Chapter 9
Emerging Numerical Methods for Atmospheric Modeling

Ramachandran D. Nair, Michael N. Levy and Peter H. Lauritzen

Abstract This chapter discusses the development of discontinuous Galerkin (DG) schemes for the hyperbolic conservation laws relevant to atmospheric modeling. Two variants of the DG spatial discretization, the modal and nodal form, are considered for the one- and two-dimensional cases. The time integration relies on a second- or third-order explicit strong stability-preserving Runge-Kutta method. Several computational examples are provided, including a solid-body rotation test, a deformational flow problem and solving the barotropic vorticity equation for an idealized cyclone. A detailed description of various limiters available for the DG method is given, and a new limiter with positivity-preservation as a constraint is proposed for two-dimensional transport. The DG method is extended to the cubed-sphere geometry and the transport and shallow water models are discussed.

9.1 Introduction

Atmospheric numerical modeling has undergone radical changes over the past decade. One major reason for this trend is the recent paradigm change in scientific computing, triggered by the arrival of petascale computing resources with core counts in the range of tens to hundreds of thousands. Due to these changes, modelers must develop or adapt grid systems and numerical algorithms which facilitate an unprecedented level of scalability on these modern highly parallel computer archi-

R. D. Nair
National Center for Atmospheric Research, 1850 Table Mesa Drive, Boulder, CO 80305, USA. e-mail: rnair@ucar.edu

M. N. Levy
Sandia National Laboratories, Albuquerque, NM 87185, USA. e-mail: mnlevy@sandia.gov

P. H. Lauritzen
National Center for Atmospheric Research, 1850 Table Mesa Drive, Boulder, CO 80305, USA. e-mail: pel@ucar.edu

tectures. The numerical algorithms which can address these challenges should have *local* properties such as a high on-processor operation (floating-point operation or flop) count and a minimal parallel communication footprint.

With the increased anount of computing resources available to modelers, it is now possible to develop global models with resolution of the order of a few kilometers. This capability bridges the gap between traditional weather and climate modeling efforts, which operate on different spatial and temporal scales, and sets the stage for the development of a unified weather and climate model. However, this opens up another challenge – switching the governing equations from the hydrostatic to non-hydrostatic regime. The equation set generally used in traditional global climate models (hydrostatic equations of motion) is not adequate at the non-hydrostatic scale. In the very high-resolution regime viable options for the governing equations of motion are the compressible (or quasi-compressible) Euler equations or Navier-Stokes equations. Also, it is highly desirable that the underlying model equations follow the physical laws of conservation for integral invariants such as mass, energy, enstrophy, etc. In order to comply with these constraints and address new computational challenges, the next generation of atmospheric models should be based on robust numerical methods which satisfy the following set of criteria:

- inherent local and global conservation
- high-order accuracy
- computational efficiency
- geometric flexibility (any type of grid system, suitable for adaptive mesh refinement)
- non-oscillatory advection (monotonic, positivity preservation)
- high parallel efficiency (local method, petascale capability).

There are several successful numerical methods, particularly in the finite-volume (FV) literature, which satisfy most of the above-mentioned properties. The FV schemes are inherently conservative but mostly low-order accurate (third-order or less). High-order extensions of the FV method are possible at the cost of wider halo regions. For example, the weighed essentially non-oscilatory (WENO) method (Shu, 1997) is a powerful approach; however, a $(k+1)$th-order accurate WENO scheme in 1D requires $2k+1$ cells (control volumes). Thus, as the order of accuracy grows the WENO scheme requires a wider computational stencil (halo region) which can seriously impede the parallel efficiency. A local method like the spectral element (SE) method has the local domain decomposition property of the finite-element (FE) method combined with high-order accuracy and the weak numerical dispersion and low numerical dissipation of spectral methods. The SE method offers excellent parallel efficiency and has become the method of choice for many practical applications. The classical SE method is not necessarily based on hyperbolic conservation laws and is not inherently conservative. Nevertheless, the conservation properties can be engineered in the SE discretization (chapter 12) much as they were in the conservative finite-difference discretization developed by Arakawa and Lamb (1977) and Simmons and Burridge (1981).

The discontinuous Galerkin (DG) method retains all the nice properties of the SE method, plus it is inherently conservative. The DG method has the potential to address all of the above-listed properties. DG algorithms for solving partial differential equations are becoming very popular in a wide range of applications in computational science and engineering. The primary focus of this chapter is on the development of the DG method for atmospheric modeling applications.

The DG method may be viewed as a hybrid approach, combining the ideas of classical FV and FE methods into a unified framework to exploit the merits of both. As a FV method, DG discretizations employ discontinuous elements (local control volumes) and flux integrals along its boundaries, guaranteeing local conservation. Similar to the FV method, DG schemes can incorporate slope limiters for controlling spurious oscillations in the solution. However, in contrast to FV methods, the DG method avoids the reconstruction process (often requiring wider stencil). The FE or SE structure (element-wise Galerkin approach) makes the DG method high-order accurate and provides the ability to handle complex geometries such as the Earth's surface or boundary conditions. However, as opposed to the FE/SE methods, the elements used for the DG methods are discontinuous, which leads to a localized discretization. This feature offers excellent parallel efficiency as well as efficient adaptive mesh refinement (AMR) capability, even with non-conforming elements.

The DG method was first introduced by Reed and Hill (1973) and later analyzed by Lesaint and Raviart (1974) for linear advection equation. A rigorous mathematical foundation for the DG method was laid by Cockburn and Shu (1989) and Cockburn et al (1990), where high-order accurate explicit Runge-Kutta (RK) time integration schemes combined with DG spatial discretizations for nonlinear systems of conservation laws were developed. The resulting RKDG method has become widely popular in different computational science and engineering disciplines (Cockburn et al, 2000; Remacle et al, 2003).

The remainder of the chapter is organized as follows: in Section 9.2 we describe the basic DG discretization in 1D, and the extension to 2D is given in Section 9.3. Section 9.4 describes various limiters for the DG method with examples. An extension of the DG method onto the sphere is given in Section 9.5, where the shallow water model for the cubed-sphere is described. Section 9.6 offers some concluding remarks.

9.2 The DG Method

Although the DG method is applicable to a variety of parabolic and elliptic problems (Rivière, 2008), our primary focus is on the DG method applied to hyperbolic conservation laws which are relevant to atmospheric numerical modeling. Before detailing the DG discretization procedure we briefly review conservation laws.

9.2.1 Conservation Laws

Systems of conservation laws are very important mathematical models for a variety of physical phenomena that appear in fluid mechanics and several other areas including atmospheric sciences. A large class of atmospheric equations of motion for compressible and incompressible flows can be written in conservation form. Conservation laws are systems of nonlinear partial differential equations (PDEs) most readily expressed in flux form and can be written:

$$\frac{\partial}{\partial t}U(\mathbf{x},t)+\sum_{j=1}^{3}\frac{\partial}{\partial x_j}F_j(U,\mathbf{x},t)=S(U), \tag{9.1}$$

where $\mathbf{x}$ is the 3D space coordinate and time $t>0$. $U(\mathbf{x},t)$ is the state vector representing conserved quantities (e.g. mass, momentum or energy). $F_j(U)$ are components of $\mathbf{F}$, a prescribed flux vector which accounts for diffusive and convective effects, and $S(U)$ is the source term representing exterior forces. The system of Euler and Navier-Stokes equations, widely used for modeling fluid motion, can be cast in this form. The mass continuity equation is an example of scalar conservation law and is a special case of (9.1), which is obtained by applying the physical principle of conservation of mass in a fluid flow:

$$\frac{\partial \rho}{\partial t}+\nabla\cdot(\rho\mathbf{V})=0,$$

where ρ is the fluid density, $\mathbf{V}$ is the velocity of the fluid, and '$\nabla\cdot$' denotes the divergence operator. Note that discretizing the equations in flux-form is important because application of the divergence theorem is straightforward and the conservation can be maintained numerically. We consider several hyperbolic conservation laws based on (9.1) in this chapter and numerically solve them by using the DG method.

9.2.2 The DG Method for 1D Problems

The basic ideas of the DG discretization may be understood in a simple 1D framework. In order to introduce the DG discretization and notations, we first consider the one-dimensional scalar conservation law:

$$\frac{\partial U}{\partial t}+\frac{\partial F(U)}{\partial x}=0 \quad \text{in} \quad \Omega\times(0,T], \tag{9.2}$$

where $U=U(x,t)$ is the conservative variable evolving in time with a known initial condition $U(x,t=0)=U_0(x),\forall x\in\Omega$, and $F(U)$ is the flux function. For a linear advection problem the flux function is $F(U)=cU$, where c is the velocity; for

the inviscid Burgers' equation, a simple non-linear problem, the flux function is $F(U) = U^2/2$.

9.2.3 Galerkin Formulation

The DG discretization consists of partitioning the global domain Ω into N_{elm} non-overlapping elements such that $\Omega = \cup_{j=1}^{N_{elm}} \Omega_j$ with $\Omega_j \equiv [x_{j-1/2}, x_{j+1/2}]$, $j = 1, \ldots, N_{elm}$. With this setup the width of the j^{th} element is $\Delta x_j = x_{j+1/2} - x_{j-1/2}$ and the midpoint is defined by $x_j = (x_{j+1/2} + x_{j-1/2})/2$. Note that the edges (interface) $x_{j\pm1/2}$ of the element Ω_j are shared by the adjacent elements in this partition, as shown schematically in Fig. 9.1.

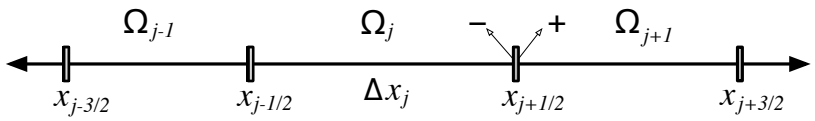

Fig. 9.1 Partition of the 1D domain Ω into non-overlapping elements $\Omega_j = [x_{j+1/2}, x_{j-1/2}]$, with element width Δx_j and edges $x_{j\pm1/2}$. The signs $(-)$ and $(+)$ indicate the left and right limits of the edge point (interface) $x_{j+1/2}$, respectively. The global solution is discontinuous at these points.

The next step is to cast the problem (9.2) into the *weak* Galerkin formulation. This is done by multiplying (9.2) by a test (weight) function $\varphi(x)$ and integrating over the element Ω_j:

$$\int_{\Omega_j} \left[\frac{\partial U}{\partial t} + \frac{\partial F(U)}{\partial x} \right] \varphi(x) dx = 0. \tag{9.3}$$

The term *weak* refers to the fact that the formulation (9.3) admits a larger class of solutions as opposed to the *strong* or classical form (9.2). Integrating the second term of (9.3) by parts (Green's method) yields

$$\int_{\Omega_j} \frac{\partial U(x,t)}{\partial t} \varphi(x) dx - \int_{\Omega_j} F(U(x,t)) \frac{\partial \varphi(x)}{\partial x} dx + \left[F(U(x,t)) \varphi(x) \right]_{x^+_{j-1/2}}^{x^-_{j+1/2}} = 0, \tag{9.4}$$

where $x^-_{j+1/2}$ is the left limit at the edge $x_{j+1/2}$, and $x^+_{j-1/2}$ is the right limit at the edge $x_{j-1/2}$ of the element Ω_j, as indicated in Fig. 9.1. While the Galerkin formulation procedure (9.4) is the same for each element Ω_j, special attention must be paid to the evaluation of fluxes at the edges because this flux is the only *connection* between the elements.

Each element Ω_j has its own approximate local solution, allowing the global solution on Ω to be discontinuous at the element interfaces $x_{j\pm1/2}$. This leads to two different values for the flux functions at each interface $x_{j+1/2}$: $F(U(x^-_{j+1/2},t))$ on the left and $F(U(x^+_{j+1/2},t))$ on the right. This discontinuity at the element edges must be addressed by employing a numerical flux (or approximate Riemann solver) $\hat{F}(U^-,U^+) = \hat{F}[U(x^-_{j+1/2},t),U(x^+_{j+1/2},t)]$, which provides the crucial coupling between the elements. Figure 9.2 describes schematically the discontinuity of the flux function at the element interface $x_{j+1/2}$.

The upwind based numerical fluxes used for DG applications are in fact identical to those developed for the finite-volume methods. A variety of numerical flux formulae are available with varying complexity, however, the Lax-Friedrichs (LF) numerical flux is cost-effective and widely used for many applications (Qiu et al, 2006). The LF flux formula is defined as follows:

$$\hat{F}(U^-,U^+) = \frac{1}{2}\left[F(U^-)+F(U^+)-\alpha_{\max}(U^+-U^-)\right] \tag{9.5}$$

where $\alpha_{\max}$ is the upper bound of $|F'(U)|$, the flux Jacobian, over the entire domain Ω (for scalar problems). If $\alpha_{\max}$ is evaluated only at the local element edges then (9.5) is known as the local Lax-Friedrichs or Russanov flux. For a linear advection problem $\alpha_{\max} = |c|$ and for the inviscid Burgers' equation $\alpha_{\max} = \max(|U^-|,|U^+|)$.

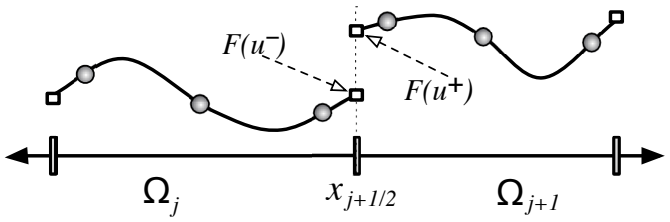

Fig. 9.2 Schematic diagram illustrating the discontinuity of the solution $U(x,t)$ and the flux function $F(U)$ at the element interface (edge) $x_{j+1/2}$. Filled circles on the smooth curves are the element-wise solution points and the open squares at the edges are the flux points. At the interface the flux function has two contributions, one from the left $F(U^-)$, and one from the right $F(U^+)$. The discontinuity of $F(U)$ at the interfaces is resolved by employing a numerical flux formula.

9.2.4 Space Discretization

In order to solve the weak Galerkin formulation (9.4) , we assume that the approximate (numerical) solution $U_h \approx U(x,t)$ and the corresponding test function φ_h are polynomial functions belonging to a finite-dimensional space V_h. This space may be

formally defined as $V_h = \{p : p\,|_{\Omega_j} \in \mathbb{P}_N(\Omega_j)\}$ where $\mathbb{P}_N$ is the space of polynomials in Ω_j with degree $\leq N$.

For the approximate solution $U_h(x,t)$, the DG spatial discretization based on the weak formulation (9.4) combined with (9.5) can now be written as follows:

$$\int_{\Omega_j} \frac{\partial U_h(x,t)}{\partial t}\,\varphi_h(x)dx = \int_{\Omega_j} F(U_h(x,t))\,\frac{\partial \varphi_h(x)}{\partial x}\,dx - \left[\hat{F}(U_h^-,U_h^+)_{j+1/2}(t)\,\varphi_h(x_{j+1/2}^-) - \hat{F}(U_h^-,U_h^+)_{j-1/2}(t)\,\varphi_h(x_{j-1/2}^+)\right], \quad (9.6)$$

where $U_h, \varphi_h \in V_h$ for all Ω_j; $j = 1,\ldots,N_{elm}$. This completes the DG formulation of problem (9.2).

In order to solve (9.6) accurately and efficiently, we need to make some judicious choices for the integrals and polynomial functions employed in (9.6). The integrals can be accurately computed using the high-order Gaussian quadrature rules. Moreover, choosing orthogonal polynomials as a basis for U_h and φ_h in (9.6) significantly enhances computational efficiency. This is because the coefficients of the time derivative in (9.6) reduce to a diagonal matrix when U_h and φ_h are orthogonal polynomials. The orthogonal basis set which spans V_h may be based on either *modal* or *nodal* expansions. We consider these two cases separately in the following sections.

9.2.4.1 Modal Formulation

The modal basis set consists of orthogonal polynomials of degree k monotonically increasing from 0 to N, and each basis function represents the moment of order k (or, equivalently, each order contributes an extra moment in the expansion (Karniadakis and Sherwin, 2005)). The Legendre polynomials $P_k(\xi), k = 0,1,\ldots,N$, $\xi \in [-1,1]$ provide an excellent choice for the orthogonal basis function in V_h. A major advantage of this choice is that the computations in (9.6) can be significantly simplified by exploiting the properties of Legendre polynomials. The first few Legendre polynomials are tabulated in Table 9.1.

Table 9.1 Legendre polynomials $P_k(\xi)$ of degree up to $k = 4$

Degree (k)	0	1	2	3	4
$P_k(\xi)$	1	ξ	$(3\xi^2-1)/2$	$\xi(5\xi^2-3)/2$	$(35\xi^4-30\xi^2+3)/8$

Higher degree $P_k(\xi)$ can be generated by the following recurrence relation:

$$P_k(\xi) = \left[\frac{2k-1}{k}\right]\xi\,P_{k-1}(\xi) - \left[\frac{k-1}{k}\right]P_{k-2}(\xi), \quad k = 2,3,4,\cdots. \quad (9.7)$$

At the edges of the interval $[-1,1]$, $P_k(-1) = (-1)^k$ and $P_k(1) = 1$, for any $k \geq 0$. In Fig. 9.3 the left panel shows the Legendre polynomials of degree up to $k = 4$. The orthogonality of $P_k(\xi)$ implies that

$$\int_{-1}^{1} P_k(\xi) P_\ell(\xi) d\xi = \frac{2}{2k+1} \delta_{k\ell}, \quad \xi \in [-1,1], \tag{9.8}$$

where $\delta_{k\ell}$ is the Kronecker delta function ($\delta_{k\ell} = 1$ if $k = \ell$, and $\delta_{k\ell} = 0$ if $k \neq \ell$).

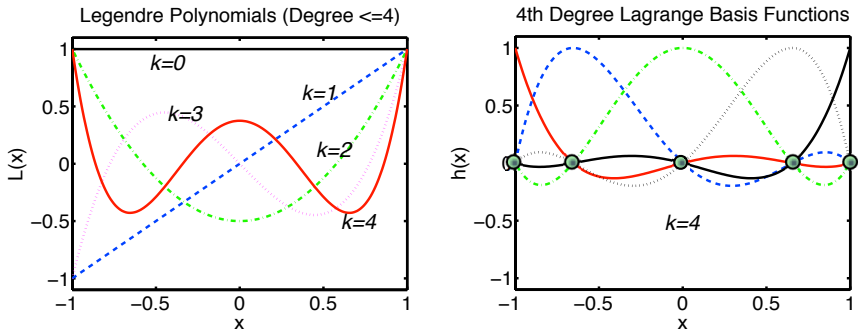

Fig. 9.3 The left panel shows Lagendre polynomials of degree from $k = 0$ to 4, which can be used as basis functions for the modal DG method. The right panel shows Lagrange-Legendre polynomials of fixed degree $k = N = 4$, whose zeros are at the Gauss-Lobatto-Legendre (GLL) quadrature points. The nodal version of DG employs GLL quadrature points, which are in the interval $[-1,1]$ and marked as filled circles.

To adopt an orthogonal basis set $\{P_k(\xi)\}_{k=0}^N$ for the DG discretization (9.6), we first need to introduce a mapping between x on each element Ω_j and the local variable $\xi \in [-1,1]$. Irrespective of the physical length Δx_j, each element Ω_j can be mapped onto a unique reference (or standard) element $Q \equiv [-1,1]$ such that

$$\xi = \frac{2(x - x_j)}{\Delta x_j}. \tag{9.9}$$

Figure (9.4) illustrates schematically the mapping between each Ω_j and the reference element Q. In terms of the new local variable $\xi = \xi(x)$, we denote the approximate solution in any element Ω_j by $U_j = U_j(\xi,t)$ and it can be expressed as

$$U_j(\xi,t) = \sum_{k=0}^{N} U_j^k(t) P_k(\xi) \quad \text{for} \quad \xi \in [-1,1], \tag{9.10}$$

where the expansion coefficients, $U_j^k(t)$, are the *moments* or degrees of freedom (*dof*) evolving in time. The explicit form of $U_j^k(t)$ is derived using (9.8) and given

by

$$U_j^k(t) = \frac{2k+1}{2}\int_{-1}^{1} U_j(\xi,t)\,P_k(\xi)\,d\xi, \quad \text{where} \quad k = 0,1,\ldots,N. \tag{9.11}$$

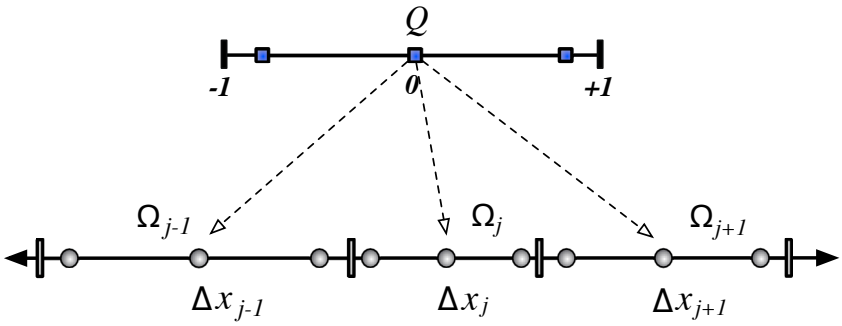

Fig. 9.4 A schematic diagram of the mapping between the unique reference element $Q = [-1,1]$ and each element Ω_j in the physical domain Ω. The filled squares on Q indicate the Gaussian quadrature points in the interval $[-1,1]$ and the filled circles are the corresponding quadrature points on the elements. All the integral and differential operations required for DG discretization are computed on Q.

Note that (9.11) may be interpreted as a transformation (or a projection operation) from the physical space to the spectral (Legendre) space with inverse transformation (9.10). It is clear from (9.11) that the zeroth moment,

$$U_j^0(t) = \overline{U}_j = \frac{1}{2}\int_{-1}^{1} U_j(\xi,t)\,d\xi, \tag{9.12}$$

is the average value $\overline{U}_j$. Similarly the first, second, and higher moments are responsible for the linear, quadratic, and higher-order variations of $U(\xi)$ in the element. The left panel in Fig. (9.3) shows the Legendre polynomials of degree up to $N = 4$; each polynomial corresponds to the k^{th} moment in the modal formulation.

We can simplify (9.6) by substituting $U_j(\xi,t)$ for $U_h(x,t)$ and $P_k(\xi)$ for $\varphi_h(x)$, however, this requires a change of variable from x to ξ in (9.6) with the new domain of integration $[-1,1]$. By using the summation (9.10) and the following relations from (9.9)

$$dx = \frac{\Delta x_j}{2}d\xi, \; \frac{\partial}{\partial x} = \frac{2}{\Delta x_j}\frac{\partial}{\partial \xi},$$

the weak Galerkin form (9.6) can be written in the semi-discrete form as given below:

$$\frac{\Delta x_j}{2}\sum_{\ell=0}^{N}\frac{d}{dt}U_j^{\ell}(t)\int_{-1}^{1}P_k(\xi)P_\ell(\xi)d\xi \;=\int_{-1}^{1}F(U_j(\xi,t))\,P_k'(\xi)\,d\xi - \left[\hat{F}_{j+1/2}(t)\,P_k(1)-\hat{F}_{j-1/2}(t)\,P_k(-1)\right] \quad (9.13)$$

where $P_k'(\xi)$ is the derivative of the Legendre polynomials (9.7). The above equation can be further simplified by employing the orthogonality relation (9.8) and the property $P_k(\pm 1)=(\pm 1)^k$ as follows:

$$\frac{1}{2k+1}\frac{d}{dt}U_j^k(t) = \frac{1}{\Delta x_j}\int_{-1}^{1}F(U_j(\xi,t))\,P_k'(\xi)\,d\xi - \frac{1}{\Delta x_j}\left[\hat{F}_{j+1/2}(t)-\hat{F}_{j-1/2}(t)(-1)^k\right], \quad (9.14)$$

where $k=0,1,\ldots,N$.

The integral appearing in (9.14) is evaluated using a high-order ($(N+1)$-node) Gaussian quadrature rule. Usually a Gauss-Legendre (GL) quadrature, which is exact for polynomials of degree $2N+1$, or a Gauss-Lobatto-Legendre (GLL) quadrature, which is exact for polynomials of degree $2N-1$, is employed; the choice of a specific quadrature is somewhat application dependent. For a given number of quadrature points, the GL quadrature is more accurate than the GLL quadrature but the former does not place nodes at the end points of the interval $[-1,1]$ (see the marked points on the reference element Q in Fig. (9.4)). We further discuss the relative merits of GL and GLL quadrature rules in a 2D context in Section 9.3.1.3.

In order to compute $\hat{F}_{j\pm 1/2}$ in (9.14), the flux $F(U(\xi))$ at the element edges $\xi=\pm 1$ must be known. In the GL case, this means that one must interpolate the solution $U(\xi)$ using (9.10). However, the GLL quadrature includes the edges where values of $U(\xi)$ are readily available, and makes the edge flux computation easy.

Regardless of the choice of quadrature, the DG solution procedure for the conservation law (9.2) on an element Ω_j reduces to solving a system of decoupled ordinary differential equations (ODEs) (9.14), which may be written in the following form.

$$\mathbf{M}_j\frac{d}{dt}\mathbf{U}_j=\mathbf{R}(\mathbf{U}_j), \quad (9.15)$$

where $\mathbf{M}_j$ is the coefficient matrix associated with the time derivative in (9.14) and formally referred to as the *mass matrix*, $\mathbf{U}_j$ is a column vector containing the moments $U^k(t), k=0,1,\ldots,N$, and $\mathbf{R}$ is the residual vector corresponding to the right-hand side of (9.14). By virtue of the orthogonality of the Legendre polynomials, the mass matrix $\mathbf{M}_j$ is strictly a diagonal matrix with non-zero entries $\{1/(2k+1)\}_{k=0}^{N}$.

This diagonal structure has great computational advantage because $\mathbf{M}_j$ can be inverted trivially and simplifies the solution process in (9.15). For the DG discretization considered here each element Ω_j relies on the same polynomial bases, therefore the mass matrix $\mathbf{M}_j=\mathbf{M}$ is identical for each element in the domain Ω. Premultiplying (9.15) by $\mathbf{M}^{-1}$ for each element results in the following system of ODEs corresponding to the problem (9.2),

$$\frac{d}{dt}\mathbf{U} = \mathbf{L}(\mathbf{U}) \quad \text{in} \quad (0,T], \tag{9.16}$$

where $\mathbf{U}$ is the global vector of degrees of freedom which evolves in time, $\mathbf{L}$ is a generic operator combining all the spatial discretizations. A DG method employing $N+1$ moments (or with polynomial bases up to degree N) is often referred to as a P^N method (Cockburn and Shu, 2001). We will consider the time discretization procedure for (9.16) in the following section.

In order to see the close link between DG and FV approaches, we consider the first few moments $k = 0,1,2$ in (9.14) as follows:

$$\frac{d}{dt}U_j^0(t) = \frac{1}{\Delta x_j}\left[\hat{F}_{j+1/2}(t) - \hat{F}_{j-1/2}(t)\right], \tag{9.17}$$

$$\frac{1}{3}\frac{d}{dt}U_j^1(t) = \frac{1}{\Delta x_j}\int_{-1}^{1} F(U_j(\xi,t))d\xi - \frac{1}{\Delta x_j}\left[\hat{F}_{j+1/2}(t) + \hat{F}_{j-1/2}(t)\right], \tag{9.18}$$

$$\frac{1}{5}\frac{d}{dt}U_j^2(t) = \frac{6}{\Delta x_j}\int_{-1}^{1} F(U_j(\xi,t))\xi d\xi - \frac{1}{\Delta x_j}\left[\hat{F}_{j+1/2}(t) - \hat{F}_{j-1/2}(t)\right]. \tag{9.19}$$

The mass matrix associated with the above system is $\mathbf{M} = \mathrm{diag}\,[1,1/2,1/5]$, and the moments $U_j^k(t)$ can be used for constructing the solution at a known time $t = t_n$ via (9.10) such that

$$U_j(\xi,t_n) = U_j^0(t_n)P_0(\xi) + U_j^1(t_n)P_1(\xi) + U_j^2(t_n)P_2(\xi). \tag{9.20}$$

For the simplest DG formulation, the P^0 case, (9.17) is the only equation to solve in time. In this case $U_j^0(t)$, the moment (*dof*) evolving in time, is nothing more than the cell-average $\overline{U}_j$ given in (9.12), which is an element-wise (or piecewise) constant. Thus the DG P^0 case reduces to the classical piecewise constant Godunov FV method (Toro 1999, chapter 8). In a similar manner one can show the DG P^1 and P^2 methods are related to the piecewise linear method (PLM, van Leer (1974)) and the piecewise parabolic method (PPM, Colella and Woodward (1984)), respectively.

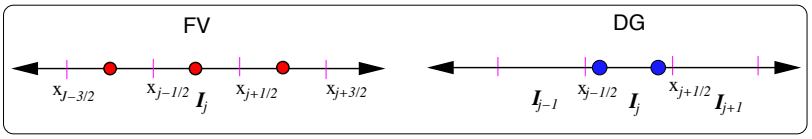

Fig. 9.5 A schematic showing a comparison between the classical 1D finite-volume (FV) and DG methods. $I_j = [x_{j-1/2}, x_{j+1/2}]$ may be interpreted as a cell in the FV method or an element in the DG (P^1) method. For the FV method, the cell-average (shown as filled circles in the left panel) is the only degree of freedom per cell evolving in time. The DG method has more degrees of freedom (marked as filled circles in the right panel) per element evolving in time, however both methods employ the same procedure to address the discontinuities at the cell boundaries $x_{j\pm1/2}$.

Nevertheless, there are subtle differences between regular FV and DG methods. In FV methods such as PLM or PPM there is only one *dof* per control volume evolving in time, namely $\overline{U}_j$, irrespective of the spatial order of accuracy of the method or the dimension of the problem. On the other hand, the DG method carries more *dofs* per element (the cell or the control volume in an FV sense) and the number of *dofs* grows with both the order of accuracy and the dimension (see Fig. 9.5). In other words, a DG method packs more information into each cell than the FV method. For example, in (9.20) three moments are required to construct the solution $U_j(\xi)$ with a P^2 method, and the moments depend only on the element Ω_j resulting in a compact computational stencil. The PPM method requires the reconstruction of parabolas of the form (9.20) by utilizing the averages $\overline{U}_j$ from the neighboring cells, resulting in a wider stencil. In both methods $U_j(\xi)$ essentially represents the sub-grid scale distribution of the solution – even though the underlying discretizations are different. However, as compared to PPM, the high accuracy and compactness of the DG P^2 method comes with additional computational cost. The DG method presented here may be viewed as a high-order compact FV method. A P^2 transport scheme is also similar to the multi-moment transport schemes developed by Prather (1986).

9.2.4.2 Nodal Formulation

The nodal expansion is based on Lagrange polynomials with roots at a set of *nodal* points, which may include the edge points. The nodal bases are widely popular in high-order spectral element methods (Karniadakis and Sherwin, 2005). An important aspect of the DG discretization is the choice of an efficient basis set (polynomials) that span V_h. Because of the inherent computational advantages associated with nodal bases, they are adopted in DG discretization for many applications (Hesthaven and Warburton, 2008). The nodal DG scheme is potentially more computationally efficient because it relies on solutions in physical (grid point) space, obviating the need to transform between spectral and physical space, which is required for the modal DG scheme (9.14).

The nodal basis set is constructed using the Lagrange polynomials $h_k(\xi)$, $\xi \in [-1,1]$, with roots at the Gauss quadrature points. The nodal points may be based on the Gauss-Legendre (GL) or the Gauss-Lobatto-Legendre (GLL) quadrature rule. However, we consistently employ the GLL quadrature for the nodal formulation considered herein. The $N+1$ GLL points $\{\xi_l\}_{l=0}^{N}$ (i.e., the nodal points including the edge point ± 1), can be generated from the relation $(1-\xi^2)P'_N(\xi) = 0$, where $P_N(\xi)$ is the Legendre polynomial of degree N. The basis functions are defined by

$$h_k(\xi) = \frac{(\xi-1)(\xi+1)P'_N(\xi)}{N(N+1)P_N(\xi_k)(\xi-\xi_k)}, \tag{9.21}$$

where $P_N(\xi)$ is the Legendre polynomial of degree N. In Fig. (9.3) the right panel shows the 4^{th}-degree nodal bases $h_k(\xi)$, and $N+1=5$ GLL points are marked as filled circles. Since $h_k(\xi)$ is a Lagrange polynomial, the following property holds at

the nodes ξ_l:

$$h_k(\xi_l) = \delta_{kl} = \begin{cases} 1 \text{ if } k = l, \\ 0 \text{ if } k \neq l. \end{cases} \tag{9.22}$$

The discrete orthogonality of $h_\ell(\xi)$ can be established through the GLL quadrature rule, given by

$$\int_{-1}^{1} f(\xi) d\xi \approx \sum_{k=0}^{N} f(\xi_k) w_k, \tag{9.23}$$

where $f(\xi)$ is an arbitrary function with known values at the nodes (quadrature points) and w_k are the weights associated with the GLL quadrature rule, defined to be

$$w_k = \frac{2}{N(N+1)\,[P_N(\xi_k)]^2}.$$

As mentioned earlier, the GLL quadrature rule (9.23) is *exact* for polynomials of degree up to $2N-1$. The discrete orthogonality of the basis function $h_\ell(\xi)$ can be derived using (9.22) and (9.23) as follows:

$$\int_{-1}^{1} h_k(\xi)\, h_l(\xi) d\xi \approx \sum_{\ell=0}^{N} h_k(\xi_\ell)\, h_l(\xi_\ell)\, w_\ell = w_k\, \delta_{kl}. \tag{9.24}$$

Note that the integrand $h_k(\xi)\, h_l(\xi)$ is a polynomial of degree $2N$, so the orthogonality does not strictly hold under exact integration. In other words, the orthogonality of the nodal expansion given in (9.24) is not as rigorous as the continuous orthogonality employed in the modal case (9.8). Fortunately, the error incurred in discrete orthogonality is of the same order as the nodal expansion so the discretization is consistent. Moreover, it is shown in Canuto et al (2007) that the discrete norm is uniformly equivalent to the continuous norm.

In the nodal expansion, the approximate solution $U_j(\xi,t)$ for an element Ω_j can be written in terms of $h_k(\xi)$ as given below:

$$U_j(\xi,t) = \sum_{k=0}^{N} U_{j,k}(t)\, h_k(\xi), \quad \xi \in [-1,1], \tag{9.25}$$

where $U_{j,k}(t) = U_j(\xi_k,t)$ are the known values of $U_j(\xi,t)$ at the GLL grid points. Also, from (9.25) it is evident that the approximate solution is expressed as a Lagrange interpolation polynomial. Analogous to the modal case, the weak Galerkin formulation (9.6) can be simplified as follows: substitute (9.25) for the approximate solution and $h_k(\xi)$ for the test function, employing the properties (9.24) and $h_k(\pm 1) = 1$. This yields the equation

$$\frac{w_k}{2} \frac{d}{dt} U_{j,k}(t) = \frac{1}{\Delta x_j} \int_{-1}^{1} F(U_j(\xi,t))\, h_k'(\xi)\, d\xi - \frac{1}{\Delta x_j} \left[\hat{F}_{j+1/2}(t) - \hat{F}_{j-1/2}(t) \right], \tag{9.26}$$

where $k = 0, 1, \ldots, N$. The right-hand side involves the derivative of the Lagrange polynomial $h_k'(\xi)$, which needs to be calculated and stored at each of the quadrature

points in order to evaluate the integral in (9.26). The resulting matrix, known as the differentiation matrix, has the following explicit form (Karniadakis and Sherwin, 2005; Canuto et al, 2007):

$$h_k'(\xi_l) = \begin{cases} \frac{L_N(\xi_k)}{L_N(\xi_l)} \frac{1}{(\xi_k - \xi_l)} & \text{if } k \neq l, \\ -\frac{(N+1)N}{4} & \text{if } k = l = 0, \\ \frac{(N+1)N}{4} & \text{if } k = l = N, \\ 0 & \text{otherwise.} \end{cases} \tag{9.27}$$

The mass matrix associated with (9.26) is a diagonal matrix $\mathbf{M}$ with non-zero entries $\{w_k/2\}_{k=0}^{N}$ and, by virtue of the GLL grids, the numerical fluxes $\hat{F}_{j\pm1/2}$ are readily available at the edges $\xi = \pm 1$. The system of ODEs (9.26) can be generalized for the whole domain Ω exactly as in (9.16),

$$\frac{d}{dt}\mathbf{U} = \mathbf{L}(\mathbf{U}) \quad \text{in} \quad (0,T],$$

where $\mathbf{U}$ is the global vector of grid point values $U_{j,k}$, $j = 1,2,\ldots,N_{elm}$ and $k = 0,1,\ldots,N$.

A remarkable difference between the nodal version (9.26) and the corresponding modal version (9.14) of the DG discretization is the absence of the spectral coefficients. In other words, the *dofs* to evolve in time in (9.26) are just the grid point values of the approximate solution $U_{j,k}(t)$, not the spectral coefficients as in the modal case. Hence, there is no need to transform between spectral and physical spaces at every time step, and this feature makes the nodal discretization computationally more efficient (Levy et al, 2007).

9.2.5 Time Integration

The modal and nodal DG discretization both reduce the one-dimensional scalar conservation law to a system of ODEs (9.16) which can be solved using a variety of time integration techniques (chapter 5). In fact, the DG discretization reduces conservation law PDEs to a system of ODEs irrespective of the spatial dimension. Therefore we consider the following general form of the ODE system:

$$\frac{d}{dt}\mathbf{U} = \mathbf{L}(\mathbf{U}) \quad \text{in} \quad (0,T].$$

The most widely used explicit time integration technique for the DG method is based on the Runge-Kutta (RK) scheme; a combination of these space and time discretization approaches is often referred to as the RKDG method (Cockburn and Shu,

2001). For the DG discretization considered in this Chapter we employ the strong stability-preserving (SSP) RK scheme, also known as the total variation diminishing RK scheme (Cockburn et al, 1997); a detailed account of SSP-RK methods is given in Gottlieb et al (2001).

The second-order (two-stage) SSP-RK is

$$\begin{aligned} \mathbf{U}^{(1)} &= \mathbf{U}^n + \Delta t\, \mathbf{L}(\mathbf{U}^n) \\ \mathbf{U}^{n+1} &= \frac{1}{2}\mathbf{U}^n + \frac{1}{2}[\mathbf{U}^{(1)} + \Delta t\, \mathbf{L}(\mathbf{U}^{(1)})], \end{aligned} \tag{9.28}$$

and the third-order (three-stage) SSP-RK is

$$\begin{aligned} \mathbf{U}^{(1)} &= \mathbf{U}^n + \Delta t\, \mathbf{L}(\mathbf{U}^n) \\ \mathbf{U}^{(2)} &= \frac{3}{4}\mathbf{U}^n + \frac{1}{4}[\mathbf{U}^{(1)} + \Delta t\, \mathbf{L}(\mathbf{U}^{(1)})] \\ \mathbf{U}^{n+1} &= \frac{1}{3}\mathbf{U}^n + \frac{2}{3}[\mathbf{U}^{(2)} + \Delta t\, \mathbf{L}(\mathbf{U}^{(2)})]. \end{aligned} \tag{9.29}$$

In both (9.28) and (9.29), $\mathbf{U}^{(1)}$ and $\mathbf{U}^{(2)}$ are intermediate stages of the RK method while the superscripts n and $n+1$ denote time levels t and $t+\Delta t$, respectively. The overall accuracy of the numerical scheme is dictated by the order of accuracy of both the spatial and temporal discretizations. For example, the DG method using polynomials of degree N along with an $N+1$ stage RK method results in an $(N+1)$th-order accurate method (Cockburn and Shu, 2001).

Higher-order RK schemes provide a wider stability region (Butcher, 2008), so a longer time step may be used in the numerical integration. Unfortunately, a high-order RK time discretization has multiple stages of function (right-hand side) evaluations and flux communications, resulting in a computationally expensive scheme (especially in a parallel computing environment). Therefore, many practical applications use a fourth- or lower-order RK scheme (Nair et al, 2005a, 2009).

The linear stability analysis for the modal DG method discussed in Cockburn and Shu (2001) may be used as a guideline for choosing the time steps. For an $(N+1)$-th -order accurate RKDG method, the CFL (Courant-Friedrichs-Lewy) stability limit is given by $c\Delta t/\Delta x \leq 1/(2N+1)$, where Δx is the element width and c is the velocity. This has been proven to be true when $N=1$, however, no theoretical proof exists when $N>1$. For $N \gg 1$ the explicit DG method is very time step restrictive, in such cases a semi-implicit or implicit time integration strategy may be desirable (chapter 5). We also note that when $N>1$, the grid spacing Δx used in calculating the CFL limit should be the *minimum* distance between the non-uniformly distributed quadrature points (see the right panel of Fig. 9.3). A detailed discussion of the CFL stability limit for advection problems for high-order Galerkin methods can be found in Chapter 6 of Karniadakis and Sherwin (2005).

9.2.6 DG 1D Computational Examples

Here we illustrate the DG method by solving two examples of the 1D conservation law (9.2). The first one is a simple linear problem involving the advection of both a Gaussian profile (smooth case) and a rectangular wave (non-smooth case). The second example is the solution of the inviscid Burgers equation, a nonlinear problem. Numerical solutions are computed using the 1D DG schemes discussed earlier. Both the modal and nodal versions of the scheme are used for the simulations. However, the results produced by these schemes are almost identical, and we show only modal or nodal solution for each test.

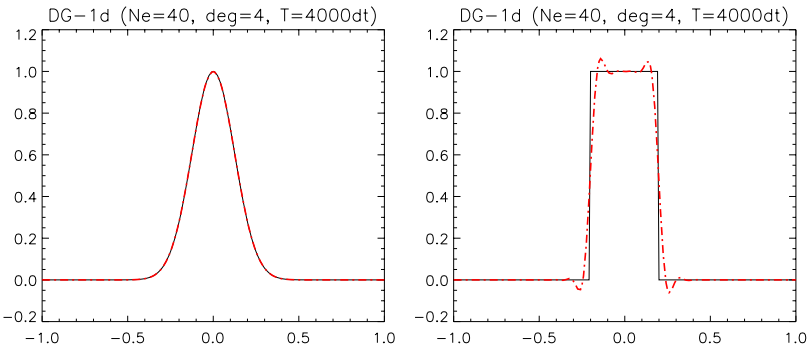

Fig. 9.6 Numerical solution (after 10 revolutions) of the 1D advection problem (9.2) with the high-order nodal DG scheme. The left panel shows the solution for the smooth case, where a Gaussian hill is used as the initial condition. The right panel shows the solution for the non-smooth case, for which a rectangular wave is used as the initial condition. The computational domain $[-1,1]$ consists of 40 elements, each with 5 GLL quadrature points.

For the linear advection problem, the domain is $\Omega = [-1,1]$ with periodic boundary conditions. The initial condition for the smooth problem is $U_0(x) = \exp(-8x^2)$, a Gaussian hill with unit height, and the wind velocity is $c = 1$. In this case the flux function in (9.2) is simply $F(U) = U$. The domain is partitioned into $N_{elm} = 40$ elements, each with $N_v = 5$ GLL quadrature points, and the nodal DG formulation (9.26) is used for the discretization. The resulting time-dependent ODE is solved with the third-order SSP-RK (9.29). 400 time steps are required for a complete revolution along the domain. Figure 9.6 shows the Gaussian hill (left panel, dashed line) after 10 revolutions; the reference solution is also plotted with a solid line but it is visually indistinguishable from the numerical solution.

For the non-smooth advection case the initial condition is a rectangular wave pattern located at the center of the domain with unit height and width of 0.5 units; other than this the boundary conditions and discretization are exactly the same as in the smooth case. The right panel in Fig. 9.6 shows the numerical solution after 10 rev-

olution, and the reference solution (initial condition) is also displayed (solid line). The DG solution suffers from oscillations at the non-smooth edges. The steep gradients at these point produces the Gibbs phenomena, however, the oscillations are confined (or local) to a narrow region even after 10 revolutions. This is a remarkable property of the DG method; other high-order approaches, such as the spectral element method, propagate the noise along the entire domain.

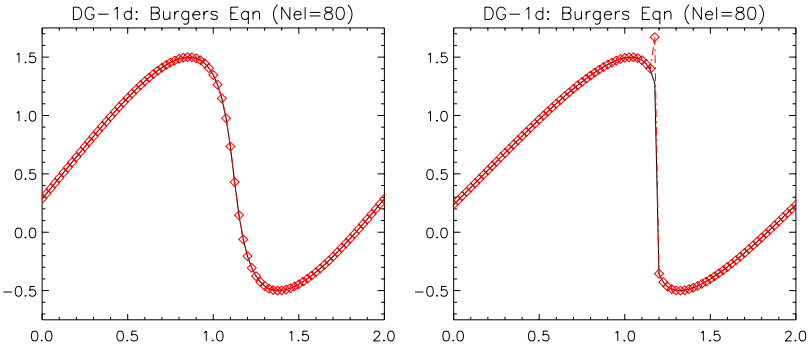

Fig. 9.7 Numerical solution for the inviscid Burgers equation with the modal DG scheme. The solid line indicates the exact solution and diamond points show the DG solution. The domain consists of 80 elements; only one value per element is plotted for clarity . The left panel shows the solution at time $t = 3/(4\pi)$; at this time the solution is still smooth and free from shocks. The right panel shows the solution at time $t = 9/(8\pi)$, at which point shocks have developed.

The inviscid Burgers equation, $U_t + (U^2/2)_x = 0$, is a special case of (9.2) with $F(U) = U^2/2$. The initial condition for this problem is $U_0(x) = 1/2 + \sin(\pi x)$ over a periodic domain $\Omega = [0,2]$. The domain is partitioned into 80 elements, and a modal version DG scheme employing 4 GLL quadrature points is used for the simulations. Time integration is performed with the third-order SSP-RK (9.29), for which a small time step of $\Delta t = 0.0015/\pi$ is used. The exact solution is known for this problem and is shown as solid narrow lines in Fig. 9.7, and the DG solution is marked as diamond points (one value for each element). The left panel in Fig. 9.7 shows the smooth solution time $t = 3/(4\pi)$ (500 time steps). Clearly, the DG solution is in good agreement with the analytic solution. However, at time $t = 9/(8\pi)$ (750 time steps) the numerical solution develops a *shock* at the steep gradient, leading to oscillations, as seen in the right panel of Fig. 9.7. As time evolves the oscillations become severe and they can pollute the numerical solution. As in the non-smooth advection case, the generation of unphysical oscillations in the numerical solution at contact discontinuities or shocks are due to the Gibbs phenomenon. Any linear numerical method higher than first-order is subject to this problem (Godunov, 1959), unless there is some measure to control or eliminate the spurious oscillations by *limiting* or

filtering the numerical solution. We discuss the limiting procedure for DG methods in the following Section.

9.3 DG for 2D Cartesian Problems

Although the DG method can be adapted to any type of domain or mesh, we choose a rectangular domain D with quadrilateral elements for simplicity. Consider the two-dimensional (2D) scalar conservation law,

$$\frac{\partial U}{\partial t} + \nabla \cdot \mathbf{F}(U) = S(U), \quad \text{in} \quad D \times (0,T), \tag{9.30}$$

where $U = U(x,y,t)$ is the conservative variable such that $(x,y) \in D$, the 2D gradient operator ∇ on D is defined as $\nabla = (\partial/\partial x, \partial/\partial y)$, $\mathbf{F} = (F_1, F_2)$ is the flux function, and $S(U)$ is the source term (if any). The intial condition for the problem is specified as $U(x,y,t=0) = U_0(x,y)$ and we assume that the rectangular domain D is periodic in both the x- and y-directions.

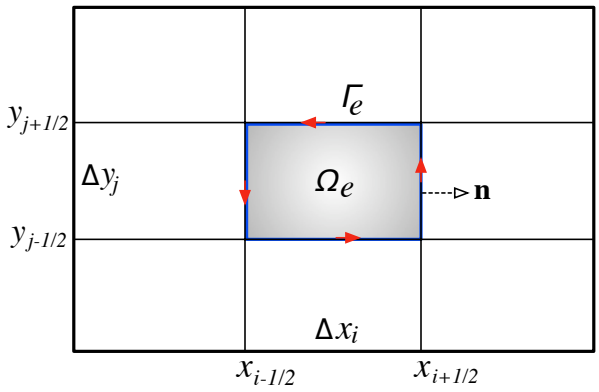

Fig. 9.8 A schematic of a 2D domain with rectangular elements. Ω_e is a generic element with boundary Γ_e and its width in the x- and y-directions are $\Delta x_i = (x_{i+1/2} - x_{i-1/2})$ and $\Delta y_j = (y_{j+1/2} - y_{j-1/2})$, respectively. The outward-facing unit normal vector is denoted by $\mathbf{n}$ and the flux integrals (line integrals) are performed along the boundary Γ_e as indicated by the arrows.

Following the steps used in the previous section the 2D extension of the DG discretization is straightforward. The domain D is partitioned into $N_{elm} = N_x \times N_y$ rectangular non-overlapping elements Ω_e such that

$$\Omega_e = \{(x,y) \,|\, x \in [x_{i-1/2}, x_{i+1/2}], y \in [y_{j-1/2}, y_{j+1/2}]\}, \quad D = \cup_{e=1}^{N_{elm}} \Omega_e, \tag{9.31}$$

where $e = e(i,j)$ is the element index and $i = 1,2,\ldots,N_x$, $j = 1,2,\ldots,N_y$. Figure 9.8 shows a simple partition of D and a general element Ω_e.

We first introduce some basic formal notations required for the discretization. Let $\mathscr{V}_h$ be a finite-dimensional space of polynomials of degree up to $k = N$ such that

$$\mathscr{V}_h = \{\varphi \in L^2(D) : \varphi|_{\Omega_e} \in \mathbb{P}_N(\Omega_e), \forall\, \Omega_e \in D\}, \tag{9.32}$$

where

$$\mathbb{P}_N = \text{span}\{x^m y^n : 0 \le m,n \le N\}.$$

The first step for the DG discretization is the weak Galerkin formulation of the problem (9.30). In general, this is achieved by multiplying (9.30) with a test function and integrating by parts (Green's method) over the domain, where both the approximate solution and test function belong to $\mathscr{V}_h$. Since the discretization procedure is the same for each element, it is only necessary to consider a generic element Ω_e with boundary Γ_e in D (as in Fig. 9.8). Thus, to find the approximate solution $U_h \in \mathscr{V}_h$, (9.30) is multiplied by a test function $\varphi_h(x,y) \in \mathscr{V}_h$ and then integrated over the element Ω_e. This results in the following integral equation (i.e., the weak formulation), analogous to (9.4):

$$\begin{aligned}&\int_{\Omega_e} \frac{\partial U_h(x,y,t)}{\partial t}\,\varphi_h(x,y)\,d\Omega - \int_{\Omega_e} \mathbf{F}[U_h(x,y,t)] \cdot \nabla\varphi(x,y)\,d\Omega \\ &+ \int_{\Gamma_e} \mathbf{F}[U_h(x,y,t)] \cdot \mathbf{n}\,\varphi_h(x,y)\,d\Gamma = \int_{\Omega_e} S[U_h(x,y,t)]\,\varphi(x,y)\,d\Omega,\end{aligned} \tag{9.33}$$

where $\mathbf{n}$ is the outward-normal unit vector on the element boundary Γ_e as shown in Fig. 9.8. A major difference between the weak formulations (9.6) of 1D and (9.33) of 2D cases is the appearance of the flux integral in the 2D case (the last term on the left-hand side of (9.33)). The flux integration should be performed along the element boundary Γ_e. The analytic flux $\mathbf{F}(U_h)\cdot\mathbf{n}$ in (9.33) is discontinuous because the solution itself is discontinuous at the element edges. Therefore, $\mathbf{F}(U_h)\cdot\mathbf{n}$ should be replaced by a numerical flux $\hat{\mathbf{F}}(U_h^-, U_h^+)$. This is addressed by employing a suitable flux formula (or approximate Riemann solver) such as the local Lax-Friedrichs flux (9.5).

The numerical flux resolves the discontinuity at the element edges and again provides the only mechanism by which adjacent elements interact. The finite-volume component of the DG method is the boundary flux integral, which in fact bridges the discontinuous elements together. The flux exchange at the boundaries is responsible for "communicating" physical information across the domain, and it preserves the local conservation properties. Thus the flux integration procedure is extremely important and its accurate evaluation is pivotal to maintaining the overall accuracy of the DG scheme. Following is a simplified version of (9.33) with the numerical flux $\hat{\mathbf{F}} = (\hat{F}_1, \hat{F}_2)$ (for brevity dependencies on (x,y) and t are omitted).

$$\frac{d}{dt}\int_{\Omega_e} U_h\,\varphi_h\,d\Omega - \int_{\Omega_e} \mathbf{F}(U_h)\cdot\nabla\varphi_h\,d\Omega + \int_{\Gamma_e} \hat{\mathbf{F}}\cdot\mathbf{n}\,\varphi_h\,d\Gamma = \int_{\Omega_e} S(U_h)\,\varphi_h d\Omega \tag{9.34}$$

9.3.1 Space Discretization

The space discretization consists of simplifying the integrals in (9.34) by choosing an appropriate set of local orthogonal basis functions. As introduced in the 1D problem, the basis set can be either a set of Legendre polynomials for the *modal* case or a set of Lagrange-Legendre polynomials for the *nodal* case. In either case, the 2D basis set can be constructed with a tensor product of 1D basis functions. This approach significantly simplifies the computational procedure. In order to exploit this option, we introduce the local independent variables (ξ,η) such that

$$\xi = \frac{2(x-x_i)}{\Delta x_i}, \quad \eta = \frac{2(y-y_j)}{\Delta y_j}, \tag{9.35}$$

where $x_i = (x_{i+1/2}+x_{i-1/2})/2$ and $y_j = (y_{j+1/2}+y_{j-1/2})/2$. The width of any element Ω_e is defined by $\Delta x_i = (x_{i+1/2}-x_{i-1/2})$ and $\Delta y_j = (y_{j+1/2}-y_{j-1/2})$ along the x- and y-directions, respectively (Fig. 9.8). Irrespective of the physical size of the rectangular element Ω_e, the transformation (9.35) maps Ω_e onto a unique element $Q \equiv [-1,1]\otimes[-1,1]$, also known as the reference element. Figure 9.9 shows the mapping between a rectangular element Ω_e and Q. Now the approximate solution, test functions and the basis functions all can be defined in terms of local coordinates on Q. Effectively Q is the computational stencil or *molecule* for the 2D DG discretization, where all the integral and differential operations required in (9.34) are performed.

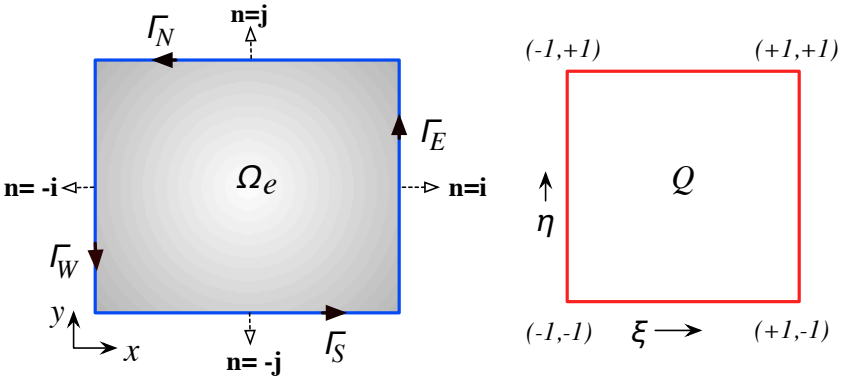

Fig. 9.9 A schematic of the mapping between a rectangular element Ω_e and the reference (standard) element Q by (9.35). The local coordinates (ξ,η) on Q are such that $-1 \le \xi,\eta \le 1$. The outward-facing unit normal vector $\mathbf{n}$ for each wall of Ω_e is marked (left panel), and the flux integrals along the boundary Γ_e can be broken into four integrals (9.36) one for each edge as described in the text.

For the rectangular elements Ω_e the boundary flux integrals in (9.34) along Γ_e can be decomposed in terms of unit vectors $\mathbf{i}$ and $\mathbf{j}$, parallel to the x- and y-axes respectively:

$$\int_{\Gamma_e} \hat{\mathbf{F}} \cdot \mathbf{n}\, \varphi_h \, d\Gamma = \int_{\Gamma_e} (\hat{F}_1 \mathbf{i} + \hat{F}_2 \mathbf{j}) \cdot \mathbf{n}\, \varphi_h \, d\Gamma,$$

where the outward-facing unit normal vector $\mathbf{n}$ takes the values $\mathbf{i}, \mathbf{j}, -\mathbf{i}$ and $-\mathbf{j}$ along the east (Γ_E), north (Γ_N), west (Γ_W), and the south (Γ_S) walls, respectively, as shown in Fig. 9.9. The boundary integrals can then be written as

$$\int_{\Gamma_e} \hat{\mathbf{F}} \cdot \mathbf{n}\, \varphi_h \, d\Gamma = \int_{\Gamma_E} \hat{F}_1 \varphi_h \, d\Gamma + \int_{\Gamma_N} \hat{F}_2 \varphi_h \, d\Gamma - \int_{\Gamma_W} \hat{F}_1 \varphi_h \, d\Gamma - \int_{\Gamma_S} \hat{F}_2 \varphi_h \, d\Gamma. \quad (9.36)$$

9.3.1.1 2D Modal Form

We first discuss the 2D discretization based on the modal basis set. In the (ξ, η) coordinate system the test function is chosen to be a tensor-product of Legendre polynomials $P_\ell(\xi) P_m(\eta)$, which belongs to $\mathbb{P}_N$ in (9.32). The approximate solution $U_h(\xi, \eta, t)$ can be written in terms of the basis functions,

$$U_h(\xi, \eta, t) = \sum_{\ell=0}^{N} \sum_{m=0}^{N} U_h^{\ell m}(t) P_\ell(\xi) P_m(\eta) \quad \text{for} \quad -1 \le \xi, \eta \le 1 \quad (9.37)$$

where $U_h^{\ell m}(t)$ are the time dependent 2D moments (*dofs*) and defined to be

$$U_h^{\ell m}(t) = \frac{(2\ell+1)(2m+1)}{4} \int_{-1}^{1} \int_{-1}^{1} U(\xi, \eta, t) P_\ell(\xi) P_m(\eta) \, d\xi \, d\eta. \quad (9.38)$$

The weak formulation (9.33) can be further simplified by mapping the integrals onto Q using the transformation (9.35), and the properties of Legendre polynomials (basis functions). The mass matrix (9.15) associated with the 2D discretization is also diagonal and can be easily inverted. The final computational form can be written as a decoupled system of time-dependent ODEs for every element Ω_e,

$$\frac{d}{dt} U_h^{\ell m}(t) = \frac{(2\ell+1)(2m+1)}{2 \Delta x_i \Delta y_j} \left[I_G + I_{F_1} + I_{F_2} + I_S \right], \quad (9.39)$$

where $0 \le \ell, m \le N$. Note that the source term $S(U) = 0$ in (9.30) for the pure advection problem; for generality we consider a non-zero source term. The integrals appearing in the right-side of (9.39) can be defined on Q as below,

$$I_G = \int_{-1}^{1}\int_{-1}^{1} \left[\Delta y_j F_1(U_h) P'_\ell(\xi) P_m(\eta) + \Delta x_i F_2(U_h) P_\ell(\xi) P'_m(\eta)\right] d\xi\, d\eta \tag{9.40}$$

$$I_{F_1} = -\Delta y_j \int_{-1}^{1} \left[\hat{F}_1(U(1,\eta,t)) - (-1)^\ell \hat{F}_1(U(-1,\eta,t)\right] P_m(\eta)\, d\eta \tag{9.41}$$

$$I_{F_2} = -\Delta x_i \int_{-1}^{1} \left[\hat{F}_2(U(\xi,1,t) - (-1)^m \hat{F}_2(U(\xi,-1,t)\right] P_\ell(\xi)\, d\xi \tag{9.42}$$

$$I_S = \frac{\Delta x_i \Delta y_j}{2} \int_{-1}^{1}\int_{-1}^{1} S(U_h(\xi,\eta,t)) P_\ell(\xi) P_m(\eta) d\xi\, d\eta, \tag{9.43}$$

where I_G and I_S are the surface integrals corresponding to the gradient and the source terms in (9.33), respectively, and I_{F_1} and I_{F_2} are boundary flux integrals (9.36) along the η and ξ-directions, respectively. $\hat{F}_1$ and $\hat{F}_2$ are the numerical fluxes at the element interfaces, which can be computed by using (9.5).

The integrals appearing in (9.39) are evaluated using high-order accurate Gaussian quadrature rules and will be discussed in the following section. The modes $U_h^{\ell m}$ are predicted at a new time level by (9.39), then the corresponding approximate solution $U_h(\xi,\eta)$ is computed from (9.37). However, this process involves transformations from the spectral to the physical space as discussed in 1D case. The ODE (9.39) can be solved by the SSP-RK procedure given in (9.29).

9.3.1.2 2D Nodal Form

The basic difference between the modal and the nodal form is the choice of basis set. The mapping between the element Ω_e and standard element Q remains the same as in the modal case. In the 2D nodal case, the test function φ_h as well as the approximate solution U_h are expanded in terms of the tensor-product of 1D functions from the nodal basis set. In the (ξ,η) coordinate system the test function is chosen to be $h_\ell(\xi) h_m(\eta)$, a tensor-product of Lagrange-Legendre polynomials (9.21) with roots at GLL quadrature points; $h_\ell(\xi) h_m(\eta)$ belongs to $\mathbb{P}_N$ in (9.32). Thus the approximate solution $U_h(\xi,\eta,t)$ can be expanded as

$$U_h(\xi,\eta,t) = \sum_{\ell=0}^{N}\sum_{m=0}^{N} U_{\ell m}(t)\, h_\ell(\xi)\, h_m(\eta), \quad \text{for} \quad -1 \le \xi,\eta \le 1, \tag{9.44}$$

where $U_{\ell m}(t)$ are the grid-point values (*dofs*) of the approximate solution at the 2D GLL points. The weak formulation (9.33) is simplified by mapping the elements onto the reference element Q, and the procedure is quite analogous to the modal case. The final approximation of (9.30) for an element Ω_e takes the form

$$\frac{d}{dt} U_{\ell m}(t) = \frac{4}{\Delta x_i \Delta y_j w_\ell w_m} \left[I_G + I_{F_1} + I_{F_2} + I_S\right], \tag{9.45}$$

where w_ℓ and w_m are the weights associated with the GLL quadrature rule and I_G is the surface integral corresponding to the gradient term. I_{F_1} and I_{F_2} are the line

integrals along the η- and ξ-directions, respectively, and they are grouped according to (9.36). The simplification (9.45) is possible because the mass matrix associated with discretization is diagonal and easily invertible.

The explicit forms of these integrals are quite similar to those for the modal case (9.40)-(9.43), however, we take an additional step and discretize them using the GLL quadrature rule. The surface (2D) integrals are approximated by a tensor-product of 1D integrals based on the N^{th}-order GLL quadrature rule. Thus on Q there are $(N+1)^2$ GLL quadrature points with coordinates (ξ_l, η_n); $l,n \in \{0,1,\ldots,N\}$. In this particular case we have the following approximations by using the discrete orthogonality relation (9.22) and the property (9.23).

$$I_G \approx \frac{\Delta y_j}{2} w_m \sum_{l=0}^{N} F_{1,lm}(t)\, h'_\ell(\xi_l) w_l + \frac{\Delta x_i}{2} w_\ell \sum_{n=0}^{N} F_{2,\ell n}(t)\, h'_m(\xi_n) w_n, \quad (9.46)$$

$$I_{F_1} \approx -\frac{\Delta y_j}{2} w_m \left[\hat{F}_1(U(1,\eta_m,t))\, \delta_{\ell N} - \hat{F}_1(U(-1,\eta_m,t))\, \delta_{\ell 0}\right], \quad (9.47)$$

$$I_{F_2} \approx -\frac{\Delta x_i}{2} w_\ell \left[\hat{F}_2(U(\xi_\ell,1,t))\, \delta_{Nm} - \hat{F}_2(U(\xi_\ell,-1,t))\, \delta_{0m}\right], \text{ and} \quad (9.48)$$

$$I_S \approx \frac{\Delta x_i \Delta y_j}{4} S_{\ell m}(t)\, w_\ell\, w_m, \quad (9.49)$$

where h'_ℓ and h'_m are the derivatives of the Lagrange polynomial as defined in (9.27) and $\delta_{\ell m}$ is the Kronecker delta function defined in (9.22).

9.3.1.3 Approximating the Integrals

The integrals appearing in the ODEs (9.39) and (9.45) are surface integrals for the internal points and line integrals for the boundaries. Approximation of these integrals has a major role in maintaining the accuracy and computational efficiency of the 2D space discretization. As we saw in the 1D case, Gaussian quadrature rules are the most accurate and efficient means for evaluating integrals. Quadrature formulas such as the Gauss-Legendre (GL) or GLL are widely used for this purpose.

The GL quadrature rule employing $N+1$ quadrature points is *exact* for polynomials of degree $2N+1$ while the GLL quadrature rule with the same number of quadrature points is exact for polynomials of degree $2N-1$. If the integrand is a polynomial of degree $2N$, as in the case of flux integrals, then the integration resulting from the GLL quadrature is *inexact* . In the analysis by Cockburn et al (1990), it is shown that, for a $(N+1)$-th order DG scheme using polynomials of degree N, the quadrature rule used for the surface (internal) integrals should be exact for polynomials of degree $2N$ and the quadrature rule used for boundary flux integrals should be exact for polynomials of degree $2N+1$. In a strict sense, this indicates that there is no single set of $N+1$ quadrature points that can be used to evaluate all the integrals to the required accuracy (Atkins and Shu, 1996). In order to meet the requirements for the exact internal integration and consistent boundary (flux)

integration, these integrals are usually treated with different orders of quadrature formulas.

Utilizing the same type of high-order quadrature rule for both internal and boundary integrals is certainly an option. This is very convenient for practical applications and leads to computationally efficient code development. Nevertheless, it is reported that, for some applications, *over-integration* resulting from keeping the boundary and flux integrals of the same order may lead to instabilities (Lomtev et al, 2000). On the other hand, computational domains with complex geometries that consist of strong curvature or curved boundaries may require more quadrature points than simple Cartesian cases; this is necessary to maintain a specific order of accuracy in the discretization. In other words, the choice of a particular quadrature rule is application dependent, and is also based on the practical consideration of computational efficiency and ease of implementation.

We now review the GL and GLL quadrature rules for the integrals. A tensor-product of 1D quadratures is usually employed to efficiently evaluate the 2D integrals (Deville et al, 2002). Figure 9.10a is a GLL grid with 4×4 quadrature points. Figures 9.10b and 9.10c are the GL grids with 3×3 quadrature points associated with the 2D GLL and GL quadrature rules, respectively; the internal (solution) points are marked as filled circles. The filled-squares along the boundaries in Fig. 9.10b and 9.10c indicate flux points which are interpolated from the solution. Technically both of the quadratures are exact for polynomials of degree up to $k = 5$, and sufficient for a third-order or P^2 DG method. The GLL grid has more points (*dofs*) than the GL case, but the internal integral is still inexact for a P^3 method.

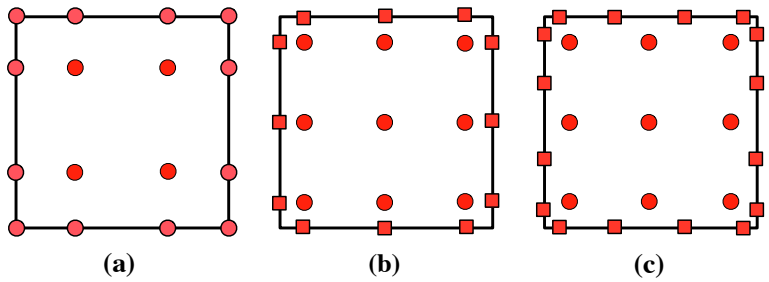

Fig. 9.10 Different types of 2D grid configurations based on Gauss-Legendre-Lobatto (GLL) and Gauss-Legendre (GL) quadrature rules on a square domain $[-1, 1]^2$. The solution points are marked by filled circles and flux points along the boundaries are marked by filled squares. (a) GLL grid with 4×4 quadrature points where the flux points on the boundary coincide with the solution points. (b) GL grid with 3×3 points for internal integrals and 3 flux points on each boundary. (c) Same as in case (b) but with 4 flux points on each boundary.

The GLL quadrature must employ more points than the GL quadrature to guarantee the same order of accuracy. However, the GLL grid has some inherent computational advantages. The GLL quadrature points include points along the boundary

lines and corners of the square domain $[-1,1]^2$ – computing the flux integrals (I_{F_1} and I_{F_2} in (9.39)) along the boundaries is trivial in this case, because the solution and flux points coincide at the quadrature points. This avoids the interpolations required for the flux evaluation, which is a significant computational savings. However, a caveat for this GLL grid configuration is that the boundary flux integral reduces to the same order of accuracy as the internal integral and leads to inexact integration. This may be an issue when the degree of the polynomial is low ($k \leq 3$) because losing an order of accuracy is not affordable, but for higher values of k the loss of an order of accuracy is often outweighed by the computational efficiency and ease of implementation (Nair, 2009). In practice, this type of GLL grid is used for many high-order nodal DG implementations (Hesthaven and Warburton, 2008).

The GL grid as shown in Fig. 9.10b is exact for the DG P^2 scheme but the boundary flux integrals have the same order of accuracy as the internal integrals. In Fig. 9.10c, the order of accuracy of the flux integrals exceeds that of the internal integral as per the theoretical requirement pointed out by Cockburn et al (1990). In order to compute the fluxes along the boundaries, interpolations are required to transfer the solution to the boundary quadrature points – the basis functions may be used for the accurate interpolation of solution (9.37). This will, of course, increase the computational expense. As previously noted, the GL quadrature rule does not use the end points ± 1 in $[-1,1]$, which means that in 2D the corner points are excluded. For rectangular domains, the problematic corner singularities may be avoided by the GL grids. So the GL quadrature may be beneficial for domains with isolated singularities such as the latitude-longitude sphere. An interesting discussion about the choice of quadrature rules can be found in a recent paper by Kopriva and Gassner (2010). In the following section we consider several examples with both GL and GLL grids.

9.3.2 Computational Examples: Advection Tests

Two standard tests for advection problems are the solid-body rotation test and deformational flow test. We examine these non-divergent test cases individually.

9.3.2.1 Solid-Body Rotation Test

To test the DG schemes discussed above we first consider a solid-body rotation problem with a smooth function on a square domain. The domain D in (9.30) is chosen to be $[-\pi,\pi]^2$ with periodic boundary conditions and the initial condition is the Gaussian hill $U(x,y,t=0) = \exp[-5((x-x_c)^2+(y-y_c)^2))]$ centered at (x_c,y_c). The velocity is prescribed as $(u,v) = (-\pi y, \pi x)$ and the flux function is $\mathbf{F}(U) = (uU, vU)$. The Gaussian hill is placed at the center of the domain $(x_c = 0, y_c = 0)$ for the convergence study so that U is continuous at the (periodic) boundaries.

The tests are conducted with both modal and nodal versions of the DG discretization and for different spatial resolution. We vary both the total number of elements ($N_{elm} \in \{20^2, 40^2, 80^2, 160^2\}$) and the polynomial degree ($k \in \{1,2,3,4\}$). The normalized standard l_2 error is computed after one complete rotation, and Fig. 9.11a shows the results with the modal version employing GLL quadrature (the nodal version gives visibly indistinguishable results). Two types of errors, *h*-error and *p*-error, are used for the convergence tests of element-based high-order Galerkin methods such as DG. The *h*-error measures the error computed by varying number of elements and keeping the polynomial degree (k) constant, while the *p*-error measures the error when the polynomial degree is varied but the number of elements is kept fixed. For a given N_{elm} the *p*-error is reduced as the polynomial degree increases, in Fig. 9.11a it is shown as black dots aligned in the vertical direction. The measures of the *p*-error vary more rapidly (at an exponential rate) than that of the *h*-error. The exponential (spectral) convergence is also reported for similar tests in Levy et al (2007).

Figure 9.11b shows the *strong scaling* results on a parallel computer architecture, a measure of parallel efficiency when the problem size is held constant. Ideally, the total work would be split evenly among processors so that doubling the number of processors would halve the runtime. This is measured by 'speed-up,' the ratio of the runtime on one processor to the runtime on a given number of processors. In this sense Fig. 9.11b shows almost perfect scaling for the nodal DG scheme (run with $N_{elm} = 80^2$ elements, each with 6×6 GLL nodes). This simulation consisted of $40,000$ time steps on a 1024 dual-node BlueGene/L cluster. Spectral convergence (for smooth problems) and excellent scaling are two remarkable properties of DG algorithms.

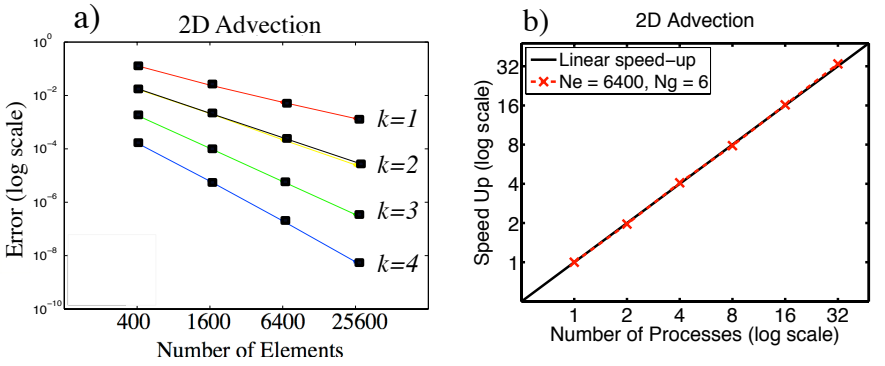

Fig. 9.11 a) Convergence results (l_2 error) for the solid-body rotation test at different resolutions and varying polynomial degree (k). b) The strong scaling results as measured with a resolution of $N_{elm} = 80^2$ and each element containing 6×6 GLL points.

9.3.2.2 Deformational Flow Test

For the deformational flow test case we consider the test proposed by Smolarkiewicz (1982). This problem is relevant to meteorology because it simulates the effect of closed vortices on warm air parcels. The test describes the advection of a scalar field (i.e., U in (9.30)), which is initially defined to be a cone of height 1 and radius 15 units located at the center of a square domain of side $L = 100$ units. The non-divergent flow field is defined by the stream function,

$$\psi(x,y) = 8\sin(4\pi x/L)\cos(4\pi y/L), \quad u = -\frac{\partial \psi}{\partial y},\ v = \frac{\partial \psi}{\partial x},$$

where u and v are the components of the wind field. Staniforth et al (1987) provide an analytical solution for this test in terms of elliptic functions and showed that there is a breaking time $T_b = 2637.6$, beyond which the length scale of the exact solution diminishes as a function of time. We examine the DG solutions at time $t = T_b/50$, when the solution exhibits very fine structures of deformation. This test is very challenging because of the severe deformation of the fields and sharp gradients which evolve in time.

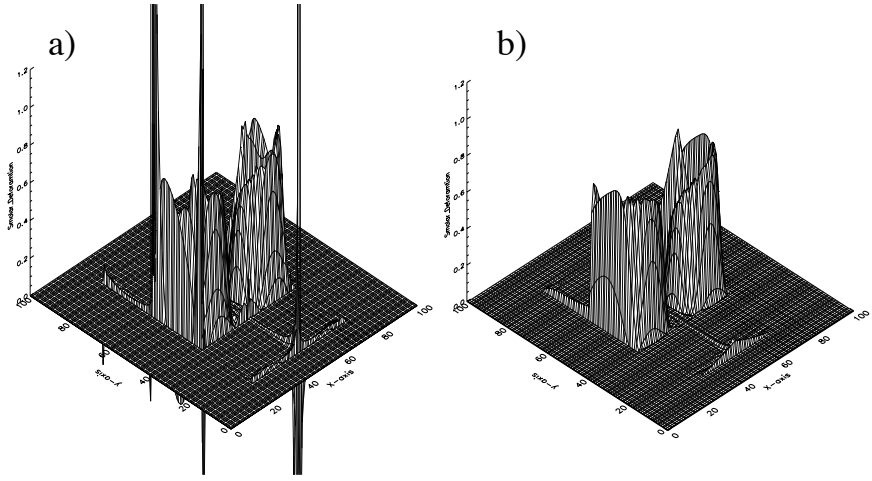

Fig. 9.12 Numerical solutions for the deformational flow test at time $t = T_b/50$ with the DG advection scheme on a 2D Cartesian domain with 40×40 elements. a) Solution with the nodal DG scheme employing 4×4 GLL points (as shown in Fig. 9.10a) on each element. The boundary flux integrals are approximated with the same order 1D GLL quadrature rule. b) Solution with the modal DG scheme employing 3×3 GL points (as shown in Fig. 9.10b or 9.10c) on each element. The flux integrals are performed with the same order GL quadrature.

The numerical results are presented in Fig. 9.12 on a 40×40 element domain employing the GLL and GL quadratures (grids) as shown in Fig. 9.10, plotted on the native computational grid to avoid interpolation errors. Figure 9.12a shows the results for the nodal DG scheme with a 4×4 GLL grid, where the boundary integrals use the same order GLL quadrature. This choice of quadrature exhibits spurious overshoots and undershoots, and the modal DG scheme with the GLL quadrature produces a similar result. Changing the spatial order of accuracy (up to 7×7 quadrature points) with the GLL nodes does not improve the results, and similar results are reported by Crowell et al (2009). Figure 9.12b shows the results with a modal version of the DG P^2 method employing 3×3 GL points. With GL grids, the solution is significantly smoother. Again, the nodal version produces similar results.

This indicates that, irrespective of the modal or nodal variant of the DG method, the GL quadrature has some qualitative advantage over the GLL quadrature; especially when the flow field is very complex. The DG schemes employing GL quadrature are more robust than those with the GLL quadrature. On the other hand, for a fixed order of accuracy, we noticed that the DG/GLL combination has a more lenient CFL stability restriction than the DG/GL combination. This is mainly due to the distribution of the internal quadrature points in the reference element (see Fig. 9.10). In the case of the GL quadrature points, the shortest distance between the internal points and the boundary is smaller than that of the GLL points, leading to relatively smaller grid spacing (Δx). In other words, the CFL estimate discussed in Section 9.2.5, $1/(2N+1)$ for the DG P^N method (Cockburn and Shu, 2001), appears to be an overestimate when the DG/GLL combination is used.

9.3.2.3 Barotropic Vorticity Equation

We now discuss a general form of (9.30) with a non-zero source term, a simple non-divergent barotropic model based on the classical barotropic vorticity equation (BVE). A barotropic atmosphere is a single-layered fluid; under this assumption there is no vertical component, so the equation to be solved is 2D. The BVE has special importance in meteorology and a historical perspective of the BVE can be found in Lynch (2008). The BVE is useful for modeling the (idealized) evolution of tropical cyclones (DeMaria, 1985), and also for the theoretical study of the interactions of vortices in close proximity. Recently, Levy et al (2009) have developed an element-based Galerkin method for solving the BVE using the DG discretization; we review this model in the present context.

The BVE can be cast in the following form (Levy et al, 2009):

$$\frac{\partial \zeta}{\partial t} + \frac{\partial}{\partial x}(u\zeta) + \frac{\partial}{\partial y}(v\zeta) = -\beta v, \tag{9.50}$$

where u and v are the horizontal components of the wind vector $\mathbf{v}$ such that $\mathbf{v} = (u, v)$, $\zeta = (\nabla \times \mathbf{v}) \cdot \hat{\mathbf{k}}$ is the relative vorticity and $\hat{\mathbf{k}}$ is a unit normal vector in the vertical direction. In (9.50), $\beta = \partial f / \partial y$ is based on the beta-plane approxima-

tion (Vallis, 2006) where f is the Coriolis parameter. The solution process involves predicting ζ at every time step, however, the (u,v) field also evolves in time and therefore needs to be computed at every new time step. Since the wind field is non-divergent it can be prescribed in terms of the stream function ψ such that $u=-\psi_y$ and $v=\psi_x$, where the suffixes denote partial differentiation. The relation $\zeta=v_x-u_y$ leads to the following Poisson equation for ψ:

$$\nabla^2\psi=\zeta. \tag{9.51}$$

Usually the initial conditions for (9.50) are prescribed in terms of the tangential velocity, from which the initial values for $\mathbf{v}$ and ζ can be derived. At every time step ζ is predicted and the corresponding stream function at the new time-level is computed by solving the Poisson problem (9.51). This is required because the wind field (u,v) must be available for the new prediction cycle; as mentioned, it can be computed directly from ψ using the relation $(u,v)=(-\psi_y,\psi_x)$.

Thus the solution process for the BVE involves solving the advection equation (9.50) and the Poisson equation (9.51) as a system. The elliptic type equation (9.51) may be solved using the DG method as described in Rivière (2008), the high-order spectral method (Kopriva, 2009), or any number of other methods. Since our focus is primarily on hyperbolic problems, we do not consider the solution procedure for (9.51) here, except to say that we adopt a spectral-element based Poisson solver (Levy, 2009) for the BVE model.

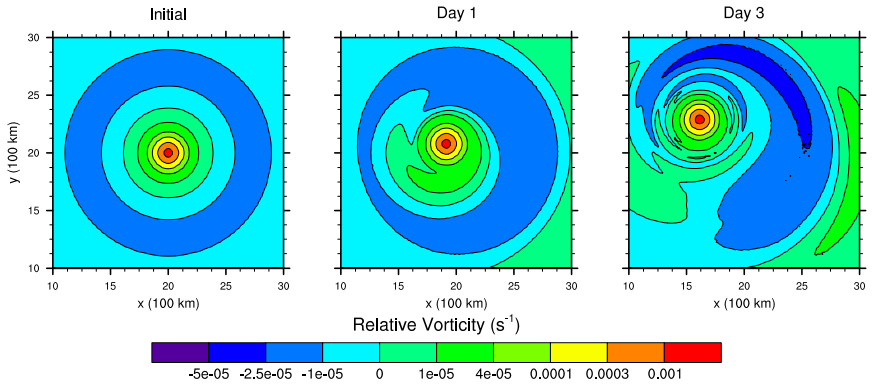

Fig. 9.13 Contours of the vorticity field (ζ) in the tropical cyclone simulation, shown after 1 and 3 days. The left panel shows the initial fields; the simulated results after 1 and 3 days are shown in the central and the right panels. Calculations are done on a square domain consisting of 100×100 elements each with 4×4 GLL points.

The initial wind profile for the vortex centered at (x_c,y_c) can be expressed in terms of tangential velocity $V(r)$ where $r=[(x-x_c)^2+(y-y_c)^2]^{1/2}$ is the radial distance from the center. The wind field and $V(r)$ are given by

$$u = -V(r)(y-y_c)/r, \quad v = V(r)(x-x_c)/r, \quad V(r) = \frac{2V_m r \exp[-a(r/r_m)^b]}{r_m[1+(r/r_m)^2]}. \tag{9.52}$$

The initial relative vorticity can be derived as

$$\zeta(r) = \begin{cases} V'(r)+V(r)/r & \text{if } r \neq 0, \\ 2V'(0) & \text{if } r = 0. \end{cases} \tag{9.53}$$

The physical dimension of the domain D is a 4000 km $\times$ 4000 km square, and D is periodic in both directions. The other parameters used in (9.52) are $V_m = 30$ m/s, $r_m = 80$ km, $a = 10^{-6}$, $b = 6$, the vortex center (x_c, y_c) positioned at $(2000, 2000)$ km, and β is computed at the latitude 20°N.

The formulation for the BVE (9.50) may be considered as a special case of the flux-form transport equation (9.30) with a non-zero source term $S(U)$. Therefore it is clear that the "conservative" variable is $U = \zeta$, the flux function is $\mathbf{F}(U) = (u\zeta, v\zeta)$ and the source is $S(U) = -\beta v$. For the DG discretization of (9.50), we employ the nodal scheme as described in Section 9.3.1.2. The computational domain consists of 100×100 elements each with 4×4 GLL points (Fig. 9.10a) so the average horizontal resolution is approximately 13.3 km. A third-order Runge-Kutta scheme (9.29) is used to solve the ODE (9.45) corresponding to (9.50), with a (sub-optimal) time step of $\Delta t = 90$ sec.

In the nodal formulation the relative vorticity ζ and stream function ψ (from (9.51)) are approximated at the GLL quadrature points (ξ_l, η_n) using the summation (9.44). To find the non-divergent wind at any time-level from the stream function fields at the GLL points the following collocation differentiation can be employed,

$$u(\xi,\eta) = -\psi_\eta \approx -\sum_{\ell=0}^{N}\sum_{m=0}^{N} \psi_{\ell m} h_\ell(\xi) h'_m(\eta),$$
$$v(\xi,\eta) = \psi_\xi \approx \sum_{\ell=0}^{N}\sum_{m=0}^{N} \psi_{\ell m} h'_\ell(\xi) h_m(\eta).$$

The numerical results are shown in Fig. 9.13. The leftmost panel shows the initial relative vorticity fields, and simulated results after 24 and 72 hours are shown in the central and right panels, respectively. As expected, the center of cyclonic vortex is well resolved and the cyclonic motion has drifted in the northwestward direction (DeMaria, 1985). Realistic hurricane simulation needs high-resolution complex 3D models capable of fast simulations. The DG methods are well-suited to address this problem because DG algorithms are known for their high parallel efficiency and adaptive mesh refinement capabilities.

9.4 Limiters for DG Methods

High-order numerical schemes will produce spurious oscillations in the vicinity of discontinuities or shocks and near under-resolved solution gradients. The unphysical oscillations not only pollute the solution but may lead to numerical instabilities. Preservation of physically realizable properties of the solution such as monotonicity (shape-preservation) or the less restrictive positivity is of great importance in atmospheric transport modeling (chapter 8). For instance, the mixing ratio (e.g., relative humidity) or density simulated by an atmospheric model should always preserve its positive sign (positive-definite). Even oscillations with small amplitudes can create negative density which in turn produce physically unacceptable negative mass — this might arise even if a minute negative density is multiplied by the volume (or integrated over a region). The process of controlling or completely eliminating the spurious oscillations in the numerical solution is often referred to as limiting. A limiter also provides nonlinear stability to the solution.

The Godunov theorem (Godunov, 1959) asserts that the "monotone linear schemes are at most first-order accurate." For high-order methods this implies that designing a monotone scheme is a daunting task because the coexistence of monotonicity and the high-order nature of the solution is difficult if not impossible. The monotonic limiting is a non-linear process that removes the oscillations from the solution at regions (points) where monotonicity is violated, and when activated the limiter reduces the oscillatory (high-order) solution to first-order. It is required that a limiter does not violate the mass conservation property (i.e., preservation of the cell-average) of the underlying conservative numerical scheme and, to the greatest extent possible, it should retain the high-order accuracy of the solution. Therefore, a limiter should be applied to the high-order scheme in a surgical manner and it should not be activated in smooth regions of the solution. Thus it is very important to have a criterion for limiting that guides *when* and *where* to limit the solution.

Another potential venue for controlling numerical noise due to under-resolved solution gradients is the application of so-called *h-p* adaptivity. Here h stands for number of elements in the domain and p is the polynomial order within in the element (Karniadakis and Sherwin, 2005). Since shocks are not really present in atmospheric model, the requirement is to prevent the generation of under-resolved gradients on the grid. The problem here is optimize the *h-p dofs* to the local structure of the solution. For example, high-order elements where high-gradients are developing can be divided into 2 elements of order $p/2$ to prevent the growth of oscillations. Ultimately one could end-up with p first order elements that are guaranteed to preserve the extrema of the solution. This approach may be more intensive on software engineering and grid refinement based on error estimators, but could be an alternative to the brute force approach of slope limiters. The DG methods are amenable to adaptive mesh refinement (AMR) strategy based on *h-p* adaptivity. Development of models based on AMR is an active area of research in geosciences (St-Cyr and Neckels, 2009; Kubatko et al, 2009).

The second-order finite-volume (FV) schemes can successfully incorporate limiters such as the slope limiters (van Leer, 1974) or flux limiters (Boris and Book,

1973). This is done either by designing a scheme which inherently prohibits oscillatory solution (Smolarkiewicz, 1984) or by applying the limiter in the reconstruction or the post-processing stage. As the order of the numerical scheme increases the limiting procedure becomes more complex and computationally expensive. A class of high-order finite-volume schemes known as essentially non-oscillatory (ENO) developed by Harten et al (1987) and its advanced variant weighted essentially non-oscillatory (WENO) by Liu et al (1994) can successfully control spurious oscillations in the solution. As the name suggests the ENO or WENO solutions are not strictly monotonic. The solution may still have oscillations of small amplitude but they do not grow with time. These schemes use adaptive stencils in the reconstruction procedure which are based on local smoothness of the numerical solution, and automatically achieve high-order accuracy and non-oscillatory properties near discontinuities.

Since the DG method has a strong FV-like connection, it may be technically possible to extend the limiters developed for FV methods to at least low-order DG methods. However, for DG schemes the direct application of a FV-based limiter such as the flux limiter is not trivial because the *dof* evolved in time per element (cell) is higher than that of the FV method. Limiting high-order DG methods on general meshes is still an open question. Here we consider the basic slope limiter (Cockburn and Shu, 1989) and the WENO-based limiting proposed by Qui and Shu (2005b) for relatively low-order DG methods.

9.4.1 The 1D Limiters for DG Methods

The basic limiter developed for the DG scheme (Cockburn and Shu, 1989) relies on the MUSCL (Monotonic Upstream Centered Schemes for Conservation Laws) slope limiting technique (van Leer, 1977). The MUSCL approach employs the piecewise linear reconstruction for the subgrid-cell distributions resulting in a second-order accurate scheme. The reconstruction process in this case is constrained to be free from spurious oscillations (monotonic) by applying the *minmod* limiter. To understand how the minmod limiter works, we consider the piecewise linear reconstruction for the 1D grid used in Section 9.2.3.

9.4.1.1 The minmod Limiter

Let $U_j(x)$ be the density distribution in a cell of width $\Delta x_j = x_{j+1/2} - x_{j-1/2}$. The piecewise linear representation of $U_j(x)$ can be expressed in terms of the slope $U_{x,j}$ and the cell-averaged density $\overline{U}_j$ (see Fig. 9.14),

$$U_j(x) = \overline{U}_j + (x - x_j)U_{x,j}, \quad \overline{U}_j = \frac{1}{\Delta x_j}\int_{x_{j-1/2}}^{x_{j+1/2}} U_j(x)dx, \tag{9.54}$$

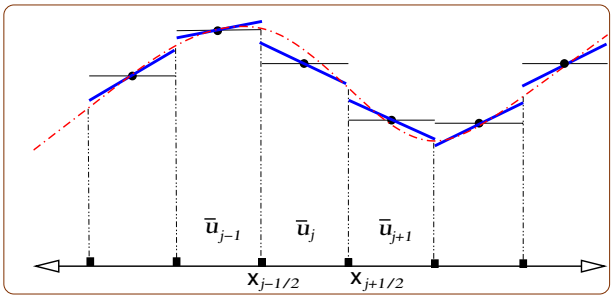

Fig. 9.14 A schematic illustration of the piecewise linear reconstruction. The cell averages $\overline{U}_j$ are shown as horizontal lines and the cell boundaries are labeled by $x_{j\pm1/2}$. The smooth dashed-line indicates the actual solution $U(x)$ which is approximated by piecewise linear distributions (broken thick lines) on each cell.

where $x_j = (x_{j-1/2} + x_{j+1/2})/2$. There are an infinite number of possibilities to choose the value of $U_{x,j}$ in (9.54) without violating mass conservation (preserving $\overline{U}_j$), nonetheless, we choose the limited slope $\tilde{U}_{x,j}$ based on the minmod approach. A minmod function has three arguments. The first argument is the slope of the cell in question and remaining arguments are the slopes of the neighboring cells. If the *left* and the *right* slopes preserve the same sign, then the minmod function returns the minimum of the absolute value of the slopes with the same sign; otherwise, if the signs are opposite, it sets the slope to zero. This can be written as follows:

$$U(x)_j = \overline{U}_j + (x - x_j)\tilde{U}_{x,j}, \quad \tilde{U}_{x,j} \Leftarrow \text{minmod}(U_{x,j}, U_{x,j-1/2}, U_{x,j+1/2}), \tag{9.55}$$

where the arrow indicates the replacement of the slope $U_{x,j}$ by the limited slope $\tilde{U}_{x,j}$; the minmod function is formally defined to be

$$\text{minmod}(a,b,c) = \begin{cases} s\,\min(|a|,|b|,|c|) & \text{if } s = \text{sign}(a) = \text{sign}(b) = \text{sign}(c), \\ 0 & \text{otherwise .} \end{cases} \tag{9.56}$$

The slopes of the neighboring cells (on a non-unifom grid) are given by

$$U_{x,j-1/2} = \frac{\overline{U}_j - \overline{U}_{j-1}}{(\Delta x_j + \Delta x_{j-1})/2}, \quad U_{x,j+1/2} = \frac{\overline{U}_{j+1} - \overline{U}_j}{(\Delta x_{j+1} + \Delta x_j)/2}.$$

This limiter falls under the class of the total variation diminishing (TVD) limiters (Toro, 1999). The minmod limiter is strictly non-oscillatory, but unfortunately it clips the legitimate extrema of smooth solutions and degrades high-order accuracy. However, the excessive limiting of the minmod function at smooth regions can be controlled to some extent by modifying (relaxing) the limiting criteria in (9.56). The resulting *modified minmod limiter* has the total variation bounded (TVB) property, which preserves high-order accuracy at smooth extrema at the cost of allowing minor oscillations in the solution. Let 'Minmod' be the modified minmod function

which is defined to be

$$\text{Minmod}(a,b,c) = \begin{cases} a & \text{if } |a| \le M_l, \\ \text{minmod}(a,b,c), & \text{otherwise}, \end{cases} \tag{9.57}$$

where M_l is a problem-dependent positive number. This parameter is more or less a *magic* number which works quite well for a few sets of problems (see, Cockburn and Shu (2001)). Smaller values of M_l introduce greater local dissipation, but larger values produce oscillations in the solution. Although there are efforts to make M_l problem independent (Ghostine et al, 2009), a generalized approach for various applications particularly in multi-dimensional systems has yet to be established.

9.4.1.2 Generalized Slope Limiter

The modal expansion (9.10) for the approximate solution $U_j(\xi)$ can be rearranged as follows (with the time dependency omitted for brevity):

$$U_j(\xi) = U_j^0 + U_j^1 \xi + \sum_{k=2}^{N} U_j^k P_k(\xi), \tag{9.58}$$

where the expansion coefficients (or moments) U_j^k are defined in (9.11). If the solution $U_j(\xi)$ is approximated as element-wise linear functions, then $U_j(\xi) = U_j^0 + U_j^1 \xi$, where the coefficients $U_j^0 = \overline{U}_j$ is the average value and $U_j^1 = U_j'(\xi)$ is the slope. This is simply the P^1 part of the solution (9.58), which is analogous to the piecewise linear reconstruction (9.54). Therefore the limited solution for the P^1 case, in terms of ξ, can be written as

$$U_j(\xi) = \overline{U}_j + \xi \tilde{U}_j^1, \quad \tilde{U}_j^1 \Leftarrow \text{minmod}\,(U_j^1, \frac{\overline{U}_j - \overline{U}_{j-1}}{\Delta\xi}, \frac{\overline{U}_{j+1} - \overline{U}_j}{\Delta\xi}), \tag{9.59}$$

where $\tilde{U}_j^1$ is the limited slope by the minmod function and $\Delta\xi = 2$. If $\tilde{U}_j^1 \neq U_j^1$ then it indicates that minmod limiter is in action; otherwise, if $\tilde{U}_j^1 = U_j^1$ then the indication is that the element is non-oscillatory and does not need limiting. In other words the minmod function may also be used to detect elements which require limiting.

Note that the left and right slopes used in the minmod function in (9.59) may be replaced with the less restrictive slopes $2(\overline{U}_j - \overline{U}_{j-1})/\Delta\xi$ and $2(\overline{U}_{j+1} - \overline{U}_j)/\Delta\xi$, respectively, (Cockburn and Shu, 2001). This leads to a simplified slope estimate at the element edges in the ξ-coordinate as employed in (9.59).

$$U_j(\xi) = \overline{U}_j + \xi \tilde{U}_j^1, \quad \tilde{U}_j^1 \Leftarrow \text{minmod}\,(U_j^1, \overline{U}_j - \overline{U}_{j-1}, \overline{U}_{j+1} - \overline{U}_j). \tag{9.60}$$

In the context of the high-order DG method, Cockburn and Shu (1989) further extended the minmod limiter to the *generalized slope limiter*. This is achieved by selectively applying the limiter (9.60) to the high-order solution (9.58) where the solution is not smooth. The selection procedure (i.e., detecting the elements which re-

quire limiting) involves finding the left and right edge values $U_{j\pm1/2} = U_j(\xi = \pm 1)$ from (9.58), and checking for oscillation using the minmod function:

$$\tilde{U}^-_{j+1/2} = \overline{U}_j + \text{minmod}\,(U^-_{j+1/2} - \overline{U}_j, \overline{U}_j - \overline{U}_{j-1}, \overline{U}_{j+1} - \overline{U}_j), \tag{9.61}$$

$$\tilde{U}^+_{j-1/2} = \overline{U}_j - \text{minmod}\,(\overline{U}_j - U^+_{j+1/2}, \overline{U}_j - \overline{U}_{j-1}, \overline{U}_{j+1} - \overline{U}_j), \tag{9.62}$$

where $U^-_{j+1/2}$ and $U^+_{j-1/2}$ denote the left and right limits (see Fig. 9.1) of the edge values $U_{j+1/2}$ and $U_{j-1/2}$, respectively.

Now the generalized slope limiter algorithm for a high-order solution (9.58) can be summarized as follows.

- First, compute the limited edge values $\tilde{U}^-_{j+1/2}$ and $\tilde{U}^-_{j+1/2}$ using (9.61) and (9.62).
- If $\tilde{U}^-_{j+1/2} = U^-_{j+1/2}$ and $\tilde{U}^+_{j-1/2} = U^+_{j-1/2}$, then it indicates that there is no spurious oscillation (or no need for limiting) in the element in question, and the solution (9.58) is acceptable as is.
- If $\tilde{U}^-_{j+1/2} \neq U^-_{j+1/2}$ and/or $\tilde{U}^+_{j-1/2} \neq U^+_{j-1/2}$, then it indicates there is oscillation is in the element and the solution should be limited by using (9.60).
- In the limited case only the limited P^1-part of solution is considered, all the high-order coefficients in (9.58) $U^k_j = 0$ for $k \geq 2$.

As discussed above the minmod limiters are dissipative, and may not be suitable for some applications. In such cases, if a solution with oscillations of small amplitude is acceptable, then it is appropriate to use the more relaxed Minmod function (9.57) instead of the regular minmod function.

9.4.1.3 The Moment Limiter

Biswas et al (1994) generalized the minmod limiter to a *moment limiter* suitable for limiting high-order DG methods. The moment limiter limits the derivative of the solution starting with the highest order, and it is given by

$$\tilde{U}^k_j = \frac{1}{2k-1}\text{minmod}\,((2k-1)U^k_j, U^{k-1}_j - U^{k-1}_{j-1}, U^{k-1}_{j+1} - U^{k-1}_j). \tag{9.63}$$

When $k = 1$, clearly the limiter (9.63) reduces to the minmod limiter in (9.60). The limiter is applied in an adaptive manner starting with the highest-order coefficient (moment) U^k_j. If $\tilde{U}^k_j = U^k_j$ then it indicates limiting is not required; if not, limiting is required and (9.63) is applied to the next lower- level coefficients U^{k-1}_j. The process stops when no modification of the coefficient occurs by applying (9.63); otherwise, the next highest order coefficient is limited. The moment limiter performs better than the generalized slope limiter at least in the 1D case; however, extending the algorithm to multi-dimension (Krivodonova, 2007) is computationally expensive.

9.4.1.4 The WENO-based Limiter

There is a novel class of limiters for DG methods recently introduced by Qui and Shu (2005b) based on the WENO method. A major advantage of this approach is its ability to retain high-order accuracy for the DG scheme while suppressing spurious oscillations. The WENO based limiting strategy for DG methods consists of two crucial steps. These are the identification of so-called *troubled cells* or the cells (elements) that need limiting, followed by a reconstruction step for the non-oscillatory solution in the troubled cells using the neighboring cell-averages. To identify the troubled cells one may use any of the slope limiting techniques described above. If, for example, the slope in a cell changes when using the minmod limiter, then that particular cell is declared a troubled cell and limiting is performed by using the WENO approach. Although the WENO limiter does not adversely affect the order of accuracy of the solution in a smooth cell, a judicious identification of troubled cells is required to avoid unnecessary computations in smooth regions.

The details of the WENO limiter implementation is given in Qui and Shu (2005b), and we do not discuss it herein. A DG P^N method is formally $(N+1)$th order accurate if the quadrature rule is exact for polynomials of degree at least $2N+1$. In order to match the same order of accuracy, a WENO reconstruction should be at least $(2N+1)$th order accurate as well. A WENO-based limiter of this order requires $2N+1$ neighboring elements $\Omega_{j-N},\ldots,\Omega_{j+N}$ to limit an element Ω_j located at the center of the stencil. Unfortunately, this requirement necessitates a wider computational stencil when $N>2$, which impedes the local nature (and, therefore, the parallel efficiency) of the combined DG-WENO scheme.

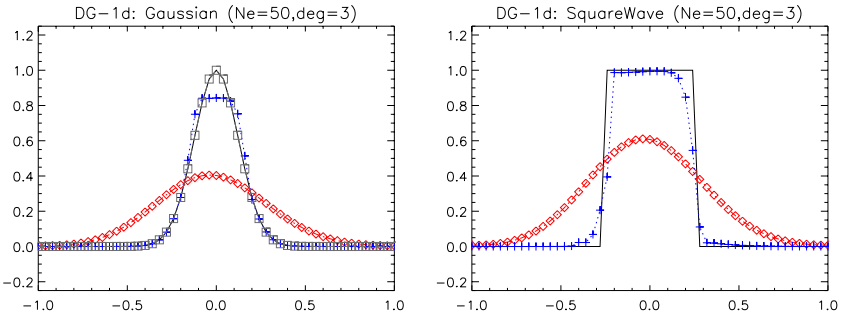

Fig. 9.15 Numerical solution after one revolution with the modal DG scheme combined with various limiters for the linear advection problem (9.2). The left and right panels show solutions for the smooth case (Gaussian hill) and non-smooth case (rectangular wave) as initial conditions, respectively, where the solid line indicates the exact solution. The diamond points ($\diamond$) show the solution with the basic minmod limiter, and '+' points indicate the solution with the generalized slope limiter employing the Minmod function with the parameter $M_l = 0.02$. On the left panel, square points show the solution with the generalized slope limiter but the parameter $M_l = 0.06$.

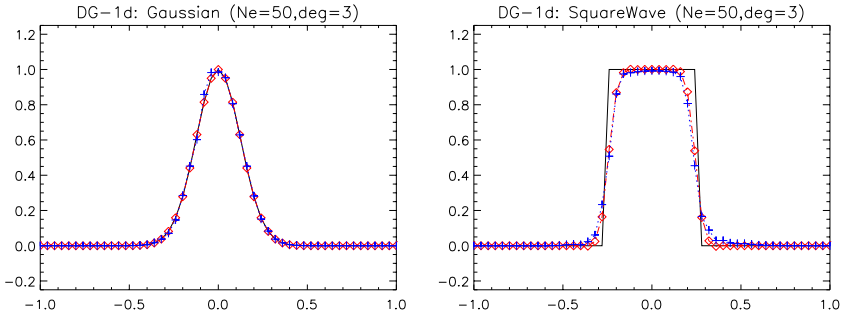

Fig. 9.16 Same as in Fig. 9.15 but with the third-order WENO limiter ('$\diamond$' points) and the moment limiter ('$+$' points). The computational domain $[-1,1]$ consists of 50 elements each with 4 GLL quadrature points.

9.4.1.5 Computational Examples with Limiters

We repeat the numerical examples used in Section 9.2.6 to demonstrate the effectiveness of the limiters as discussed above. First, the simple linear advection problem $U_t + U_x = 0$ is solved with initial conditions representing two extreme cases, a Gaussian hill (smooth case) and a rectangular wave (non-smooth case). A modal version of the DG discretization is employed with 50 elements, each with 4 GLL quadrature points, in the domain $[-1,1]$, and 400 time steps are used for a complete revolution. Ideally, the challenge for a limiter is to preserve high-order accuracy in smooth regions of the solution while eliminating spurious oscillations *only* from the non-smooth regions.

Figure 9.15 shows the numerical solutions with the basic minmod limiter (9.55) and the generalized slope limiter combined with the modified Minmod limiter (9.57). Only one point per element is plotted for clarity. The Minmod function employs a problem-dependent parameter M_l which controls the limiting operation. The exact solution (initial condition) is shown as solid lines in Fig. 9.15, and the solution with the basic minmod limiter is very diffused in both cases ('$\diamond$' points). For the non-smooth case (Fig. 9.15 right panel), relatively *better* solutions are obtained with the generalized slope limiter ('$+$' points) for which the parameter value is $M_l = 0.02$; however, for $M_l > 0.02$ the limiter reintroduces oscillations. For the same value $M_l = 0.02$, the generalized slope limiter clips the peak smooth regions of the Gaussian hill as seen in the left panel of Fig. 9.15 ('$+$' points). Nevertheless, when M_l is increased to 0.06 the limiter further relaxes without destroying the legitimate extrema of the Gaussian hill (square points). Although these limiters are simple and easy to implement, a major drawback is that they have a strong dependence on the parameter M_l. Moreover, the basic minmod (P^1) limiter is unacceptably diffusive for high-order DG methods for practical applications (Iskandarani et al, 2005).

Now we consider the same experiment with the moment limiter (9.63) and a third-order WENO- based limiter. For the computational examples considered here we employ a DG P^2 case combined with a WENO limiter employing the GLL quadrature rule with 4 points. Figure 9.16 shows the limited solution with the WENO limiter ('$\diamond$' points) and moment limiter ('$+$' points). Both the limiters perform very well for the two extreme cases. However, the WENO based limiter is very robust and performs slightly better than the moment limiter in terms of the symmetry of the solution (shape preservation). The WENO limiter unfortunately comes with a higher computational cost because for the third-order (P^2) case a 5-element wide stencil is required for the reconstructions.

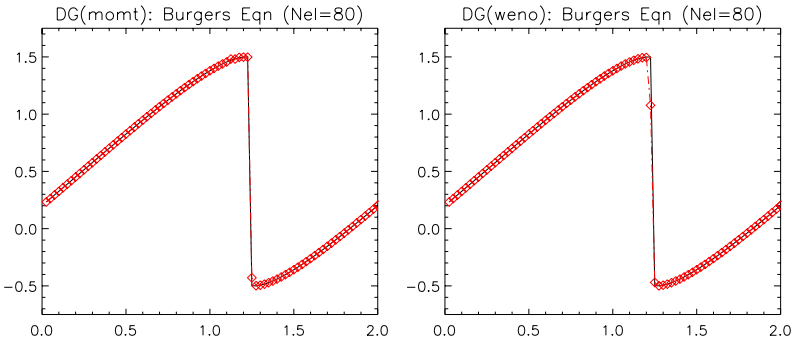

Fig. 9.17 Limited numerical solution for the inviscid Burgers equation at time $t = 3/(2\pi)$ with the modal DG scheme. The solid-line indicates the exact solution and '$\diamond$' points show the limited DG solution. The left panel shows solution by DG scheme combined with the moment limiter and right panel shows DG solutions combined with the WENO limiter.

The moment limiter (9.63) and the third-order WENO limiter are applied to the P^2 DG scheme for solving the inviscid Burgers equation $U_t + (U^2/2)_x = 0$, with the initial condition $U_0(x) = 1/2 + \sin(\pi x)$. As mentioned in Section 9.2.6, this is a simple non-linear case where a shock wave develops during the integration, but the analytic solution is known at any time. The computational domain $[0,2]$ consists of 80 elements, each containing 4 GLL points. The limited numerical solution at time $t = 3/(2\pi)$ (1000 time steps) is shown in Fig. 9.17, where the left and right panels show solutions with the moment and WENO limiters, respectively. For clarity, only one point per element is sampled for displaying the numerical results. Both limiters successfully eliminate spurious oscillations near the shock (as seen in Fig. 9.7), and the computed solutions are very similar to the reference solutions. Note that in Fig. 9.7, for which no limiting is employed, shocks develop during the integration and oscillations appear at $t = 9/(8\pi)$ (750 time steps). Eventually the growing spurious oscillations contaminate the numerical solution in this case.

9.4.2 2D limiters for the DG Method

The 1D limiters used for the high-order DG method are quite successful in eliminating spurious oscillations. Unfortunately, extending these limiters to 2D problems is not trivial. In addition to the slopes (derivatives), the high-order derivatives and cross-derivative terms are also subject to limiting, making the limiting process algorithmically complex and computationally expensive. The development of limiting techniques for high-order DG methods is an active area of research, and two promising approaches in this direction are based on the moment limiter (Biswas et al, 1994) and the WENO limiter (Qui and Shu, 2005b). Recently, the moment limiter has been rigorously extended to 2D problems with high computational expense (Krivodonova, 2007). A major advantage of this limiter is that it only needs information from the nearest neighbors of the element which is to be limited. The 1D WENO limiter can be extended to 2D problems in a tensor-product form as demonstrated in Levy et al (2007).

However, recently a compact limiter based on the Hermite WENO (or H-WENO) method has been proposed by Qui and Shu (2005a). This new limiter has been successfully implemented in applications involving system of conservation laws (Balsara et al, 2007; Luo et al, 2007). The H-WENO limiter not only exploits the cell-averages but also the readily available derivative information (high-order moments) from the nearest neighboring cells. This enables the WENO reconstruction process to rely on narrow stencils, and as a result the limiter is computationally attractive. However, for the 2D case we only consider the moment limiter combined with a positivity-preserving slope limiter.

9.4.2.1 A Limiter for the DG P^2 Method

We consider a third-order (P^2) modal DG scheme with the expansion (9.37) employing the basis set $\mathscr{B} = \{1, \xi, \eta, \xi\eta, (3\xi^2-1)/2, (3\eta^2-1)/2\}$. The approximate solution $U_{ij}(\xi,\eta)$ corresponding to element Ω_{ij} is then given as

$$U_{ij}(\xi,\eta) = U_{ij}^{0,0} + U_{ij}^{1,0}\,\xi + U_{ij}^{0,1}\,\eta + U_{ij}^{1,1}\,\xi\,\eta + U_{ij}^{2,0}\,(3\xi^2-1)/2 + U_{ij}^{0,2}\,(3\eta^2-1)/2, \tag{9.64}$$

where the coefficients $U_{ij}^{\ell,m}$ correspond to the moments (9.38), and $U_{ij}^{0,0}$ is the average value over Ω_{ij}. In the tensor-product expansion (9.37) for the P^2 case, the basis set employs additional basis functions $P_2(\xi)P_1(\eta)$, $P_1(\xi)P_2(\eta)$ and $P_2(\xi)P_2(\eta)$ in $\mathscr{B}$. However, for the sake of simplicity we exclude additional basis functions in (9.64).

The moment limiter (9.63) introduced for the 1D case can be extended for the 2D case (Biswas et al, 1994; Krivodonova, 2007). We denote the limited coefficients in (9.64) as $\tilde{U}_{ij}^{\ell,m}$ which are modified by a generalized version of the minmod limiter (9.63):

$$
\begin{aligned}
\tilde{U}_{ij}^{2,0} &= \text{minmod}\left[U_{ij}^{2,0}, \alpha_l(U_{ij}^{1,0} - U_{i-1,j}^{1,0}), \alpha_l(U_{i+1,j}^{1,0} - U_{ij}^{1,0})\right], \\
\tilde{U}_{ij}^{0,2} &= \text{minmod}\left[U_{ij}^{0,2}, \alpha_l(U_{ij}^{0,1} - U_{i,j-1}^{0,1}), \alpha_l(U_{i,j+1}^{0,1} - U_{ij}^{0,1})\right], \\
\tilde{U}_{ij}^{1,1} &= \text{minmod}\Big[U_{ij}^{1,1}, \alpha_l(U_{ij}^{1,0} - U_{i,j-1}^{1,0}), \alpha_l(U_{i,j+1}^{1,0} - U_{ij}^{1,0}), \\
&\qquad\qquad \alpha_l(U_{ij}^{0,1} - U_{i-1,j}^{0,1}), \alpha_l(U_{i+1,j}^{0,1} - U_{ij}^{0,1})\Big], \qquad (9.65) \\
\tilde{U}_{ij}^{1,0} &= \text{minmod}\left[U_{ij}^{1,0}, \alpha_l(U_{ij}^{0,0} - U_{i-1,j}^{0,0}), \alpha_l(U_{i+1,j}^{0,0} - U_{ij}^{0,0})\right], \\
\tilde{U}_{ij}^{0,1} &= \text{minmod}\left[U_{ij}^{0,1}, \alpha_l(U_{ij}^{0,0} - U_{i,j-1}^{0,0}), \alpha_l(U_{i,j+1}^{0,0} - U_{ij}^{0,0})\right],
\end{aligned}
$$

where α_l is a parameter in $[0,1]$ which controls the effect (dissipation) of limiting. Smaller α_l values reduce the effect of limiting. Note that the minmod function used in (9.65) has five arguments, but it acts as the standard minmod function defined in (9.56): it returns the minimum of the absolute value of arguments if all of the arguments have the same sign, otherwise it returns zero. The limiting algorithm for (9.64) can be summarized in the following steps.

- If $\tilde{U}_{ij}^{2,0} = U_{ij}^{2,0}$ and $\tilde{U}_{ij}^{0,2} = U_{ij}^{0,2}$ then there is no need for limiting and the limiting process can be stopped. If not, replace the coefficients $U_{ij}^{2,0}$ and $U_{ij}^{0,2}$ by the corresponding limited coefficients and move to the next step.
- If $\tilde{U}_{ij}^{1,1} = U_{ij}^{1,1}$ then stop limiting, otherwise replace the coefficient $U_{ij}^{1,1}$ by the limited coefficient $\tilde{U}_{ij}^{1,1}$ and move to the last step.
- If $\tilde{U}_{ij}^{1,0} = U_{ij}^{1,0}$ and $\tilde{U}_{ij}^{0,1} = U_{ij}^{0,1}$ then stop limiting. If not, replace the coefficient by the corresponding limited coefficients (i.e., slopes).

Limiting an element Ω_{ij} using the above algorithm requires information from the nearest-neighboring four elements ($\Omega_{i\pm1,j}$, $\Omega_{i,j\pm1}$). The most influential factor controlling the quality of the limited solution is the set of coefficients corresponding to the slopes ($U_{ij}^{0,1}$ and $U_{ij}^{1,0}$) used in the last step. As shown in the 1D case, excessive use of the minmod slope limiter (MUSCL) makes the solution very dissipative. For the moment limiter described above, the limiting hierarchy starts with the highest order coefficients and prevents excessive slope limiting at the last step. We also examine a positivity-preserving limiter (which is less restrictive than the minmod slope limiter) in the following section as an alternative to the slope limiter at the last step of the limiting algorithm.

9.4.2.2 A Positivity-Preserving Slope Limiter

A positivity-preserving (PP) scheme guarantees that the cell-averages which evolve in time will lie in a certain range governed by the initial conditions. Although the solution may contain minor oscillations within this range, it is less dissipative than the rigorous monotonic case. Recently, Zhang and Shu (2010) introduced a uniformly high-order accurate PP scheme for the compressible Euler equations. This

scheme avoids creating negative pressure and density in the solution at a reasonable computational cost.

The PP solution is acceptable for practical applications such as atmospheric tracer transport modeling, where positivity preservation is a highly desirable property. The minmod limiter introduced in the MUSCL scheme is strictly monotonic; unfortunately, it is very diffusive too. However, the 2D PP limiter introduced by Suresh (2000) for FV methods is less restrictive than the basic minmod limiter (9.56), and, unlike the modified Minmod limiter (9.57), does not have a problem-dependent parameter.

We adapt this PP limiter as a replacement for the minmod slope limiter used in the above-mentioned limiting process for the coefficients $U_{ij}^{0,1}$ and $U_{ij}^{1,0}$. The 2D PP limiter requires information from the nearest neighbors as well as the corner elements ($\Omega_{i\pm1,j\pm1}$), which create a 3×3 halo region with Ω_{ij} at the center. The average value of the solution on Ω_{ij} is denoted $\overline{U}_{ij}$. To understand how the PP scheme works we use the linear part of (9.64), which can be written as

$$U_{ij}(\xi,\eta) = \overline{U}_{ij} + U_{ij}^{1,0}\,\xi + U_{ij}^{0,1}\,\eta. \tag{9.66}$$

In order to advance in time, the MUSCL scheme requires a reconstruction step (9.66) which involves computing new slopes $U_{ij}^{1,0}$ and $U_{ij}^{0,1}$ from the neighboring cell averages $\overline{U}_{i\pm1,j}$ and $\overline{U}_{i,j\pm1}$. The minmod slope limiter is constrained in such a way that the U_{ij} in (9.66) lie in the range of $\overline{U}_{ij}$ and four independent cell averages $\overline{U}_{i\pm1,j}$ and $\overline{U}_{i,j\pm1}$. The PP limiter essentially extends this range by adding the corner cell-averages. The slopes are then restricted so that the reconstructed values at the corner points also lie within the new ranges (Suresh, 2000). The modification of the slopes is done so that they continuously depend on the neighboring data. The following procedure briefly outlines the process of modifying the slopes.

We first construct a 3×3 matrix $\mathbf{D}^{(\varepsilon)}$ consisting of the differences between the averages of each element in the halo region and the average value of the element Ω_{ij} that require limiting:

$$\mathbf{D}^{(\varepsilon)} = \begin{bmatrix} \overline{U}_{i-1,j+1}-\overline{U}_{ij} & \overline{U}_{i,j+1}-\overline{U}_{ij} & \overline{U}_{i+1,j+1}-\overline{U}_{ij} \\ \overline{U}_{i-1,j}-\overline{U}_{ij} & \varepsilon & \overline{U}_{i+1,j}-\overline{U}_{ij} \\ \overline{U}_{i-1,j-1}-\overline{U}_{ij} & \overline{U}_{i,j-1}-\overline{U}_{ij} & \overline{U}_{i+1,j-1}-\overline{U}_{ij} \end{bmatrix}, \tag{9.67}$$

where ε is a small positive number ($O(10^{-20})$) in order to make the algorithm robust. The extreme values of $\mathbf{D}^{(\varepsilon)}$ are computed as

$$V_{\min} = \min[\mathbf{D}^{(-\varepsilon)}], \quad V_{\max} = \max[\mathbf{D}^{(+\varepsilon)}].$$

The corner values of the reconstructed solution (9.66) can be effectively bounded within in the interval $[V_{\min}, V_{\max}]$ by restricting the slopes $|U_{ij}^{0,1}| + |U_{ij}^{1,0}|$. In other words we rescale the slopes using the ratio

$$V_s = \frac{\min(|V_{\min}|, |V_{\max}|)}{|U_{ij}^{0,1}| + |U_{ij}^{1,0}|}.$$

The final PP limited slopes are given by

$$\tilde{U}_{ij}^{0,1} = \min(1, V_s) U_{ij}^{0,1}, \quad \tilde{U}_{ij}^{1,0} = \min(1, V_s) U_{ij}^{1,0} \tag{9.68}$$

The modified slopes in (9.68) may be used as a substitute for the slopes computed by the minmod in the moment limiter.

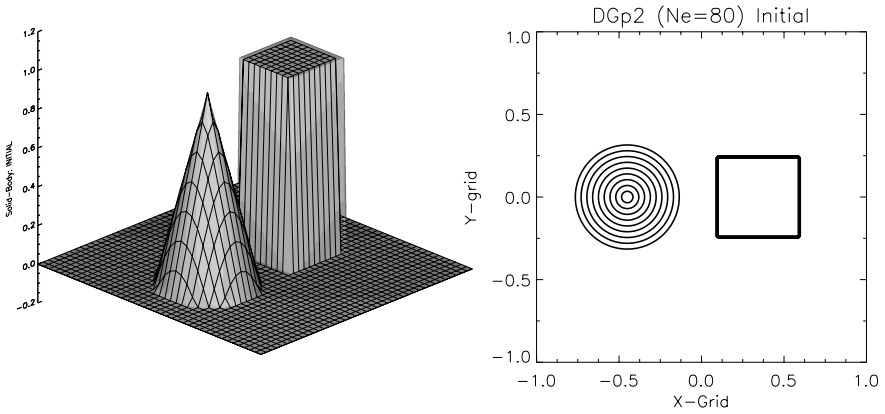

Fig. 9.18 Initial conditions for the solid-body rotation test. The initial scalar field consists of a quasi-smooth cone and a non-smooth square block whose height range form 0 to 1. The domain is $[-1,1]^2$ with 80 elements in each direction. Only one value per element is sampled in the plots for clarity.

9.4.2.3 2D Numerical Experiments

To test the limiter we use the 2D advection problem (9.30) for a solid-body rotation test. The test consists of quasi-continuous data and provides an excellent test for the monotonicity of the advecting field (LeVeque, 2002; Cheruvu et al, 2007). The velocity field is given by $(u,v) = (y,-x)$ on a square domain D where $x,y \in [-1,1]$, and the initial condition is defined in a piecewise fashion: $U(x,y,t=0) = U_0 = 0$ except in a square region where $U_0 = 1$ and a circular region where U_0 is cone-shaped, growing to the maximum value 1 at the center. Formally,

$$U_0(x,y) = \begin{cases} 1 & \text{if } 0.1 < x < 0.6 \quad \text{and} \quad -0.25 < y < 0.25, \\ 1 - \rho_c/0.35 & \text{if } \rho_c = \sqrt{(x+0.45)^2 + y^2} < 0.35, \\ 0 & \text{otherwise.} \end{cases} \tag{9.69}$$

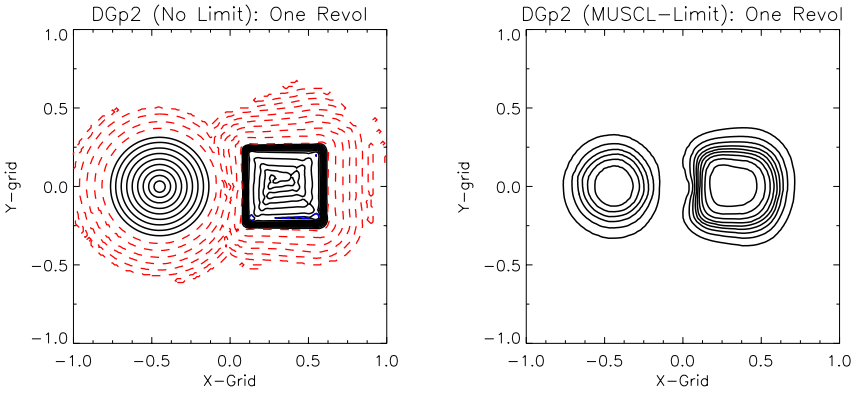

Fig. 9.19 Numerical solution with a third-order DG scheme after one revolution. The left panel shows the solution without limiting where the dashed lines correspond to the zero-contours, indicating spurious undershoots. The right panel shows limited monotonic solution with a moment limiter, where a MUSCL type minmod limiter is employed for limiting the coefficients $U^{0,1}$ and $U^{1,0}$ corresponding to the slopes.

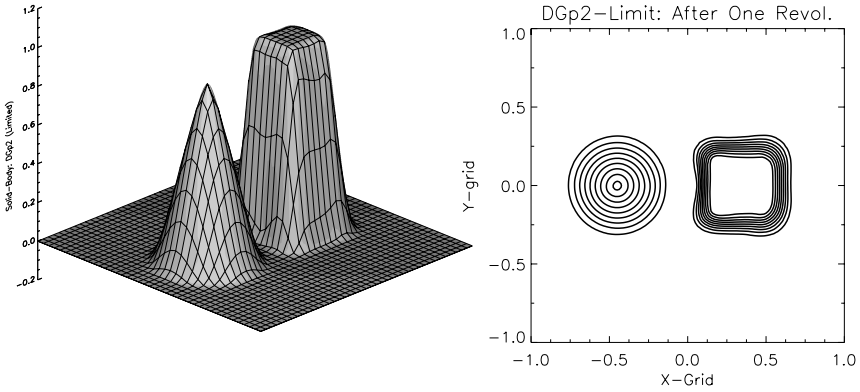

Fig. 9.20 Numerical solution with a third-order DG scheme combined with the moment limiter after one revolution. The coefficients $U^{0,1}$ and $U^{1,0}$ (corresponding to the slopes) are limited using the positivity-preserving limiter.

The initial conditions are shown in Fig. 9.18. The domain consists of 80^2 elements and the time step is $\Delta t = 2\pi/1000$ so 1000 iterations are required for one complete revolution.

Figure 9.19 shows the solution after one revolution with and without the moment limiter. The left panel of Fig.9.19 shows the DG P^2 numerical solution without any limiting, and the dashed lines indicate oscillations. The right panel shows the limited solution with the moment limiter where the slopes $U_{ij}^{0,1}$ and $U_{ij}^{1,0}$ are limited with a minmod limiter. The solution is very diffusive, the cone height has been reduced to about 60% of its initial height, and the square-block has been smoothly deformed. In Fig. 9.20 the numerical solution with the moment limiter combined with the PP slope limiter is shown. The PP limiter (9.68) is only used as a substitute for the minmod limiter in the last step of the limiting algorithm. There is a significant improvement in the solution as compared to Fig. 9.19: the cone and square-block both preserve their maximum height, although the numerical solution still suffers from slight diffusion.

9.5 The DG Methods on the Sphere

There are several geometrical options for discretizing a sphere for global modeling. The choice of a particular spherical grid system is based on various factors including the numerical method being considered (Williamson, 2007). For element-based Galerkin approaches such as the spectral element or DG method, the cubed-sphere geometry provides an excellent choice. The cubed-sphere topology introduced by Sadourny (1972) consists of a rectangular (quasi-uniform) tiling of the sphere $\mathscr{S}$, representing the planet Earth, which facilitates an efficient implementation of the DG method on the sphere. As an application of the DG method on the sphere, we consider the global shallow water model as reviewed below.

9.5.1 The Shallow Water Model on the Sphere

The shallow water (SW) equations are a system of hyperbolic PDEs. They are widely used for studying horizontal aspects of atmospheric dynamics (Vallis, 2006), and also serve as a testbed to evaluate various discretization techniques (Williamson et al, 1992). The flux-form (or conservative form) SW equations on a rotating sphere can be written as

$$\frac{\partial h\mathbf{v}}{\partial t} + \nabla\cdot(\mathbf{v}h\mathbf{v}) = -f\hat{\mathbf{k}}\times h\mathbf{v} - gh\nabla(h+h_s) \tag{9.70}$$

$$\frac{\partial h}{\partial t} + \nabla\cdot(h\mathbf{v}) = 0 \tag{9.71}$$

Here, h is the depth of the fluid above the solid surface and is related to the free surface geopotential height (above sea level) $\Phi = g(h_s + h)$, where h_s denotes the height of the underlying topography and g is the gravitational acceleration. $\mathbf{v}$ is the horizontal wind vector, f is the Coriolis parameter, and $\hat{\mathbf{k}}$ is the unit vector along the outward radial direction. The 2D divergence ($\nabla\cdot$) and gradient (∇) operators are general and not specific to a particular spherical grid system. Note that $\mathbf{v}h\mathbf{v}$ is a *dyadic* (or a second-order tensor) term and can also be written in the tensor-product notation $h\mathbf{v}\otimes\mathbf{v}$. Although (9.70) is widely used in computational fluid dynamics, for meteorological modeling application a simplified version of the momentum equations, the so-called "vector invariant form" is popular and is given by (Sadourny, 1972; Arakawa and Lamb, 1977),

$$\frac{\partial \mathbf{v}}{\partial t} + \nabla(\Phi + \frac{1}{2}\mathbf{v}\cdot\mathbf{v}) = -(\zeta + f)\hat{\mathbf{k}}\times\mathbf{v}, \tag{9.72}$$

where $\zeta = \hat{\mathbf{k}}\cdot(\nabla\times\mathbf{v})$ is the relative vorticity. The vector invariant form (9.72), as the name suggests, preserves its formal form under coordinate transformations. In a rigorous sense (9.72) is not in momentum conserving form, and when combined with (9.71) it leads to a weakly hyperbolic SW system (Toro, 2001). Nevertheless, (9.72) is still in flux-form, although the fluxes being addressed are the energy fluxes $\Phi + \mathbf{v}\cdot\mathbf{v}/2$, rather than the momentum fluxes $h\mathbf{v}$ as used in (9.70).

9.5.2 The Cubed-Sphere Geometry

Here we consider the cubed-sphere geometry employing the equiangular central (gnomonic) projection as described in Nair et al (2005b). The physical domain $\mathscr{S}$ is partitioned into six identical regions (sub-domains), which are obtained by the central projection of the faces of the inscribed cube onto the surface of $\mathscr{S}$, (see Fig. 9.21(a)). Each of the local coordinate systems is free of singularities, employs identical metric terms, and creates a non-orthogonal curvilinear coordinate system on $\mathscr{S}$. However, the edges of the six faces are discontinuous.

Because of the non-orthogonal nature of the grid system on $\mathscr{S}$, a tensorial form is convenient for describing the local vectors and the fluid motion in general. In order to be consistent with tensor notations, we choose (x^1, x^2) as the independent variables, which are the central angles of the gnomonic projection (Nair et al, 2005b). Thus the local coordinates for each face are $x^1 = x^1(\lambda,\theta)$, $x^2 = x^2(\lambda,\theta)$ such that $x^1, x^2 \in [-\pi/4, \pi/4]$, where λ and θ are the longitude and latitude, respectively, of a sphere with radius R. The metric tensor, G_{ij}, associated with the transformation is

$$G_{ij} = \frac{R^2}{\rho^4 \cos^2 x^1 \cos^2 x^2} \begin{bmatrix} 1+\tan^2 x^1 & -\tan x^1 \tan x^2 \\ -\tan x^1 \tan x^2 & 1+\tan^2 x^2 \end{bmatrix}, \tag{9.73}$$

where $i, j \in \{1,2\}$ and $\rho^2 = 1 + \tan^2 x^1 + \tan^2 x^2$. The Jacobian of the transformation (the metric or curvature term) is $\sqrt{G} = [\det(G_{ij})]^{1/2}$, which is identical for each face

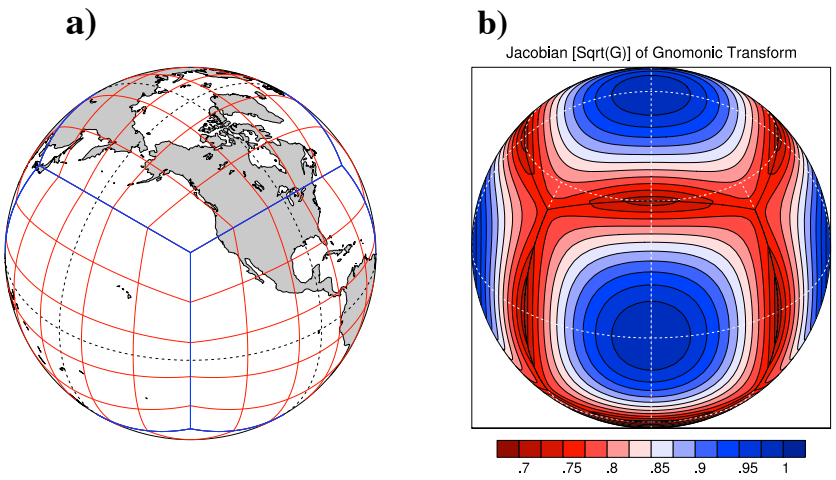

Fig. 9.21 a) A cubed-sphere with 5×5 elements on each face, so 150 elements span the entire surface of the sphere. b) The Jacobian $\sqrt{G}$ (also refered to as the metric or curvature term) associated with the gnomonic transformation from a cube onto a sphere. For a unit sphere $\sqrt{G}$ has a maximum value of 1 at the center of each face, and has a minimum value $1/\sqrt{2}$ at the center of the edges. The cubed-sphere gridlines are great-circle arcs and they are orthogonal only at the center of each panel.

of the cubed-sphere. For a unit sphere the curvature term has a maximum value of 1 at the center of each panel and a minimum of $1/\sqrt{2}$ at the center of the edges (see Fig. 9.21(b)). Although the cells are uniform on the cube, the quadrilateral cell on the sphere is most deformed at the corners of the cubed-sphere and the ratio between the maximum and minimum grid width for the gnomonic cubed-sphere has an upper bound approximately 1.3 at any resolution (Rančić et al, 1996).

9.5.3 The Shallow Water Model on the Cubed-Sphere

On the cubed-sphere the SW equations are treated in tensor form with covariant (u_1, u_2) and contravariant (u^1, u^2) wind vectors. These vectors are related through the matrix equations:

$$\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{bmatrix} \begin{bmatrix} u^1 \\ u^2 \end{bmatrix}, \quad \begin{bmatrix} u^1 \\ u^2 \end{bmatrix} = \begin{bmatrix} G^{11} & G^{12} \\ G^{21} & G^{22} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}, \tag{9.74}$$

where $G^{ij} = G_{ij}^{-1}$ and can be computed from (9.73).

The orthogonal components of the spherical wind vector $\mathbf{v}(\lambda,\theta) = (u,v)$ — i.e., the physical zonal and meridional components of the horizontal wind — can be expressed in terms of contravariant vectors (u^1,u^2) as follows:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \mathbf{A} \begin{bmatrix} u^1 \\ u^2 \end{bmatrix}, \mathbf{A} = \begin{bmatrix} R\cos\theta\,\partial\lambda/\partial x^1 & R\cos\theta\,\partial\lambda/\partial x^2 \\ R\,\partial\theta/\partial x^1 & R\,\partial\theta/\partial x^2 \end{bmatrix}; \mathbf{A}^T\mathbf{A} = G_{ij}. \quad (9.75)$$

The details of the local transformation laws and the transformation matrix $\mathbf{A}$ for each face of the cubed-sphere can be found in Nair et al (2005b).

The SW equations of a thin layer of fluid in 2D are the horizontal momentum equations and the continuity equation for the height h. The momentum equations are cast in terms of covariant (u_1,u_2) vectors, which leads to a flux-form formulation suitable for methods based on hyperbolic conservation laws (Nair et al, 2005a). Note that this particular formulation preserves the vector invariant form of momentum equations (9.72). Thus the prognostic variables are u_1, u_2 and h, and the shallow water equations on $\mathscr{S}$ can be written in a compact form following the inviscid formulation described in Nair (2009):

$$\frac{\partial}{\partial t}\mathbf{U} + \frac{\partial}{\partial x^1}\mathbf{F}_1(\mathbf{U}) + \frac{\partial}{\partial x^2}\mathbf{F}_2(\mathbf{U}) = \mathbf{S}(\mathbf{U}), \quad (9.76)$$

where the state vector $\mathbf{U}$ and the flux vectors $\mathbf{F}_1, \mathbf{F}_2$ are defined by

$$\mathbf{U} = \left[u_1, u_2, \sqrt{G}h\right]^T, \quad \mathbf{F}_1 = \left[E, 0, \sqrt{G}hu^1\right]^T, \quad \mathbf{F}_2 = \left[0, E, \sqrt{G}hu^2\right]^T,$$

and $E = \Phi + \frac{1}{2}(u_1u^1 + u_2u^2)$ is the energy term. The divergence δ and relative vorticity ζ on $\mathscr{S}$ are defined as

$$\delta = \frac{1}{\sqrt{G}}\left[\frac{\partial\sqrt{G}u^1}{\partial x^1} + \frac{\partial\sqrt{G}u^2}{\partial x^2}\right], \quad \zeta = \frac{1}{\sqrt{G}}\left[\frac{\partial u_2}{\partial x^1} - \frac{\partial u_1}{\partial x^2}\right] \quad (9.77)$$

The source term, $\mathbf{S}$, is a function of the relative vorticity ζ, the Coriolis parameter f, and the contravariant wind vector (u^1,u^2), and is defined as

$$\mathbf{S}(\mathbf{U}) = \left[\sqrt{G}u^2(f+\zeta), -\sqrt{G}u^1(f+\zeta), 0\right]^T.$$

9.5.4 The Computational Domain

The spherical SW equations can be discretized either in physical space or in the computational (transformed) space. Since the SW equations (9.76) are already in the computational (x^1,x^2) space (due to the central projection), it makes sense to discretize the system in the same space. The computational domain may be considered as the surface of a logical cube $\mathscr{C}$ such that each face of $\mathscr{C}$ is defined in

terms of local orthogonal Cartesian coordinates $x^1, x^2 \in [-\pi/4, \pi/4]$, as shown in Fig. 9.22 . Thus $\mathscr{C}$ is essentially a union of six non-overlapping sub-domains (faces) and any point on $\mathscr{C}$ can be uniquely represented by the ordered triple (x^1, x^2, ν) where $\nu = 1, \ldots, 6$, is the cube-face or panel index. The projections and the logical orientation of the cube panels are described in Nair et al (2005b) and Lauritzen et al (2010).

The equiangular central projection results in a uniform element width ($\Delta x^1 = \Delta x^2$) on $\mathscr{C}$, which is an advantage for practical implementation. Figure 9.22 provides a schematic diagram of the mapping between the physical domain $\mathscr{S}$ (cubed-sphere) and the computational domain $\mathscr{C}$ (cube).

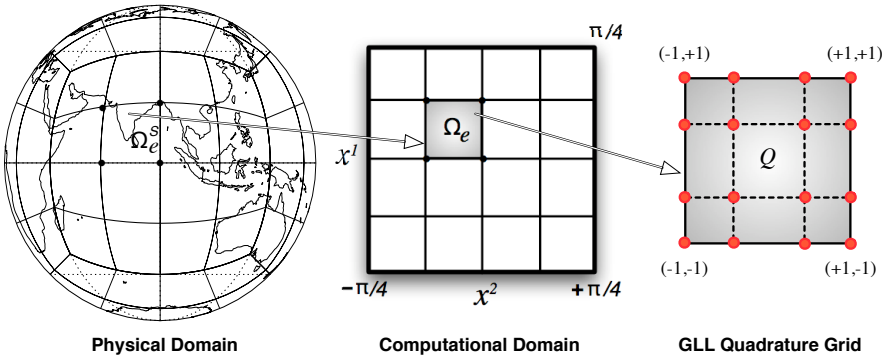

Fig. 9.22 A schematic diagram showing the mapping between each spherical tile (element) Ω_e^S of the physical domain (cubed-sphere) $\mathscr{S}$ onto a planar element Ω_e on the computational domain $\mathscr{C}$ (cube). For a DG discretization each element on the cube is further mapped onto a unique reference element Q, which is defined by the Gauss-Lobatto-Legendre (GLL) quadrature points. The horizontal discretization of the HOMME dynamical cores relies on this grid system.

The cubed-sphere has the attractive feature that the domain $\mathscr{S}$ is naturally decomposed into non-overlapping quadrilateral elements (tiles) Ω_e^S. This topology is well-suited for high-order element-based methods such as spectral element or DG methods, and amenable to efficient parallel implementation. Each face of the cubed-sphere has $N_e \times N_e$ elements, thus $N_{elm} = 6N_e^2$ elements span the entire spherical domain such that $\mathscr{S} = \cup_{e=1}^{N_{elm}} \Omega_e^S$; in Fig. 9.22 N_e is 4. There exists a one-to-one correspondence between the spherical element Ω_e^S on $\mathscr{S}$ and the planar element Ω_e on $\mathscr{C}$ as depicted in Fig. 9.22. The element-wise continuous mapping allows us to perform integrations on the sphere in a mapped (local) Cartesian geometry rather than on the surface of the sphere. The High-Order Method Modeling Environment (HOMME) developed at NCAR relies on this grid system (Dennis et al, 2005).

9.5.5 The DG Discretization of the SW Equations

The SW model developed in Nair et al (2005a) is based on a modal DG discretization, however, here we consider the nodal inviscid version of the SW model as implemented in HOMME (Nair, 2009). The discretization process for a multidimensional system of equations (9.76) is quite similar to the 2D case considered in Section 9.3. However, as we discuss in Section 9.5.5.1, the flux operations (Riemann solvers) along the cubed-sphere edges are not trivial to implement. For notational simplicity, we consider a generic component of the system (9.76) as follows,

$$\frac{\partial \psi}{\partial t} + \nabla_c \cdot \mathbf{F}(\psi) = S(\psi), \quad \text{in} \quad \mathscr{C} \times (0,T], \tag{9.78}$$

where $\mathbf{F} = (F_1, F_2)$ is the flux function and T is the prescribed time of integration. The Cartesian gradient operator ∇_c on $\mathscr{C}$ is defined to be

$$\nabla_c \equiv \left(\frac{\partial}{\partial x^1}, \frac{\partial}{\partial x^2} \right) \quad \Rightarrow \quad \nabla_c \cdot \mathbf{F} = \frac{\partial F_1}{\partial x^1} + \frac{\partial F_2}{\partial x^2}$$

For example, (9.78) may be considered the continuity equation (or the flux-form transport equation) for the SW system (9.76); in this case $\psi = \sqrt{G}h, \mathbf{F} = (\psi u^1, \psi u^2)$ and the source term is $S = 0$. Similarly, the components of the momentum equation in (9.76) can be cast in the Cartesian form (9.78).

Analogous to the 2D case considered earlier, the weak Galerkin form corresponding to (9.78) on any element Ω_e with boundary Γ_e on $\mathscr{C}$ can be written as follows:

$$\begin{aligned} \frac{d}{dt} \int_{\Omega_e} \psi_h \varphi_h \, d\Omega - \int_{\Omega_e} \mathbf{F}(\psi_h) \cdot \nabla_c \varphi_h \, d\Omega + \int_{\Gamma_e} \hat{\mathbf{F}} \cdot \mathbf{n} \, \varphi_h \, d\Gamma \\ = \int_{\Omega_e} S(\psi_h) \, \varphi_h \, d\Omega, \end{aligned} \tag{9.79}$$

where ψ_h is the approximate solution and φ_h is a test function in $\mathscr{V}_h$. $\hat{\mathbf{F}}$ is the numerical flux, $\mathbf{n}$ is the outward-facing unit normal vector on the element boundary Γ_e and the element of integration is $d\Omega = dx^1 dx^2$. For the numerical flux we employ the local Lax-Friedrichs flux formula as follows:

$$\hat{\mathbf{F}}(\psi_h) = \frac{1}{2} \left[(\mathbf{F}(\psi_h^-) + \mathbf{F}(\psi_h^+)) - \alpha^i_{\max} (\psi_h^+ - \psi_h^-) \right], \tag{9.80}$$

where $\alpha^i_{\max}$ is the absolute maximum of the eigenvalues of the flux Jacobian; ψ_h^- and ψ_h^+, respectively, are the left and right limits of ψ_h along the boundary Γ_e.

Recall that for each component of the system (9.76) the weak formulation (9.79) is valid, however, $\alpha^i_{\max}$ must be computed for the entire system. Nair et al (2005a) derived the flux Jacobian for the SW system on the cubed-sphere, which is a 3×3 matrix, and its maximum eigenvalues along the x^1 and x^2-directions are,

$$\alpha_1 = |u^1| + \sqrt{G^{11}\Phi}, \quad \alpha_2 = |u^2| + \sqrt{G^{22}\Phi}. \tag{9.81}$$

These values are nothing but the maximum phase speed of the SW system in the curvilinear coordinate directions. From (9.81) the local maximum values computed from both sides along the element wall (Γ_e), are $\alpha^1_{\max} = \max(\alpha^-_1, \alpha^+_1)$ and $\alpha^2_{\max} = \max(\alpha^-_2, \alpha^+_2)$, as required in (9.80).

9.5.5.1 Flux Exchanges at the Cubed-Sphere Edges

For DG methods, the flux exchanges at the element edges are managed by the numerical flux formulas such as (9.5), and this is the only mechanism by which the adjacent elements *communicate*. Because local coordinates are discontinuous at the cubed-sphere edges, the flux exchange across the edges require special attention. The local transformation of vectors using (9.75) at the cubed-sphere edges can be used for exchanging vector quantities including fluxes. For example, consider a point on the cubed-sphere edge separated by two neighboring faces 'm' and 'n'. The local vector on the point $(u^1, u^2)_m$ belonging to a face m can be transformed into the global spherical components $(u, v)_s$ using (9.75), and then transformed back to the local vector $(u^1, u^2)_n$ of the adjacent edges on the face n.

The flux operations on the cubed-sphere edges also follow a similar procedure. To compute the flux on an edge (or interface) using (9.5), both the left, $\mathbf{F}^-$, and the right, $\mathbf{F}^+$, contributions of $\mathbf{F} = (F_{u^1}, F_{u^2})$ are required. For instance, if $\mathbf{F}^-$ on the panel m is available then the corresponding $\mathbf{F}^+$ belongs to the adjacent panel n, and can be transformed in terms of the local vectors in the panel m by employing the following dual transformation,

$$\begin{bmatrix} F_{u^1} \\ F_{u^2} \end{bmatrix}^+_m = \mathbf{A}^{-1}_m \mathbf{A}_n \begin{bmatrix} F_{u^1} \\ F_{u^2} \end{bmatrix}^+_n, \tag{9.82}$$

where the suffixes m, n indicate the adjacent panel indices such that $m, n \in \{1, 2, \ldots, 6\}$. $\mathbf{A}_m$, $\mathbf{A}_n$ are transformation matrices defined in (9.75), and for the sake of computational efficiency the dual transformation matrices $\mathbf{A}^{-1}_m \mathbf{A}_n$ in (9.82) as well as the metric terms can be pre-computed.

9.5.5.2 Numerical Integration of the SW Model

The integral and the differential operators required in the DG discretization (9.79) of the SW system can be approximated on each Ω_e with boundary Γ_e. The element-wise discretization is quite similar to the 2D case considered earlier, therefore, we just outline the procedure in terms of the weak form (9.79) and the SW system (9.76).

Here we adopt the nodal basis set used for the HOMME dynamical core (Nair, 2009). In order to take advantage of efficient quadrature rules, new independent variables $\xi^i = \xi^i(x^i)$, $i \in \{1, 2\}$ are introduced such that $\xi^i \in [-1, 1]$. This leads to a mapping of each element $\Omega_e \in \mathscr{C}$ to a unique reference element $Q = [-1, 1] \otimes$

$[-1,1]$, as illustrated schematically in Fig. 9.22. The nodal basis functions are the Lagrange polynomials $h_\ell(\xi^i)$, with roots at the GLL quadrature points. The nodal basis set is chosen to be a tensor-product of polynomials $h_k(\xi^1)h_\ell(\xi^2)$. Now the approximate solution ψ_h and test function φ_h in $\mathscr{V}_h$ can be expanded in terms of a tensor-product of the Lagrange basis functions, and, in the case of ψ_h, such that

$$\psi_h(\xi^1,\xi^2) = \sum_{k=0}^{N}\sum_{\ell=0}^{N} \psi_h(\xi_k^1,\xi_\ell^2)\, h_k(\xi^1)h_\ell(\xi^2), \tag{9.83}$$

where $\{\xi_\ell^i\}_{\ell=0}^N$ are the GLL quadrature points on the reference element Q. In other words, there are $N_v \times N_v$ GLL points on Q (where $N_v = N+1$), therefore the total degrees of freedom on $\mathscr{C}$ is $6N_e^2N_v^2$. The equivalent resolution of the cubed-sphere with respect to the regular latitude-longitude sphere at the equator is approximately $90^\circ/(N_e \times N)$. However, a latitude-longitude spherical grid with the same resolution at the equator will have approximately 30% more grid points. For the sake of computational efficiency we use the same order GLL quadrature rule for the internal integrals in Ω_e and the boundary flux integrals along Γ_e, at the cost of nominal loss of accuracy due to inexact integration (see Section 9.3.1.3).

Substitution of the expansion (9.83) for ψ_h and φ_h in the weak formulations and further simplification leads to a system of ODEs in time corresponding to the continuous problem (9.76),

$$\frac{d\mathbf{U}}{dt} = L(\mathbf{U}) \text{ in } (0,T], \tag{9.84}$$

where $\mathbf{U}$ are the time dependent nodal gridpoint values for the SW system (9.76). In the present study we use the third-order accurate explicit strong stability-preserving (SSP) Runge-Kutta as discussed in Section 9.2.5.

9.5.6 Numerical Experiments

Discussion of the solutions to the SW equations on the cubed-sphere based on the DG method with the Williamson et al (1992) test suite can be found in Nair et al (2005a,b) or, with a viscous SW model, in Nair (2009). In this section we consider a new deformational test and the barotropic instability test case proposed by Galewsky et al (2004).

9.5.6.1 Advection Test

The flux-form advection equation (9.71) on the cubed-sphere can be written as

$$\frac{\partial}{\partial t}\sqrt{G}\phi + \frac{\partial}{\partial x^1}(\sqrt{G}\phi u^1) + \frac{\partial}{\partial x^2}(\sqrt{G}\phi u^2) = 0, \tag{9.85}$$

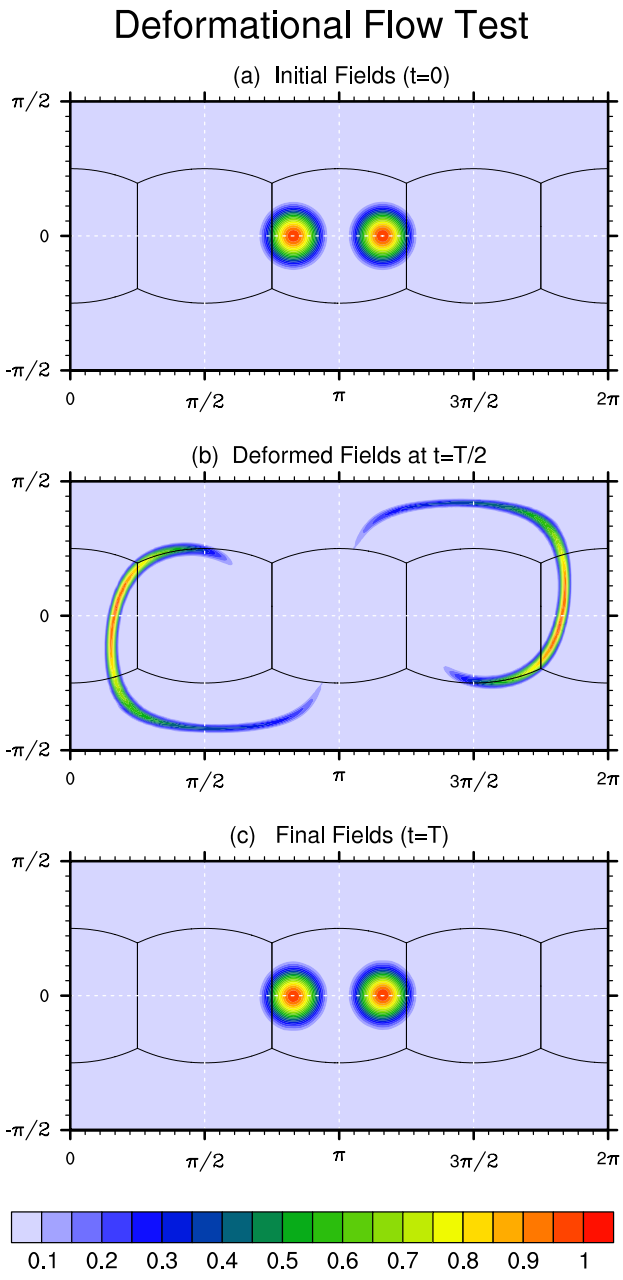

Fig. 9.23 Deformational flow test with the DG transport scheme on the cubed-sphere. The equivalent resolution at the equator (with $N_e = 20$) is approximately 1.5°. a) The initial positions of the scalar field (cosine bells) centered at $(\lambda_i, \theta_i) = (5\pi/6, 0)$ and $(7\pi/6, 0)$. b) Deformed scalar fields at half-time $(t = T/2)$ of the simulation. c) The scalar fields (numerical solution) return back to the initial positions at the final time $(t = T)$.

where ϕ is the scalar field and the advecting wind is given by the contravariant vector field (u^1, u^2). In fact, this is the continuity equation in the SW system (9.76). If we introduce $\psi = \sqrt{G}\phi$ and the fluxes $F_1 = \psi u^1$ and $F_2 = \psi u^2$ then (9.85) can be written in a form analogous to the 2D Cartesian case.

9.5.6.2 Deformational Flow Test

We consider a new deformational flow test introduced in Nair and Lauritzen (2010). For this problem, the initial distributions undergo severe deformation for a prescribed time and then the flow reverses its course, returning the deforming fields to their initial states (the "boomerang effect"). A special feature of this test is that the trajectories of the flow are non-trivial (not along a circle or straight line) and consequently the deformation is severe, making the test very challenging.

This test is prescribed on a unit sphere and quasi-smooth cosine-bell patterns (a C^1 function) are used as the initial scalar fields. Two symmetrically located cosine bells are defined by

$$\phi(\lambda,\theta) = \frac{1}{2}[1+\cos(\pi r_i/r)] \quad \text{if} \quad r_i < r, \tag{9.86}$$

where $r = 1/2$ is base radius of the bells, $r_i = r_i(\lambda,\theta)$ is the great-circle distance between (λ,θ) and a specified center (λ_i,θ_i) of the cosine bell, which is given by

$$r_i(\lambda,\theta) = \cos^{-1}[\sin\theta_i \sin\theta + \cos\theta_i \cos\theta \cos(\lambda-\lambda_i)]. \tag{9.87}$$

The scalar values are initially set to zero ($\phi(\lambda,\theta) = 0$), and then two cosine bells (cones) are generated using (9.86) at known points $(\lambda_1,\theta_1) = (5\pi/6, 0)$ and $(\lambda_2,\theta_2) = (7\pi/6, 0)$ as the bell centers. The flow field is non-divergent and the time dependent velocity fields $\mathbf{v}(\lambda,\theta,t)$ are prescribed in longitude-latitude coordinates,

$$u(\lambda,\theta,t) = \kappa \sin^2(\lambda)\sin(2\theta)\cos(\pi t/T) \tag{9.88}$$

$$v(\lambda,\theta,t) = \kappa \sin(2\lambda)\cos(\theta)\cos(\pi t/T), \tag{9.89}$$

where the parameter $\kappa = 2$ and the final time of the simulation is $T = 5$ non-dimensional.

The DG transport scheme employs a 4×4 GLL grid with $N_e = 20$. This corresponds to an approximate resolution of 1.5° at the equator. The third-order SSP RK scheme (9.29) is used with a time step $\Delta t = 5/1200$ for the simulations (1200 time steps are required for the total simulation). Figure 9.23 shows the initial conditions and simulated results for the deformational test with the DG scheme. The cosine bells move away from the initial positions (Fig. 9.23a) and deform into thin spiral shapes at time $t = T/2$ (Fig. 9.23b). The trajectories for the non-divergent flow are complex and the cosine bells pass along the edges and corners, covering the six faces of the cubed-sphere. The DG scheme successfully simulates the deformations and retains the initial position as well as shape of the distribution at the end

of the simulation ($t = T$), as shown in Fig. 9.23c. Since the final solution is identical to the initial conditions by design, the global standard errors norms l_1, l_2 and l_∞ (Williamson et al, 1992) can be computed.

9.5.6.3 Solid-Body Rotation Test

The cosine-bell problem proposed by Williamson et al (1992) is widely used to test advection schemes on the sphere. The same test has been considered in Nair et al (2005b) for verifying the accuracy and conservation properties of the DG schemes as well as the accuracy of various central projections for the cubed-sphere system. Here we employ this test to demonstrate the effectiveness of the monotonic limiter designed for the DG P^2 transport scheme in Section 9.4.2. The initial scalar field is a cosine bell defined as follows,

$$\phi(\lambda,\theta) = \begin{cases} (h_0/2)\,[1+\cos(\pi r/r_0)] & \text{if} \quad r < r_0 \\ 0 & \text{if} \quad r \geq r_0, \end{cases} \tag{9.90}$$

where r is the great-circle distance between (λ,θ) and the bell center $(3\pi/2,0)$ as given in (9.87). The cosine-bell radius is $r_0 = R/3$ and the maximum height of the bell is $h_0 = 1000$ m, where $R = 6.37122 \times 10^6$ m is the Earth's radius. The velocity components of the advecting wind field are

$$\begin{aligned} u &= u_0\,(\cos\alpha_0\cos\theta + \sin\alpha_0\cos\lambda\sin\theta), \\ v &= -u_0\sin\alpha_0\sin\lambda, \end{aligned}$$

where $u_0 = 2\pi R/(12\text{ days})$, and α_0 is the flow orientation parameter which controls the direction of the flow on the sphere along a great-circle trajectory. When the value of α_0 is equal to zero or $\pi/2$, the flow direction is along the equator or in the north-south (meridional) direction, respectively. For the cubed-sphere, flow along the north-east direction ($\alpha_0 = \pi/4$) is more challenging because the cosine-bell pattern passes over four vertices and two edges of the cube during a complete revolution (in a 12-day period). The exact solution for $\phi(\lambda,\theta)$ is known for this test and is equal to the initial value. Ideally, after a complete revolution the cosine-bell pattern should return to the initial position without incurring any deformation.

The DG P^2 scheme with $N_e = 45$ is used for the numerical simulation, this corresponds to 1° resolution (approximately) at the equator. The second-order SSP RK scheme (9.28) is applied for 1600 time steps to complete one revolution. Figure 9.24 shows the numerical solution (left panel) and the limited solution (right panel). As expected the non-limited solution is oscillatory, however, oscillations are confined to a smaller region around the cosine-bell. The monotonic limiter removes spurious oscillations but slightly deforms the shape of the bell. The additional computational expense required for the limiter is nominal, for the cosine-bell advection test it is found to be less than 5%.

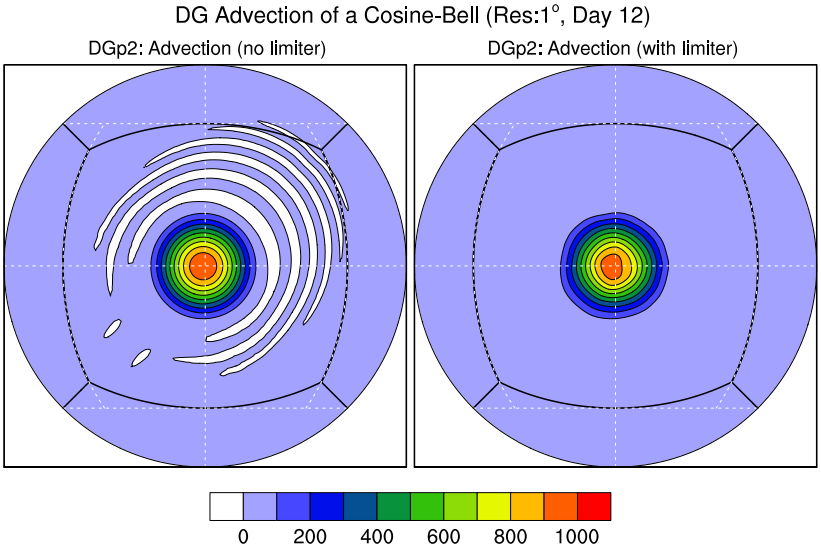

Fig. 9.24 The cosine-bell advection test on the sphere. The left panel shows the DG (P^2) numerical solution after a complete revolution along northeast direction, where spurious oscillations in the solution can be seen for the zero contour value. The right panel shows the limited solution by applying a monotonic limiter that completely removes the oscillations.

9.5.6.4 Barotropic Instability Test

The barotropic instability test proposed by Galewsky et al (2004) is an interesting test for the SW models developed on the cubed-sphere grids. The test describes the evolution of a barotropic wave in the northern hemisphere and exhibits continuous nonlinear transfer of energy at the midlatitudes from large to small scales. The test is particularly challenging on the cubed-sphere because the vigorous barotropic instability activities are located at the discontinuous edges of the top panel of the cubed-sphere grid. This test *exposes* artifacts from wave number 4 due to the cube-edge discontinuities at low resolutions for various SW models (St-Cyr et al, 2008; Chen and Xiao, 2008; Levy, 2009).

The initial conditions are zonally symmetric, and nearly in balance but physically unstable. This introduces a strong zonal jet along the midlatitudes; details can be found in Galewsky et al (2004). The test recommends a simulation time of 6 days with and without diffusion. Fine features of the vorticity fields can be captured at a resolution of about 1.25° or higher (St-Cyr et al, 2008), and the DG results agree with this observation. Figure 9.25 shows a high-resolution DG simulation of relative vorticity (ζ) at days 4 and 6, respectively. The approximate equatorial resolution is 0.64° $(N_e = 20, N_v = 8)$ and a time step $\Delta t = 6$s is used for these simulations. The fine features of the vortex are well captured by the DG SW model and comparable to

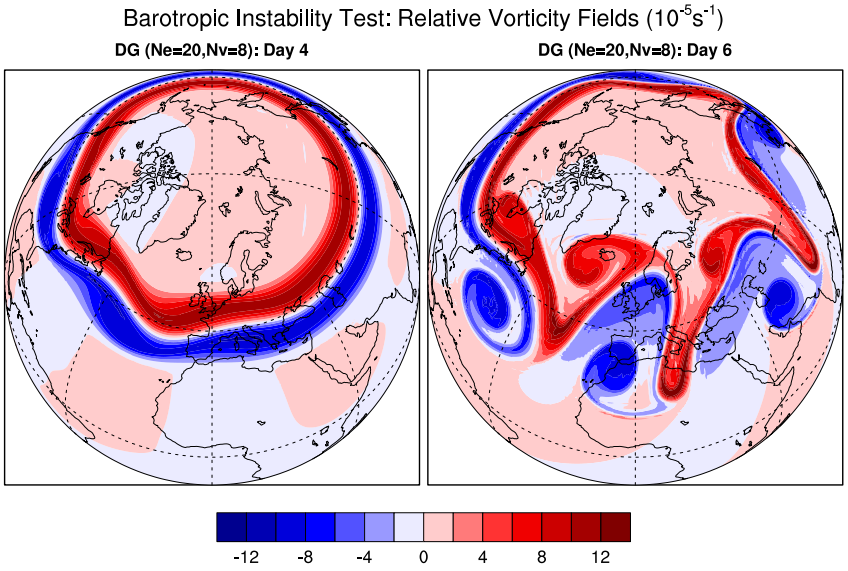

Fig. 9.25 The simulated relative vorticity fields (ζ) for the barotropic instability test at a high-resolution. The left panel shows ζ at day 4 and the right panel, ζ at day 6.

the reference solution given in Galewsky et al (2004). Small-scale noise at the sharp gradients in Fig. 9.25 can be effectively controlled by using a diffusion scheme.

We briefly outline the diffusion process as used for DG methods in the context of the barotropic vorticity evolution. Diffusion and dissipation mechanisms are inevitable for practical atmospheric models. For example, momentum diffusion transfers energy from the resolved scales into the unresolved scales. However, in a discrete climate model, diffusion tries to mimic the effects of unresolved scales on the resolved fluid flow (chapter 13). Moreover, the diffusion process prevents spurious accumulation of energy and enstrophy at the model grid scale. The DG method is amenable to efficient implementation of robust diffusion schemes. This is based on the so-called Local DG or LDG method by Cockburn and Shu (1998), which is a generalization of the explicit diffusion scheme proposed originally by Bassi and Rebay (1997). Recently, Nair (2009) developed a second-order LDG diffusion scheme for the viscous SW model on the cubed-sphere. The vorticity evolution results shown in Nair (2009) confirm that the LDG based diffusion mechanism removes small-scale noise such that the solution converges monotonically to a diffused state. The convergence is dependent on the coefficient of diffusion.

9.6 Concluding Remarks

The DG method combined with explicit strong stability-preserving Runge-Kutta time-stepping is particularly attractive for wave propagation problems because of the ability to use local high-order polynomial approximations for the solution, providing an efficient way to control phase and dissipation errors. The DG method is becoming popular in geophysical fluid dynamics modeling, with several efforts to develop global SW models based on DG methods (Giraldo et al, 2002; Nair et al, 2005a; Läuter et al, 2008; Nair, 2009). Very recently, DG methods have been further extended to hydrostatic (Nair et al, 2009) and non-hydrostatic (Giraldo and Restelli, 2008; St-Cyr and Neckels, 2009) atmospheric models. Currently there are new efforts by various research groups to develop sophisticated DG-based atmospheric models, including some with adaptive meshes. Motivations for choosing the DG method as the primary numerical technique for these model developments are based on various factors such as the high-order accuracy, conservation, geometric flexibility and parallel efficiency. Nevertheless, there are some computational issues associated with the explicit DG discretization.

A major drawback of the DG algorithm is the severe CFL stability restriction associated with explicit time-stepping. For practical climate models and high resolution non-hydrostatic NWP models, overall computational efficiency is very much contingent on the model's ability to take larger time steps. A moderate order DG scheme employing third- or fourth-order spatial discretization (i.e., a P^2 or P^3 method) can address the stringent stability requirement to some extent. Implicit time integration approaches are also popular for DG methods in CFD applications (Diosady and Darmofal, 2009; Bassi et al, 2009). The numerical algorithms for such methods are far more complex and require considerably more computational resources than explicit schemes. If such techniques permit at least 3-fold longer time steps for unsteady problems as compared to the explicit method, then they may be worth considering for atmospheric modeling applications.

Development of efficient time integration methods for DG methods is an active area of research. The semi-implicit time integration method for a DG non-hydrostatic model introduced by Restelli and Giraldo (2009) appears to be promising. The recent novel time integration approaches such as the ADER (Arbitrary high order DERivatives) by Käser et al (2007) and IMEX (implicit explicit) RK methods by Kanevsky et al (2007) have been shown to be efficient time integration options for DG methods. These new time integration techniques could be extended to DG atmospheric models.

Acknowledgments The authors are thankful to IMAGe (NCAR) colleagues, particularly Dr. Duane Rosenberg for an internal review of the manuscript. The authors would also like to thank two anonymous reviewers for several helpful suggestions. This project is partially supported by the U.S. Department of Energy under the awards DE-FG02-07ER64464 and DE-SC0001658. The National Center for Atmospheric Research is sponsored by the National Science Foundation.

References

Arakawa A, Lamb VR (1977) Computational design of the basic dynamical process of the UCLA general circulation model. In: Chang J (ed) Methods in Computational Physics, Academic Press, pp 173–265

Atkins HL, Shu CW (1996) Quadrature-free implementation of the discontinuous Galerkin method for hyperbolic equations. In: 2nd AIAA/CEAS Aeroacoustic Conference, Paper 96-1683.

Balsara DS, Altman C, Munz CD, Dumbser M (2007) Sub-cell based indicator for troubled zones in RKDG schemes and a novel class of hybrid RKDG+HWENO schemes. J Comput Phys 226(1):586–620

Bassi F, Rebay S (1997) A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier-Stokes equations. J Comput Phys 131:267–279

Bassi F, Ghidoni A, Rebay S, Tesini P (2009) High-order accurate *p*-multigrid discontinuous Galerkin solution of the Euler equations. Int J Numer Meth Fluids 60:847–865, doi:10.1002/fld.1917

Biswas R, Devine K, Flaherty J (1994) Parallel adaptive finite-element methods for conservation laws. Appl Num Math 14:255–283

Boris JP, Book DL (1973) Flux-Corrected Transport. I. SASHSTA, a fluid transport algorithm that works. J Comput Phys 11(1):38–69

Butcher JC (2008) Numerical Methods for Ordinary Differential equations, Second edn. Wiley, ISBN 978-0-470-72335-7 463 pp.

Canuto C, Hussaini MY, Quarteroni A, Zang TA (2007) Spectral Methods: Evolution of Complex Geometries and Application to Fluid Dynamics. Springer, ISBN 978-3-540-30727-3, 596 pp.

Chen C, Xiao F (2008) Shallow water model on cubed-sphere by multi-moment finite volume method. J Comput Phys 227(10):5019–5044

Cheruvu V, Nair RD, Tufo HM (2007) A spectral finite volume transport scheme on the cubed-sphere. Appl Num Math 57:1021–1032

Cockburn B, Shu CW (1989) TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservative laws II. Math Comp 52:411–435

Cockburn B, Shu CW (1998) The local discontinuous Galerkin method for time-dependent convection-diffusion schemes. SIAM J Numer Anal 35:2440–2463

Cockburn B, Shu CW (2001) The Runge-Kutta discontinuous Galerkin method for convection-dominated problems. J Sci Computing 16:173–261

Cockburn B, Hou S, Shu CW (1990) TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws IV. Math Comp 54:545–581

Cockburn B, Johnson C, Shu CW, Tadmor E (1997) Advanced Numerical Approximation of Nonlinear Hyperbolic Equations. Springer, LNM 1697

Cockburn B, Karniadakis GE, Shu CW (2000) The development of discontinuous Galerkin methods. In: Cockburn B, Karniadakis GE, Shu CW (eds) Discontinuous Galerkin Methods: Theory, Computation, and Applications. Lecture Notes in Computational Science and Engineering, vol 11, Springer, 470 pp.

Colella P, Woodward PR (1984) The Piecewise Parabolic Method (PPM) for gas-dynamical simulations. J Comput Phys 54:174–201

Crowell S, Williams D, Marviplis C, Wicker L (2009) Comparison of traditional and novel discretization methods for advection models in numerical weather prediction. In: Lecture Notes in Computer Science, vol 5545, Springer-Verlag, pp 263–272, ICCS 2009, Part II.

DeMaria M (1985) Tropical cyclone motion in a nondivergent barotropic model. Mon Wea Rev 113:119–1210

Dennis J, Fournier A, Spotz WF, St-Cyr A, Taylor MA, Thomas SJ, Tufo H (2005) High-resolution mesh convergence properties and parallel efficiency of a spectral element atmospheric dynamical core. Int J High Perf Computing Appl 19(3):225–235

Deville MO, Fisher PF, Mund EM (2002) High-Order Methods for Incompressible Fluid Flow. Cambridge University Press, ISBN 0-521-45309-7, 499 pp.

Diosady LT, Darmofal DL (2009) Preconditioning methods for discontinuous Galerkin solutions of the compressible Navier-Stokes equations. J Comput Phys 228:3917–3835

Galewsky J, Polvani LM, Scott RK (2004) An initial-value problem to test numerical models of the shallow water equations. Tellus 56A:429–440

Ghostine R, Kessewani G, Mosé R, Vazquez J, Ghenaim A (2009) An improvement of classical slope limiters for high-order discontinuous Galerkin method. Int J Numer Meth Fluids 59:423–442

Giraldo FX, Restelli M (2008) A study of spectral element and discontinuous Galerkin methods for the Navier-Stokes equations in nonhydrostatic mesoscale atmospheric modeling: Equation sets and test cases. J Comput Phys 227:3849–3877

Giraldo FX, Hesthaven JS, Wartburton T (2002) Nodal high-order discontinous Galerkin methods for the shallow water equations. J Comput Phys 181:499–525

Godunov SK (1959) A difference method for numerical calculation of discontinuous solutions of the equations of hydrodynamics. Mat Sb 47:271–306

Gottlieb S, Shu CW, Tadmor E (2001) Strong stability preserving high-order time discretization methods. SIAM Rev 43:89–112

Harten A, Engquist B, Osher S, Chakravarthy S (1987) Uniformly high order essentially non-oscillatory schemes, III. J Comput Phys 71:231–303

Hesthaven JS, Warburton T (2008) Nodal Discontinuous Galerkin Methods: Algorithms, Analysis, and Applications. Springer, ISBN 978-0-387-72065-4, 500 pp.

Iskandarani M, Levin J, Choi BJ, Haidgovel D (2005) Comparison of advection schemes for high-order h-p finite element and finite volume methods. Ocean Modeling 10:233–252

Kanevsky A, Carpenter MH, Gottlieb D, Hasthevan JS (2007) Application of implicit-explicit high order Runge-Kutta methods to discontinuous-Galerkin schemes. J Comput Phys 225:1753–1781

Karniadakis GE, Sherwin S (2005) Spectral/hp Element Methods for Computational Fluid Dynamics. Oxford University Press, ISBN 0-19-852869-8, 657 pp.

Käser M, Dumbser M, Puente J, Igel H (2007) An arbitrary high-order discontinuous Galerkin method for elastic waves on unstructured meshes-III. Geophys J Int 168:224–242

Kopriva DA (2009) Implementing Spectral Methods for Partial Differential Equations. Springer, ISBN 978-90-481-2260-8, 394 pp.

Kopriva DA, Gassner G (2010) On the quadrature and weak form choices in collocation type discontinuous Galerkin spectral element methods. J Sci Comput 44:136–155

Krivodonova L (2007) Limiters for high-order dicontinuous Galerkin methods. J Comput Phys 226:879–896

Kubatko EJ, Bunya S, Dawson C, Westerink JJ (2009) Dynamic p-adaptive Runge–Kutta discontinuous Galerkin methods for the shallow water equations. Comput Methods Appl Mech Engrg 198:1766–1774

Lauritzen PH, Nair RD, Ullrich PA (2010) A conservative semi-Lagrangian multi-tracer transport scheme (CSLAM) on the cubed-sphere grid. J Comput Phys 229:1401–1424

Läuter M, Giraldo FX, Handorf D, Dethloff K (2008) A discontinuous Galerkin method for shallow water equations in spherical traingular coordinates. J Comput Phys 227:10,226–10,242

van Leer B (1974) Towards the ultimate conservative difference scheme. II. Monotonicity and conservation combined in a second-order scheme. J Comput Phys 14:361–370

van Leer B (1977) Towards the ultimate conservative difference scheme. IV. A new approach to numerical convection. J Comput Phys 23:276–299

Lesaint P, Raviart P (1974) Mathematical Aspects of Finite Elements in Partial Differential Equations, Academic Press, New York, chap On a finite element method for solving neutron transport eqaution, pp 89–123

LeVeque RJ (2002) Finite Volume Methods for Hyperbolic Problems. Cambridge University Press, ISBN 0-19-00924-3, 558 pp.

Levy MN (2009) A high-order element-based Galerkin method for the global shallow water equations. PhD thesis, University of Colorado at Boulder, Department of Applied Mathematics, 108 pp.

Levy MN, Nair RD, Tufo HM (2007) High-order Galerkin methods for scalable global atmospheric models. Computers & Geosciences 33(8):1022–1035

Levy MN, Nair RD, Tufo HM (2009) A high-order element-based Galerkin method for the barotropic vorticity equation. Int J Numer Meth Fluids 59(12):1369–1387

Liu XD, Osher S, Chan T (1994) Weighted essentially non-oscillatory schemes. J Comput Phys 115:200–212

Lomtev I, Kirby RM, Karniadakis GE (2000) A discontinuous Galerkin method in moving domains. In: Cockburn B, Karniadakis GE, Shu CW (eds) Discontinuous Galerkin Methods: Theory, Computation, and Applications. Lecture Notes in Computational Science and Engineering, vol 11, Springer, 470 pp.

Luo H, Baum JD, Löhner R (2007) A Hermite WENO-based limiter for discontinuous Galerkin method on unstructured grids. J Comput Phys 225(1):686–713

Lynch P (2008) The origins of computer weather prediction and climate modeling. J Comput Phys 227:3431–3444

Nair RD (2009) Diffusion experiments with a global discontinuous Galerkin shallow-water model. Mon Wea Rev 137:3339–3350

Nair RD, Lauritzen PH (2010) A class of deformational flow test cases for linear transport problems on the sphere. J Comput Phys 229:8868–8887

Nair RD, Thomas SJ, Loft RD (2005a) A discontinuous Galerkin global shallow water model. Mon Wea Rev 133:876–888

Nair RD, Thomas SJ, Loft RD (2005b) A discontinuous Galerkin transport scheme on the cubed-sphere. Mon Wea Rev 133:814–828

Nair RD, Choi HW, Tufo HM (2009) Computational aspects of a scalable high-order discontinuous Galerkin atmospheric dynamical core. Computers and Fluids 38:309–319

Prather MJ (1986) Numerical advection by conservation of second-order moments. J Geophys Res 91:6671–6681

Qiu J, Khoo BC, Shu CW (2006) A numerical study for the performance of the Runge-Kutta discontinuous Galerkin method based on different numerical fluxes. J Comput Phys 212(2):540–565

Qui J, Shu CW (2005a) Hermite WENO schemes and their application as limiters for Runge-Kutta discontinuous Galerkin methods II: Two dimesional case. Computers & Fluids 34:642–663

Qui J, Shu CW (2005b) Runge-Kutta discontinuous Galerkin methods using WENO limiters. SIAM J Sci Computing 26:907–927

Rančić M, Purser R, Mesinger F (1996) A global shallow water model using an expanded spherical cube. Q J R Meteorol Soc 122:959–982

Reed WH, Hill TR (1973) Triangular mesh method for neutron transport equation. Technical Report LA-UR-73-479, Los Alamos Scientific Laboratory

Remacle JF, Flaherty JE, Sheppard MS (2003) An adaptive discontinuous Galerkin technique with an orthogonal basis applied to compressible flow problems. SIAM Review 45:53–72

Restelli M, Giraldo FX (2009) A conservative discontinuous Galerkin semi-implicit formulation for the Navier-Stokes equations in nonhydrostatic mesoscale modeling. SIAM J Sci Comput 31:2231–2257

Rivière B (2008) Discontinuous Galerkin Method for Solving Elliptic and Parabolic Equations: Theory and Implementation. SIAM, ISBN 978-0-898716-56-6, 187 pp.

Sadourny R (1972) Conservative finite-difference approximations of the primitive equations on quasi-uniform spherical grids. Mon Wea Rev 100:136–144

Shu CW (1997) Advanced Numerical Approximation of Nonlinear Hyperbolic Equations, Springer, chap Essentially non-oscillatory and weighted essentially non-oscillatory schemes for conservation laws, pp 324–432. LNM 1697

Simmons AJ, Burridge DM (1981) An energy and angular-momentum conserving vertical finite-difference scheme and hybrid vertical coordinates. Mon Wea Rev 109:758–766

Smolarkiewicz PK (1982) The multi-dimensional Crowley advection scheme. Mon Wea Rev 110:1968–1983

Smolarkiewicz PK (1984) A fully muliti-dimensional positive definite advection transport algorithm with small implicit diffusion. J Comput Phys 54:325–362

St-Cyr A, Neckels D (2009) A fully implicit Jacobian-free high-order discontinuous Galerkin mesoscale flow solver. In: Lecture Notes in Computer Science, vol 5545, Springer-Verlag, pp 243–252, ICCS 2009, Part II.

St-Cyr A, Jablonowski C, Dennis JM, Tufo HM, Thomas SJ (2008) A comparison of two shallow water models with nonconforming advaptive grids. Mon Wea Rev 136:1898–1922

Staniforth A, Coté J, Pudickiewicz J (1987) Comments on "Smolarkiewicz's deformational flow". Mon Wea Rev 115:894–900

Suresh A (2000) Positivity preserving schemes in multidimensions. SIAM J Sci Comput 22:1184–1198

Toro EF (1999) Riemann Solvers and Numerical Methods for Fluid Dynamics. A Practical Introduction (2nd Ed.). Springer-Verlag, New York

Toro EF (2001) Shock-Capturing Methods for Free-Surface Shallow Flows. John Wiley & Sons, England, ISBN 0-471-98766-2, 305 pp.

Vallis GK (2006) Atmopsheric and Ocenaic Fluid Dynamics. Cambridge University Press, ISBN 978-0-521-84969-2, 745 pp.

Williamson DL (2007) The evolution of dynamical cores for global atmospheric models. J Meteo Soc of Japan 85:241–269

Williamson DL, Drake JB, Hack JJ, Jakob R, Swarztrauber PN (1992) A standard test set for numerical approximations to the shallow water equations in spherical geometry. J Comput Phys 102:211–224

Zhang X, Shu CW (2010) On positivity-preserving high-order discontinuous Galerkin schemes for compressible Euler equations on rectangular meshes. J Comput Phys 229(23):8918–8934, doi:10.1016/j.jcp.2010.08.016