

Efficient Implicit LU-SGS Algorithm for High-Order Spectral Difference Method on Unstructured Hexahedral Grids

Yuzhi Sun* and Z.J. Wang†

Department of Aerospace Engineering, Iowa State University, Ames, IA 50011

Yen Liu‡

NASA Ames Research Center, Moffett Field, CA 94035

C.L. Chen§

Rockwell Scientific, Thousand Oaks, CA 91360

An efficient implicit lower-upper symmetric Gauss-Seidel (LU-SGS) solution algorithm has been developed for a high order multi-domain spectral difference method on unstructured hexahedral grids. The LU-SGS solver is preconditioned by the block element matrix, and the system of equations is then solved with an exact LU decomposition approach. In addition, the effects of several parameters on the convergence rate in the implicit scheme have been investigated for both external and internal flows. The implicit scheme has shown a speed up factor of more than an order of magnitude relative to the multi-stage explicit Runge-Kutta scheme for several demonstration problems.

I. Introduction

The advantage of high-order methods (order of accuracy > 2) over first and second-order ones is well known in the CFD community. Generally speaking, with the same number of degrees-of-freedom (DOFs) or solution unknowns, high-order methods are capable of producing much more accurate results. For problems requiring very high accuracy, e.g., wave propagation problems in computational aeroacoustics, high-order methods have been the main choice. Many high-order methods were developed for structured grids, e.g., ENO/WENO methods¹ compact methods²⁻³, optimized methods⁴, to name just a few. In the last two decades, there have been intensive research efforts on high-order methods for unstructured grids since many real world applications have complex geometries. An incomplete list of notable examples includes the spectral element method⁵, multi-domain spectral method⁶⁻⁷, k-exact finite volume method⁸, WENO methods⁹, discontinuous Galerkin (DG) method¹⁰⁻¹², high-order residual distribution methods¹³, spectral volume (SV)¹⁴⁻¹⁷ and spectral difference (SD) methods¹⁸⁻²¹. Among those methods, some are based on the weighted residual form of the governing equations, for instance the DG method¹⁰⁻¹². Some are based on the integral form of the governing equations, e. g., the k-exact finite volume method⁸ and SV methods¹⁴⁻¹⁷. Others, such as the staggered grid multi-domain spectral method⁶⁻⁷ and the SD method¹⁸⁻²¹, are based on the differential form.

When one chooses a particular method for three-dimensional applications, the cost and the complexity in implementing 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. We recently developed a high order SD method²² for the three dimensional Navier-Stokes equations on unstructured hexahedral grids. High order accuracy and spectral convergence are achieved for several benchmark problems. It was also shown that the wall boundaries must be approximated with high-order surfaces. An explicit Runge-Kutta time integration scheme was used in the implementation. Although the explicit scheme is easy to implement and has high-order accuracy in time, it suffered from slow convergence, especially for viscous grids which are clustered in the viscous boundary layer. It is well-known that high-order methods are restricted to a smaller CFL number than low order ones. In addition, they also

* Postdoc Research Associate of Aerospace Engineering, 2271 Howe Hall, sunyuzhi@iastate.edu, AIAA Member.

† Associate Professor of Aerospace Engineering, 2271 Howe Hall, zjw@iastate.edu, Associate Fellow of AIAA

‡ Research Scientist, Yen.Liu@nasa.gov, Mail Stop T27B-1.

§ Manager of Applied Computational Physics Department, Rockwell Scientific, cchen@rws.com,

possess much less numerical dissipation. Therefore it takes excessive CPU to reach a state-steady solution with explicit high-order schemes. The computation cost of high-order explicit methods for many steady-state problems is so high that they become less efficient than low-order implicit methods in terms of the total CPU time given the same level of solution error. It is therefore imperative to develop efficient implicit solution approaches for high-order methods to fully realize the potentials, which is the objective of the present study.

Implicit time-integration schemes are highly desired for improved efficiency since they can advance the solution with significantly larger time steps comparing with the explicit methods. Many implicit schemes have been developed and applied successfully to unstructured grids to accelerate convergence to steady state²³⁻²⁸ in the last one and a half decades. In this paper, an efficient implicit lower-upper symmetric Gauss-Seidel (LU-SGS) solution algorithm has been developed for the high order spectral difference method on unstructured hexahedral grids.

The paper is organized as follows. In the next section, the formulation of the 3D spectral difference method including both explicit and implicit schemes is described for a hexahedral element. In Section 3, several representative test cases are selected to demonstrate the efficiency of the implicit scheme, and study the effects of several parameters on the convergence rate. Conclusions and possible future works are summarized in Section 4.

II. 3D Formulation of High-Order Multi-domain Spectral Difference Method

We assume the computational domain is discretized into unstructured hexahedral cells, which are transformed into the standard cube in the computational domain for efficient implementation, as shown in Figure 1. In each element, two sets of points are defined, namely the solution points and flux points, as shown in Figure 2. Solution unknowns are defined at the solution points, while fluxes are computed at the flux points. Solution points in 1D are the Gauss quadrature points, and the flux points are the Gauss-Labatto points. The solutions are discontinuous across element boundaries, and finite-volume type Riemann solvers are used to compute a common interface flux, thus providing element-wise coupling.

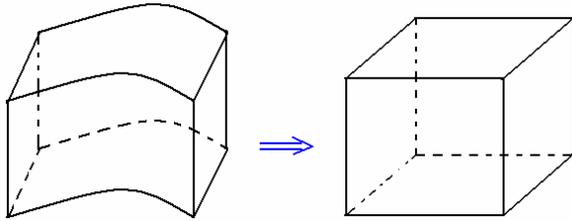


Figure 1. Transformation from a physical element to a standard element

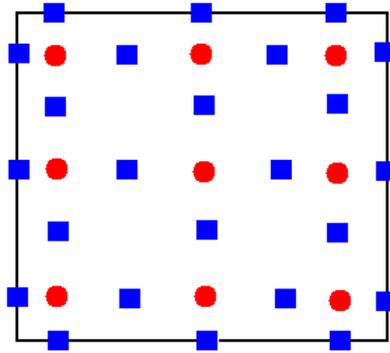


Figure 2. Distribution of solution points (circles) and flux points (squares) in a standard element for a 3rd order SD scheme.

After the transformation, the Navier-Stokes equations in the standard element can be written as

$$\frac{\partial \tilde{Q}}{\partial t} + \frac{\partial \tilde{F}}{\partial X} + \frac{\partial \tilde{G}}{\partial Y} + \frac{\partial \tilde{H}}{\partial Z} = 0. \quad (1)$$

The residual is defined as

$$R(\tilde{Q}) = -\left(\frac{\partial \tilde{F}}{\partial X} + \frac{\partial \tilde{G}}{\partial Y} + \frac{\partial \tilde{H}}{\partial Z} \right). \quad (2)$$

These equations can be viewed to be point-wise or cell-wise (element-wise) equations. The construction of the solution and flux in the unit cube can be expressed as follows:

$$\tilde{Q}(X, Y, Z) = \sum_{k=1}^N \sum_{j=1}^N \sum_{i=1}^N \tilde{Q}_{i,j,k} h_i(X) \cdot h_j(Y) \cdot h_k(Z),$$

$$\begin{aligned}\tilde{F}(X, Y, Z) &= \sum_{i=1}^{N+1} \sum_{k=1}^N \sum_{j=1}^N \tilde{F}_{i+\frac{1}{2}, j, k} l_{i+\frac{1}{2}}(X) \cdot h_j(Y) \cdot h_k(Z) \\ \tilde{G}(X, Y, Z) &= \sum_{j=1}^{N+1} \sum_{i=1}^N \sum_{k=1}^N \tilde{G}_{i, j+\frac{1}{2}, k} h_i(X) \cdot l_{j+\frac{1}{2}}(Y) \cdot h_k(Z) \\ \tilde{H}(X, Y, Z) &= \sum_{k=1}^{N+1} \sum_{j=1}^N \sum_{i=1}^N \tilde{H}_{i, j, k+\frac{1}{2}} h_i(X) \cdot h_j(Y) \cdot l_{k+\frac{1}{2}}(Z)\end{aligned}$$

where N is the given number of Gauss quadrature points in a coordinate direction, $N-1$ is the degree of construction polynomials, and

$$\begin{aligned}h_i(X) &= \prod_{s=1, s \neq i}^N \left(\frac{X - X_s}{X_i - X_s} \right), & X_s &= \frac{1}{2} \left(1 - \cos\left(\frac{2s-1}{2N} \cdot \pi\right) \right), \quad s = 1, 2, \dots, N \\ l_{i+\frac{1}{2}}(X) &= \prod_{s=1, s \neq i}^{N+1} \left(\frac{X - X_{s+\frac{1}{2}}}{X_{i+\frac{1}{2}} - X_{s+\frac{1}{2}}} \right), & X_{s+\frac{1}{2}} &= \frac{1}{2} \left(1 - \cos\left(\frac{s-1}{N} \cdot \pi\right) \right), \quad s = 1, 2, \dots, N+1\end{aligned}$$

For details of the numerical method including the computation of the inviscid and viscous fluxes, refer to Reference 22.

A. Explicit scheme

At cell c , using the forward Euler difference, (1) can be written as

$$\tilde{Q}_c^{n+1} = \tilde{Q}_c^n + dt * R_c(\tilde{Q}^n).$$

Similarly a three-stage TVD Runge-Kutta scheme can be written as

$$\begin{aligned}\tilde{Q}_c^{(1)} &= \tilde{Q}_c^n + dt * R_c(\tilde{Q}^n). \\ \tilde{Q}_c^{(2)} &= \frac{3}{4} \tilde{Q}_c^n + \frac{1}{4} \left[\tilde{Q}_c^{(1)} + dt * R_c(\tilde{Q}^{(1)}) \right]. \\ \tilde{Q}_c^{n+1} &= \frac{1}{3} \tilde{Q}_c^n + \frac{2}{3} \left[\tilde{Q}_c^{(2)} + dt * R_c(\tilde{Q}^{(2)}) \right].\end{aligned}$$

B. Implicit scheme

At each cell c , using the backward Euler difference, (1) can be written as

$$\frac{\tilde{Q}_c^{n+1} - \tilde{Q}_c^n}{\Delta t} - \left[R_c(\tilde{Q}^{n+1}) - R_c(\tilde{Q}^n) \right] = R_c(\tilde{Q}^n) \quad (3)$$

Let $\Delta \tilde{Q}_c = \tilde{Q}_c^{n+1} - \tilde{Q}_c^n$ and linearizing the residual, we obtain

$$R_c(\tilde{Q}^{n+1}) - R_c(\tilde{Q}^n) \approx \frac{\partial R_c}{\partial \tilde{Q}_c} \Delta \tilde{Q}_c + \sum_{nb \neq c} \frac{\partial R_c}{\partial Q_{nb}} \Delta \tilde{Q}_{nb}, \quad (4)$$

where nb indicates all the neighboring cells contributing to the residual of cell c . Therefore, the fully linearized equations for (3) can be written as

$$\left(\frac{I}{\Delta t} - \frac{\partial R_c}{\partial \tilde{Q}_c} \right) \Delta \tilde{Q}_c - \sum_{nb \neq c} \frac{\partial R_c}{\partial Q_{nb}} \Delta \tilde{Q}_{nb} = R_c(\tilde{Q}^n). \quad (5)$$

However, it costs too much memory to store the LHS implicit Jacobian matrices. Therefore, we employ a LU-SGS scheme to solve (5), i.e., we use the most recent solution for the nb cells,

$$\left(\frac{I}{\Delta t} - \frac{\partial R_c}{\partial \tilde{Q}_c} \right) \Delta \tilde{Q}_c^{(k+1)} = R_c(\tilde{Q}^n) + \sum_{nb \neq c} \frac{\partial R_c}{\partial Q_{nb}} \Delta \tilde{Q}_{nb}^*. \quad (6)$$

The matrix

$$D = \left(\frac{I}{\Delta t} - \frac{\partial R_c}{\partial \tilde{Q}_c} \right) \quad (7)$$

is the element (or cell) matrix, which is not too large for 2nd to 4th order SD schemes. (6) is then solved with a direct LU decomposition solver. Since we do not want to store the matrices $\frac{\partial R_c}{\partial Q_{nb}}$, (6) is further manipulated as follows.

Note that

$$\begin{aligned} R_c(\tilde{Q}^n) + \sum_{nb \neq c} \frac{\partial R_c}{\partial \tilde{Q}_{nb}} \Delta \tilde{Q}_{nb}^* &= R_c(\tilde{Q}_c^n, \{\tilde{Q}_{nb}^n\}) + \sum_{nb \neq c} \frac{\partial R_c}{\partial \tilde{Q}_{nb}} \Delta \tilde{Q}_{nb}^* \\ &\approx R_c(\tilde{Q}_c^n, \{\tilde{Q}_{nb}^*\}) \approx R_c(\tilde{Q}_c^*, \{\tilde{Q}_{nb}^*\}) - \frac{\partial R_c}{\partial \tilde{Q}_c} \Delta Q_c^* = R_c(\tilde{Q}^*) - \frac{\partial R_c}{\partial \tilde{Q}_c} \Delta Q_c^*. \end{aligned} \quad (8)$$

Let $\Delta^* Q_c^{(k+1)} = \Delta Q_c^{(k+1)} - \Delta Q_c^{(k)}$. The combining (6) and (8) together, we obtain

$$\left(\frac{I}{\Delta t} - \frac{\partial R_c}{\partial \tilde{Q}_c} \right) \Delta^* \tilde{Q}_c^{(k+1)} = R_c(\tilde{Q}^*) - \frac{\Delta Q_c^*}{\Delta t}. \quad (9)$$

Eq. (9) is then solved with multiple symmetric forward and backward sweeps with a prescribed convergence tolerance ε . Note that if (9) is solved to machine zero, the unsteady residual $R_c(\tilde{Q}) - \frac{\Delta Q_c}{\Delta t}$ is zero at each time step. The initial guess for \tilde{Q}_c^{n+1} can be set to \tilde{Q}_c^n . Therefore, the initial “unsteady residual” is the same as the steady residual at the last time step, i.e., $R_c(\tilde{Q}^n)$. The unsteady residual is then monitored for convergence. For steady state problems, it is not necessary to drive the unsteady residual to machine zero. In fact, it may be more efficient to set a maximum number of sweeps for (9), *inner_sweep* to just a few (3 – 10).

III. Numerical Results and Discussions

In order to demonstrate the efficiency of the implicit LU-SGS scheme and analyze the effect of various parameters on the convergence rate, three flow problems are chosen as the demonstration problems. All of the simulations start from the free stream and converge to machine zero.

A. Inviscid flow over a sphere

An inviscid flow over a sphere with a free stream Mach number of 0.2535 is selected as the first test to demonstrate the efficiency of the implicit scheme, and also to study the effects of *CFL* number and inner iteration control parameters on convergence characteristics. Figure 3 shows the computational grid used in the simulation, which includes 768 hexahedral cells. Figure 4 depicts the Mach contour distribution computed with the 4th order SD scheme.

Both the three-stage Runge-Kutta explicit and the LU-SGS implicit schemes with 2nd, 3rd, and 4th order spatial accuracy were employed in the simulation. The implicit scheme dramatically accelerates the convergence rates to the steady state for this external flow. This is illustrated in the Figure 5, which displays the

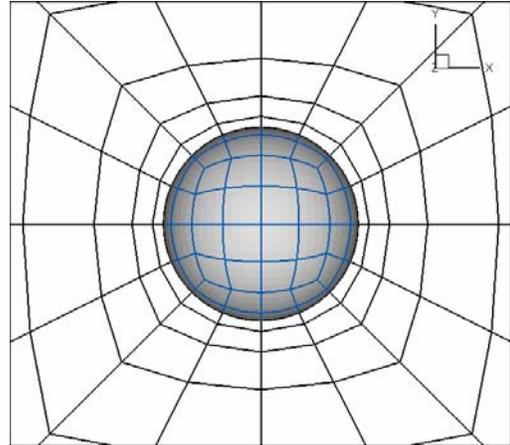


Figure 3. Grid for inviscid flow over a sphere

convergence histories in terms of CPU time. The convergence rate with the implicit scheme is more than an order of magnitude faster than the explicit scheme.

Next, we study the effects of several parameters on the convergence rate of the simulation. Obviously the CFL number is an important convergence parameter. In the present study, the CFL number is computed based on the following power law form and bounded by the minimum and maximum CFL number:

$$CFL = MIN(CFL_{min} \cdot \alpha^n, CFL_{max})$$

where $\alpha \geq 1$ is the amplification factor, and n is the iteration number. In the first test, CFL_{min} and α are

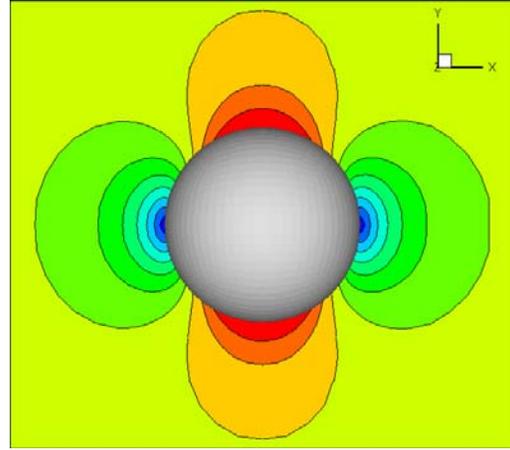


Figure 4. Mach contours for flow over a sphere

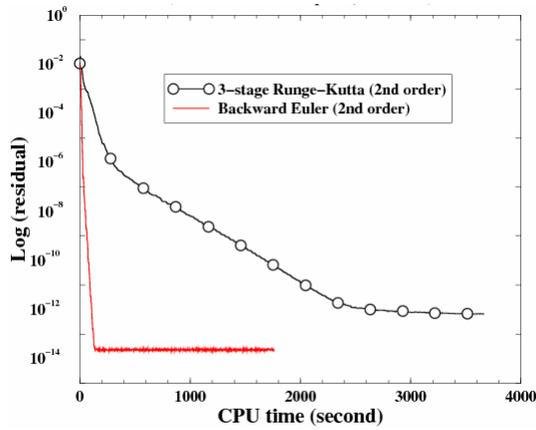


Figure 5a. Residual history of inviscid flow over a sphere with the 2nd order spatial accuracy

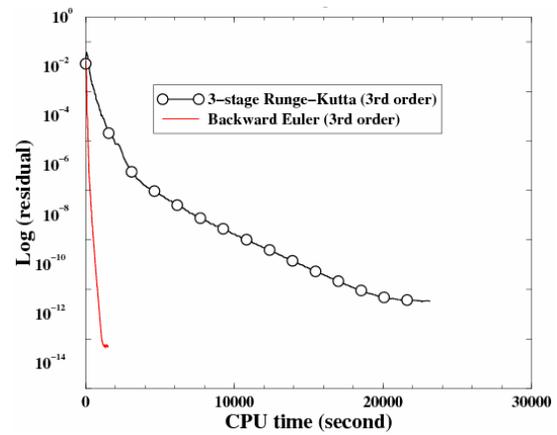


Figure 5b. Residual history of inviscid flow over a sphere with 3rd spatial accuracy

fixed at certain values, while CFL_{max} is a variable.

The effects of CFL_{max} on the convergence rates are showed in Figure 6a with $CFL_{min} = 1.0$ for the 3rd order SD scheme, and Figure 6b with $CFL_{min} = 0.5$ for the 4th order scheme. The amplification factor is set to be 1.25 for both schemes. It is easily observed that the convergence rate strongly depends on the CFL number. The larger CFL number results in higher convergence rate. However, we also want to emphasize that a too large CFL number can cause the simulation to diverge.

The second parameter on the convergence rate is the number of the inner iterations, i.e. the number of forward and backward Gauss-Seidel sweeps in the LU-SGS approach. One sweep is defined to include both the forward and backward sweeps here, and is denoted by *inner_sweep*. The unsteady residual in each time iteration step can be driven to machine zero if *inner_sweep* is big enough. In the present simulations, *inner_sweep* = 3 is the smallest number to guarantee stability and convergence to the steady state. In this test, we

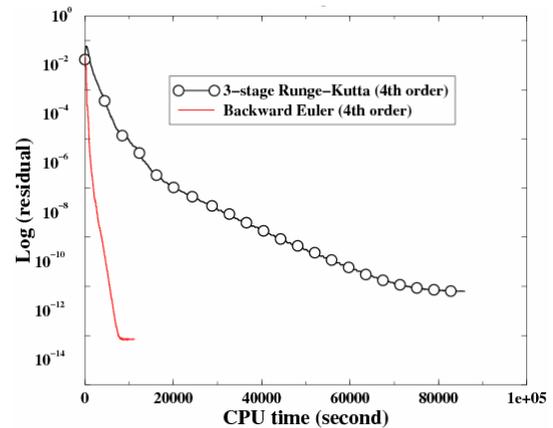


Figure 5c. Residual history of inviscid flow over a sphere with 4th order spatial accuracy

let *inner_sweep* vary and fix the other parameters. From Figure 7, it seems that the number of inner iterations doesn't strongly influence the convergence rate for the inviscid flow over a sphere.

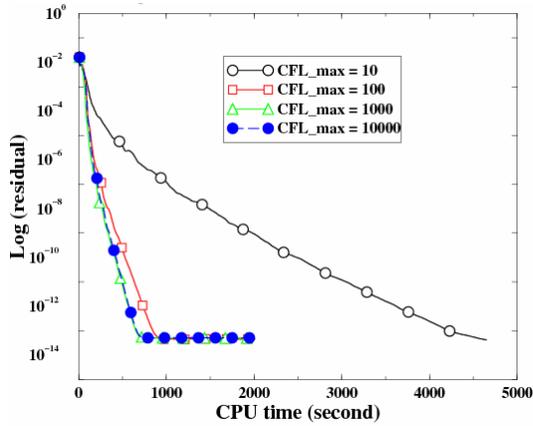


Figure 6a. Effect of CFL number (3rd order)

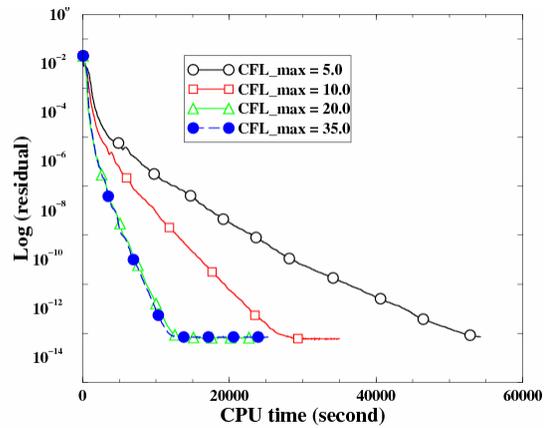


Figure 6b. Effect of CFL number (4th order)

In the LU-SGS approach, the computation of the element Jacobian matrix is quite time consuming since its size is quite large. One idea to improve the efficiency is to freeze this matrix for several time steps. The frequency in which this matrix is updated is denoted by *F_ITIME*. For example, *F_ITIME* = 5 means the element Jacobian matrix is computed every 5 time steps. Figure 8 shows the convergence histories with different *F_ITIME* with a 3rd order SD scheme. In this test, the *inner_sweep* is set to be 5, and CFL ranges from 1 to 1,000,000. It can be observed that the bigger *F_ITIME* results in higher efficiency.

Finally the effect of the amplification factor on the convergence rate is studied. In this test, α varies from 1.25 to 3 with a 0.25 interval. Other parameters are set as follows: $CFL_{min} = 1$, $CFL_{max} = 1,000,000$, *inner_sweep* = 5 and *F_ITIME* = 40. In Figure 9, the convergence rates are plotted together for $1.25 \leq \alpha \leq 2.25$. It appears the convergence rate does not strongly depend on the amplification factor when $1.25 \leq \alpha \leq 2.25$. However, the simulation diverged when $\alpha = 2.5$ or $\alpha = 3.0$.

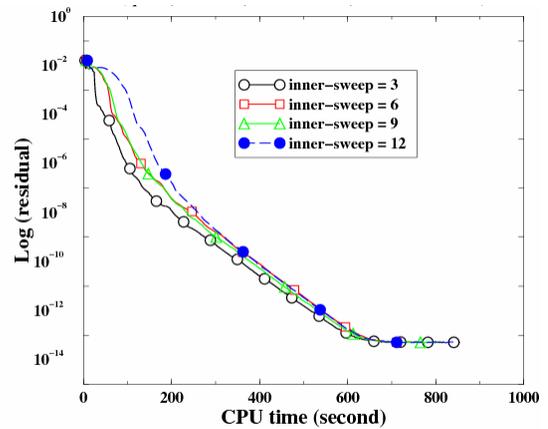


Figure 7. Effect of inner iteration (3rd)

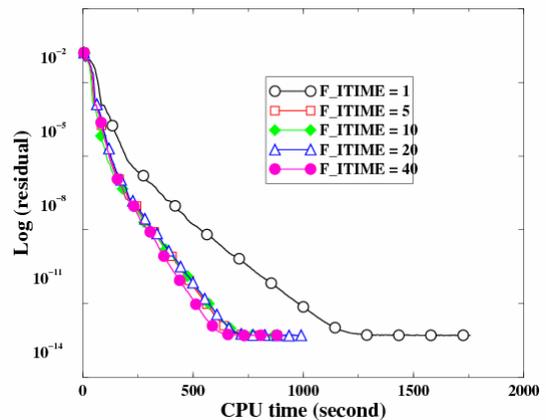


Figure 8. Effect of matrix freezing frequency

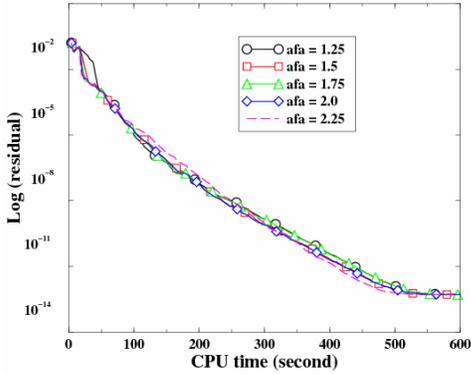


Figure 9. Effect of the amplification factor

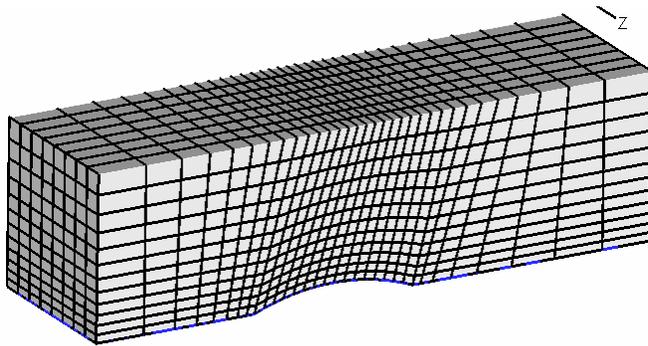


Figure 10. Grid for inviscid flow over a 3D bump

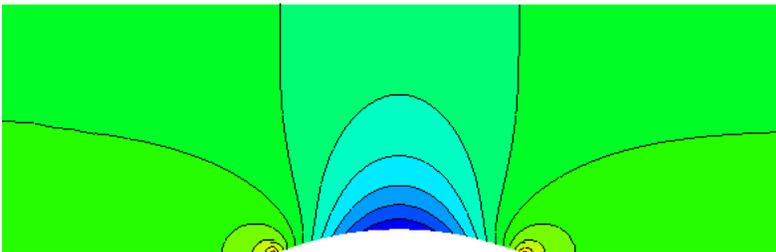


Figure 11. Pressure contours at the middle cutting plane

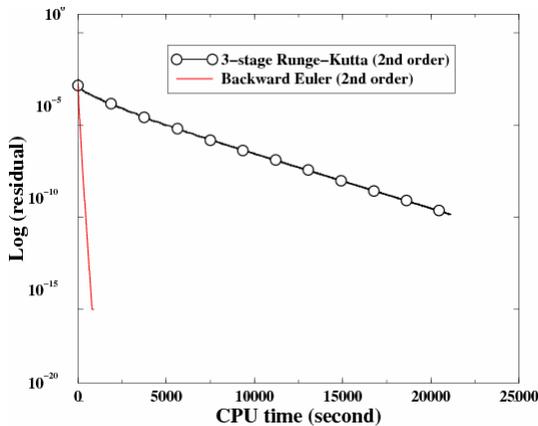


Figure 12a. Residual history of inviscid flow over a 3d-bump with 2nd order spatial accuracy

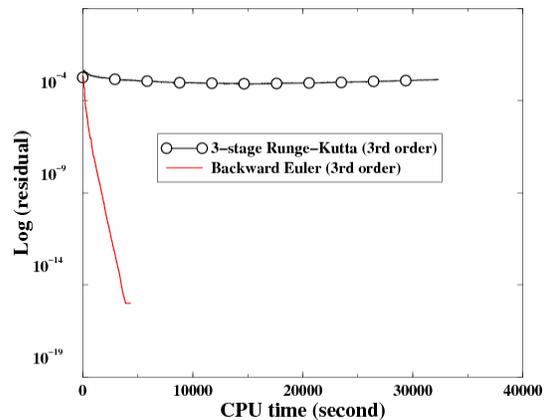


Figure 12b. Residual history of inviscid flow over a 3d-bump with 3rd spatial accuracy

B. Inviscid flow over a 3d bump

Inviscid flow over a 3D bump was selected to represent internal flow problems. Figure (10) shows the computation grid with 3,072 hexahedral cells. Figure 11 shows the steady state pressure contours at the middle cutting plane computed with the 4th order SD scheme.

Both the three-stage Runge-Kutta explicit and LU-SGS implicit schemes were employed for this problem with 2nd, 3rd, and 4th order spatial accuracy. Figure 12 shows the convergence histories. From Figure 12a, we can observe that the convergence to the steady state is accelerated by more than 20 times using the 2nd order SD scheme. For 3rd and 4th order SD schemes, the three-stage Runge-Kutta schemes failed to converge the simulations, as shown in Figures 12b and 12c.

The effects of the maximum CFL number are illustrated in Figure 13. For these simulations, α in the power law is set to be 1.25, and $CFL_{min} = 1$. Note again that the larger CFL_{max} results in higher convergence rate, which is consistent with the sphere case.

The effect of the number of inner iterations on the convergence rate is the same as the sphere case, i.e., the different values of *inner_sweep* don't make much difference on the convergence rate, which is shown in the Figure 14.

Next, the effect of matrix freezing frequency is shown in Figure 15. Note also that the larger F_ITIME is, the higher the efficiency.

Finally the effect of the amplification factor on the convergence rate is shown in Figure 16. Again the convergence rate is not strongly dependent on this parameter, as in the case of flow over the sphere.

C. Inviscid flow over a NACA wing

The case of inviscid flow over a NACA wing is chosen to further analyze the effects of different parameters on the convergence rate. Figure 17 shows the computational grid with 1,248 hexahedral cells, and Figure 18 displays the steady state Mach contours computed with the 4th order SD scheme.

To test the effect of the maximum CFL number, all other parameters are fixed as follows: $CFL_{min} = 0.5$, $\alpha = 1.25$, and the element Jacobian matrix is updated every 40 time steps. The convergence histories are plotted in Figure 19 for the different values of CFL_{max} . Again, we see that a bigger value of maximum CFL results in a higher efficiency.

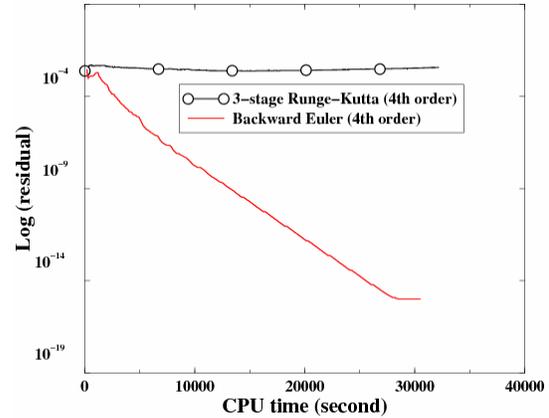


Figure 12c. Residual history of inviscid flow over a 3d-bump with 4th order SD scheme

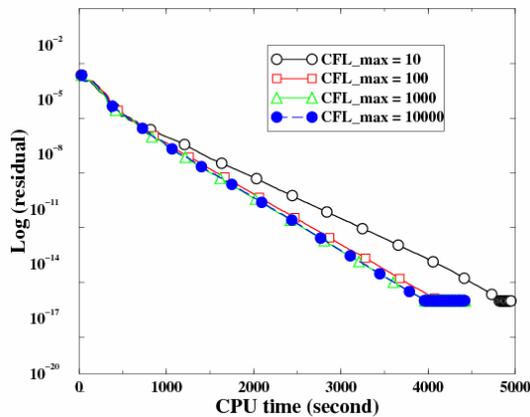


Figure 13. Effect of maximum CFL number

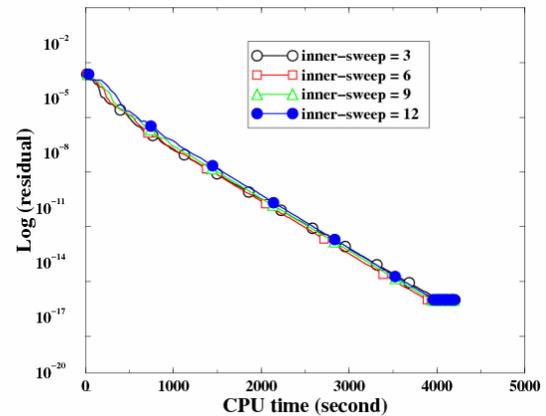


Figure 14. Effect of number of inner iteration

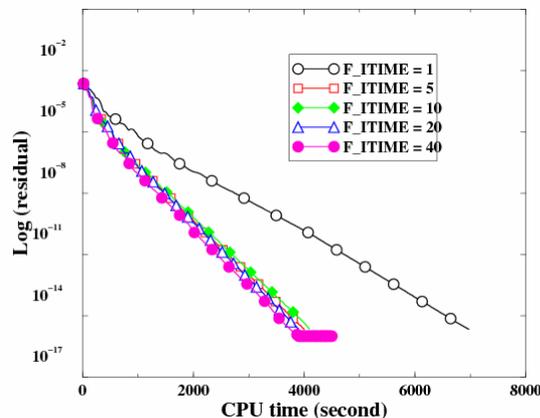


Figure 15. Effect of matrix freezing frequency

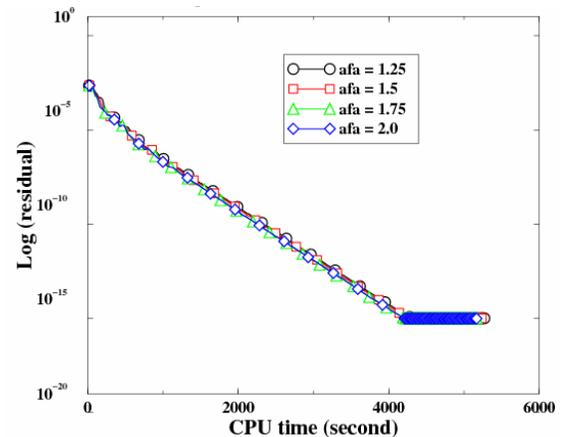


Figure 16. Effect of amplification factor

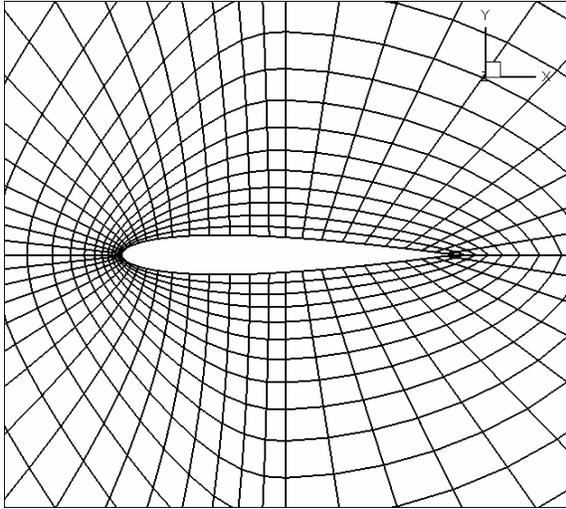


Figure 17. Grid for inviscid flow over a NACA wing

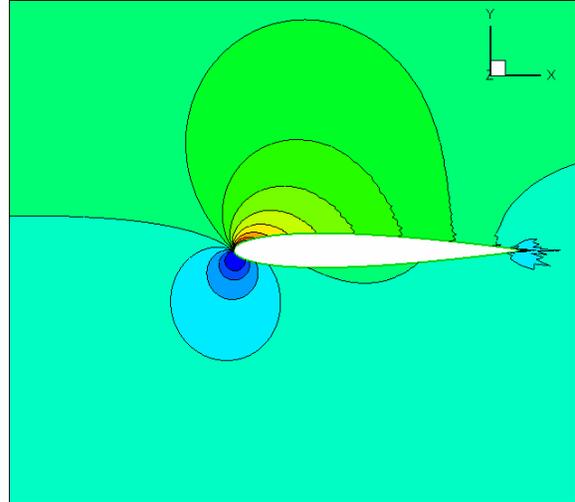


Figure 18. Mach contours start at Mach = 0.2 with a 0.025 interval for NACA wing

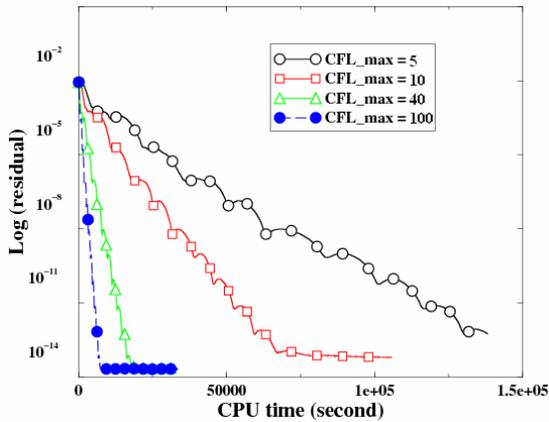


Figure 19. Effect of maximum CFL number

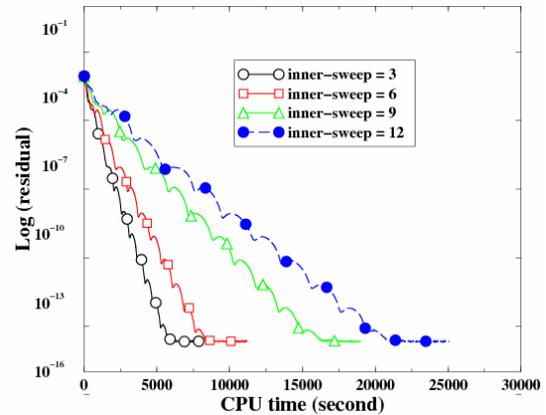


Figure 20. Effect of number of inner iterations

The effect of the number of inner sweeps is shown in Figure 20. Obviously for this case, the less the number of inner sweeps, the more efficient the simulation is.

Finally the effect of the matrix freezing frequency on the convergence rate is shown in Figure 21. In this simulation, the other parameters are set as follows: $CFL_{\min} = 0.5$, $CFL_{\max} = 100$ and $\alpha = 1.25$. The figure confirms that the less frequently the matrix is updated, the more efficient the simulation is.

IV. Conclusions and Future Work

In this paper, an efficient implicit lower-upper symmetric Gauss-Seidel (LU-SGS) solution algorithm has been developed for a high order multi-domain spectral difference method on unstructured hexahedral grids. The implicit scheme has shown more than an order of magnitude of speed-up relative to the multi-stage Runge-Kutta explicit time integration scheme for several

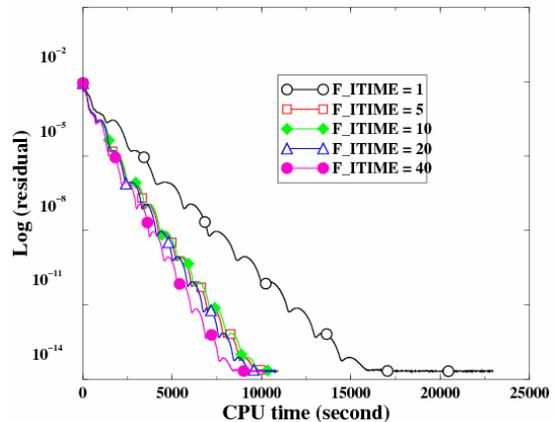


Figure 21. Effect of matrix freezing frequency

demonstration problems. In addition, the effects of several parameters on the convergence rate have been investigated numerically for both external and internal flows. Generally speaking, larger CFL number, less frequent matrix update and smaller number of inner iterations result in the faster convergence. We are currently extending the approach to viscous flow problems and the results will be presented in a future publication.

Acknowledgements

The study has been funded by Rockwell Scientific/DARPA under contract W911NF-04-C-0102, and partially supported by the Air Force Office of Scientific Research. 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, AFOSR, or the U.S. Government.

References

- ¹Shu, C.-W., Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws, in *Advanced Numerical Approximation of Nonlinear Hyperbolic Equations*, edited by A. Quarteroni, Lecture Notes in Mathematics (Springer-Verlag, Berlin/New York, 1998), Vol. 1697, P. 325.
- ²Lele, S.K., "Compact Finite Difference Schemes with Spectral-Like Resolution," *Journal of Computational Physics*, Vol. 103, 1992, pp. 16-42.
- ³Visbal, M. and Gaitonde, D., Shock Capturing Using Compact- Differencing- Based Methods, AIAA-2005-1265.
- ⁴Tam, C.K.W., and Webb, J.C., "Dispersion-Relation-Preserving Finite difference Schemes for Computational Acoustics," *Journal of Computational Physics*, Vol. 107, No. 2, 1993, pp. 262-281.
- ⁵Patera, A.T., A Spectral element method for fluid dynamics: Laminar flow in a channel expansion, *J. Comput. Phys.* 54, 468(1984).
- ⁶Kopriva, D.A. and Kolas, J.H., A conservative staggered –grid Chebyshev multidomain method for compressible flows, *J. Comput. Phys.* 125, 244(1996).
- ⁷Kopriva, D.A., A Staggered-Grid Multidomain Spectral Method for the Compressible Navier–Stokes Equations, *Journal of Computational Physics*, Volume 143, pp. 125-158, 1998.
- ⁸Barth, T.J. and Frederickson, P.O., High-Order Solution of the Euler Equations on Unstructured Grids using Quadratic Reconstruction, AIAA Paper No. 90-0013(1990).
- ⁹Hu, C. and Shu, C.-W., Weighted essentially non-oscillatory schemes on triangular meshes, *J. Comput. Phys.* 150, 97 (1999).
- ¹⁰Cockburn, B. and Shu, C.-W., TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws II: General framework, *Math. Comput.* 52,411(1989).
- ¹²Cockburn, B. and Shu, C.-W., The Runge-Kutta discontinuous Galerkin method for conservation laws V: multidimensional systems, *J. Comput. Phys.*, 141, 199 - 224, (1998).
- ¹²Bassi, F. and Rebay, S., High-order accurate discontinuous finite element solution of the 2D Euler equations, *J. Comput. Phys.* 138, 251-285 (1997).
- ¹³Abgrall, R. and Roe, P.L., High Order Fluctuation Schemes on Triangular Meshes, *Journal of Scientific Computing*, Volume 19, pp. 3 – 36, 2003.
- ¹⁴Wang, Z.J. and Liu, Yen, Spectral (Finite) Volume Method for Conservation Laws on Unstructured Grids III: Extension to One-Dimensional Systems, *J. Scientific Computing*, Vol. 20 No. 1, pp.137-157 (2004).
- ¹⁵Wang, Z.J., Zhang, L., and Liu, Yen, Spectral (finite) volume method for conservation laws on unstructured grids IV: extension to two-dimensional systems, *J. Comput. Phys.* 194, 716-741(2004).
- ¹⁶Liu, Y., Vinokur, M., and Wang, Z.J., Spectral (Finite) Volume Method for Conservation Laws on Unstructured Grids V: Extension to Three-Dimensional Systems, *Journal of Computational Physics* Vol. 212, pp. 454-472 (2006).
- ¹⁷Sun, Yuzhi, Wang, Z.J., and Liu, Yen, Spectral (Finite) Volume Method for Conservation Laws on Unstructured Grids VI: Extension to Viscous Flow, *Journal of Computational Physics*, Vol. 215, pp.41-58 (2006).
- ¹⁸ Liu, Y., Vinokur, M., and Wang, Z.J., Discontinuous Spectral Difference Method for Conservation Laws on Unstructured Grids, in *Proceeding of the 3rd International Conference in CFD*, Toronto, Canada July 2004.
- ¹⁹Liu, Yen, Vinokur, M., and Wang, Z.J., Multi-Dimensional Spectral Difference Method for Unstructured Grids, AIAA-2005-0320.
- ²⁰Wang, Z. J., and Liu, Yen, The Spectral Difference Method for the 2D Euler Equations on Unstructured Grids, AIAA-2005-5112.
- ²¹Huang, P.G., Wang, Z.J., and Liu, Yen, An Implicit Space-Time Spectral Difference Method for Discontinuity Capturing Using Adaptive Polynomials, AIAA-2005-5255.
- ²²Sun, Yuzhi, Wang, Z.J., and Liu, Yen, High-Order Multi-domain Spectral Difference Method for the Navier-Stokes Equations on Unstructured Hexahedral Grids, *Communications in Computational Physics* Vol. 2, No. 2, pp. 310-333 (2007), and also AIAA-2006-301.
- ²³Venkatakrisnan, V., and Mavriplis, D.J., Implicit Solvers for Unstructured Meshes, *Journal of Computational Physics*, Vol. 105, No. 1, 1993, pp.83-91.
- ²⁴Venkatakrisnan, V., and Mavriplis, D.J., Implicit Method for the Computation of Unsteady Flows on Unstructured Grids, *Journal of Computational Physics*, Vol. 127, No.2, 1996, pp.380-397.

²⁵Soetrisno, M., Imlay, S.T. and Roberts, D.W., A zonal Implicit Procedure for Hybrid Structured-Unstructured Grids, AIAA Paper 94-0645, Jan. 1994.

²⁶Frink, N.T., Assessment of an Unstructured-Grid Method for Predicting 3-D Turbulent Viscous Flows, AIAA Paper 96-0292, Jan. 1996.

²⁷Blanco, M., and Zingg, D.W., A Fast Solver for the Euler Equation on Unstructured Grids Using a Newton-GMRES Method, AIAA Paper 97-0331, Jan. 1997.

²⁸Chen, R.F. and Wang, Z.J., Fast, Block Lower-Upper Symmetric Gauss-Seidel Scheme for Arbitrary Grids.