. Braaten, S. D. Connel, Three-dimensional unstructured adaptive multigrid scheme for the Navier-Stokes equations, AIAA Journal 34 (2) (1996) 281–290.
[57] H. Deconinck, R. Abgrall, Introduction to residual distribution methods, in: 34th VKI CFD Lecture Series Very-High Order Discretization Methods, VKI Lecture Series, 2005.
[58] R. Abgrall, M. Mezine, Construction of second order accurate monotone and stable residual distribution schemes for unsteady flow problems, Journal of Computational Physics 188 (2002) 16–55.
[59] M. Ricchiuto, R. Abgrall, H. Deconinck, Construction of very high order residual distribution schemes for unsteady scalar advection: Preliminary results, in: 33rd Computational Fluid Dynamics-Novel Methods for Solving Convection Dominated Systems, VKI Lecture Series, 2003.
[60] M. Ricchiuto, R. Abgrall, Explicit Runge-Kutta residual distribution schemes for time dependent problems: Second order case, Journal of Computational Physics 229 (2010) 5653–5691.
[61] H. Nishikawa, B. Diskin, J. L. Thomas, Critical study of agglomerated multigrid methods for diffusion, AIAA Journal 48 (4) (2010) 839–847.
[62] D. J. Mavriplis, Multigrid techniques for unstructured meshes, in: VKI Lecture Series VKI-LS 1995-02, Von Karman Institute for Fluid Dynamics, Rhode-Saint-Genese, Belgium, 1995.
[63] H. Nishikawa, B. Diskin, J. L. Thomas, Development and application of agglomerated multigrid methods for complex geometries, in: 40th AIAA Fluid Dynamics Conference and Exhibit, AIAA Paper 2010-4731, Chicago, 2010.
[64] B. Cockburn, C. W. Shu, The Runge-Kutta discontinuous Galerkin method for conservation laws: V. multidimensional systems, Journal of Computational Physics 141 (1998) 199–224.
[65] H. L. Atkins, C.-W. Shu, Analysis of the discontinuous Galerkin method applied to the diffusion operator, in: 14th AIAA Computational Fluid Dynamics Conference, AIAA Paper 99-3306, Norfolk, 1999.
[66] Z. J. Wang, Y. Liu, Spectral (finite) volume method for conservation laws on unstructured grids: II. extension to two-dimensional scalar equation, Journal of Computational Physics 179 (2002) 665–697.
[67] O. C. Zienkiewicz, R. L. Taylor, The Finite Element Method, Volume 1, McGraw-Hill Company, 1994.
[68] T. J. Barth, Numerical aspects of computing viscous high reynolds number flows on unstructured meshes, AIAA Paper 91-0721, 1991.