

An Implicit Algorithm for High-Order DG/FV Schemes for Compressible Flows on 2D Arbitrary Grids

Laiping Zhang^{1,2,*}, Ming Li², Wei Liu^{1,2} and Xin He¹

¹ State Key Laboratory of Aerodynamics, Mianyang, Sichuan, 621000, China.

² China Aerodynamics Research and Development Center, Mianyang, Sichuan, 621000, China.

Received 9 November 2013; Accepted (in revised version) 28 July 2014

Communicated by Chi-Wang Shu

Abstract. A Newton/LU-SGS (lower-upper symmetric Gauss-Seidel) iteration implicit method was developed to solve two-dimensional Euler and Navier-Stokes equations by the DG/FV hybrid schemes on arbitrary grids. The Newton iteration was employed to solve the nonlinear system, while the linear system was solved with LU-SGS iteration. The effect of several parameters in the implicit scheme, such as the CFL number, the Newton sub-iteration steps, and the update frequency of Jacobian matrix, was investigated to evaluate the performance of convergence history. Several typical test cases were simulated, and compared with the traditional explicit Runge-Kutta (RK) scheme. Firstly the Couette flow was tested to validate the order of accuracy of the present DG/FV hybrid schemes. Then a subsonic inviscid flow over a bump in a channel was simulated and the effect of parameters was investigated also. Finally, the implicit algorithm was applied to simulate a subsonic inviscid flow over a circular cylinder and the viscous flow in a square cavity. The numerical results demonstrated that the present implicit scheme can accelerate the convergence history efficiently. Choosing proper parameters would improve the efficiency of the implicit scheme. Moreover, in the same framework, the DG/FV hybrid schemes are more efficient than the same order DG schemes.

AMS subject classifications: 65C20, 65L60, 65M08, 65M60, 68U20, 76M10, 76M12

Key words: Implicit algorithm, discontinuous Galerkin method, DG/FV hybrid Scheme, Newton iteration, Gauss-Seidel iteration.

1 Introduction

The use of unstructured meshes for computational fluid dynamics problems has become widespread due to their ability to discretize arbitrarily complex geometries and the ease

*Corresponding author. *Email addresses:* zhanglp_cardc@126.com (L. Zhang), mli@sina.com (M. Li), lw4992@gmail.com (W. Liu), fantasy2003@gmail.com (X. He)

of adaptation in enhancing the solution accuracy and efficiency through the use of adaptive refinement techniques. In recent years, significant progress has been made in developing numerical algorithms for the solution of the Euler and Navier-Stokes equations on unstructured grids. Nearly all production flow solvers are based on second-order numerical methods, either finite volume method (FVM), finite difference method (FDM) or finite element method (FEM). Nevertheless, many types of problems, such as computational aeroacoustics (CAA), vortex-dominant flows and large eddy simulation (LES) of turbulent flows, call for higher order accuracy (third-order and higher). The main deficiency of widely available, second-order methods for the accurate simulations of the above-mentioned flows is the excessive numerical diffusion and dissipation of vorticity. Applications of higher-order accurate, low-diffusion and low dissipation numerical methods can significantly alleviate this deficiency of the traditional second-order methods, and improve predictions of vortical and other complex, separated, unsteady flows. Therefore, various higher-order methods have been developed in the last two decades [1–5], especially for unstructured grids, including the well-known discontinuous Galerkin (DG) method [3–7], the spectral volume (SV) method [8–11], and the spectral difference (SD) method [12–14]. Interested readers can refer to the comprehensive review articles for higher-order methods by Ekaterinaris [15] and Wang [16].

As the leader of higher-order numerical methods for compressible flow computations in aerospace applications, the DG method has recently become popular for problems with both complex physics and geometry. The DG method was originally developed by Reed and Hill to solve the neutron transport equation [3]. The development of higher-order DG methods for hyperbolic conservation laws was pioneered by Cockburn, Shu and their collaborators in a series of papers on the Runge-Kutta DG (RKDG) method [4–7]. Many other researchers have made significant contributions in the development of the DG methods. Refer to [17] for a comprehensive review on the DG method history and literature.

However, the DG method does have a number of weaknesses, including the huge memory requirement and high computational cost. In order to improve the efficiency in both memory and CPU time for 3D realistic complex configurations, many hybrid approaches have been proposed, including 1) different schemes for inviscid and viscous flux discretization [18, 19]; 2) hybrid approach based on domain decomposition [20]; 3) hybrid approach based on local polynomial reconstruction, such as the $P_N P_M$ schemes [21–23], the reconstructed-based DG (RDG) scheme [24, 25], the Hermite WENO (HWENO) schemes [26, 27].

In our previous work [28–30], by comparing the DG methods, the k -exact FV methods and the lift collocation penalty (LCP) methods [31, 32], a concept of “static reconstruction” and “dynamic reconstruction” was introduced for higher-order numerical methods. Based on this concept, a class of high order DG/FV hybrid schemes was presented for 1D and 2D conservation law using a “hybrid reconstruction” approach. In the DG/FV hybrid schemes, the lower-order derivatives of the piecewise polynomial are computed locally in a cell by the DGM based on Taylor basis functions [33] (called as “dynamic

reconstruction”), while the higher-order derivatives are re-constructed by the “static reconstruction” of the FVM, using the known lower-order derivatives in the cell itself and its adjacent face neighboring cells.

The idea of DG/FV hybrid schemes came from that of the $P_N P_M$ schemes proposed by Dumbser et al. [21–23]. However, unlike the $P_N P_M$ schemes, the present DG/FV hybrid schemes are different from the $P_N P_M$ schemes in the following aspects. Firstly, the orthogonalized Taylor basis functions are adopted in this paper, while the traditional Lagrange finite element or hierarchical node-based functions are adopted in the $P_N P_M$ schemes. The purpose of choosing orthogonalized Taylor basis functions is to handle arbitrary grids using a unified basis function. Secondly, the higher-order derivatives (DOFs) are reconstructed directly by the Green-Gauss formulae since we let $M = N + 1$, instead of using another set of orthogonal basis functions as in the $P_N P_M$ schemes or least-squares reconstruction in the RDG schemes. The purpose of choosing $M = N + 1$ is to ensure the grid compactness and the reconstruction robustness. Thirdly, the semi-discretization of the governing equations is considered only for discretization of the convection term, while a local continuous space-time Galerkin method is used in the $P_N P_M$ schemes to construct a fully discrete scheme. The purpose of choosing the semi-discretization is to develop the implicit algorithm to improve the convergence history further.

The DG/FV hybrid schemes had been successfully applied to solve the 2D Euler and Navier-Stokes equations. The numerical results have demonstrated that, compared with the same order DGM, the memory requirement and CPU time are reduced by approximately 30% and 50%, respectively [29, 30]. We can expect the DG/FV hybrid scheme to be more efficient for the three-dimensional cases, which is very important for realistic applications.

Nevertheless, it is well known that higher-order spatial operators are much stiffer than lower-order ones. The allowable CFL number decreases with increasing order of accuracy for explicit schemes. For viscous problems with highly clustered mesh to resolve the viscous boundary layer, explicit higher-order methods are severely limited by the time step, and usually not competitive against implicit methods in terms of efficiency. Therefore, a major pacing item in the CFD community is the development of efficient time integration/iteration solution approaches for higher-order methods.

It has long been recognized that implicit algorithms are necessary to overcome the time step limit suffered by explicit algorithms. Many types of implicit algorithms have been successfully developed for flow solvers since 1970s, e.g., alternating direction implicit (ADI) algorithm [34, 35], Newton iteration [36], the element Jacobi [37], Gauss-Seidel (GS), preconditioned GMRES [38, 39], matrix-free Krylov [40], lower-upper symmetry Gauss-Seidel (LU-SGS) [41, 42], point-wise relaxation algorithms [43], and implicit-explicit Runge-Kutta iteration [44, 45]. Interested readers can refer to Ref. [46] for a comprehensive review on implicit algorithms. More recently, some of these implicit algorithms have been applied to high-order spatial discretization mentioned earlier [36, 38, 40, 42, 43, 47–49]. For example, Xia et al. proposed a GMRES based implicit algorithm [47], and then an implicit approach based on automatic differentiation [48] for their WENO

RDG method; Sun et al. developed an implicit approach based on LU-SGS for their SD method [42]. However, the pursuit of a faster and more efficient implicit algorithm for the DG methods, as well as the DG/FV hybrid schemes, still remains one of active research areas, especially for the high-order solvers on unstructured grids.

In this work, an implicit algorithm is divided into three iteration levels hierarchically. The first level is the iteration of real-time advancing. For steady flows, the first-order backward-difference Euler scheme is enough, because we do not care about the time accuracy during reaching the converged solution. In general, it will result in a nonlinear system about unknowns, and can't be solved directly. Therefore, a second level iteration strategy is employed with Newton iteration. Finally, a third level iteration is employed to solve the large linear system generated by the second level iteration. In the third level iteration, many schemes can be employed, such as the one-step lower or upper Gauss Seidel, the lower-upper symmetric Gauss Seidel, Jacobian iteration, and GMRES, etc.

According to the hierarchical implicit framework, a Newton/LU-SGS iteration algorithm was presented in this paper for the DG/FV hybrid schemes developed in our previous work. The non-linear system of equations was "linearized" by the Newton iteration algorithm, and the linear system was then solved with the well-known LU-SGS iteration approach. Furthermore, the effects on the convergence rate of several parameters in the implicit scheme were investigated with a set of typical test cases. The simulations were carried out on unstructured grids and triangular/quadrilateral (or Cartesian) hybrid grids to demonstrate the capability for arbitrary grids. The numerical results demonstrated that the implicit scheme speeds up the convergence rate more than one order of acceleration, compared with the traditional explicit multi-stage Runge-Kutta scheme. Choosing proper parameters (CFL-ratio, Newton sweep steps, and update frequency of the mass matrix) would improve the convergence history. Moreover, in the same framework, the DG/FV hybrid schemes are more efficient than the same order DG schemes.

The remainder of this paper is organized as follows. In Section 2, the DG/FV hybrid schemes based on Taylor basis functions are reviewed briefly. In Section 3, the implicit algorithm is presented in detail, including Newton iteration, LU-SGS iteration, matrix-free strategy, and the calculation of diagonal Jacobian matrix. In Section 4, some typical 2D test cases are simulated to validate the performance of the implicit scheme. Finally, some concluding remarks are given in Section 5.

2 Spatial discretization of DG method and DG/FV hybrid schemes

2.1 Flux divergence reconstruction procedure

For simplicity, consider the following two-dimensional Euler equations:

$$\frac{\partial u_i}{\partial t} + \nabla \cdot F_i(u) = 0, \quad (2.1)$$

where i is the index of variables, the state variable $u = (u_i)$. $F_i(u) = (f_i(u), g_i(u))$ is the flux vector. Assume that the domain Ω is subdivided into a collection of non-overlapping elements Ω_e . In [31], Huynh introduced a new approach to higher-order accuracy for the numerical solution of conservation laws by using flux reconstruction (FR). In the FR framework, a flux polynomial was defined by interpolating the values at the solution points. Then a flux correction function was employed to account for the flux transportation between adjacent cells. In the further study [32], Wang introduced the penalty idea to construct the correction function. In this subsection, we unify DG methods and the DG/FV hybrid schemes via flux divergence reconstruction (FDR). Different from the FR framework, FDR constructs flux divergence directly locally using Riemann flux as boundary condition. In each Ω_e , the flux divergence $\nabla \cdot F_i$ is projected to a polynomial space of degree n , P_n , noted as $\hat{\nabla} \cdot F_i$. It means that $\hat{\nabla} \cdot F_i$ can be expressed locally as a piecewise polynomial of degree n

$$\hat{\nabla} \cdot F_i = \sum_{p=0}^{N-1} d_{i,p}^{(e)} B_p^{(e)}, \quad (2.2)$$

in which $N = (n+1)(n+2)/2$ for two-dimensional cases. In the following context, for convenience, we sometimes omit the superscript of cell marker $(\cdot)^{(e)}$, when there is no confusion. Here, B_p are the finite element basis functions, $d_{i,p}$ are the coefficients of flux divergence, which can be calculated via L_2 projection procedure.

Multiplying Eq. (2.2) by the test function B_q , integrating over the domain Ω_e , and performing integration by parts, we have

$$\int_{\Omega_e} d_{i,p} B_p B_q d\Omega = \int_{\Omega_e} F_i \cdot \nabla B_q d\Omega - \int_{\Gamma_e} (F_i \cdot \mathbf{n}) B_q d\Gamma, \quad (2.3)$$

where Γ_e denotes the boundary of Ω_e , \mathbf{n} is the unit outward normal vector.

Replacing the flux function $F_i \cdot \mathbf{n}$ with the approximate Riemann flux $h_i = \hat{F}_i(u^+, u^-, \mathbf{n})$, we have

$$M_i D_i = Rhs_i, \quad (2.4)$$

in which $M_i = (\langle B_p B_q \rangle)$ is the local mass matrix. $\langle B_p B_q \rangle$ is the inner product of basis functions in Ω_e , and $D_i = (d_{i,p})$ is the vector of the i th unknown. The right-hand-side term $Rhs_i = (rhs_{i,p})$

$$rhs_{i,p} = \int_{\Omega_e} F_i \cdot \nabla B_p d\Omega - \int_{\Gamma_e} h_i B_p d\Gamma. \quad (2.5)$$

Riemann flux \hat{F}_i is regarded as the boundary condition to reconstruct flux divergence $\hat{\nabla} \cdot F_i$ locally, illuminated in Fig. 1.

2.2 Taylor basis functions

In formula (2.5), the selection of basis function B_p is an issue. In [33], Luo et al. developed a DG method based on Taylor basis functions. Unlike the traditional DG methods,

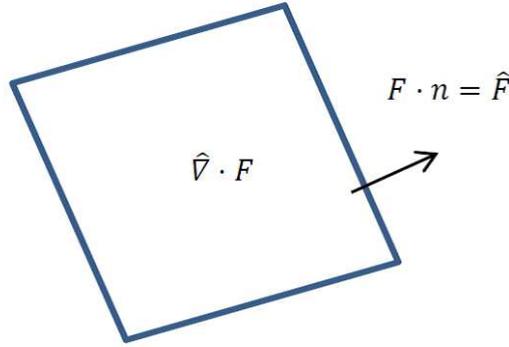


Figure 1: Flux divergence reconstruction.

this DG method represents the numerical polynomial solutions using a Taylor series expansion at the centroid of cell. The advantage of Taylor basis is uniform for different element shape. In this paper, we choose an orthogonal, and hierarchical basis for arbitrary grid based on linear transformation. And the hierarchical structure is suitable for the complementation of p -multigrid approach [49].

If we take Taylor series expansion for function f at the centroid of the cell Ω_e , the piece-wise polynomial solution f_h can be expressed as follows:

$$f_h = \sum_{q=0}^k \frac{1}{q!} \left(\xi \frac{\partial}{\partial \xi} + \eta \frac{\partial}{\partial \eta} \right)^q f_c = \sum_{q=0}^k \sum_{r=0}^q C_q^r \xi^{q-r} \eta^r \frac{\partial^q f_c}{\partial \xi^{q-r} \partial \eta^r}, \tag{2.6}$$

in which f_c means the centroid value, ξ, η are the local coordinates

$$\begin{cases} \xi = \frac{x-x_c}{h}, \\ \eta = \frac{y-y_c}{h}, \end{cases} \tag{2.7}$$

(x_c, y_c) denotes the centroid coordinate, $h := \sqrt{|\Omega_e|}$ is defined as cell characteristic scale. Furthermore, f_h can be expressed as the cell-averaged value and its derivatives about local coordinates at the centroid of the cell

$$f_h = \bar{f} + \sum_{q=0}^k \sum_{r=0}^q C_q^r (\xi^{q-r} \eta^r - \langle \xi^{q-r} \eta^r \rangle) \frac{\partial^q f_c}{\partial \xi^{q-r} \partial \eta^r}. \tag{2.8}$$

Here, \bar{f} is the cell-averaged value. $\langle \xi^m \eta^n \rangle$ is called as the centroid moment with $m-n$ rank on Ω_e . It was defined as

$$\langle \xi^m \eta^n \rangle = \frac{1}{|\Omega_e|} \int_{\Omega_e} \xi^m \eta^n d\Omega = \frac{1}{|\Omega_e|} \int_{\Omega_e} \frac{(x-x_c)^m (y-y_c)^n}{h^{m+n}} d\Omega. \tag{2.9}$$

$\langle \xi^m \eta^n \rangle$ can be calculated exactly with Gauss quadrature procedure, stored at the initial step.

For the fourth order DG scheme in two-dimensional case, the total number of degree of freedoms, $N = 10$, and the Taylor basis functions can be written explicitly as

$$B_0 = 1, \quad B_1 = \xi, \quad B_2 = \eta, \quad B_3 = \frac{1}{2}(\xi^2 - \langle \xi^2 \rangle), \quad (2.10a)$$

$$B_4 = \xi\eta - \langle \xi\eta \rangle, \quad B_5 = \frac{1}{2}(\eta^2 - \langle \eta^2 \rangle), \quad B_6 = \frac{1}{6}(\xi^3 - \langle \xi^3 \rangle), \quad (2.10b)$$

$$B_7 = \frac{1}{2}(\xi^2\eta - \langle \xi^2\eta \rangle), \quad B_8 = \frac{1}{2}(\xi\eta^2 - \langle \xi\eta^2 \rangle), \quad B_9 = \frac{1}{6}(\eta^3 - \langle \eta^3 \rangle). \quad (2.10c)$$

2.3 Review of DG method and DG/FV hybrid schemes

Another issue in formula (2.5) is how to construct the state variable distribution to compute the convection term and the Riemann flux at Gauss quadrature points. Like flux divergence, we project the state variables to P_n space locally,

$$u_i^{(e)} = \sum_{p=0}^{N-1} u_{i,p}^{(e)} B_p^{(e)}. \quad (2.11)$$

$u_{i,p}^{(e)}$ is called as the degrees of freedom (DOFs). The discretization form of governing equation can be re-written as

$$\int_{\Omega_e} \left(\frac{du_i^{(e)}}{dt} + \hat{\nabla} \cdot F_i^{(e)} \right) d\Omega = 0. \quad (2.12)$$

Substitute into Eq. (2.2),

$$M_i \frac{d}{dt} (u_{i,p}) = -rhs_{i,p}. \quad (2.13)$$

Eq. (2.13) is equivalent to the weak formulation of DGM. As well-known, one of the distinguished properties of DGM is the ‘‘compactness’’, since it constructs the higher-order piece-wise polynomial locally, instead of extending the grid stencil like the k -exact FVM. As mentioned in Section 2.1, constructing piecewise polynomial in each cell is the key step for numerical method. In DGMs, all DOFs are updated as time evolution (called as ‘‘dynamic reconstruction’’ in Refs. [28–30]), while in FVMs, the high-order derivatives are computed at the final output time (called as ‘‘static reconstruction’’). Generally, the ‘‘dynamic reconstruction’’ needs more CPU time and memory requirement to deal with all the DOFs. However, the ‘‘static reconstruction’’ operator is unnecessary to update all DOFs at each time step, most of the DOFs can be computed by reconstruction with the help of neighboring cells, so a larger grid stencil should be adopted for higher-order reconstruction, which results in a non-compact method. Naturally, a ‘‘hybrid reconstruction’’ approach can be adopted to achieve some balance between the ‘‘compact’’ property

and the computational cost. A higher-order polynomial distribution of state variables can be written as

$$v_i^{(e)} = \sum_{p=0}^{M-1} v_{i,p}^{(e)} B_p^{(e)}. \quad (2.14)$$

Thanks to the hierarchy of Taylor basis functions, $v_{i,p}^{(e)} = u_{i,p}^{(e)}$, when $0 \leq p \leq N-1$. Therefore, the higher-order coefficients $v_{i,p}^{(e)}$, ($N \leq p \leq M-1$) can be reconstructed like the traditional FVMs. For example, the first order derivatives can be taken to construct the second order derivatives with Green's theorem, such as:

$$u_{xx} = \frac{1}{|\Omega_G|} \int_{\Omega_G} \frac{\partial^2 u}{\partial x^2} d\Omega = \frac{1}{|\Omega_G|} \sum_{\Gamma_l \in \partial\Omega_G} \int_{\Gamma_l} \frac{\partial u}{\partial x} n_x d\Gamma. \quad (2.15)$$

Similarly,

$$u_{yy} = \frac{1}{|\Omega_G|} \int_{\Omega_G} \frac{\partial^2 u}{\partial y^2} d\Omega = \frac{1}{|\Omega_G|} \sum_{\Gamma_l \in \partial\Omega_G} \int_{\Gamma_l} \frac{\partial u}{\partial y} n_y d\Gamma. \quad (2.16)$$

The cross-derivatives can be calculated by Eq. (2.17) or Eq. (2.18). In our code, we take an arithmetic mean to approximate the cross-derivatives

$$u_{xy} = \frac{1}{|\Omega_G|} \int_{\Omega_G} \frac{\partial^2 u}{\partial x \partial y} d\Omega = \frac{1}{|\Omega_G|} \sum_{\Gamma_l \in \partial\Omega_G} \int_{\Gamma_l} \frac{\partial u}{\partial x} n_y d\Gamma, \quad (2.17)$$

$$u_{yx} = \frac{1}{|\Omega_G|} \int_{\Omega_G} \frac{\partial^2 u}{\partial y \partial x} d\Omega = \frac{1}{|\Omega_G|} \sum_{\Gamma_l \in \partial\Omega_G} \int_{\Gamma_l} \frac{\partial u}{\partial y} n_x d\Gamma. \quad (2.18)$$

The choice of the Gauss integral region Ω_G is the main issue of this static reconstruction. In this paper, we take the target cell as Ω_G , it means $\Omega_G = \Omega_e$. The first-order derivatives at nodes of the target cell are calculated from the neighboring cells using a weighted-average approach. Here, the weight function is chosen as the inverse of the cell-center-to-node distance. We call this static reconstruction operator as the cell-vertex type scheme. The designed accuracy of this kind DG/FV hybrid scheme is the third order one, and named as DG/FV3. Naturally, we can compute $u_{xxx}, u_{xxy}, u_{xyy}, u_{yyy}$ using in the same way, which is designed to achieve the fourth order, and named as DG/FV4. Anyway, other kinds of Gauss integral region can be adopted here, which had been shown in [29]. Moreover, the least-squares approach can be adopted also to calculate the higher-order derivatives, as shown in [24, 29].

Following this idea, the authors had successfully developed a class of DG/FV hybrid schemes for one-dimensional linear/non-linear scalar equation and Euler equations in Ref. [28], and for two-dimensional scalar equations and Euler equations on arbitrary grids in Refs. [29, 30], respectively. More details can be found in these papers. For Navier-Stokes equations, the second order spatial derivatives for viscous term in the governing equations are eliminated by introducing a first-order auxiliary variable $S = \nabla u$. Thus, the

original system is now rewritten as a component form of two first-order equations. In this paper, the viscous flux is computed with the BR2 scheme (more details can be found in Ref. [38]).

3 Implicit algorithm based on Newton/LU-SGS iteration

Assembling all DOFs of each state variable, each cell, Eq. (2.13) can be written as the following ODE form,

$$M \frac{dU}{dt} = R(U), \quad (3.1)$$

in which $U = (U_e)$, $U_e = (U_i^{(e)})$, $U_i^{(e)} = (u_{i,p}^{(e)})$ and $R(U) = (Rhs_e)$, $Rhs_e = (Rhs_i^{(e)})$, $Rhs_i^{(e)} = (rhs_{i,p}^{(e)})$. M is the global mass matrix, which is block diagonal. The explicit multi-stage Runge-Kutta iteration scheme is usually adopted to solve the above ODE formula. However, the explicit method is severely limited by the time step because of the restriction of the CFL-stability. In this paper, we focus on the development of implicit time-integration schemes to solve Eq. (3.1). Generally, an implicit algorithm can be divided into three levels as follows.

3.1 The first level: temporal iteration

If we choose the first-order backward Euler difference scheme for time iteration, Eq. (3.1) can be written as

$$M \frac{\Delta U^{n+1}}{\Delta t} = R(U^{n+1}), \quad (3.2)$$

where Δt is the time increment and ΔU is the difference of an unknown vector between the n th and $(n+1)$ th time steps. Obviously, in the first level, the first-order backward-difference Euler scheme is enough for steady flows, since we do not care about the time accuracy during iteration procedure. To ensure the time accuracy for unsteady flows, other kinds of time integration schemes, such as the multi-step backward difference formulations, the second-order accurate Crank-Nicolson scheme, or the implicit multi-stage Runge-Kutta schemes are alternative. After the temporal iteration, it will result in a nonlinear system about U^{n+1} . Then, the second level iteration strategy is employed to solve the nonlinear system.

3.2 The second level: Newton iteration

Considering (3.2) as an example, we define a non-linear unsteady residual $Res(W)$, for the corresponding backward-difference Euler scheme as

$$Res(W) = W - U^n - \Delta t M^{-1} R(W). \quad (3.3)$$

The solution of the scheme can be achieved by solving the non-linear problem if $R(W)=0$ at each time step. Then, Newton iteration scheme can be written as

$$\begin{cases} \frac{\partial Res}{\partial W} \Delta W^{m+1} = -Res(W^m), \\ W^{m+1} = W^m + \Delta W^{m+1}, \end{cases} \quad m=0,1,2,\dots, \quad (3.4)$$

where m refers to the sub-iteration index for the linearized system in the Newton iteration scheme. The vector W represents intermediate solutions in the Newton solver for the solution of U^{n+1} . The initial guess for W is set as $W^0 = U^n$. Note that if Eq. (3.4) is solved to machine zero, the unsteady residual $Res(W^{m+1})=0$, then W^{m+1} will converge to U^{n+1} . If only one step of the Newton iteration is employed, Eq. (3.4) is equivalent to residual linearization. For steady state problems, it is not necessary to drive the unsteady residual to machine zero in each time step. In fact, if only one Newton sweep is adopted here, it also can achieve a convergence solution for steady flows. In the numerical test, we found that properly more Newton sweeps would be more efficient even for steady flows. The main purpose of choosing Newton iteration algorithm is to unify the implicit treatment of steady and unsteady solvers, instead of the traditional dual-time stepping technique [51]. For unsteady cases, a higher-order temporal scheme could be employed in first level.

The term $\partial Res/\partial W$ represents symbolically Jacobian matrix of unsteady residual,

$$\frac{\partial Res}{\partial W} = I - \Delta t M^{-1} \frac{\partial R}{\partial W}, \quad (3.5)$$

where $\partial R/\partial W$ is the Jacobian matrix of spatial discretization procedure, which is a large, sparse block matrix, since the spatial discretization of both DG methods and the DG/FV hybrid schemes are only depended on adjacent cells. Eq. (3.4) represents a system of linear algebraic equations and needs to be solved at each Newton sweep. It is impossible to solve directly by matrix inversion because of the large Jacobian matrix. Then a third level iteration should be employed to solve this linear system.

3.3 The third level: lower-upper symmetric Gauss-Seidel iteration

The lower-upper symmetric Gauss-Seidel (LU-SGS) algorithm proposed firstly by Jameson and Yoon [52] on structured grids has been successfully generalized for FVMs. The LU-SGS algorithm is attractive because of its good stability properties and competitive computation cost in comparison with explicit method. So we employ a LU-SGS sweep to solve Eq. (3.4) approximately in this work. In LU-SGS, the Jacobian matrix of residual $\partial R/\partial W$, can be split into three (block) matrices, a strict lower matrix, a diagonal matrix, and a strict upper matrix, respectively,

$$\frac{\partial R}{\partial W} = \mathcal{L} + \mathcal{D} + \mathcal{U}, \quad (3.6)$$

where

$$\mathcal{L} = (r_{ek})_{e < k} = \frac{\partial R(W_e, W_k)}{\partial W_k}, \quad (3.7a)$$

$$\mathcal{D} = (r_{ek})_{e=k} = \frac{\partial R(W_e, W_k)}{\partial W_k}, \quad (3.7b)$$

$$\mathcal{U} = (r_{ek})_{e > k} = \frac{\partial R(W_e, W_k)}{\partial W_k}. \quad (3.7c)$$

Then, the Newton scheme can be written as

$$(I - \Delta t M^{-1}(\mathcal{L} + \mathcal{D} + \mathcal{U})) \Delta W^{m+1} = -Res(W^m). \quad (3.8)$$

If we choose a LU-SGS iteration to solve Eq. (3.8), we can employ the following forward and backward sweeps:

$$(I - \Delta t M^{-1}(\mathcal{L} + \mathcal{D})) \Delta W^* = -Res(W^m), \quad (3.9a)$$

$$(I - \Delta t M^{-1}(\mathcal{U} + \mathcal{D})) \Delta W^{m+1} = -Res(W^m) + \Delta t M^{-1} \mathcal{L} \Delta W^*. \quad (3.9b)$$

3.4 The matrix-free strategy

To avoid saving the lower and upper matrix, the matrix-free strategy is employed here. Following the idea of a matrix-free approach, the product of Jacobian matrix and the incremental vector is approximated by the increment of the flux vector

$$\begin{cases} \sum_{e > k} r_{ek} \Delta W_k^{m+1} \approx Rhs_e(W_e^m, W_{e < k}^m, W_{e > k}^{m+1}) - Rhs_e(W_e^m, W_k^m), \\ \sum_{e < k} r_{ek} \Delta W_k^{m+1} \approx Rhs_e(W_e^m, W_{e < k}^{m+1}, W_{e > k}^m) - Rhs_e(W_e^m, W_k^m), \end{cases} \quad m = 0, 1, 2, \dots \quad (3.10)$$

3.5 Calculation of the diagonal Jacobian matrix

The diagonal Jacobian matrix can be calculated by an analytical or numerical method, at least for Euler equations. In this work, the numerical method is employed following the idea in Ref. [42]

$$r_{ee} = \frac{\partial rhs_{i,p}(u_{j,q}^{(e)}, u_{j,q}^{(k)})}{\partial u_{j,q}^{(e)}} = \frac{rhs_{i,p}(u_{j,q}^{(e)} + \varepsilon, u_{j,q}^{(k)}) - rhs_{i,p}(u_{j,q}^{(e)}, u_{j,q}^{(k)})}{\varepsilon}, \quad (3.11)$$

where ε is a small parameter, e.g., $\varepsilon = 1 \times 10^{-8}$. Although this approach is very easy to implement for an arbitrarily complex residual operator like viscous term, or turbulence model, it is quite expensive because each variable, as well as each degree of freedom, has to be computed. Fortunately, it is not necessary to compute the matrix in each iteration. In practice, we update the matrices every 10~100 time steps. Numerical tests in the following section show that this matrix-frozen approach does not significantly degrade the convergence rate for steady flow.

4 Numerical results and discussion

In this section, several typical steady cases are selected to illustrate the performance of the proposed implicit algorithm for the 2D Euler and Navier-Stokes equations. The computational cost in the test cases represent the actual CPU time (second) on an Intel® core™ i5-760(2.8GHZ) machine running the Windows system.

4.1 Couette flow

Couette flow between two parallel walls is used to evaluate the accuracy of the DG/FV hybrid method on triangle grids. The upper wall moves at a constant speed, and the lower one is static. The computational domain is a 4×2 rectangle as shown in Fig. 2. The exact solution for this case is

$$u = \frac{y}{H}, \quad v = 0, \quad (4.1a)$$

$$p = \text{const} = \frac{1}{\gamma Ma_\infty^2}, \quad (4.1b)$$

$$T = T_0 + \frac{y}{H}(T_1 - T_0) + \frac{\text{Pr}(\gamma - 1)}{2Ma_\infty^2} \frac{y}{H} \left(1 - \frac{y}{H}\right). \quad (4.1c)$$

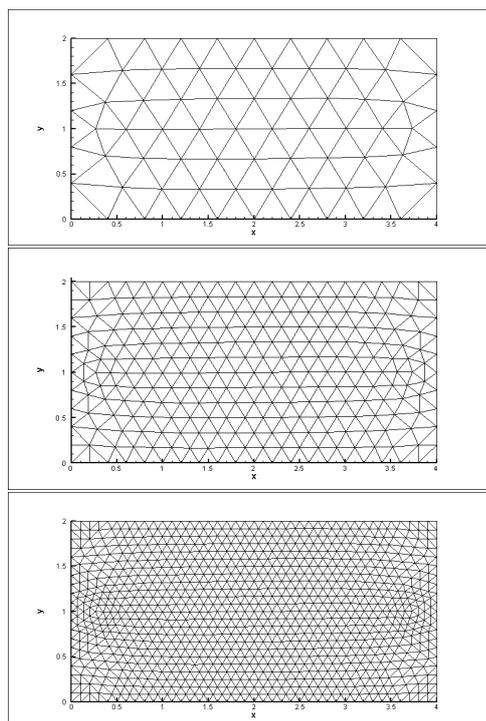


Figure 2: Computational grids for Couette flow (Top: Grid1; Middle: Grid2; Bottom: Grid3).

Table 1: Accuracy study and Comparison of efficiency by DGM and DG/FV for Couette flow.

		L_2 error	Order	CPU cost on Grid1
2^{nd} DGM	Grid level1	5.63E-06	—	3.68E-02
	Grid level2	1.41E-06	1.99	
	Grid level3	3.56E-07	1.99	
3^{rd} DGM	Grid level1	1.47E-08	—	6.46E-02
	Grid level2	1.83E-09	2.99	
	Grid level3	2.31E-10	2.99	
4^{th} DGM	Grid level1	2.70E-11	—	6.46E-02
	Grid level2	1.79E-12	3.87	
	Grid level3	1.40E-13	3.68	
3^{rd} DG/FV	Grid level1	8.31E-08	—	4.66E-02
	Grid level2	7.72E-09	2.89	
	Grid level3	1.01E-09	2.93	
4^{th} DG/FV	Grid level1	3.15E-10	—	8.28E-02
	Grid level2	1.39E-11	3.89	
	Grid level3	9.36E-13	3.90	

The following parameters are chosen as [53]: the temperature of the lower wall $T_0 = 0.8$, the upper wall temperature $T_1 = 0.85$, the domain size in y direction $H = 2$. The Mach number Ma_∞ of upper wall is set to be 0.1 and Reynolds number is set to be 100. The flow variables at boundary faces are fixed to the exact solution, and the initial condition is set to be a uniform flow, i.e. $(\rho, u, v, T) = (1, 1, 0, 1)$. Three meshes generated by an h-refinement manner are used to take an accuracy study, as shown in Fig. 2. The pressure is used to measure the order of accuracy, since it is constant as an exact solution. The accuracy of results for different DG and DG/FV hybrid schemes, and CPU cost per time step are shown in Table 1. We can see that the DG/FV hybrid schemes have achieved the designed 3rd and 4th order of accuracy, even though the absolute errors of DG/FV hybrid schemes are relatively larger than those of same-order DG scheme. It is reasonable, since the DG/FV hybrid schemes extend the reconstruction stencil to the neighboring cells, and DGM is constructed in the local cell. However, the reconstruction cost of DG/FV is more efficient than that of DGM, because less Gauss points are required to calculate the right hand side in the DG/FV hybrid schemes. Fig. 3 shows the convergence histories of different schemes on Grid1 for both the 3-stage Runge-Kutta explicit scheme and the present Newton/LU-SGS implicit scheme. We can observe that the convergence rate with the implicit scheme is more efficient than the explicit one. In addition, the DG/FV hybrid schemes consume less CPU time compared with the same order DG schemes, even though they are almost the same with respect to the iteration steps.

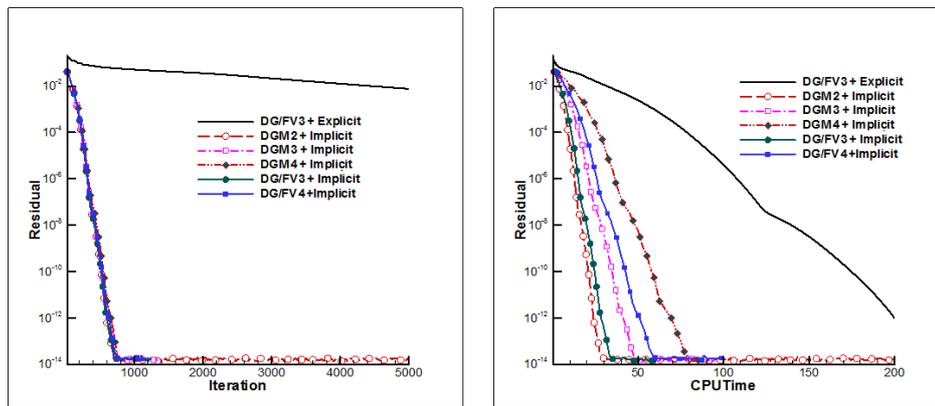


Figure 3: Residual histories of Couette flow for different schemes on Grid1 (Left: Residual vs. iteration step; Right: Residual vs. CPU Time.)

4.2 Internal inviscid flow over a smooth bump

The bump shape is defined as

$$y = 0.0625e^{-25x^2}.$$

The numerical simulations are carried out on a triangular mesh as shown in Fig. 4. The finer grids are generated recursively by refining each coarser grid cell into four finer grid cells, denoted as level-1, level-2, level-3, level-4 grids, respectively. Anyway, the grid points on the bump are modified onto the curve boundary. The level-1 grid has 120 cells. All the simulations start from a uniform free stream with Mach number 0.5 everywhere on the four level grids. The relative L_2 norm of the density residual is taken as a criterion to test the convergence history. Fig. 5 shows the Mach number contours by DGM2 and DG/FV4 on level-2 grid. The Mach number contours by DGM3, DGM4 and DG/FV3 are very similar to those by DG/FV4, so we do not show them here. It can be seen that the flow field is more symmetric with accuracy increasing.

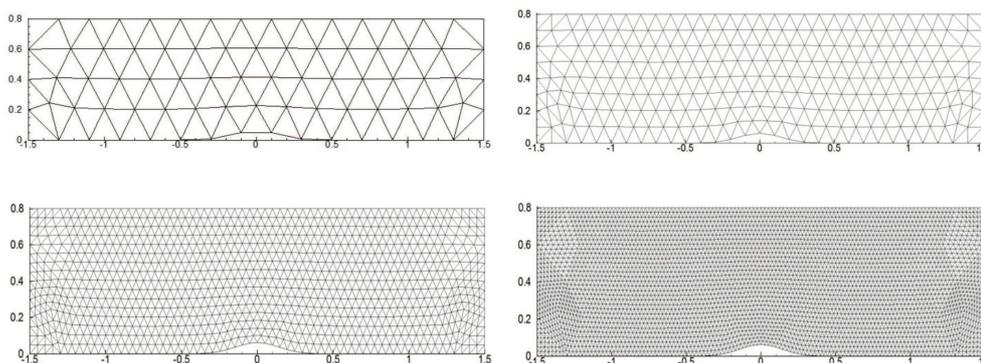


Figure 4: The computation grid for bump problem.

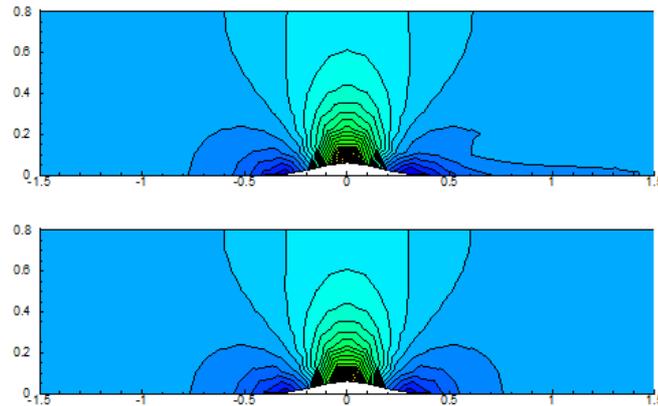


Figure 5: Mach number contours on level-2 grid (0.45~0.78 with 30 lines, top: DGM2, bottom: DG/FV4).

To quantify the results, we take the L_2 errors in entropy production as the accuracy indicator until achieving steady state, which is defined as:

$$Error_{L_2(\Omega)} = \sqrt{\frac{\int_{\Omega} (S - S_{\infty})^2 d\Omega}{\int_{\Omega} S_{\infty}^2 d\Omega}},$$

where the entropy $S = p/\rho^{\gamma}$. Table 2 lists the computed errors of entropy production. Note that the optimal accuracy has been almost achieved, except DGM4 and DG/FV4. Generally, the curve boundaries are represented with straight line segment in the 2nd order CFD solvers. However, for higher-order solvers, this simple geometry representation don't match with spatial discretization, which may result in non-physical solution, even make the solvers unstable. In [54], a curve modification technique was proposed to implement solid wall boundary conditions. In this work, the Algorithm II in [54] is adopted, where the curved element are not required, instead of an accurate representation of the boundary normal at the quadrature points to impose the solid wall boundary condition for curve geometries. In our implementation, the normal at the quadrature points are computed using the local true surface normal based on the analytically defined boundary geometries. Indeed, the curve modification is helpful to improve flow quality, and can match with the 3rd order schemes. However, for the 4th order schemes, the accuracy degrades. So more accurate algorithm for curve boundary should be studied further.

Fig. 6 shows the convergence histories of different schemes on level-4 grid for both the 3-stage Runge-Kutta explicit scheme and the present Newton/LU-SGS implicit scheme. From Fig. 6, we can observe that the convergence rate with the implicit scheme is more than an order of magnitude faster than the explicit one. In addition, the DG/FV hybrid schemes consume less CPU time, compared with the same order DG schemes (see Fig. 6), even though they are almost same with respect to the iteration steps. Fig. 7 displays the convergence histories of DGM4 on different grids (level-1 to level-4), in which we employ

Table 2: Accuracy study of DGMs and the DG/FV hybrid schemes for the bump case.

		L_2 error	Order
2^{nd} DGM	Grid level1	3.017E-03	—
	Grid level2	5.636E-04	2.42
	Grid level3	1.206E-04	2.22
	Grid level4	2.429E-05	2.31
3^{rd} DGM	Grid level1	4.865E-04	—
	Grid level2	4.491E-05	3.44
	Grid level3	4.841E-06	3.21
	Grid level4	4.966E-07	3.29
4^{th} DGM	Grid level1	6.961E-05	—
	Grid level2	2.559E-06	4.77
	Grid level3	2.249E-07	3.51
	Grid level4	3.368e-08	2.74
3^{rd} DG/FV	Grid level1	4.884E-04	—
	Grid level2	4.779E-05	3.35
	Grid level3	5.533E-06	3.11
	Grid level4	6.810E-07	3.02
4^{th} DG/FV	Grid level1	6.900E-05	—
	Grid level2	3.429E-06	4.33
	Grid level3	3.491E-07	3.30
	Grid level4	4.295E-08	3.02

the logarithm coordinate of iteration steps and CPU time in the x -axis to demonstrate the convergence history more clearly. It can be seen that the performance of this implicit schemes are very similar on different grids. Next, we study the effects of several parameters on the convergence rate. These simulations are carried out on level-2 grid. Obviously the maximum allowable time step is an important convergence parameter. In this work, the CFL ratio is taken as the criterion of implicit maximum time, it is defined as

$$CFL_ratio = \frac{dt_{imp}}{dt_{exp}}$$

in which dt_{exp} is the maximum allowable time step of the RK explicit scheme. The effects of CFL_ratio on the convergence rates are shown in Fig. 8 for DGM2. It is easily observed that the convergence rate strongly depends on CFL_ratio . The larger CFL_ratio results in better convergence rate. However, excessively large CFL_ratio may also cause relatively lower convergence rate to steady state. The same conclusion is illustrated in Figs. 9-12, which show the convergence histories of other spatial schemes. Generally, the $CFL_ratio = 40\sim 100$ is appropriate. The second parameter on the convergence rate is the number

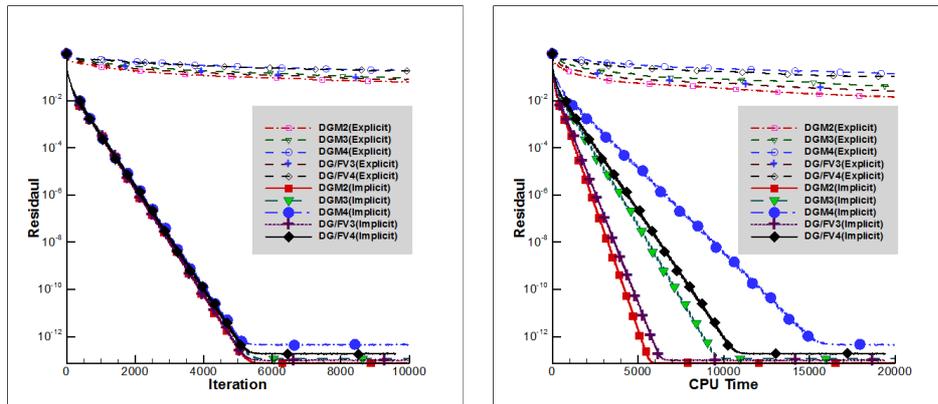


Figure 6: Residual histories of bump problem for different schemes on level-4 grid (Left: Residual vs. iteration step; Right: Residual vs. CPU Time).

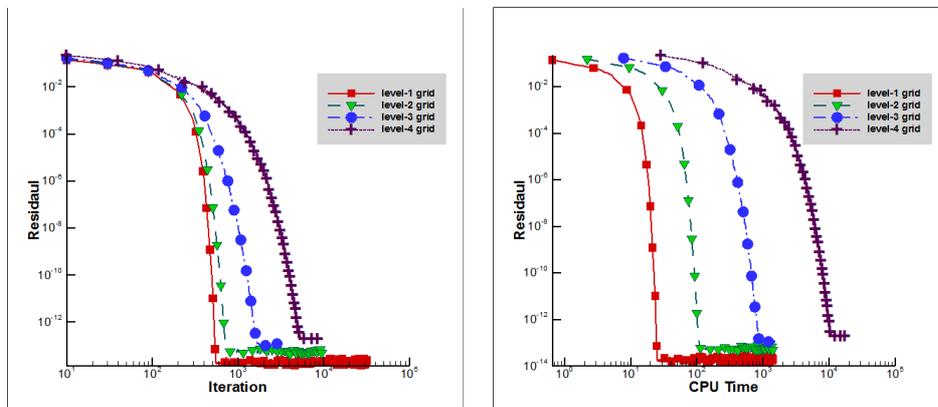


Figure 7: Residual histories of bump problem on different grids with DGM4 (Left: Residual vs. iteration step; Right: Residual vs. CPU Time).

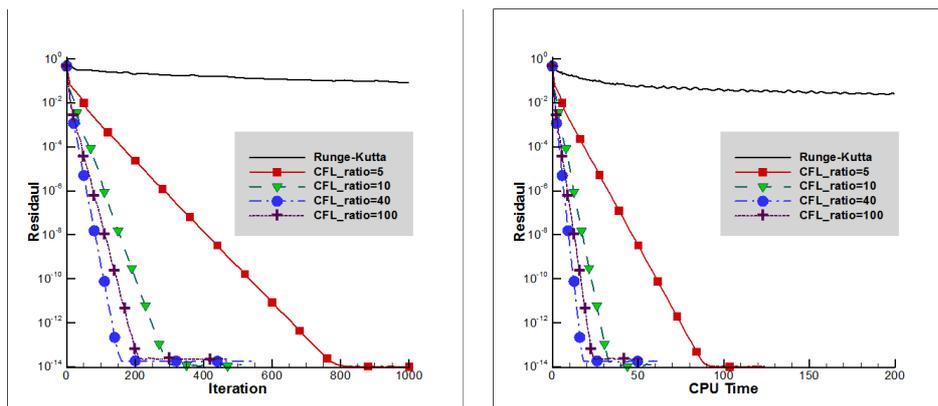


Figure 8: Effect of CFL_ratio for DGM2 (Left: Residual vs. iteration step; Right: Residual vs. CPU Time).

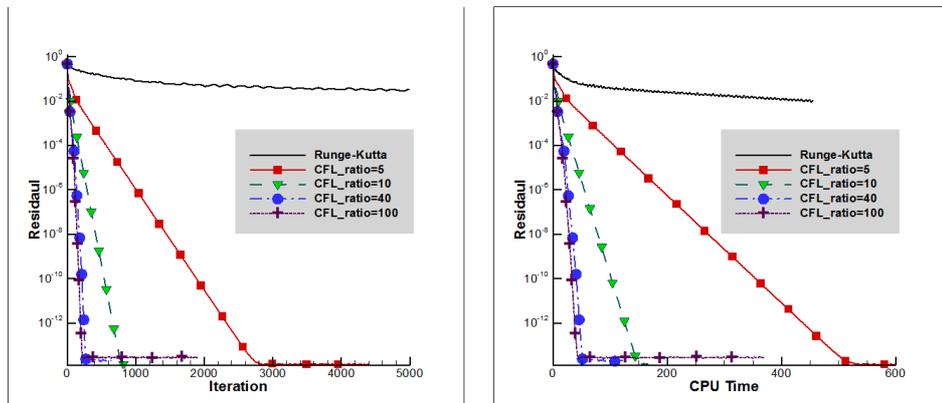


Figure 9: Effect of CFL_ratio for DGM3 (Left: Residual vs. iteration step; Right: Residual vs. CPU Time).

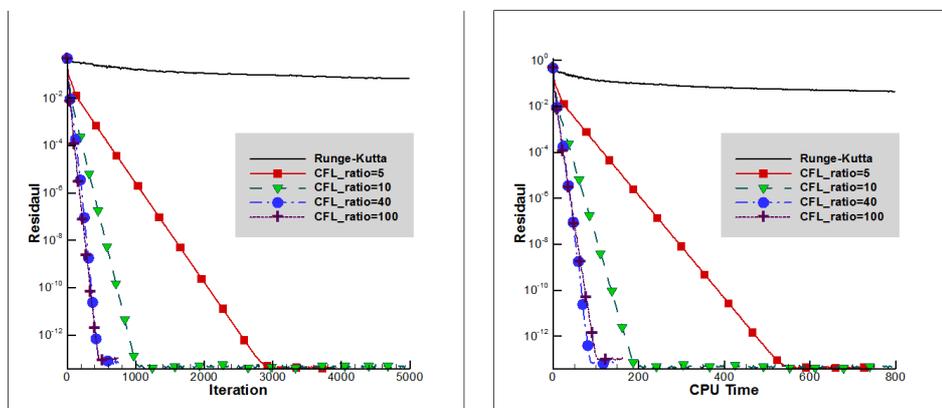


Figure 10: Effect of CFL_ratio for DGM4 (Left: Residual vs. iteration step; Right: Residual vs. CPU Time).

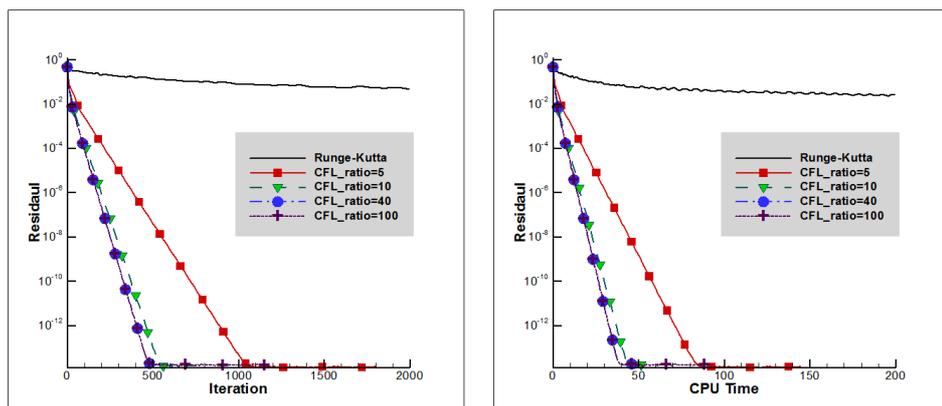
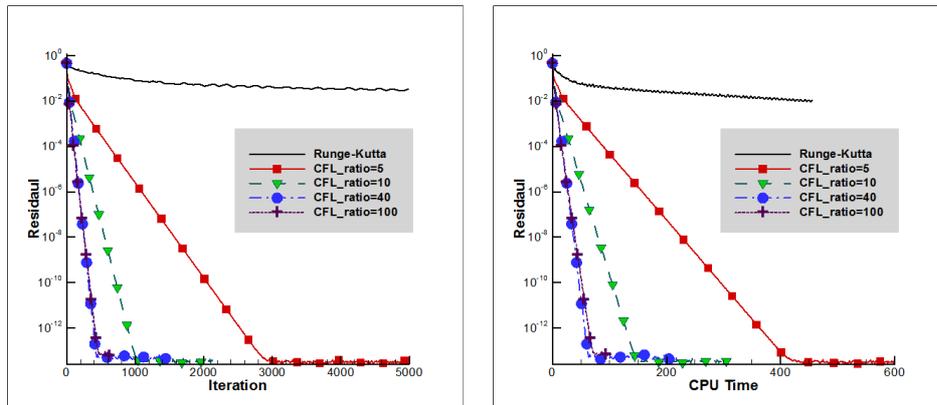
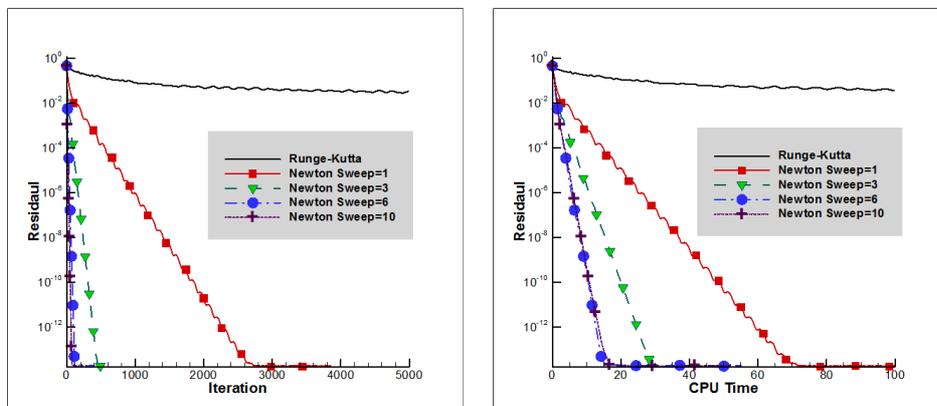


Figure 11: Effect of CFL_ratio for DG/FV3 (Left: Residual vs. iteration step; Right: Residual vs. CPU Time).

Figure 12: Effect of CFL_ratio for DG/FV4 (Left: Residual vs. iteration step; Right: Residual vs. CPU Time).Figure 13: Effect of $Newton_sweep$ for DGM2 (Left: Residual vs. iteration step; Right: Residual vs. CPU Time).

of Newton sweep steps in Section 3.2, denoted as $Newton_sweep$. For unsteady flow, the unsteady residual in each time step should be driven to machine zero by enough Newton sweeps. For steady state problems, it is unnecessary. In fact, if only one Newton sweep is adopted, it also can achieve a convergent solution. In this numerical test, we study the effect of Newton sweep steps for steady flow when CFL_ratio is set to be 40. The convergence histories are illustrated in Figs. 13-17. We can observe that the convergence rate depends either on $Newton_sweep$ even for steady flow. When $Newton_sweep = 1$, which means residual linearization, every time step consumes less CPU time, but more iteration steps should be carried out to steady state, the total convergence rate is not optimal. With $Newton_sweep$ increasing, the convergence efficiency is improved. Certainly, too much $Newton_sweep$ also results in lower convergence efficiency as shown in these figures. Generally, $Newton_sweep = 3\sim 10$ is preferable.

The computation of the Jacobian matrix is quite expensive as Eq. (3.11). To improve

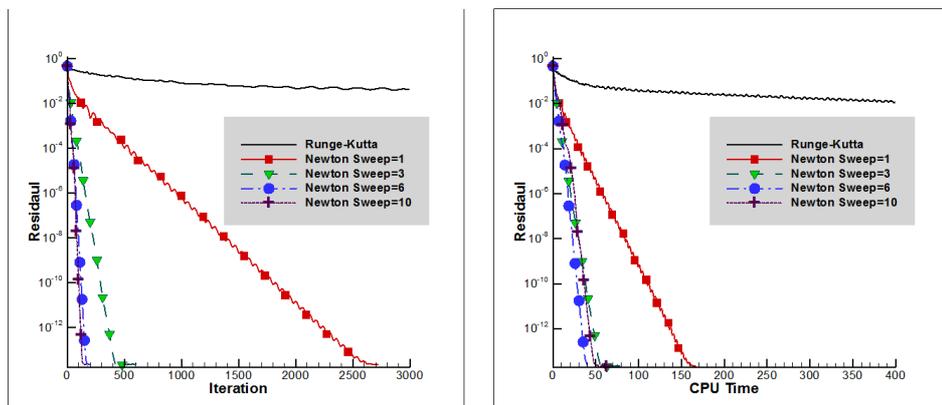


Figure 14: Effect of *Newton_sweep* for DGM3 (Left: Residual vs. iteration step; Right: Residual vs. CPU Time).

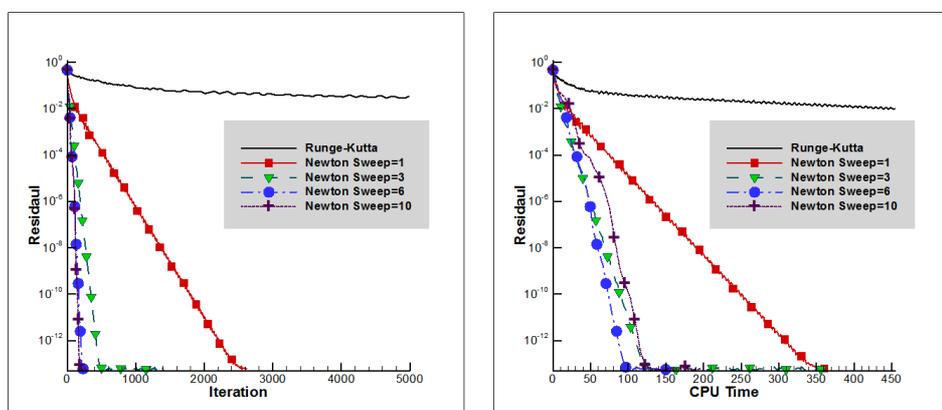


Figure 15: Effect of *Newton_sweeps* for DGM4 (Left: Residual vs. iteration step; Right: Residual vs. CPU Time).

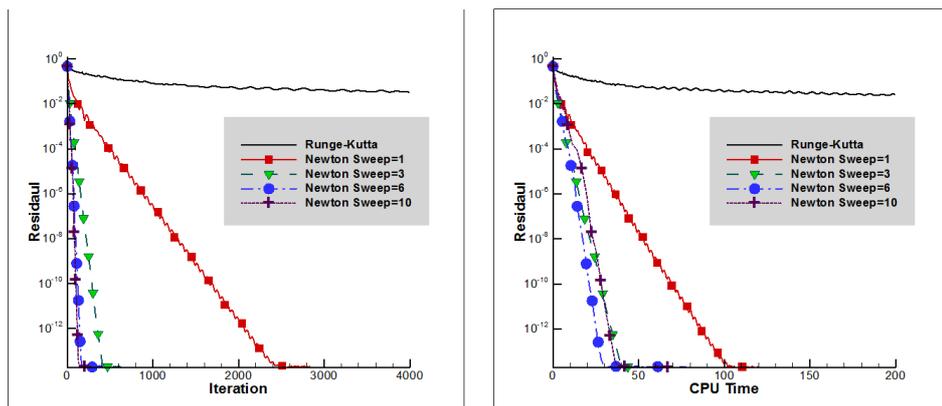


Figure 16: Effect of *Newton_sweep* (Left: Residual vs. iteration step; Right: Residual vs. CPU Time).

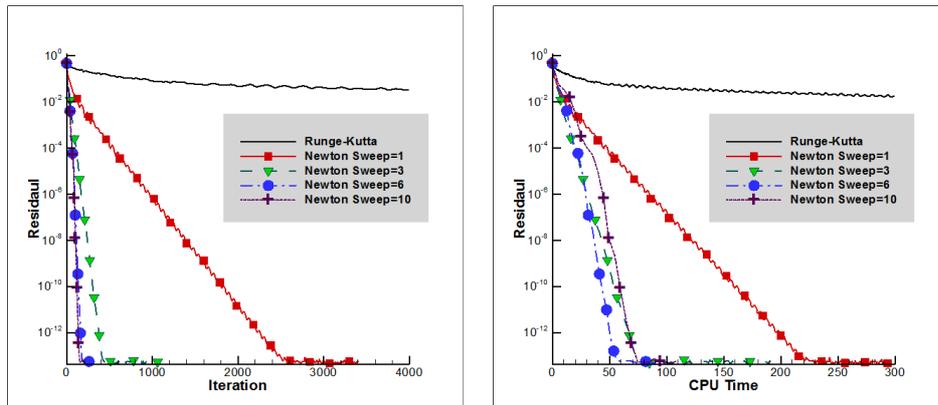


Figure 17: Effect of *Newton_sweep* for DG/FV4 (Left: Residual vs. iteration step; Right: Residual vs. CPU Time).

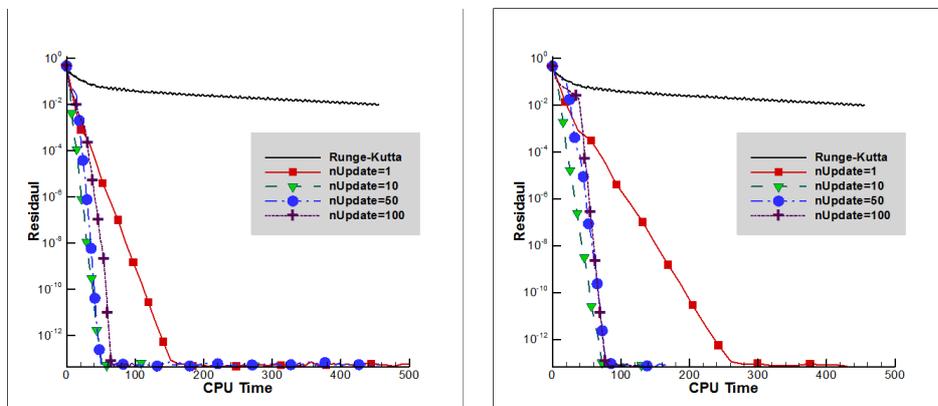


Figure 18: Effects of *nUpdate* for DGM4 and DG/FV4 (Left: Residual vs. iteration step; Right: Residual vs. CPU Time).

efficiency, the matrix can be frozen for several time steps. The update frequency of Jacobian matrix is denoted as *nUpdate*. Fig. 18 shows the convergence histories with different *nUpdate* for DGM4 and DG/FV4. In this test, the *CFL_ratio* is set to be 50, and *Newton_sweep*=6. We can observe that the convergence rate weakly depends on *nUpdate* when *nUpdate* \approx 10. When *nUpdate* = 1, it means that the matrix is computed every time step, which results in lower efficiency. With *nUpdate* increasing (more than 10), the computation efficiency is almost similar, since the saved cost of updating the matrix is counteracted during the residual convergence.

4.3 Subsonic flow over a cylinder

In this test case, we consider a subsonic flow around a circular cylinder on a hybrid triangular/quadrilateral grid to demonstrate the applicability of the implicit method for

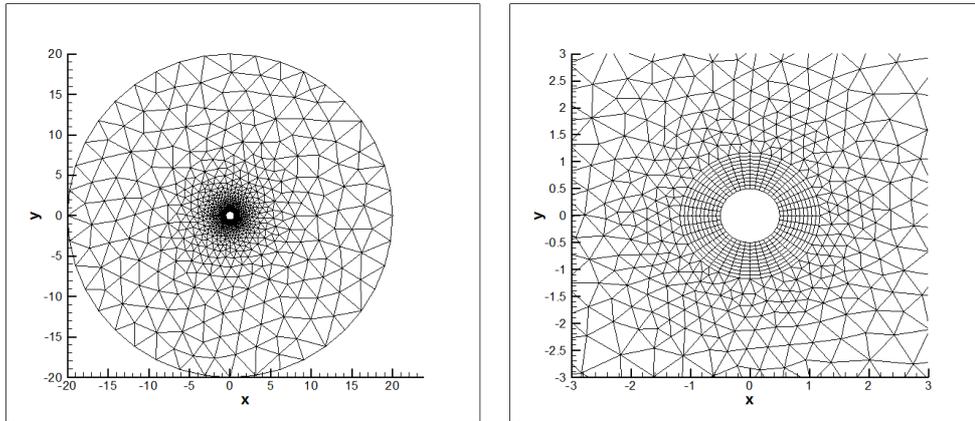
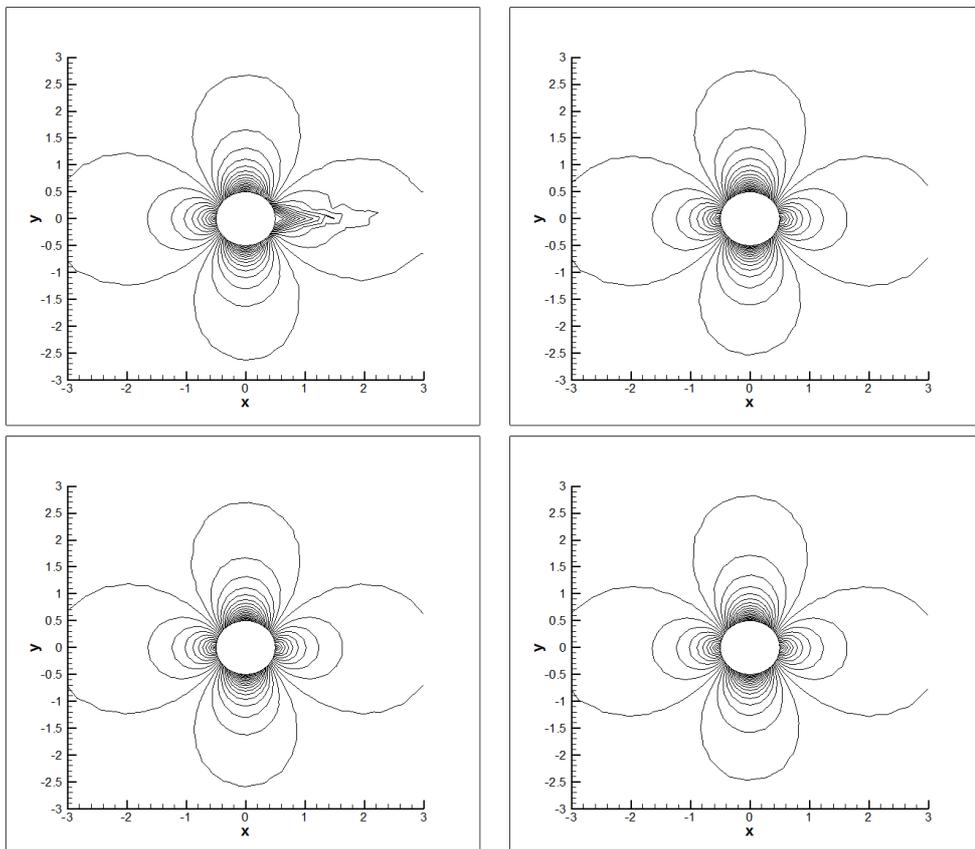


Figure 19: Hybrid grid over circular cylinder.

Figure 20: Mach number contours over a circular cylinder ($0 \sim 0.9$ with 30 lines). Top & Left: DGM2, Top & Right: DGM4, Bottom & Left: DG/FV3 Bottom & Right: DG/FV4.

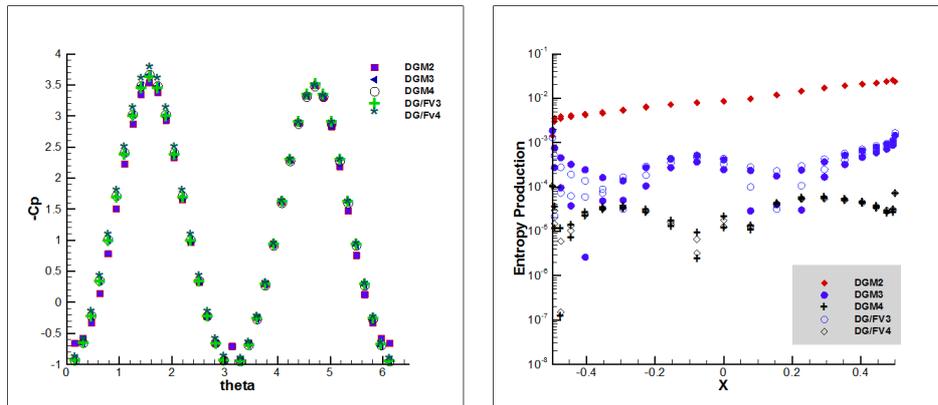


Figure 21: Pressure coefficient on the surface (left) and entropy production on the surface (right).

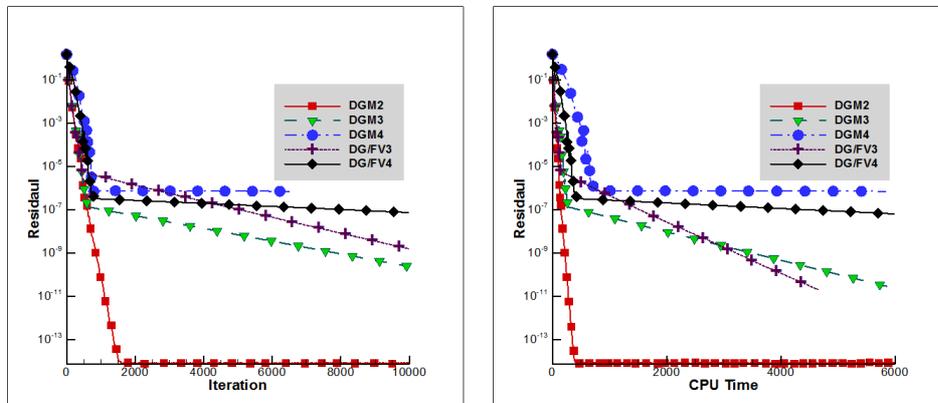


Figure 22: Convergence histories for flow over a circular cylinder.

hybrid grids. The Mach number of incoming flow is set to be 0.38. The mesh is shown in Fig. 19 with 800 quadrilateral cells and 2170 triangles. The quality of the numerical solution can be checked by the left-to-right symmetry of the computed Mach number contours around the cylinder, which is displayed in Fig. 20. It can be seen that the flow field by DG/FV4 is indeed more accurate than that by DGM2. Fig. 21(left) shows the pressure coefficients on the super surface of the cylinder. The entropy production distribution of each scheme on the upper surface is provided in Fig. 21(right). It can be seen that the distribution becomes more symmetric for higher-order schemes. Obviously, the higher-order schemes (both DG and DG/FV) result in less entropy production for inviscid flows as we expected. The residual convergence histories are illustrated in Fig. 22. We can observe that the residual can achieve machine zero efficiently for DGM2. For higher-order schemes, the convergence rate is very efficient before the residual is driven to about 1.0^{-7} , afterwards, the residuals drop down slowly. We guess that it be related to the curve boundary modification algorithm, as mentioned in the bump case (Section 4.2).

In that case, the curve boundary modification can work well for residual convergence since the curvature of the bump is small.

4.4 Low speed viscous flow in a square cavity

The last test involves a low speed laminar flow in a square cavity whose top wall moves with a uniform velocity in its own plane. This test is performed with Mach number of 0.1, and Reynolds number (Re) of 1000 and 10000, and by several DG and DG/FV schemes. For the case of $Re = 1000$, a hybrid grid with 640 quadrangular cells and 478 triangular cells is adopted, as shown in Fig. 23(left), while for the case of $Re = 10000$, a hybrid grid with 1440 quadrangular cells and 1110 triangular cells is adopted, as shown in Fig. 23(right). The results are compared with those in Ghia [55] with second-order FD scheme on 129×129 and 257×257 uniform structured grids, respectively. The computed streamlines agree well with the results in Ghia [55] as shown in Fig. 24 and Fig. 25. It can be seen clearly from Fig. 24 that, for the case of $Re = 1000$, there is a primary vortex in the center and two secondary vortices in left-bottom corner and right-bottom corner respectively. For the case of $Re = 10000$, there is a vortex in the left-top corner and additional corner vortices in the left-bottom and right-bottom corner. Fig. 26 and Fig. 27 show the velocity profiles for u along vertical line and v along horizontal line passing through the geometric center of the cavity when $Re = 1000$ and $Re = 10000$, respectively. It can be observed that the computed velocity profiles with the 4th order DG/FV4 scheme show good agreement with the results in Ghia [55] and by DGM4, even though the grids are one-order less than those in Ghia [55]. The residual convergence histories are illustrated in Fig. 28 and Fig. 29 for the cases of $Re = 1000$ and $Re = 10000$, respectively. Compared with the same-order DG schemes, the DG/FV hybrid schemes are more efficient, and this conclusion is the same as the former tests.

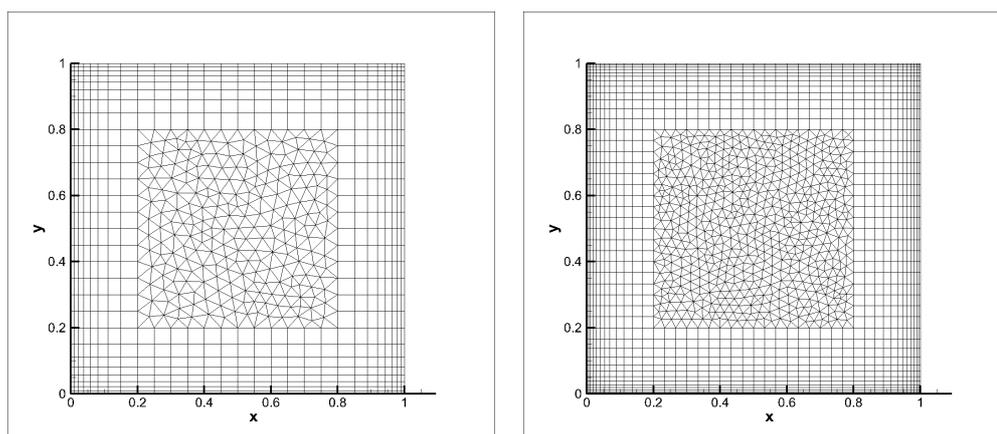


Figure 23: The hybrid grid in a square cavity. Left: Grid for the case of $Re = 1000$, Right: Grid for the case of $Re = 10000$.

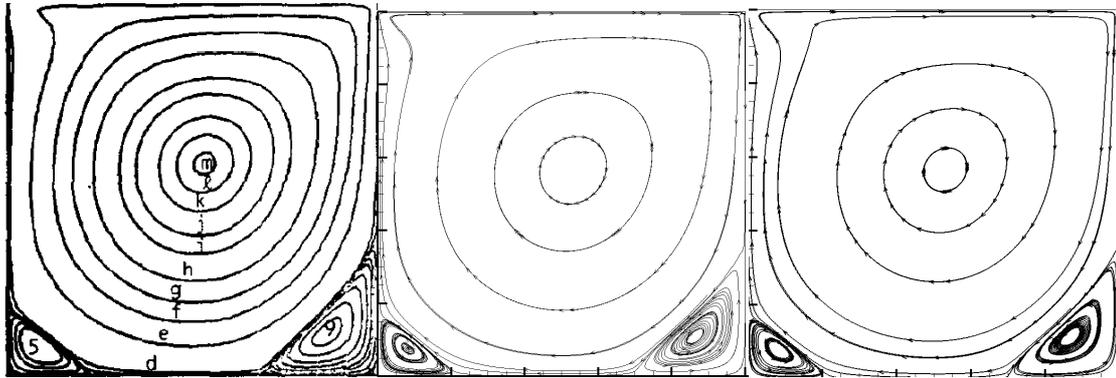


Figure 24: Streamlines of computation with $Re = 1000$.

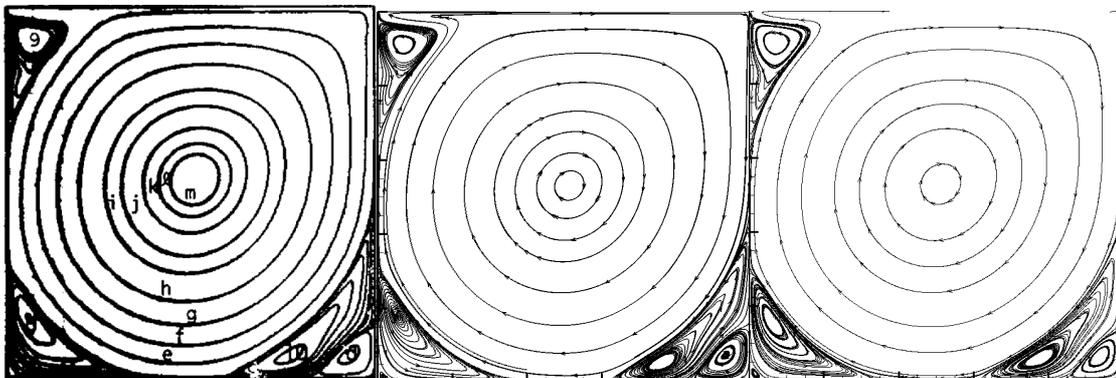


Figure 25: Streamlines of computation with $Re = 10000$.

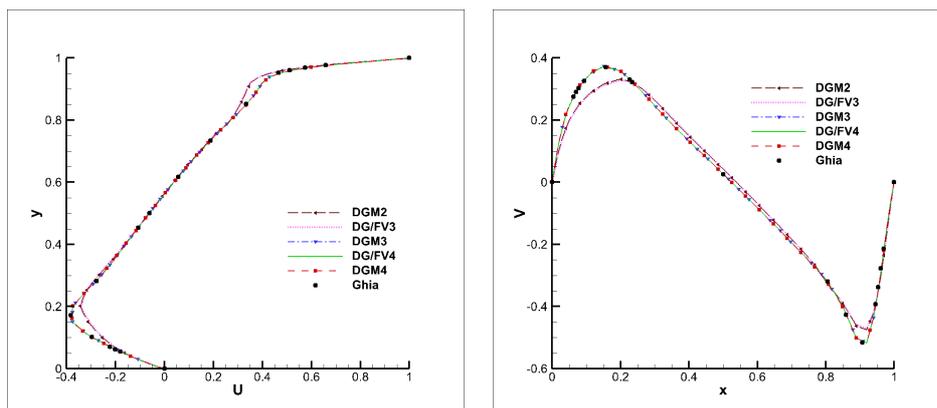


Figure 26: Velocity profile of computation with $Re = 1000$. Left: u at $x = 0.5$, Right: v at $y = 0.5$.

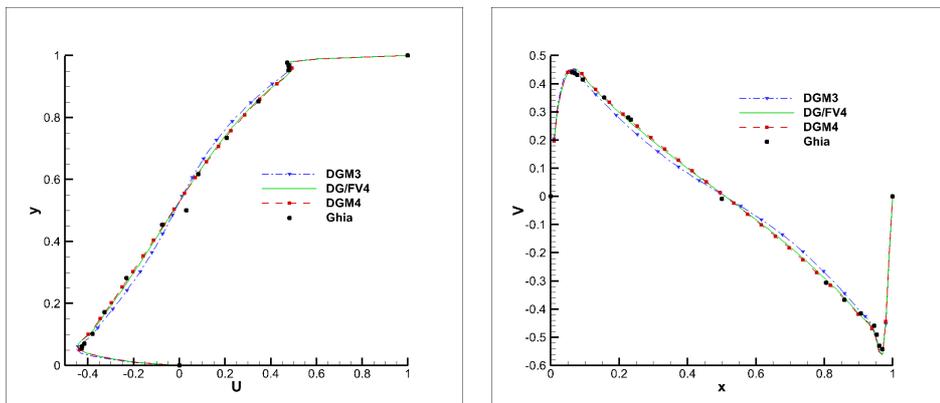


Figure 27: Velocity profile of computation with $Re=10000$. Left: u at $x=0.5$, Right: v at $y=0.5$.

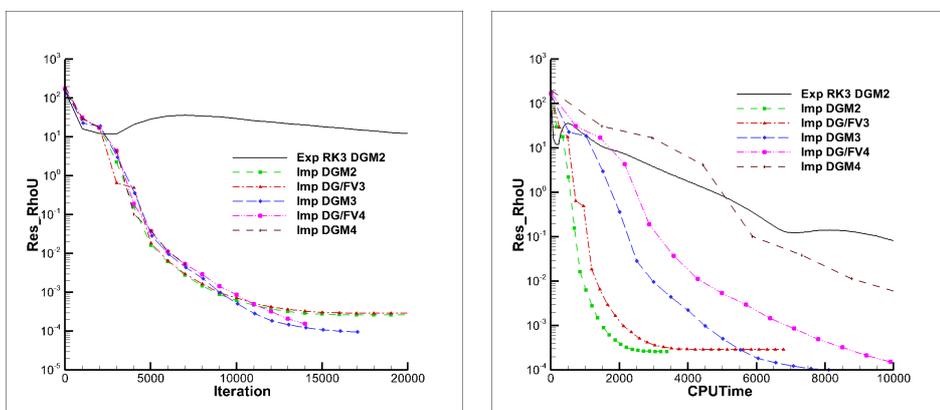


Figure 28: Convergence history of computation ($Re=1000$).

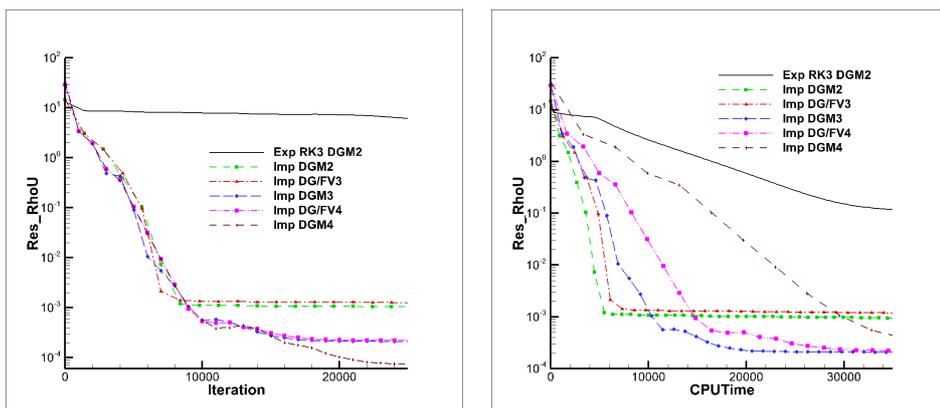


Figure 29: Convergence history of computation ($Re=10000$).

5 Concluding remarks

In this work, an efficient implicit method based on Newton and LU-SGS iteration has been developed for the high-order DG/FV hybrid schemes on two-dimensional unstructured/hybrid grids. The numerical results indicate that the Newton/LU-SGS implicit method leads to a significant improvement of convergence rate over the traditional Runge-Kutta explicit method. In addition, the effects of several parameters in the implicit scheme, such as the CFL number, the Newton sub-iteration steps, and the update frequency of mass-matrix have been investigated numerically. The results demonstrate that the CFL number is the dominate parameter of efficiency of the implicit algorithm. Properly increasing Newton sweeps are helpful for convergence rate. Less update frequency of the Jacobian matrix will results in faster convergence. Moreover, the DG/FV hybrid schemes are more efficient than the same order DG schemes. We are currently extending the implicit method to three-dimensional cases, and the results will be presented in a future publication.

Acknowledgments

This work is supported partially by National Basic Research Program of China (Grant No. 2009CB723800) and by National Science Foundation of China (Grant Nos. 11402290 and 91130029).

References

- [1] Harten A., Engquist B., Osher S. and Chakravarthy S., Uniformly high-order essentially non-oscillatory schemes III, *J. Comput. Phys.*, 1987, 71: 231-303.
- [2] Jiang G. and Shu C-W., Efficient implementation of weighted ENO schemes, *J. Comput. Phys.*, 1996, 126: 202-228.
- [3] Reed W.H., Hill T.R., Triangular mesh methods for the neutron transport equation, Technical Report LA-UR-73-479, Los Alamos Scientific Laboratory, 1973.
- [4] Cockburn B., Shu C-W., TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws II: general framework, *Math. Comp.*, 1989, 52: 411-435.
- [5] Cockburn B., Lin S.Y., Shu C-W., TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws III: one-dimensional systems, *J. Comput. Phys.*, 1989, 84: 90-113.
- [6] Cockburn B., Hou S., Shu C-W., TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws IV: the multidimensional case, *Math. Comp.*, 1990, 54: 545-581.
- [7] Cockburn B., Shu C-W., Runge-Kutta discontinuous Galerkin methods for convection-dominated problems, *J. Sci. Comput.*, 2001: 173-261.
- [8] Wang Z.J., Spectral (finite) volume method for conservation laws on unstructured grids: basic formulation, *J. Comput. Phys.*, 2002, 178: 210-251.
- [9] Wang Z.J. and Liu Y., Spectral (finite) volume method for conservation laws on unstructured grids II: extension to two-dimensional scalar equation, *J. Comput. Phys.*, 2002, 179: 665-697.

- [10] Wang Z.J. and Liu Y., Spectral (finite) volume method for conservation laws on unstructured grids III: one-dimensional systems and partition optimization, *J. Sci. Comput.*, 2004, 20: 137-157.
- [11] Wang Z.J., Zhang L.P., Liu Y., Spectral (finite) volume method for conservation laws on unstructured grids IV: extension to two-dimensional Euler equations, *J. Comput. Phys.*, 2004, 194: 716-741.
- [12] Liu Y., Vinokur M., and Wang Z.J., Discontinuous spectral difference method for conservation laws on unstructured grids, *J. Comput. Phys.*, 2006, 216: 780-801.
- [13] Venkatakrisnan V., Allmaras S.R., Kamenetskii D.S., Johnson F.T., Higher order schemes for the compressible Navier-Stokes equations, *AIAA paper 2003-3987*, 2003.
- [14] May G. and Jameson A., A spectral difference method for the Euler and Navier-Stokes equations, *AIAA paper 2006-304*, 2006.
- [15] Ekaterinaris J.A., High-order accurate, low numerical diffusion methods for aerodynamics, *Prog. Aero. Sci.*, 2005, 41: 192-300.
- [16] Wang Z.J., High-order methods for the Euler and Navier-Stokes equations on unstructured grids, *Prog. Aero. Sci.*, 2007, 43: 1-41.
- [17] Cockburn B., Karniadakis G.E., Shu C-W., *Discontinuous Galerkin methods*, Berlin: Springer, 2000.
- [18] Thareja R.R., Stewart J.R., A point implicit unstructured grid solver for the Euler and Navier-Stokes equations, *Inter. J. Numer. Meth. Fluids*, 1989, 9: 405-425.
- [19] Luo H., Baum J.D. and Löhner R., High-Reynolds number viscous computations using an unstructured-grid method, *J. Aircraft*, 2005, 42: 483-492.
- [20] He L.X., Zhang L.P., Zhang H.X., A finite element/finite volume mixed solver on hybrid grids, *Proceedings of the Fourth International Conference on Computational Fluid Dynamics*, 10-14 July, 2006, Ghent, Belgium, edited by Herman Deconinck and Erik Dick, Springer Press, 2006: 695-700.
- [21] Dumbser M., Balsara D.S., Toro E.F., A unified framework for the construction of one-step finite volume and discontinuous Galerkin schemes on unstructured meshes, *J. Comput. Phys.*, 2008, 227: 8209-8253.
- [22] Dumbser M., Arbitrary high order $P_N P_M$ schemes on unstructured meshes for the compressible Navier-Stokes equations, *Computers and Fluids*, 2010, 39: 60-76.
- [23] Dumbser M., Zanotti O., Very high order $P_N P_M$ schemes on unstructured meshes for the resistive relativistic MHD equations, *J. Comput. Phys.*, 2009, 228: 6991-7006.
- [24] Luo H., Luo L., Norgaliev R., Mousseau V.A. and Dinh N., A reconstructed discontinuous Galerkin method for the compressible Navier-Stokes equations on arbitrary grids, *J. Comput. Phys.*, 2010, 229: 6961-6978.
- [25] Luo H., Luo L.P., Ali A., Norgaliev R. and Cai C., A parallel, reconstructed discontinuous Galerkin method for the compressible flows on arbitrary grids, *Commun. Comput. Phys.*, 2011, 9(2): 363-389.
- [26] Qiu J.X., Shu C-W., Hermite WENO schemes and their application as limiters for Runge-Kutta discontinuous Galerkin method: one-dimensional case, *J. Comput. Phys.*, 2003, 193: 115-135.
- [27] Qiu J.X., Shu C-W., Hermite WENO schemes and their application as limiters for Runge-Kutta discontinuous Galerkin method II: Two-dimensional case, *Computers and Fluids*, 2005, 34: 642-663.
- [28] Zhang L.P., Liu W., He L.X., Deng X.G., Zhang H.X., A class of DG/FV hybrid methods for conservation laws I: Basic formulation and one-dimensional systems, *J. Comput. Phys.*,

- 2012, 231: 1081-1103.
- [29] Zhang L.P., Liu W., He L.X., Deng X.G., Zhang H.X., A class of DG/FV hybrid methods for conservation laws II: Two-dimensional Cases, *J. Comput. Phys.*, 2012, 231: 1104-1120.
- [30] Zhang L.P., Liu W., He L.X., Deng X.G., Zhang H.X., A class of DG/FV hybrid methods for conservation laws III: Two-dimensional Euler equations, *Commun. Comput. Phys.*, 2012, 12(1): 284-314.
- [31] Huynh H.T., A flux reconstruction approach to high-order schemes including discontinuous Galerkin methods, *AIAA paper 2007-4079*, 2007.
- [32] Wang Z.J., Gao H.Y., A unifying lifting collocation penalty formulation for the Euler equations on mixed grids, *AIAA paper 2009-0401*, 2009.
- [33] Luo H., Baum J.D., Löhner R., A discontinuous Galerkin method based on a Taylor basis for the compressible flows on arbitrary grids, *J. Comput. Phys.*, 2008, 227: 8875-8893.
- [34] Briley W.R., McDonald H., Solution of the three dimensional compressible Navier-Stokes equations by an implicit technique. In: *Proceedings of the fourth international conference on numerical methods in fluid dynamics*, Lecture Notes in Physics, vol.35. Berlin: Springer-Verlag; 1975.
- [35] Beam R.W., Warming R.F., An implicit finite-difference algorithm for hyperbolic systems in conservation law form. *J. Comput. Phys.*, 1976, 22: 87-109.
- [36] Wang L., Mavriplis D.J., Implicit solution of the unsteady Euler equation for high-order accurate discontinuous Galerkin discretizations, *J. Comput. Phys.*, 2007, 225:1994-2005.
- [37] Mavriplis D.J., Multigrid strategies for viscous flow solvers on anisotropic unstructured meshes, *J. Comput. Phys.*, 1998; 145:141-165.
- [38] Bassi F., Crivellini A., Rebay S., Savini M., Discontinuous Galerkin solutions of the Reynolds-averaged Navier-Stokes and $k\omega$ turbulence model equations, *Computers and Fluids*, 2005, 34: 507-540.
- [39] Saad Y., Schultz M.H., GMRES: A generalized minimal residual algorithm for solving non-symmetric linear systems, *SIAM J. Sci. Stat. Comput.*, 1986, 7: 865-884.
- [40] Rasetarinera P., Hussaini M.Y., An efficient implicit discontinuous Galerkin method, *J. Comput. Phys.*, 2001, 172: 718-738.
- [41] Sharov D., Nakahashi K., Low speed preconditioning and LU-SGS scheme for 3D viscous flow computations on unstructured grids, *AIAA paper 98-0614*, 1998.
- [42] Sun Y.Z., Wang Z.J., Liu Y. and Chen C.L., Efficient implicit LU-SGS algorithm for high-order spectral difference method on unstructured hexahedral grids, *AIAA paper 2007-0313*, 2007.
- [43] Yasue K., Furudate M., Ohnishi N., Sawada K., Implicit discontinuous Galerkin method for RANS simulation utilizing pointwise relaxation algorithm, *Commun. Comput. Phys.*, 2010, 7(3): 510-533.
- [44] Kanevsky A., Carpenter M.H., Gottlieb D., Hesthaven J.S., Application of implicit-explicit high order Runge-Kutta methods to discontinuous-Galerkin schemes, *J. Comput. Phys.*, 2007, 225: 1753-1781.
- [45] Ascher U.M., Ruuth S.J., Spiteri R.J., Implicit-explicit Runge-Kutta methods for time-dependent partial differential equations, *Appl. Numer. Math.*, 1997, 25: 151-167.
- [46] MacCormack R.W., Implicit methods for fluid dynamics, *Computers and Fluids*, 2011, 41: 72-81.
- [47] Xia Y.D., Luo H., Nourgaliev R., An implicit method for a reconstructed discontinuous Galerkin method on tetrahedron grids, *AIAA paper 2012-834*, 2012.
- [48] Xia Y.D., Luo H., Frisbey M., Nourgaliev R., A set of parallel, implicit methods for a reconstructed discontinuous Galerkin method for compressible flows on 3D hybrid grids, *Com-*

- puters & Fluids, 2014, 98: 134-151.
- [49] Crivellini A., Bassi F., An implicit matrix-free discontinuous Galerkin solver for viscous and turbulent aerodynamic simulations, *Computers & Fluids*, 2011, 50(1): 81-93.
 - [50] Nastase C.R., Mavriplis D.J., High-order discontinuous Galerkin methods using an hp-multigrid approach, *J. Comput. Phys.*, 2006, 213: 330-357.
 - [51] Jameson A., Time dependent calculations using multigrid, with applications to unsteady flows past airfoils and wings, AIAA paper 91-1596, 1991.
 - [52] Jameson A. and Yoon S., Lower-upper implicit schemes with multiple grids for the Euler equations, *AIAA J.*, 1987. 25(7): 929-935.
 - [53] Wang Z.J. and Gao H.Y., A unifying lifting collocation penalty formulation including the discontinuous Galerkin, spectral volume/difference methods for conservation laws on mixed grids, *J. Comput. Phys.*, 2009, 228: 8161-8186.
 - [54] Krivodonova L., Berger M., High-order accurate implementation of solid wall boundary condition in curved geometries, *J. Comput. Phys.*, 2006, 211: 492-512.
 - [55] Ghia U., Ghia K.N., Shin C.T., High-Re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method, *J. Comput. Phys.*, 1982, 48: 387-411.