COMMUNICATIONS IN COMPUTATIONAL PHYSICS
Vol. **2**, No. 2, pp. 310-333

Commun. Comput. Phys.
April 2007

High-Order Multidomain Spectral Difference Method for the Navier-Stokes Equations on Unstructured Hexahedral Grids

Yuzhi Sun[1], Z. J. Wang[1,*] and Yen Liu[2]

[1] *Department of Aerospace Engineering, Iowa State University, 2271 Howe Hall, Ames, IA 50011, USA.*
[2] *NASA Ames Research Center, Mail Stop T27B-1, Moffett Field, CA 94035, USA.*

Received 15 May 2006; Accepted (in revised version) 9 August 2006

Available online 30 September 2006

Abstract. A high order multidomain spectral difference method has been developed for the three dimensional Navier-Stokes equations on unstructured hexahedral grids. The method is easy to implement since it involves one-dimensional operations only, and does not involve surface or volume integrals. Universal reconstructions are obtained by distributing solution and flux points in a geometrically similar manner in a unit cube. The concepts of the Riemann solver and high-order local representations are applied to achieve conservation and high order accuracy. In this paper, accuracy studies are performed to numerically verify the order of accuracy using flow problems with analytical solutions. High order of accuracy and spectral convergence are obtained for the propagation of an isotropic vortex and Couette flow. The capability of the method for both inviscid and viscous flow problems with curved boundaries is also demonstrated.

AMS subject classfications: 65M70, 76M20, 76M22

Key words: High order, unstructured grids, spectral difference, Navier-Stokes.

1 Introduction

It is well known that the computation of the aerodynamic flow field around a helicopter is a considerable challenge [10] because of the following difficulties: the strong interaction between the moving blades, vortices and wakes; the disparate length scales in the flow turbulence under flight conditions; the complex and moving geometries. Although low order (first and second order) finite volume methods have become the main choice of

*Corresponding author. *Email addresses:* sunyuzhi@iastate.edu (Y. Sun), zjw@iastate.edu (Z. J. Wang), Yen.Liu@nasa.gov (Y. Liu)

many commercial Computational Fluid Dynamics (CFD) codes, and proven successful in tackling a wide variety of flow problems [5,11,24] in engineering design, many in the CFD community believe vortex-dominated flows require high-order methods. This is mainly because lower order methods usually dissipate propagating vortices too quickly. For example, to study the classical blade-vortex interaction problem, it is estimated that the blade vortex should be well preserved for several revolutions. However, low order methods may completely "consume" the vortex after one or two revolutions, therefore making the computation results very inaccurate.

Many high-order methods (order > 2) have been developed in CFD for a wide range of applications, such as large eddy simulation, direct numerical simulation, computational aeroacoustics, etc. Most high-order methods were developed for structured grids, e.g., ENO/WENO methods [29], compact methods [18,34,35], optimized methods [32]. In particular, high-order compact methods have been successfully employed to tackle vortex dominated problems, including vortex breakdown over a delta wing [35]. For complex configurations, it is often very time-consuming to generate smooth structured grids required by these high-order methods. Our focus in this study is therefore on high-order methods for unstructured grids, and in particular unstructured hexahedral grids.

There have been intensive research efforts on high-order methods for flow simulation on unstructured grids in the last two decades. An incomplete list of notable examples includes the spectral element method [25], multi-domain spectral method [16,17], k-exact finite volume method [3], WENO methods [13], discontinuous Galerkin method [4,7,8], high-order residual distribution methods [1], spectral volume (SV) [21,31,36,37] and spectral difference (SD) methods [14,19,20,23,38]. Among those methods, some are based on the weighted residual form of the governing equations, for instance the discontinuous Galerkin (DG) method. Some are based on the integral form of the governing equations, e.g., the k-exact finite volume method and the SV methods. Others, such as staggered grid multi-domain spectral method and SD method are based on the differential form.

When selecting a method to implement for three-dimensional problems, the cost and the complexity of the method is often an important factor. It is obvious that methods based on the differential form are the easiest to implement since they do not involve surface or volume integrals. This is particularly true when high-order curved boundaries need to be dealt with. Based on our experiences with the DG, SV and SD methods on 2D triangular meshes, the SD method seems the easiest to implement and most efficient for the 2D Euler equations. Therefore in the present study, the SD method is selected to solve the 3D Navier-Stokes equations on unstructured hexahedral grids. The use of hexahedral grids is again a compromise between flexibility and efficiency. Although tetrahedral grids are easier to generate for complex 3D configurations, hexahedral grids have been shown to possess higher efficiency and accuracy for viscous boundary layers [22].

The SD method and the staggered-grid multi-domain spectral method on hexahedral grids actually converge to the same method. The solution unknowns or degrees-of-freedom (DOFs) are the conserved variables at the Gauss points, while fluxes are evaluated at Lobatto points to generate the flux derivatives to update the DOFs. On a hexahedral element, all the operations can be performed in a one-dimensional manner, resulting in higher efficiency and less implementation cost. As in the Godunov-type finite volume method [12], Riemann solvers [26, 27] are used at element interfaces to couple the discontinuous elements together, and provide the necessary numerical dissipation, which makes the method conservative and stable.

The paper is organized as follows. In the next section, the formulation of the 3D SD method is described for a hexahedral element. Numerical results including accuracy studies for both inviscid and viscous flows are presented in Section 3, together with results for several often used demonstration cases. Conclusions and possible future work are outlined in Section 4.

2 Formulation of the 3D spectral difference method on hexahedral grids

2.1 Governing equation

Consider the unsteady compressible 3D Navier-Stokes equations in conservative form written as

$$\frac{\partial Q}{\partial t}+\frac{\partial F}{\partial x}+\frac{\partial G}{\partial y}+\frac{\partial H}{\partial z}=0 \tag{2.1}$$

where Q is the vector of conserved variables, and F, G, H are the total fluxes including both the inviscid and viscous flux vectors, i.e., $F=F^i-F^v$, $G=G^i-G^v$, $H=H^i-H^v$, which take the following form

$$\begin{aligned}
&Q=\left\{\begin{array}{c}\rho\\ \rho u\\ \rho v\\ \rho w\\ E\end{array}\right\}, \quad F^i=\left\{\begin{array}{c}\rho u\\ p+\rho u^2\\ \rho uv\\ \rho uw\\ u(E+p)\end{array}\right\},\\
&G^i=\left\{\begin{array}{c}\rho v\\ \rho uv\\ p+\rho v^2\\ \rho vw\\ v(E+p)\end{array}\right\}, \quad H^i=\left\{\begin{array}{c}\rho w\\ \rho uw\\ \rho vw\\ p+\rho w^2\\ w(E+p)\end{array}\right\},
\end{aligned} \tag{2.2}$$

and

$$F^v=\left\{\begin{array}{c}0\\ \tau_{xx}\\ \tau_{yx}\\ \tau_{zx}\\ u\tau_{xx}+v\tau_{yx}+w\tau_{zx}+\frac{\mu C_p}{P_r}T_x\end{array}\right\},$$

$$G^v=\left\{\begin{array}{c}0\\ \tau_{xy}\\ \tau_{yy}\\ \tau_{zy}\\ u\tau_{xy}+v\tau_{yy}+w\tau_{zy}+\frac{\mu C_p}{P_r}T_y\end{array}\right\}, \tag{2.3}$$

$$H^v=\left\{\begin{array}{c}0\\ \tau_{xz}\\ \tau_{yz}\\ \tau_{zz}\\ u\tau_{xz}+v\tau_{yz}+w\tau_{zz}+\frac{\mu C_p}{P_r}T_z\end{array}\right\}.$$

In (2.1)-(2.3), ρ is the density, u, v and w are the velocity components in x, y and z directions, p is the pressure, and E is the total energy, μ is dynamic viscosity, C_p is the specific heat at constant pressure, P_r is the Prandle number, and T is the temperature. For a perfect gas, the pressure is related to the total energy by

$$E=\frac{p}{\gamma-1}+\frac{1}{2}\rho(u^2+v^2+w^2), \tag{2.4}$$

with a constant ratio of specific heats γ, which is taken to be 1.4 for air. The stress tensor in (2.3) takes the following form

$$\begin{aligned}
&\tau_{xx}=2\mu\left(u_x-\frac{u_x+v_y+w_z}{3}\right), \qquad &&\tau_{yy}=2\mu\left(v_y-\frac{u_x+v_y+w_z}{3}\right),\\
&\tau_{zz}=2\mu\left(w_z-\frac{u_x+v_y+w_z}{3}\right), &&\tau_{xy}=\tau_{yx}=\mu(v_x+u_y),\\
&\tau_{yz}=\tau_{zy}=\mu(w_y+v_z), &&\tau_{zx}=\tau_{xz}=\mu(u_z+w_x).
\end{aligned} \tag{2.5}$$

2.2 Coordinate transformation

We employ non-overlapping unstructured hexahedral cells or elements to fill the computational domain. The use of hexahedral cells for viscous boundary layers is preferred over tetrahedral cells because of the efficiency and accuracy. In order to handle curved boundaries, both linear and quadratic isoparametric elements are employed, with linear elements used in the interior domain and quadratic elements near high-order curved boundaries. To achieve an efficient implementation, all elements are transformed from

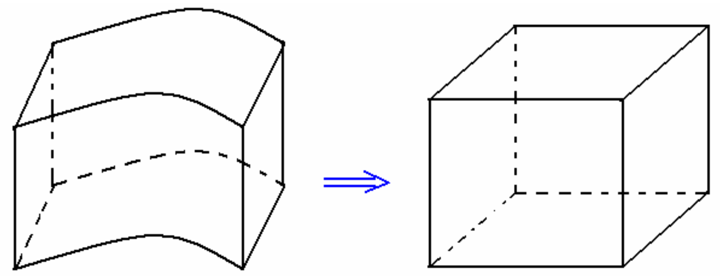


Figure 1: Transformation from a physical element to a standard element.

the physical domain (x,y,z) into a standard cubic element $(\xi,\eta,\varsigma)\in[0,1]\times[0,1]\times[0,1]$ as shown in Fig. 1. The transformation can be written as

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \sum_{i=1}^{K} M_i(\xi,\eta,\varsigma) \begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix}, \tag{2.6}$$

where K is the number of points used to define the physical element, (x_i,y_i,z_i) are the Cartesian coordinates of those points, and $M_i(\xi,\eta,\varsigma)$ are the shape functions. For the transformation given in (2.6), the Jacobian matrix J takes the following form

$$J=\frac{\partial(x,y,z)}{\partial(\xi,\eta,\zeta)}=\begin{bmatrix} x_\xi & x_\eta & x_\zeta \\ y_\xi & y_\eta & y_\zeta \\ z_\xi & z_\eta & z_\zeta \end{bmatrix}.$$

For a non-singular transformation, its inverse transformation must also exist, and the Jocobian matrices are related to each other according to

$$\frac{\partial(\xi,\eta,\zeta)}{\partial(x,y,z)}=\begin{bmatrix} \xi_x & \xi_y & \xi_z \\ \eta_x & \eta_y & \eta_z \\ \zeta_x & \zeta_y & \zeta_z \end{bmatrix}=J^{-1}.$$

Therefore the metrics can be computed according to

$$\begin{aligned} &\xi_x=(y_\eta z_\zeta-y_\zeta z_\eta)/|J|, \quad \xi_y=(x_\zeta z_\eta-x_\eta z_\zeta)/|J|, \quad \xi_z=(x_\eta y_\zeta-x_\zeta y_\eta)/|J|, \\ &\eta_x=(y_\zeta z_\xi-y_\xi z_\zeta)/|J|, \quad \eta_y=(x_\xi z_\zeta-x_\zeta z_\xi)/|J|, \quad \eta_z=(x_\zeta y_\xi-x_\xi y_\zeta)/|J|, \\ &\zeta_x=(y_\xi z_\eta-y_\eta z_\xi)/|J|, \quad \zeta_y=(x_\eta z_\xi-x_\xi z_\eta)/|J|, \quad \zeta_z=(x_\xi y_\eta-x_\eta y_\xi)/|J|. \end{aligned}$$

The governing equations in the physical domain are then transformed into the computational domain (standard element), and the transformed equations take the following form

$$\frac{\partial\tilde{Q}}{\partial t}+\frac{\partial\tilde{F}}{\partial\xi}+\frac{\partial\tilde{G}}{\partial\eta}+\frac{\partial\tilde{H}}{\partial\varsigma}=0, \tag{2.7}$$

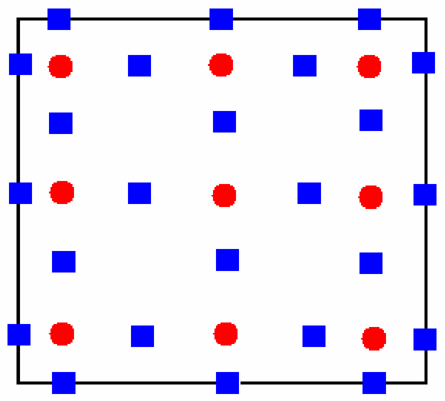


Figure 2: Distribution of solution points (circles) and flux points (squares) in a standard element for a third-order SD scheme.

where

$$\tilde{Q}=|J|\cdot Q, \quad \tilde{F}=\tilde{F}^i-\tilde{F}^v,$$
$$\tilde{G}=\tilde{G}^i-\tilde{G}^v, \quad \tilde{H}=\tilde{H}^i-\tilde{H}^v,$$

$$\begin{bmatrix}\tilde{F}^i\\ \tilde{G}^i\\ \tilde{H}^i\end{bmatrix}=|J|\begin{bmatrix}\xi_x & \xi_y & \xi_z\\ \eta_x & \eta_y & \eta_z\\ \zeta_x & \zeta_y & \zeta_z\end{bmatrix}\cdot\begin{bmatrix}F^i\\ G^i\\ H^i\end{bmatrix}, \quad \begin{bmatrix}\tilde{F}^v\\ \tilde{G}^v\\ \tilde{H}^v\end{bmatrix}=|J|\begin{bmatrix}\xi_x & \xi_y & \xi_z\\ \eta_x & \eta_y & \eta_z\\ \zeta_x & \zeta_y & \zeta_z\end{bmatrix}\cdot\begin{bmatrix}F^v\\ G^v\\ H^v\end{bmatrix}.$$

Let $\vec{S}_\xi=|J|(\xi_x,\xi_y,\xi_z),\vec{S}_\eta=|J|(\eta_x,\eta_y,\eta_z), \vec{S}_\zeta=|J|(\zeta_x,\zeta_y,\zeta_z)$. Then we have $\tilde{F}=\vec{f}\bullet\vec{S}_\xi$, $\tilde{G}=\vec{f}\bullet\vec{S}_\eta$, $\tilde{H}=\vec{f}\bullet\vec{S}_\zeta$, with $\vec{f}=(F,G,H)$. In our implementation, $|J|$ is stored at the solution points, while $\vec{S}_\xi$, $\vec{S}_\eta$, $\vec{S}_\zeta$ are stored at flux points to minimize memory usage.

2.3 Space discretization

In the standard element, two sets of points are defined, namely the solution points and the flux points, illustrated in Fig. 2 for a 2D element. The solution unknowns or degrees-of-freedom (DOFs) are the conserved variables at the solution points, while fluxes are computed at the flux points. In order to construct a degree $(N-1)$ polynomial in each coordinate direction, solutions at N points are required. The solution points in 1D are chosen to be the Gauss points defined by

$$X_s=\frac{1}{2}\left[1-\cos\left(\frac{2s-1}{2N}\cdot\pi\right)\right], \quad s=1,2,\cdots,N. \tag{2.8}$$

The flux points are selected to be the Gauss-Lobatto points given by

$$X_{s+1/2}=\frac{1}{2}\left[1-\cos\left(\frac{s}{N}\cdot\pi\right)\right], \quad s=0,1,\cdots,N. \tag{2.9}$$

Using the N solutions at the solution points, a degree $N-1$ polynomial can be built using the following Lagrange basis defined as

$$h_i(X)=\prod_{s=1,s\neq i}^{N}\left(\frac{X-X_s}{X_i-X_s}\right). \tag{2.10}$$

Similarly, using the $N+1$ fluxes at the flux points, a degree N polynomial can be built for the flux using a similar Lagrange basis defined as

$$l_{i+1/2}(X)=\prod_{s=0,s\neq i}^{N}\left(\frac{X-X_{s+1/2}}{X_{i+1/2}-X_{s+1/2}}\right). \tag{2.11}$$

The reconstructed solution for the conserved variables in the standard element is just the tensor products of the three one-dimensional polynomials, i.e.,

$$Q(\xi,\eta,\varsigma)=\sum_{k=1}^{N}\sum_{j=1}^{N}\sum_{i=1}^{N}\frac{\tilde{Q}_{i,j,k}}{|J_{i,j,k}|}h_i(\xi)\cdot h_j(\eta)\cdot h_k(\varsigma). \tag{2.12}$$

Similarly, the reconstructed flux polynomials take the following form:

$$\begin{aligned}
\tilde{F}(\xi,\eta,\varsigma)&=\sum_{k=1}^{N}\sum_{j=1}^{N}\sum_{i=0}^{N}\tilde{F}_{i+1/2,j,k}l_{i+1/2}(\xi)\cdot h_j(\eta)\cdot h_k(\varsigma),\\
\tilde{G}(\xi,\eta,\varsigma)&=\sum_{k=1}^{N}\sum_{j=0}^{N}\sum_{i=1}^{N}\tilde{G}_{i,j+1/2,k}h_i(\xi)\cdot l_{j+1/2}(\eta)\cdot h_k(\varsigma),\\
\tilde{H}(\xi,\eta,\varsigma)&=\sum_{k=0}^{N}\sum_{j=1}^{N}\sum_{i=1}^{N}\tilde{H}_{i,j,k+1/2}h_i(\xi)\cdot h_j(\eta)\cdot l_{k+1/2}(\varsigma).
\end{aligned} \tag{2.13}$$

The reconstructed fluxes are only element-wise continuous, but discontinuous across cell interfaces. For the inviscid flux, a Riemann solver, such as the Rusanov [27] or Roe flux [26], is employed to compute a common flux at interfaces to ensure conservation and stability. In summary, the algorithm to compute the inviscid flux derivatives consists of the following steps:

1. Given the conserved variables at the solution points $\{\tilde{Q}_{i,j,k}\}$, compute the conserved variables at the flux points $\{Q_{i+1/2,j,k},\ Q_{i,j+1/2,k},\ Q_{i,j,k+1/2}\}$ using (2.12) (Note that $h_m(X_n)=\delta_{mn}$);
2. Compute the inviscid fluxes at the interior flux points using the solutions computed at Step 1, i.e., $\{\tilde{F}^i_{i+1/2,j,k},\ i=1,\cdots,N-1\}$, $\{\tilde{G}^i_{i,j+1/2,k},\ j=1,\cdots,N-1\}$, $\{\tilde{H}^i_{i,j,k+1/2},\ k=1,\cdots,N-1\}$;

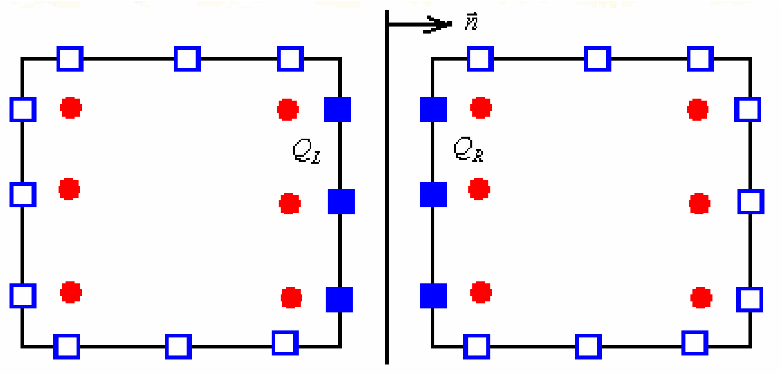


Figure 3: The interface between two elements and the locations of solution (circles) and flux (solid squares) points.

3. Compute the inviscid flux at element interfaces using a Riemann solver, such as the Rusanov solver, in terms of the left and right conserved variables of the interface (as shown in Fig. 3). Given the normal direction of the interface $\vec{n}$, and the averaged normal velocity component $\bar{V}_n$ and sound speed $\bar{c}$, the Rusanov flux on the interface is computed with

$$\begin{aligned}
\tilde{F}^i &= \frac{1}{2}(\tilde{F}^i_L + \tilde{F}^i_R - (|\bar{V}_n| + \bar{c}) \cdot (Q_R - Q_L) \cdot \left|\vec{S}_\xi\right| \cdot sign\left(\vec{n} \bullet \vec{S}_\xi\right),\\
\tilde{G}^i &= \frac{1}{2}(\tilde{G}^i_L + \tilde{G}^i_R - (|\bar{V}_n| + \bar{c}) \cdot (Q_R - Q_L) \cdot \left|\vec{S}_\eta\right| \cdot sign\left(\vec{n} \bullet \vec{S}_\eta\right),\\
\tilde{H}^i &= \frac{1}{2}(\tilde{H}^i_L + \tilde{H}^i_R - (|\bar{V}_n| + \bar{c}) \cdot (Q_R - Q_L) \cdot \left|\vec{S}_\zeta\right| \cdot sign\left(\vec{n} \bullet \vec{S}_\zeta\right);
\end{aligned}$$

4. Compute the derivatives of the fluxes at all the solution points according to

$$\begin{aligned}
\left(\frac{\partial \tilde{F}}{\partial \xi}\right)_{i,j,k} &= \sum_{r=0}^{N} \tilde{F}_{r+1/2,j,k} \cdot l'_{r+1/2}(\xi_i),\\
\left(\frac{\partial \tilde{G}}{\partial \eta}\right)_{i,j,k} &= \sum_{r=0}^{N} \tilde{G}_{i,r+1/2,k} \cdot l'_{r+1/2}(\eta_j),\\
\left(\frac{\partial \tilde{H}}{\partial \varsigma}\right)_{i,j,k} &= \sum_{r=0}^{N} \tilde{H}_{i,j,r+1/2} \cdot l'_{r+1/2}(\varsigma_k).
\end{aligned} \tag{2.14}$$

The viscous flux is a function of both the conserved variables and their gradients, i.e., $\tilde{F}^v_{i+1/2,j,k} = \tilde{F}^v(Q_{i+1/2,j,k}, \nabla Q_{i+1/2,j,k})$. Therefore the key is how to compute the solution gradients at the flux points. One approach is given in [17], and is named the average approach, and the other approach is a local DG-like approach [9,40]. Both approaches are employed and tested in the present study. The gradient of the conserved variables in the physical domain can be easily computed using

$$\nabla Q=\frac{\partial Q}{\partial \xi}\nabla\xi+\frac{\partial Q}{\partial \eta}\nabla\eta+\frac{\partial Q}{\partial \zeta}\nabla\zeta=\frac{1}{|J|}\left[\frac{\partial\left(Q\vec{S}_\xi\right)}{\partial\xi}+\frac{\partial\left(Q\vec{S}_\eta\right)}{\partial\eta}+\frac{\partial\left(Q\vec{S}_\zeta\right)}{\partial\zeta}\right]. \tag{2.15}$$

In deriving (2.15), we have used the following identity

$$\frac{\partial\vec{S}_\xi}{\partial\xi}+\frac{\partial\vec{S}_\eta}{\partial\eta}+\frac{\partial\vec{S}_\zeta}{\partial\zeta}=0.$$

The derivatives along each coordinate direction are computed using

$$\begin{aligned}
\left[\frac{\partial\left(Q\vec{S}_\xi\right)}{\partial\xi}\right]_{j,k} &=\sum_{r=0}^{N}\left(Q\vec{S}_\xi\right)_{r+1/2,j,k}\cdot l'_{r+1/2}(\xi),\\
\left[\frac{\partial\left(Q\vec{S}_\eta\right)}{\partial\eta}\right]_{i,k} &=\sum_{r=0}^{N}\left(Q\vec{S}_\eta\right)_{i,r+1/2,k}\cdot l'_{r+1/2}(\eta),\\
\left[\frac{\partial\left(Q\vec{S}_\varsigma\right)}{\partial\zeta}\right]_{i,j} &=\sum_{r=0}^{N}\left(Q\vec{S}_\varsigma\right)_{i,j,r+1/2}\cdot l'_{r+1/2}(\zeta).
\end{aligned} \tag{2.16}$$

In the average approach, the following steps are taken to compute the viscous fluxes:

1. Same as Step 1 for the inviscid flux computations;

2. When computing the derivatives using (2.16), the solution Q at the cell interface is not uniquely defined. The solution at the interface is simply the average of the left and right solutions,
$$\hat{Q}=\frac{Q_L+Q_R}{2}.$$

3. Compute the gradients of the solution at the solution points using the solutions at the flux points with (2.15) and (2.16). Then the gradients are interpolated from the solution points to the flux points using the same Lagrangian interpolation approach given in (2.12).

4. Compute the viscous flux at the flux points using the solutions and their gradients at the flux points. Again at cell interfaces, the gradients have two values, one from the left and one from the right. The gradients used in the viscous fluxes at the cell interface are simply the averaged ones, i.e.,
$$\tilde{F}^v=\tilde{F}^v\left(\frac{Q_L+Q_R}{2},\frac{\nabla Q_L+\nabla Q_L}{2}\right).$$

The local-DG like algorithm to compute the viscous flux takes the following steps:

1. Same as Step 1 for the inviscid flux computations;

2. When computing the derivatives using (2.16), the solution Q at the cell interface is not uniquely defined. Following the local-DG approach, a numerical "flux" for Q at the interface can be determined based on its local orientation. For example, we could use
$$\hat{Q} = Q_R$$
to compute the gradients using (2.16), where $\hat{Q}$ is the "numerical flux".

3. Compute the gradients of the solution at the solution points using the solutions at the flux points with (2.15) and (2.16). Then the gradients are interpolated from the solution points to the flux points using the same Lagrangian interpolation approach given in (2.12).

4. Compute the viscous flux at the flux points using the solutions and their gradients at the flux points. Again at cell interfaces, the solution and their gradients have two values, one from the left and one from the right. The solutions at the interface are obtained by averaging the left and right solutions on the face. Based on the LDG approach, the viscous flux uses the gradients from the left, i.e.,
$$\tilde{F}^v = \tilde{F}^v\left(\frac{Q_L+Q_R}{2}, \nabla Q_L\right).$$

Once all fluxes are computed at the flux points, the flux polynomials are built according to (2.13), and the derivatives of the fluxes are then evaluated at each solution point to update the DOFs, i.e.,

$$\frac{\partial \tilde{Q}_{i,j,k}}{\partial t} = -\left(\frac{\partial \tilde{F}}{\partial \xi} + \frac{\partial \tilde{G}}{\partial \eta} + \frac{\partial \tilde{H}}{\partial \varsigma}\right)_{i,j,k}. \tag{2.17}$$

For time integration, we employ a multistage TVD [28] or SSP (strong-stability-preservation) [30] scheme.

3 Numerical results

3.1 Accuracy study using an isotropic vortex propagating problem

In order to quantify the numerical order of accuracy of the SD method for the Euler equations with both grid-refinement (*h*-refinement) and order refinement (*p*-refinement), an isotropic vortex propagation problem is chosen since it has an analytical solution. This is an idealized problem for the Euler equations, which was used by Shu [29].

The mean flow is $\{\rho,u,v,w,p\}=\{1,1,1,0,1\}$. An isotropic vortex is then added to the mean flow, i.e., with perturbations in u, v, and temperature $T=p/\rho$, and no perturbation in velocity w and entropy $S=p/\rho^{\gamma}$:

$$\begin{aligned}(\delta u,\delta v,\delta w)&=\frac{\varepsilon}{2\pi}e^{0.5(1-r^2)}(-y,x,0),\\ \delta T&=-\frac{(\gamma-1)\varepsilon^2}{8\gamma\pi^2}e^{1-r^2},\quad \delta S=0,\end{aligned}$$

where $r^2=x^2+y^2$, and the vortex strength $\varepsilon=5$. In the numerical simulation, the computational domain is taken to be $[-5,5]\times[-5,5]\times[-5,5]$, with characteristic inflow and outflow boundary conditions imposed on the boundaries in x and y directions, and extrapolation in the z direction. It can be readily verified that the Euler equations with the above initial conditions admit an exact solution that moves with the speed (1,1,0) in the diagonal direction on the x-y plane.

For the h-refinement study, the order of accuracy N was set to 6. Four meshes were employed. Fig. 4 shows the time-independent errors between the numerical solution and the analytical solution in L_∞, L_1 and L_2 norms at $t=0.1$. The three-stage Runge-Kutta scheme was employed for the time integration. Note that the average slope is about 6, indicating that the numerical order of accuracy is 6^{th} order.

For the p-refinement study, a coarse grid with 100 ($10\times10\times1$) elements was used. Then the order of the polynomial basis was increased. The numerical errors are computed at $t=0.1$, and displayed in Fig. 5. Note that the exponential decay of error with respect to the order of accuracy is achieved.

Lastly, the time integration accuracy was also studied on a fine mesh (25 elements, $N=25$). The numerical errors with different time-step sizes are displayed at $t=0.1$ in Fig. 6. Note that the order of accuracy of the time integration scheme is indeed 3^{rd} order.

3.2 Accuracy study using Couette flow

Couette flow is an analytical solution for the Navier-Stokes equations, and was selected to study the accuracy of the 3D Navier-Stokes solver. This problem models the viscous flow in the positive x direction between two parallel plates locate at $y=0$, and $y=H$. The plate at $y=0$ is stationary with a fixed temperature T_0, while the plate at $y=H$ has a different temperature T_1, and moves at speed of U. This problem has an exact solution under the simplification that the viscosity coefficient μ is a constant, which can be expressed as

$$\begin{aligned}&u=\frac{U}{H}y,\quad v=0,\quad w=0,\\ &T=T_0+\frac{y}{H}(T_1-T_0)+\frac{\mu U^2}{2k}\cdot\frac{y}{H}\left(1-\frac{y}{H}\right),\\ &p=const,\quad \rho=\frac{p}{R\cdot T},\end{aligned}$$

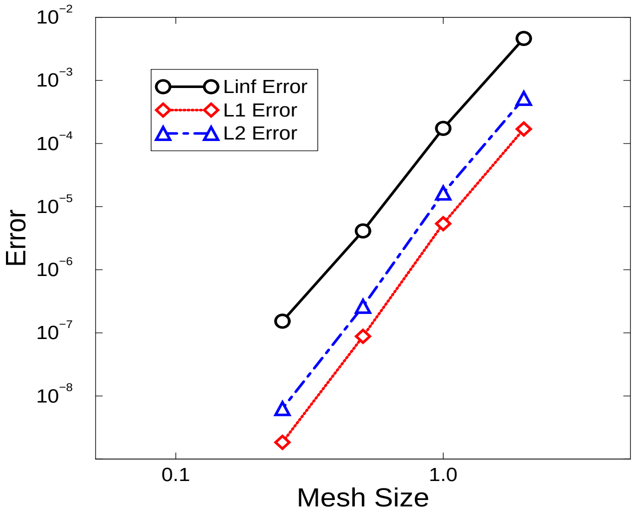


Figure 4: Solution errors with h-refinement for the vortex propagation problem.

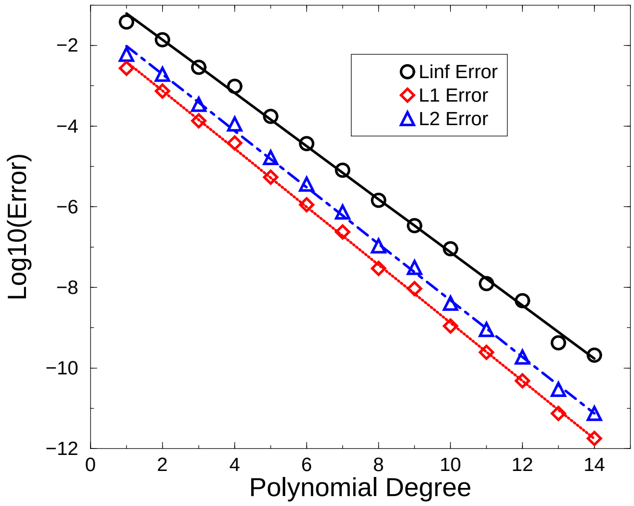


Figure 5: Solution errors with p-refinement for the vortex propagation problem.

where k is the thermal conductivity, and R is the gas constant. In the present computations, we have chosen $U=1.0$, $H=2$, $T_0=0.8$, $T_1=0.85$, $\mu=0.01$. The computational domain is $[0,4]\times[0,2]\times[0,4]$.

In the grid refinement study, four different grids were employed with several SD schemes of various accuracy, and the results are presented in Tables 1 and 2 using the average approach and the LDG approach for the viscous fluxes, respectively. Note that the expected accuracy was achieved with both approaches, and the average approach gave slightly more accurate results. Therefore, it is used in the rest of the computations. A

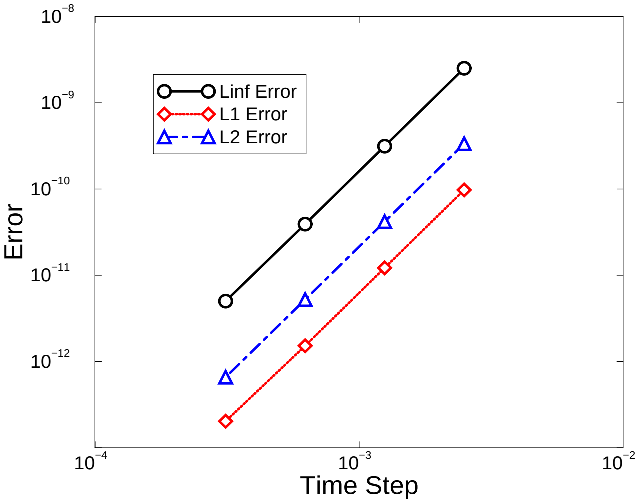


Figure 6: Solution errors with temporal refinement for the vortex propagation problem.

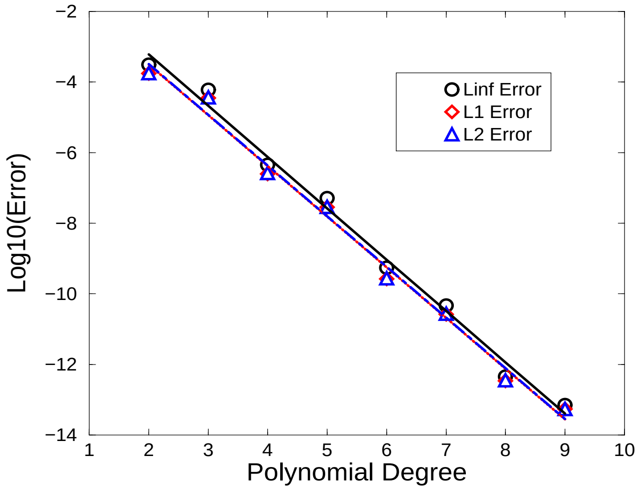


Figure 7: Solution errors with p-refinement for Couette flow problem.

p-refinement study was performed on a coarse grid with 16 (4x4x1) elements. The order of the polynomial basis was increased up to 9, and the 3-stage Runge-Kutta scheme was employed for time integration. The exponential decay of numerical errors in L_∞, L_1, and L_2 norms was again observed in Fig. 7.

3.3 Flow over a sphere

To test the capability of the SD method for handling curved boundaries, both inviscid and viscous flows over a sphere were computed. In order to perform a grid refinement

Table 1: Grid refinement study for Couette flow using the LDG approach for viscous flux computation.

Order of Accuracy	Grid	L_∞ error	L_∞ order	L_1 error	L_1 order	L_2 error	L_2 order
3	2x2x1	3.26E-04	—	8.48E-05	—	1.29E-04	—
	4x4x1	4.94E-05	2.72	1.03E-05	3.04	1.55E-05	3.06
	8x8x1	7.97E-06	2.63	1.45E-06	2.83	1.98E-06	2.96
	16x16x1	1.36E-06	2.55	2.32E-07	2.64	2.97E-07	2.74
4	2x2x1	6.78E-05	—	2.35E-05	—	3.33E-05	—
	4x4x1	5.24E-06	3.70	1.72E-06	3.77	2.34E-06	3.83
	8x8x1	4.41E-07	3.57	1.43E-07	3.59	1.75E-07	3.74
	16x16x1	4.04E-08	3.45	1.24E-08	3.52	1.40E-08	3.64
5	2x2x1	2.43E-06	—	4.71E-07	—	7.90E-07	—
	4x4x1	1.17E-07	4.38	1.52E-08	4.95	2.76E-08	4.84
	8x8x1	5.53E-09	4.40	4.52E-10	5.07	9.33E-10	4.89
	16x16x1	2.89E-10	4.26	2.03E-11	4.48	3.69E-11	4.66

Table 2: Grid refinement study for Couette flow using the average approach for viscous flux computation.

Order of Accuracy	Grid	L_∞ error	L_∞ order	L_1 error	L_1 order	L_2 error	L_2 order
3	2x2x1	3.06E-04	—	8.72E-05	—	1.26E-04	—
	4x4x1	3.96E-05	2.95	9.34E-06	3.22	1.37E-05	3.20
	8x8x1	5.10E-06	2.96	1.14E-06	3.03	1.56E-06	3.13
	16x16x1	7.05E-07	2.86	2.08E-07	2.46	2.44E-07	2.68
4	2x2x1	6.49E-05	—	2.35E-05	—	3.29E-05	—
	4x4x1	4.79E-06	3.76	1.84E-06	3.68	2.40E-06	3.78
	8x8x1	3.80E-07	3.66	1.70E-07	3.43	1.98E-07	3.60
	16x16x1	3.73E-08	3.35	1.77E-08	3.26	1.90E-08	3.38
5	2x2x1	2.00E-06	—	4.69E-07	—	7.19E-07	—
	4x4x1	7.35E-08	4.77	1.29E-08	5.19	2.03E-08	5.14
	8x8x1	2.44E-09	4.91	3.28E-10	5.30	5.10E-10	5.32
	16x16x1	8.30E-11	4.88	1.28E-11	4.68	1.63E-11	4.97

study, a coarse and a fine grid were employed in the simulations. Figs. 8a and 8b show both the coarse and fine computational grids. The coarse grid has 768 total cells with 96 surface quadrilaterals, and there are 8 cells between the sphere surface and the far-field outer boundary, which is 10 diameters away from the center of the sphere. The fine grid is generated from the coarse grid through global refinement in all directions, resulting in a grid with 8 times the number of cells (6,144 cells). At these grid resolution, a linear facet representation of the sphere geometry is very crude, and large solution errors are expected. To remedy this problem, the surface is approximated with a quadratic

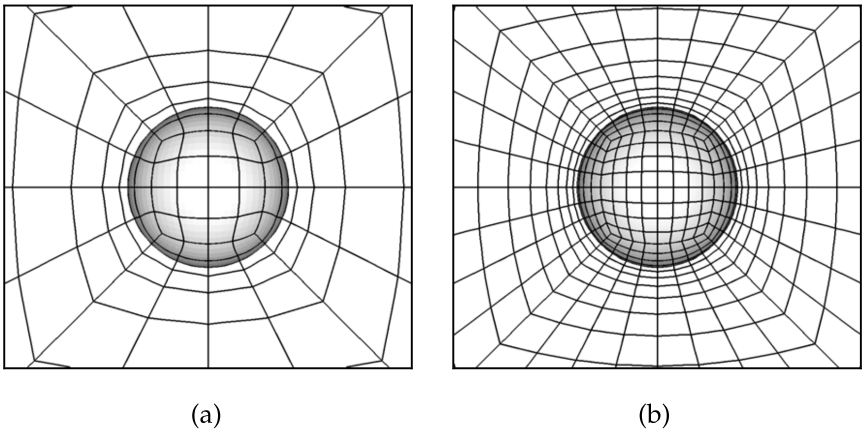


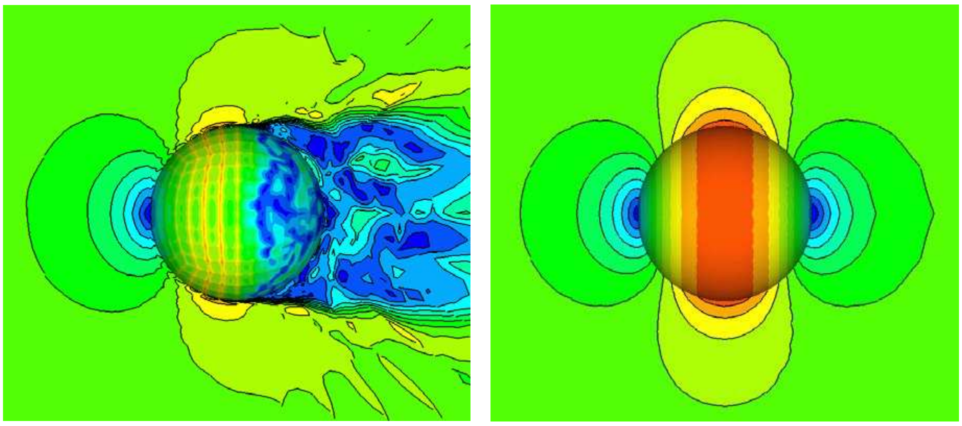
Figure 8: Coarse and fine grids for a sphere with quadratic boundary.

representation, as shown in Fig. 8.

In order to show the importance of high-order geometry representation, linear and quadratic boundaries were used in an inviscid flow simulation using the 3^{rd} order SD scheme. The inflow Mach number was 0.2535. With a linear boundary representation, a steady solution was not possible to obtain, and the flow field is not stable because of the sharp corners present in the surface grid. The instantaneous Mach number contours are displayed in Fig. 9a, which clearly shows that the flow is not stable. With the quadratic boundary representation, the simulation converged to machine-zero, and the Mach contours are presented in Fig. 9b. Obviously the flow is very symmetric, indicating the isotropic nature of the flow. For here on, quadratic boundaries were used in all the rest of the simulations.

Next, p-refinement studies were performed on both the coarse and fine grids. The 2^{nd} to 4^{th} order SD schemes were employed in these studies. The computed Mach contours on the coarse grids are displayed in Fig. 10, while the corresponding fine grid results are presented in Fig. 11. In these contours, the Mach number is discontinuous across cell interfaces, and the solution quality can be visibly judged from the magnitude of the discontinuities across cell interfaces, since the flow is theoretically continuous. Obviously the 2^{nd} order solution is very crude on the coarse grid. Large entropy generation is evident from the non-symmetric Mach contours. The higher-order results show marked improvements. The 4^{th} order results on the coarse mesh are very smooth and symmetric. The computed Mach contours on the fine mesh are much smoother than the corresponding ones on the coarse mesh. It appears p-refinement is more effective than h-refinement (grid-refinement), since it is obvious that the 3^{rd} and 4^{th} order solutions on the coarse mesh are visibly smoother and more symmetric than the 2^{nd} and 3^{rd} order solutions on the fine mesh respectively.

For the viscous flow simulation, the Reynolds number based on the diameter was chosen to be 118 since an experimental streamline picture is available for comparison [33].



(a) Linear boundary (b) Quadratic boundary

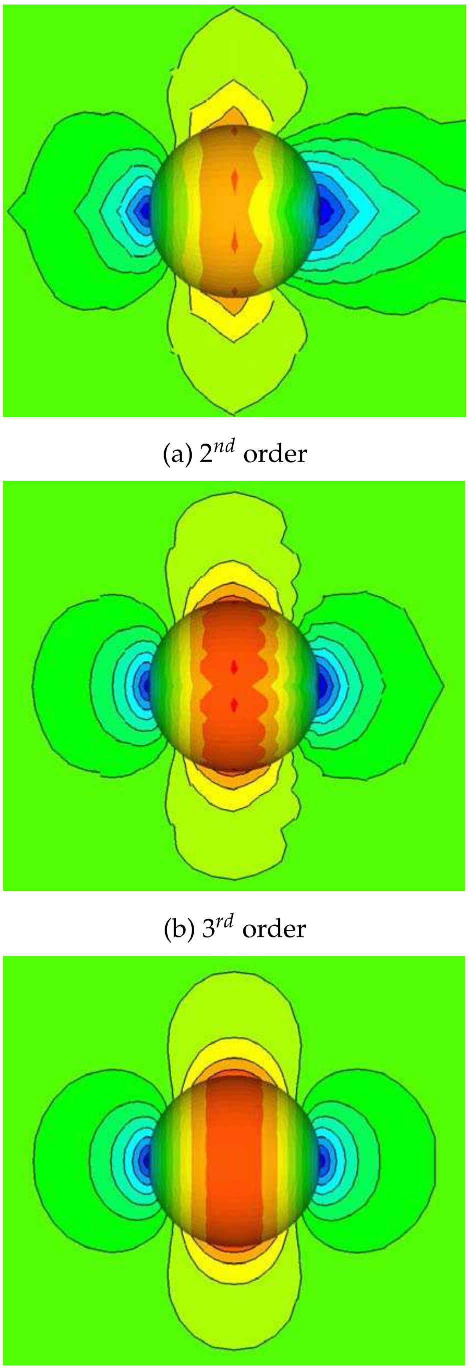
Figure 9: Computed Mach contours for subsonic flow around a sphere with linear and quadratic boundaries using a 3^{rd} order SD scheme on the fine mesh. Mach contours start at Mach $= 1/30$ with a $1/30$ interval.

This viscous flow computation was performed on the fine grid using the 4^{th} and 6^{th} order SD schemes to assess the numerical error. The computed Mach number contours using both schemes are shown in Fig. 12. Note that the contours are nearly on top of each other, indicating an excellent agreement. The computational streamlines are compared with the experimental streamlines in Fig. 13. In both plots the steady separation bubble is readily observed, and the size of the separation region in the computation agrees very well with that of the experiment, at least in the "eye ball" norm. The skin friction coefficients at the wall computed with both schemes are plotted in Fig. 14, and they are right on top of each other. In fact, the skin friction coefficients differ less than 0.1% between the 4^{th} and 6^{th} order results. The predicted separation angle from both schemes is 123.6 degrees (the wind side stagnation point has an angle of 0), and the length of the separation region is 1.04D.

3.4 Unsteady viscous flow over a cylinder

Finally, unsteady viscous flow over a circular cylinder was selected to demonstrate the SD method for time-accurate flow computations. The computational grid has 590 cells, and is shown in Fig. 15. The outer boundaries are at least ten diameters away from the center of the cylinder (the origin). Since this is a two-dimensional problem, only one cell was used in the z-direction.

Many experimental and computational studies have been carried out to study the unsteady flow around a cylinder at various Reynolds numbers [2,6,15,39]. In the present study, the Reynolds number was chosen to be 160, and both the 3^{rd} and 4^{th} order SD schemes were employed in the simulation. The unsteady von Karman vortex street was predicted by both the 3^{rd} and 4^{th} order schemes. By monitoring the pressure histories

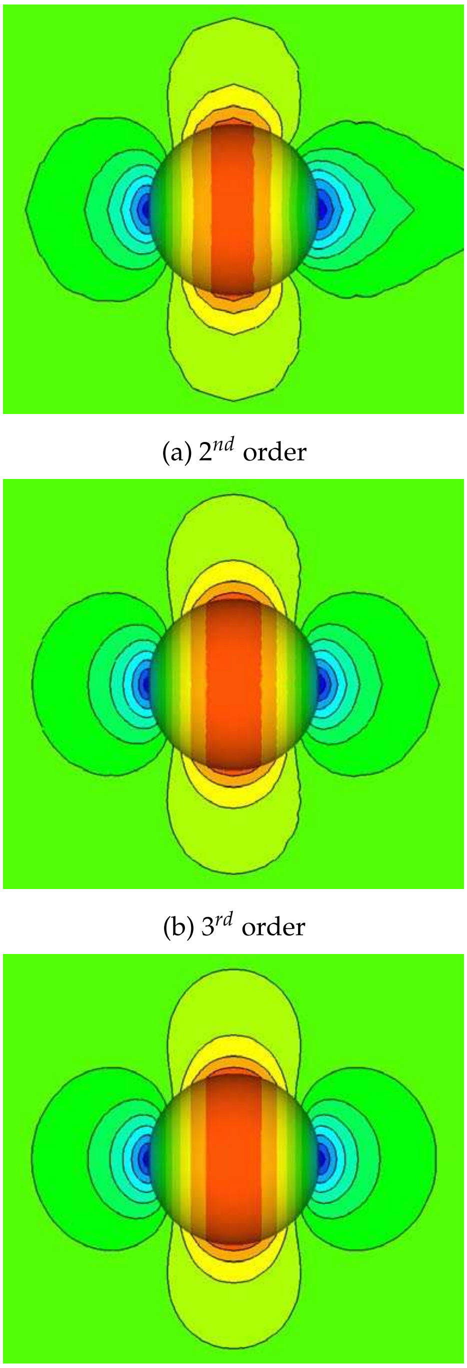


(a) 2^{nd} order

(b) 3^{rd} order

(c) 4^{th} order

Figure 10: Computed Mach contours for flow around a sphere with quadratic boundaries using 2^{nd}-4^{th} order SD schemes on the coarse mesh (768 cells). Mach contours start at Mach $= 1/30$ with a $1/30$ interval.



(a) 2^{nd} order

(b) 3^{rd} order

(c) 4^{th} order

Figure 11: Computed Mach contours for flow around a sphere with quadratic boundary using 2^{nd}-4^{th} order SD schemes on the fine mesh (6,144 cells). Mach contours start at Mach $= 1/30$ with a $1/30$ interval.

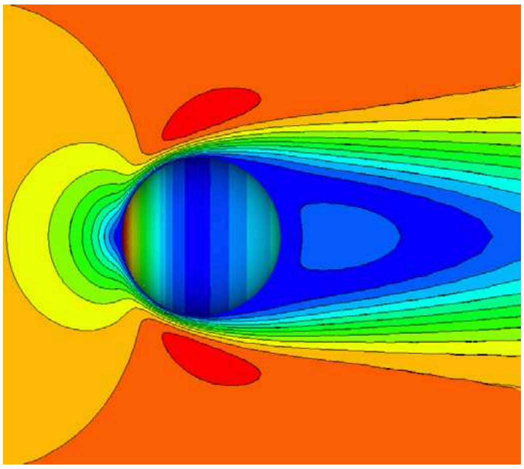
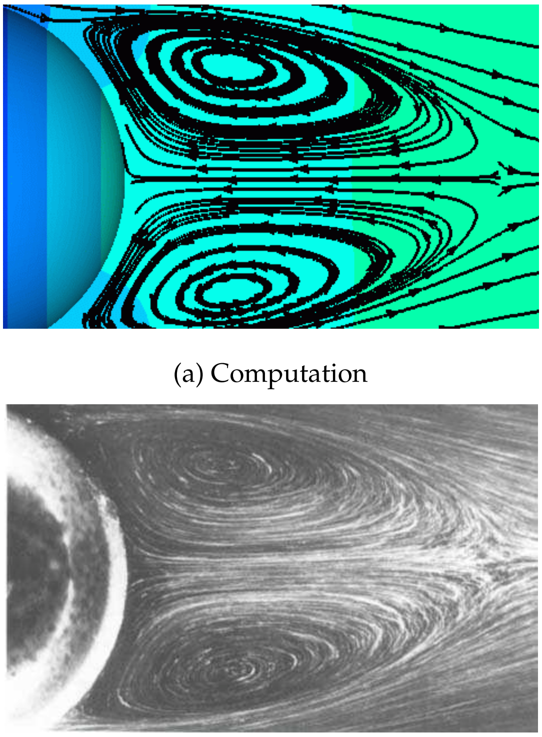


Figure 12: Computed pressure (on the sphere) and Mach number (on z = 0 plane) distributions using the 4^{th} and 6^{th} order SD schemes. Mach contours start at Mach = 1/40 with a 1/40 interval.



(a) Computation

(b) Experiment

Figure 13: Comparison of streamlines of the flow field from computation (a) and experiment (b).

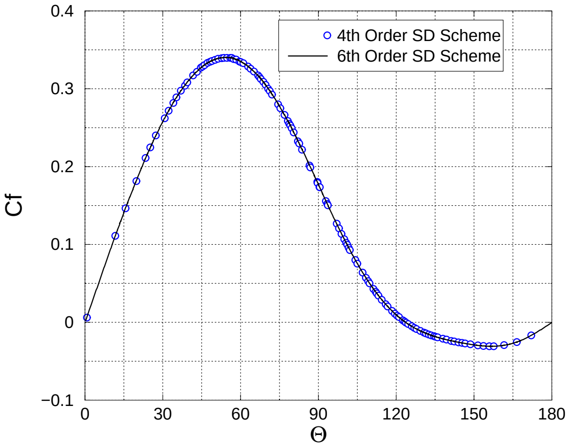


Figure 14: Predicted skin friction coefficient profiles with the 4^{th} and 6^{th} order SD schemes.

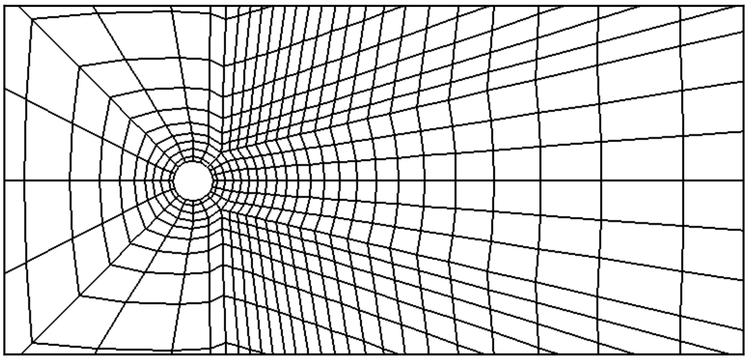


Figure 15: Computational grid for unsteady flow over a circular cylinder.

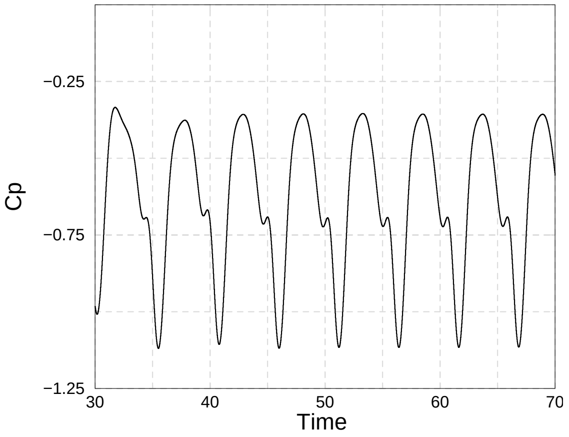


Figure 16: Pressure history at (10, 1) for flow over a cylinder at Re = 160.

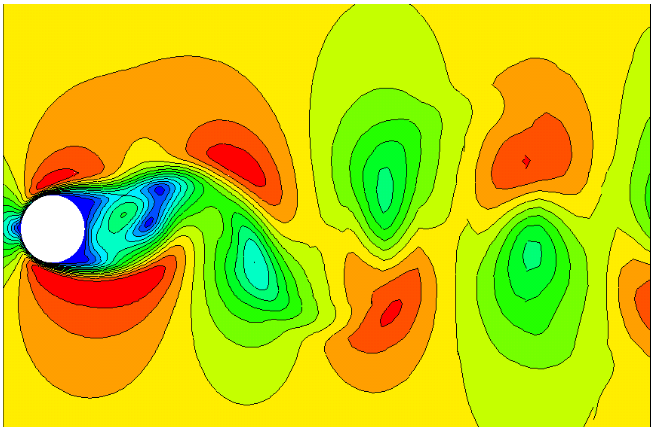


Figure 17: Mach number contours for flow over a cylinder.

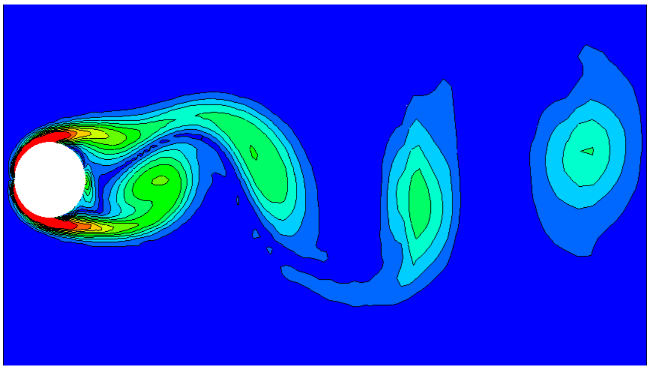


Figure 18: Vorticity magnitude contours for flow over a cylinder.

at selected locations in the flow field, we observed that the flow reached a periodic state after about 40 non-dimensional time units (scaled by diameter and incoming velocity). For example, the Cp history at point (10D, 1D) is shown in Fig. 16. From the Cp histories, we can easily identify the vortex shedding frequency, and then compute the Strouhal number. The computed Strouhal number is 0.194 using the 3^{rd} order SD scheme, and 0.192 using the 4^{th} order SD scheme, indicating a very close agreement between the two simulations. In comparison, the experimentally measured Strouhal number is 0.186 [39]. Two computations using a spectral element method [15] and a finite difference method [39] gave a Strouhal number of 0.198 and 0.175 respectively. The present computation appears to have the closest agreement with the experiment data, and is between the two other computed results. The computed Mach and vorticity magnitude using the 4^{th} order scheme are displayed in Figs. 17 and 18. The von Karman vortex street can be clearly observed.

4 Conclusions and future work

In this paper, the multidomain spectral difference method has been successfully extended to 3D for solving the full Navier-Stokes equations on unstructured hexahedral grids. Because the SD method is based on the differential form of the governing equations, the implementation is straightforward even for high-order curved boundaries. All the operations are basically one-dimensional in each coordinate direction, resulting in improved efficiency. The high order accuracy of the SD method is numerically verified in both space and time, and the exponential decay of the numerical error with respect to p-refinement has been achieved for the vortex propagation problem and Couette flow. Numerical experiments for flow over a cylinder and a sphere have demonstrated the capability of the SD method in obtaining high quality results with high order curved boundaries. We are currently demonstrating the SD method for more complex flow problems, and working on efficient implicit and multigrid solution algorithms.

Acknowledgments

The study was partially funded by Rockwell Scientific/DARPA under contract W911NF-04-C-0102, by DOE grant DE-FG02-05ER25677 and AFOSR grant FA9550-06-1-0146. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of DARPA, DOE, AFOSR, or the U.S. Government.

References

[1] R. Abgrall and P. L. Roe, High order fluctuation schemes on triangular meshes, J. Sci. Comput., 19 (2003), 3-36.

[2] H. M. Badr, M. Coutanceau, S. Dennis and C. Menard, Unsteady flow past a rotating circular cylinder at Reynolds numbers 10^3 and 10^4, J. Fluid Mech., 220 (1990), 459-484.

[3] T. J. Barth and P. O. Frederickson, High-order solution of the Euler equations on unstructured grids using quadratic reconstruction, AIAA Paper 90-0013, 1990.

[4] F. Bassi and S. Rebay, High-order accurate discontinuous finite element solution of the 2D Euler equations, J. Comput. Phys., 138 (1997), 251-285.

[5] R. Biswas and R. C. Strawn, A dynamic mesh adaptation procedure for unstructured hexahedral grids, AIAA Paper 96-0027, 1996.

[6] R. Bouard and M. Coutanceau, The early stages of development of the wake behind an impulsively started cylinder for $40 < Re < 10^4$, J. Fluid Mech., 101 (1980), 583-607.

[7] B. Cockburn and C. W. Shu, TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws II: General framework, Math. Comput., 52 (1989), 411-435.

[8] B. Cockburn and C. W. Shu, The Runge-Kutta discontinuous Galerkin method for conservation laws V: multidimensional systems, J. Comput. Phys., 141 (1998), 199-224.

[9] B. Cockburn and C. W. Shu, The local discontinuous Galerkin method for time-dependent convection diffusion system, SIAM J. Numer. Anal., 35 (1998), 2440-2463.

[10] A. T. Conlisk, Modern helicopter aerodynamics, Annu. Rev. Fluid. Mech., 29 (1997), 515-567.

[11] E. P. N. Duque, R. Biswas and R. C. Strawn, A solution adaptive structured/unstructured overset grid solver with applications to helicopter rotor flows, AIAA Paper 95-1766, 1995.

[12] S. K. Godunov, A finite-difference method for the numerical computation of discontinuous solutions of the equations of fluid dynamics, Mat. Sb., 47 (1959), 271-290.

[13] C. Hu and C. W. Shu, Weighted essentially non-oscillatory schemes on triangular meshes, J. Comput. Phys., 150 (1999), 97-127.

[14] P. G. Huang, Z. J. Wang and Y. Liu, An implicit space-time spectral difference method for discontinuity capturing using adaptive polynomials, AIAA Paper 2005-5255, 2005.

[15] G. E. Karniadakis and G. S. Triantafyllou, Frequency selection and asymptotic states in laminar wakes, J. Fluid Mech., 199 (1989), 441-469.

[16] D. A. Kopriva and J. H. Kolias, A conservative staggered-grid Chebyshev multidomain method for compressible flows, J. Comput. Phys., 125 (1996), 244-261.

[17] D. A. Kopriva, A staggered-grid multidomain spectral method for the compressible Navier-Stokes equations, J. Comput. Phys., 143 (1998), 125-158.

[18] S. K. Lele, Compact finite difference schemes with spectral-like resolution, J. Comput. Phys., 103 (1992), 16-42.

[19] Y. Liu, M. Vinokur and Z. J. Wang, Discontinuous spectral difference method for conservation laws on unstructured grids, in: C. Groth, D. W. Zingg (Eds.), Proceeding of the 3rd International Conference in CFD, Toronto, Springer, 2004, pp. 449-454.

[20] Y. Liu, M. Vinokur and Z. J. Wang, Spectral difference method for unstructured grids I: Basic formulation, J. Comput. Phys., 216 (2006), 780-801.

[21] Y. Liu, M. Vinokur and Z. J. Wang, Spectral (finite) volume method for conservation laws on unstructured grids V: Extension to three-dimensional systems, J. Comput. Phys., 212 (2006), 454-472.

[22] H. Luo, D. Sharov, J. D. Baum and R. Lohner, On the computation of compressible turbulent flows on unstructured grids, AIAA Paper 2000-0927, 2000.

[23] G. May and A. Jameson, A spectral difference method for the Euler and Navier-Stokes equations on unstructured grids, AIAA Paper 2006-304, 2006.

[24] R. L. Meakin, Moving body overset grid methods for complete tiltrotor simulations, AIAA Paper 93-3350-CP, 1993.

[25] A. T. Patera, A Spectral element method for fluid dynamics: Laminar flow in a channel expansion, J. Comput. Phys., 54 (1984), 468-488.

[26] P. L. Roe, Approximate Riemann solvers, parameter vectors, and difference schemes, J. Comput. Phys., 43 (1981), 357-372.

[27] V. V. Rusanov, Calculation of interaction of non-steady shock waves with obstacles, J. Comput. Math. Phys., USSR, 1 (1961), 267-279.

[28] C. W. Shu, Total-Variation-Diminishing time discretizations, SIAM J. Sci. Stat. Comput., 9 (1988), 1073-1084.

[29] C. W. Shu, Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws, in: A. Quarteroni (Ed.), Advanced Numerical Approximation of Nonlinear Hyperbolic Equations, Lecture Notes in Mathematics, Springer-Verlag, Berlin/New York, volume 1697, 1998, pp. 325.

[30] R. J. Spiteri and S. J. Ruuth, A new class of optimal high-order strong-stability-preserving time discretization methods, SIAM J. Numer. Anal., 40 (2002), 469-491.

[31] Y. Sun, Z. J. Wang and Y. Liu, Spectral (finite) volume method for conservation laws on unstructured grids VI: Extension to viscous flow, J. Comput. Phys., 215 (2006), 41-58.

[32] C. K. W. Tam and J. C. Webb, Dispersion-relation-preserving finite difference schemes for computational acoustics, J. Comput. Phys., 107 (1993), 262-281.

[33] S. Taneda, Experimental investigations of the wake behind a sphere at low Reynolds numbers., J. Phys. Soc. Jpn., 11 (1956), 1104-1108.

[34] M. Visbal and D. Gaitonde, Shock capturing using compact-differencing-based methods, AIAA Paper 2005-1265, 2005.

[35] M. Visbal, P. E. Morgan and D. P. Rizzetta, An implicit LES approach based on high-order compact differencing and filtering schemes, AIAA Paper 2003-4098, 2003.

[36] Z. J. Wang and Y. Liu, Spectral (finite) volume method for conservation laws on unstructured grids III: Extension to one-dimensional systems, J. Sci. Comput., 20 (2004), 137-157.

[37] Z. J. Wang, L. Zhang and Y. Liu, Spectral (finite) volume method for conservation laws on unstructured grids IV: Extension to two-dimensional systems, J. Comput. Phys., 194 (2004), 716-741.

[38] Z. J. Wang and Y. Liu, The spectral difference method for the 2D Euler equations on unstructured grids, AIAA Paper 2005-5112, 2005.

[39] C. H. K. Williamson, Oblique and parallel modes of vortex shedding in the wake of a cylinder at low Reynolds number, J. Fluid Mech., 206 (1989), 579-628.

[40] Y. Xu and C. W. Shu, Local discontinuous Galerkin methods for the Kuramoto-Sivashinsky equations and the Ito-type coupled KdV equations, Comput. Method Appl. Mech. Engrg., 195 (2006), 3430-3447.