

A High-Order Accurate Unstructured GMRES Algorithm for Inviscid Compressible Flows

A. Nejat* and C. Ollivier-Gooch†

Department of Mechanical Engineering, The University of British Columbia, 2054-6250 Applied Science Lane, Vancouver, BC V6T 1Z4, Canada

A preconditioned matrix-free GMRES method is presented for unstructured higher-order computation of 2D inviscid compressible flows. Lower-Upper Symmetric Gauss-Seidel (LU-SGS) has been used as the preconditioning strategy, and an approximate first order Jacobian as the preconditioning matrix. The numerical algorithm demonstrates promising convergence performance for both the second and the third order discretization methods in supersonic flow.

I. Introduction

RECENT research on structured mesh flow solver for aerodynamic problems shows that for practical levels of accuracy using a higher order accurate method can reduce computation time and memory usage and improve quality of the solution compared to a second-order scheme.^{1,2} To take advantage of the high flexibility in mesh generation and adaptation for unstructured meshes, we want to apply higher-order accurate methods for unstructured meshes. This approach combines benefits of higher-order methods and unstructured meshes. Although high-order accurate methods for unstructured meshes are reasonably well established,³⁻⁶ application of these methods for physically complicated flows is still a challenge due to very slow convergence. This eliminates the efficiency benefits of higher-order unstructured discretization and limits their application for practical purposes. Consequently, convergence acceleration becomes the key issue for the practical usage of higher-order unstructured solvers.

Newton-GMRES family solvers⁷⁻¹¹ are a commonly used method due to their property of fast convergence. Since GMRES algorithm only needs matrix vector products and these products can be computed by matrix free approach, matrix-free GMRES⁹ is a very attractive technique for dealing with the complicated Jacobian matrices. This approach saves memory usage considerably and, removes the problem of explicit forming of the Jacobian matrix. Recent results of an unstructured mesh solver for Poisson's equation¹² clearly showed the possibility of reducing computational cost required for a given level of solution accuracy using higher-order methods and matrix free GMRES as a convergence acceleration technique. As Poisson's equation is simple and has a linear flux, the resultant Jacobian matrix is well-conditioned, and the convergence rate for GMRES technique is good. In case of more complex problems, including CFD problems, the Jacobian is typically ill-conditioned and effective preconditioning is necessary for good GMRES convergence. Several authors have studied the effect of various preconditioning methods on convergence of matrix-free GMRES both for structured and unstructured meshes.^{7-9,11,13,14} Their research shows incomplete lower-upper (ILU) factorization of the approximate Jacobian is a very efficient preconditioning strategy for CFD problems. Delanaye et al.¹⁵ presented an ILU preconditioned matrix-free GMRES solver for Euler and Navier-Stokes equations on unstructured adaptive grids using quadratic reconstruction. A totally matrix-free implicit method was introduced by Luo et al.¹⁶ for 3D compressible flows using GMRES-LUSGS (Lower-Upper Symmetric Gauss-Seidel). They completely eliminated the storage of the preconditioning Jacobian matrix by approximating the Jacobian with numerical fluxes.

Our objective in this research is to develop an efficient and accurate unstructured solver for inviscid compressible fluid flows. This paper presents preliminary results of using Lower-Upper Symmetric Gauss-Seidel (LU-SGS) for preconditioning of the matrix-free GMRES to compute the higher-order solution of Euler equations.

* PhD Candidate, nejat@mech.ubc.ca, Student Member AIAA.

† Associate Professor, cfog@mech.ubc.ca, Member AIAA.

II. Governing Equations

The unsteady (2D) Euler equations which model compressible inviscid fluid flows, are conservation equations for mass, momentum, and energy. The finite-volume formulation of Euler equations for an arbitrary control volume can be written in the following form of a volume and a surface integral:

$$\frac{d}{dt} \int_{cv} U dv + \oint_{cs} F dA = 0 \quad (1)$$

where

$$U = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ E \end{bmatrix}, \quad F = \begin{bmatrix} \rho u_n \\ \rho u u_n + P \hat{n}_x \\ \rho v u_n + P \hat{n}_y \\ \rho(E + P)u_n \end{bmatrix} \quad (2)$$

In (2), $u_n = u\hat{n}_x + v\hat{n}_y$ and $(\rho \ \rho u \ \rho v \ E)^T$ are the densities of mass, x-momentum, y-momentum, and energy, respectively. The energy is related to the pressure by the perfect gas equation of state: $E = P/(\gamma - 1) + \rho(u^2 + v^2)/2$, with γ the ratio of specific heats for the gas.

III. Algorithm Description

The integral form of Eq. (1) (for control volume “ i ”) can be written in the form of Eq. (3-1) where R (i.e. residual of the governing equations) represents the spatial discretization operator. Linearization in time and applying implicit time integration leads to implicit time advance formula (Eq. (3-2)):

$$\frac{dU_i}{dt} + R_i = 0 \quad (3-1)$$

$$\left(\frac{I}{\Delta t} + \frac{\partial R}{\partial U}\right) \delta U_i = -R_i, \quad \delta U_i = U_i^{n+1} - U_i^n \quad (3-2)$$

where $\frac{\partial R}{\partial U}$ in Eq. (3-2) is the Jacobian matrix resulting from the residual linearization. Eq. (3-2) is a large linear system of equations which should be solved at each time step to obtain an update for the vector of unknowns. As we are only interested in steady state solution, the time marching process continues till the residual of the linear system practically converges to zero.

A. Spatial Discretization

For spatial discretization, a higher-order accurate least-square reconstruction scheme⁶ has been used in the interior of the domain to compute flow quantities at flux integration points to the desired accuracy. Fluxes at control volume boundaries are calculated by Roe’s flux difference-splitting formula¹⁷ using reconstructed flow quantities. Having computed the fluxes, we use Gauss quadrature to integrate the fluxes to the same order of accuracy as the reconstruction. High-order accurate boundary conditions are also used.⁶

B. Linear System Solver

Direct methods for solving the large linear system of Eq.(3-2) resulting from the discretization of governing equations at each time step are not computationally tractable. Therefore iterative techniques such as Krylov family methods are being used for that purpose.⁹ The linear system is asymmetric both in fill and values making GMRES¹⁸

the obvious first choice among iterative solvers. The linear system arising from a high-order discretization has four to five times as many non-zero entries as a second-order scheme. Because of the size of these matrices and the difficulty in computing their entries analytically (even for the second order), we use a matrix-free implementation of GMRES⁹. In the case of Euler equations with nonlinear flux function and possible discontinuities in the solution, using a high-order discretization makes the Jacobian matrix even more off-diagonally dominant and quite ill-conditioned. This degrades or stalls the convergence of GMRES, which is highly dependent on the condition number of the Jacobian matrix. Therefore using a preconditioner for GMRES becomes necessary for practical purposes.

C. Preconditioned GMRES

As previously mentioned, in order to enhance the convergence performance of GMRES solver for complicated linear systems, it is necessary to apply preconditioning. In principal preconditioning is to solve a modified linear system which is relatively better conditioned than the original system. Equation (4) shows the modified system using right-preconditioning.

$$AM^{-1}(MX) = b \quad (4)$$

M in (4) is an approximation to matrix A which has simpler structure and better condition number and consequently is easier to invert. If M is a good approximation to A , AM^{-1} becomes close to identity matrix, increasing the performance of the linear solver. We have chosen to use a matrix-free implementation of flexible GMRES¹⁹ (FGMRES). FGMRES satisfies the residual norm minimization property over the preconditioned Krylov subspace like standard GMRES, but allows variation in the preconditioner! As a preconditioner, we use LU-SGS, an iterative linear solver that has recently received a fair amount of attention both as a preconditioner and as a solver^{16,20}, and has been shown to be more efficient than ILU in some cases.^{16,19}

D. Jacobian Matrix Approximation

Unlike the original linear system matrix A , the preconditioner matrix, M , must be built explicitly. We compute an approximate form of the first-order Jacobian matrix, reducing the size and complexity of the preconditioner matrix M . To build the Jacobian (preconditioner matrix) we first define the residual for a typical cell in terms of flux functions at the control volume faces. For the cell “ i ” with the direct neighbors of N_1 , N_2 and N_3 (Fig. 1), the

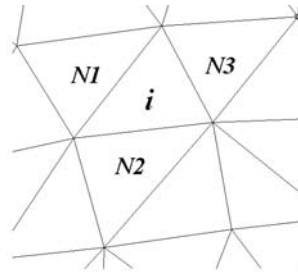


Figure 1. A typical control volume with its first neighbors

residual can be recast in the form of Eq. (5), where \hat{n} is an outward normal vector for each face and l is the face length.

$$R_i = \sum_{faces} F_i \hat{n} ds = F(U_i, U_{N_1})(\hat{n}l)_{i,N_1} + F(U_i, U_{N_2})(\hat{n}l)_{i,N_2} + F(U_i, U_{N_3})(\hat{n}l)_{i,N_3} \quad (5)$$

The next step is taking derivative of the residual function respect to the solution vector of U at control volume “ i ” and its neighbors. Eq. (6-1) through Eq. (6-4) represent the row “ i ” entries of the Jacobian matrix. Here, we only consider the first neighbors as the Jacobian matrix is being computed to the first order of accuracy.

$$J(i, N_1) = \frac{\partial R_i}{\partial U_{N_1}} = \frac{\partial F(U_i, U_{N_1})}{\partial U_{N_1}} (\hat{n}l)_{i, N_1} \quad (6-1)$$

$$J(i, N_2) = \frac{\partial R_i}{\partial U_{N_2}} = \frac{\partial F(U_i, U_{N_2})}{\partial U_{N_2}} (\hat{n}l)_{i, N_2} \quad (6-2)$$

$$J(i, N_3) = \frac{\partial R_i}{\partial U_{N_3}} = \frac{\partial F(U_i, U_{N_3})}{\partial U_{N_3}} (\hat{n}l)_{i, N_3} \quad (6-3)$$

$$J(i, i) = \frac{\partial R_i}{\partial U_i} = \frac{\partial F(U_i, U_{N_1})}{\partial U_i} (\hat{n}l)_{i, N_1} + \frac{\partial F(U_i, U_{N_2})}{\partial U_i} (\hat{n}l)_{i, N_2} + \frac{\partial F(U_i, U_{N_3})}{\partial U_i} (\hat{n}l)_{i, N_3} \quad (6-4)$$

The fluxes at control volume faces are calculated based on Roe’s flux formula,¹⁷ for example, for the face between cell “ i ” and its neighbor, cell “ N_1 ”, the flux function can be expressed by Eq. (7).

$$F(U_i, U_{N_1}) = \frac{1}{2} (F(U_i) + F(U_{N_1})) - \frac{1}{2} \left| \tilde{A} \right|_{(i, N_1)} (U_{N_1} - U_i) \quad (7)$$

Therefore the flux function derivative terms in Eq. (6-1) through Eq. (6-4) simply can be computed by ignoring changes in $\left| \tilde{A} \right|$ matrix. Equation (8-1) and Eq. (8-2) show examples of such derivatives.

$$\frac{\partial F(U_i, U_{N_1})}{\partial U_{N_1}} = \frac{1}{2} \left(\left(\frac{\partial F}{\partial U} \right)_{N_1} - \left| \tilde{A} \right|_{(i, N_1)} \right) \quad (8-1)$$

$$\frac{\partial F(U_i, U_{N_1})}{\partial U_i} = \frac{1}{2} \left(\left(\frac{\partial F}{\partial U} \right)_i + \left| \tilde{A} \right|_{(i, N_1)} \right) \quad (8-2)$$

E. Start up process and Newton-GMRES iteration

As convergence performance and stability of the Newton-GMRES technique, especially for compressible flows, are quite sensitive to the start up, we have to do several implicit iterations before switching to Newton-GMRES iteration. Equation (3-2) is used for implicit pre-iterations. For the Left hand side we use the approximate first order Jacobian, while the residuals in the right hand side are being computed to the right order of accuracy. The LU-SGS algorithm is used as a liner solver for these pre-iterations. Then we switch to Newton-GMRES iteration; at this stage

the $\frac{I}{\Delta t}$ term in Eq. (3-2) is removed (i.e. taking infinite time step). GMRES is used to solve the resultant linear

system at each Newton iteration. However completely solving the linear system at each Newton iteration does not necessarily accelerate overall convergence. Because of the non-linearity of the equations, Newton iteration would not accurately represent the solution and solving the system completely would result in excessive but not helpful computations. Wasted computation can be reduced by using Inexact-Newton method,¹¹. Multiple GMRES inner iterations at each Newton outer iteration are applied to decrease the residual of the linear system by some factor

using restart. Our experience (at least with the presented test cases) shows that normally GMRES iteration without restart would provide us enough drop in residual of the linear system. Therefore we perform one GMRES iteration per Newton iteration. With this approach we do not achieve the quadratic convergence rate of Newton method but we do reach convergence in less CPU-Time.

IV. Results

To investigate the convergence performance and robustness of the purposed GMRES solver with a higher order unstructured discretization, different supersonic flow cases have been studied. Here for brevity, we present two of them which show the general convergence behavior of our numerical experiments. The first test case (Fig. 2) is a supersonic duct with a 12.5 (deg) ramp. The ramp starts at $x=0.04\text{m}$ and ends at $x=0.22\text{m}$. The total length of the duct is 1.2m and has a height of 0.3m at the inlet section. The inflow Mach number is equal to 2. We have used a coarse mesh of 1578 control volumes for this case with some refinement at the ramp's start and end points to capture the shock and expansion waves properly at those locations. The second test case is a Diamond airfoil (15% thickness) with 7722 control volumes. We use 80 control volumes along the chord with the proper refinement at leading and trailing edges and also apex of the airfoil (Fig. 3). The far field Mach number is again 2 and the angle of attack is zero. For both cases, 300 implicit pre-iterations have been performed with the starting CFL=1.0 which increases gradually to 100, and a fixed Krylov-subspace size of 40 is used.

F. Supersonic duct

The residual convergence history both in terms of CPU-Time and residual evaluations has been shown in Fig. 4 and Fig. 5. GMRES-LUSGS decreases the residual by 10 order of magnitude in 16 iterations for 2nd order and 17 iterations for 3rd order. For both orders of accuracy, L2-norm of the residual is reduced by 10 orders of magnitude in 1000 residual evaluations. However as each residual evaluation for the 3rd order discretization is more expensive than for 2nd order, the 3rd order CPU-time is larger. Table 1 shows the summary of the convergence performance for this case.

Table 1. Convergence summary for supersonic duct

	2 nd order	3 rd order
CPU-Time (Sec)	189	286
Residual Evaluations	956	997
GMRES Iterations.	16	17

G. Diamond airfoil

Supersonic flow field around the diamond airfoil has been computed, using second and third order discretization. The convergence summary is shown in Table. 2. This is the case that actually using higher-order discretization has improved the overall performance of the computation, and both CPU-Time and number of residual evaluations for the 3rd order discretization are less than these quantities for the 2nd order discretization. Figure 6 and Fig. 7 demonstrate residual convergence history in terms of CPU-Time and residual evaluations respectively. Steeper slope for the 3rd order scheme in Fig. 7 means the linear system arising from the 3rd order discretization at each GMRES (Newton) iteration is a better representative for the original non-linear system. Figure 8 shows convergence history in terms of iteration number. The first 300 iterations are implicit pre-iterations, taking just small portion of CPU-Time (Fig. 6). They do not decrease the residuals, but they are necessary for the start up. Figure. 9 compares computed and the exact pressure coefficient over the surface along the chord of the airfoil. Except the overshoot (at the leading edge) and undershoot (at the apex), agreement with the exact solution is reasonably good. It is expected by adding limiter to the present scheme these over and under shoots would be cured considerably. Figure 10 and Fig. 11 display Mach contours of the flow field for the 2nd and 3rd order discretizations. The captured shock waves at leading and trailing edges and the expansion fan at the apex of the airfoil are clearly visible. The quality of the

captured discontinuities and high gradient regions in the flow field is noticeably better for the 3rd order discretization respect to the 2nd order case.

Table 2. Convergence summary for diamond airfoil

	2 nd order	3 rd order
CPU-Time (Sec)	1922	1730
Residual Evaluations	2186	1366
GMRES Iterations	46	26

V. Conclusion

A preconditioned matrix-free LUSGS-GMRES algorithm has been presented for higher-order computation of solution of 2D Euler equations. The results show that LUSGS-GMRES works almost as efficiently for the third order discretization as for the second order one. In fact it showed that in some cases, convergence rate can be increased using higher-order discretization. We are currently working on extension of the proposed approach for the 4th order discretization, followed by a detailed study of solution accuracy and resource requirements for all orders of accuracy.

Acknowledgements

This research was supported by the Canadian Natural Science and Engineering Research Council under Grant OPG-0194467.

References

- ¹De Rango, S., and Zingg D. W., "Aerodynamic Computations Using a Higher-Order Algorithm," AIAA Conference Paper 99-0167, 1999.
- ²Zingg, D. W., De Rango, S., Nemec, M., and Pulliam, T. H., "Comparison of Several Spatial Discretizations for the Navier-Stokes Equations," Journal of Computational Physics, Vol.160, 2000, pp. 683-704.
- ³Barth, T. J., Fredrickson P. O., and Stuke M., "Higher-Order Solution of the Euler Equations on Unstructured Grids Using Quadratic Reconstruction," AIAA Conference Paper 90-0013, 1990.
- ⁴Barth, T. J., "Recent Development in High-Order K-Exact Reconstruction on Unstructured Meshes," AIAA Conference Paper 93-0668, 1994.
- ⁵Delanaye, M., and Essers, J. A., "An Accurate Finite Volume Scheme for Euler and Navier-Stokes Equations on Unstructured Adaptive Grids," AIAA Conference Paper, 95-1710, 1995.
- ⁶Ollivier-Gooch, C., and Van Altena M., "A Higher-Order Accurate Unstructured Mesh Finite-Volume Scheme for the Advection-Diffusion Equation, Journal of Computational Physics," Vol. 181, 2002, pp. 729-752.
- ⁷Venkatakrishnan, V., and Mavriplis, D., "Implicit Solvers for Unstructured Meshes," AIAA Conference Paper 91-537, 1991.
- ⁸Orkwis, P. D., "Comparison of Newton's and Quasi-Newton's Method Solvers for Navier-Stokes Equations," AIAA Journal, Vol.31, 1993, pp. 832-836.
- ⁹Barth, T. J., and Linton, S. W., "An Unstructured Mesh Newton Solver for Compressible Fluid Flow and Its Parallel Implementation," AIAA Conference Paper 95-0221, 1995.
- ¹⁰Ollivier-Gooch, C., "Toward Problem-Independent Multigrid Convergence Rates for Unstructured Mesh Methods," In 6th International Symposium on Computational Fluid Dynamics, 1995.
- ¹¹Pueyo, A., and Zingg, D. W., "Improvement to a Newton-Krylov Solver for Aerodynamic Flows," AIAA Conference Paper 98-0619, 1998.
- ¹²Nejat, A., and Ollivier-Gooch C., "A High-Order Accurate Unstructured GMRES Solver for Poisson's Equation," CFD 2003 Conference Proceeding, 2003, pp. 344-349.
- ¹³Blanco, M., and Zingg, D. W., "A Fast Solver for the Euler Equations on Unstructured Grids Using a Newton-GMRES Method," AIAA Conference Paper 97-0331, 1997.
- ¹⁴Manzano, L. M., Lassaline, J. V., Wong, P., and Zingg D. W., "A Newton-Krylov Algorithm for the Euler Equations Using Unstructured Grids," AIAA Conference Paper 2003-0274, 2003.
- ¹⁵Delanaye, M., Geuzaine, Ph., Essers J. A., and Rogiest, P., "A Second-Order Finite-Volume Scheme Solving Euler and Navier-Stokes Equations on Unstructured Adaptive Grids with an Implicit Acceleration Procedure," AGARD 77th Fluid

Dynamics panel Symposium on Progress and Challenges in Computational Fluid Dynamic Methods and Algorithms, Seville, Spain, 1995.

¹⁶Luo, H., Baum, J. D., and Lohner, R., "A Fast Matrix-free implicit Method for compressible Flows on Unstructured Grids," *Journal of Computational Physics*, Vol 146, 1998, pp. 664-690.

¹⁷Roe, P. L., "Approximate Riemann Solvers, Parameter vectors, and difference schemes," *Journal of Computational Physics*, Vol. 43, 1981, pp. 357-372.

¹⁸Saad, Y., and Schultz M. H., "A Generalized Minimal Residual Algorithm for Solving Non-Symmetric Linear Systems," *SIAM J. Sci., Stat. Comp.* Vol. 7, 1986, pp. 856-869.

¹⁹Saad, Y., "A Flexible Inner-Outer Preconditioned GMRES Algorithm," *SIAM J. Sci.,Stat. Comp.* Vol. 14, 1993, pp. 461-469.

²⁰Chen, R. F., and Wang Z. J., "Fast, Block Lower-Upper Symmetric Gauss-Seidel Scheme for Arbitrary Grids," *AIAA Journal*, Vol. 38, 2000, pp. 2238-2245.

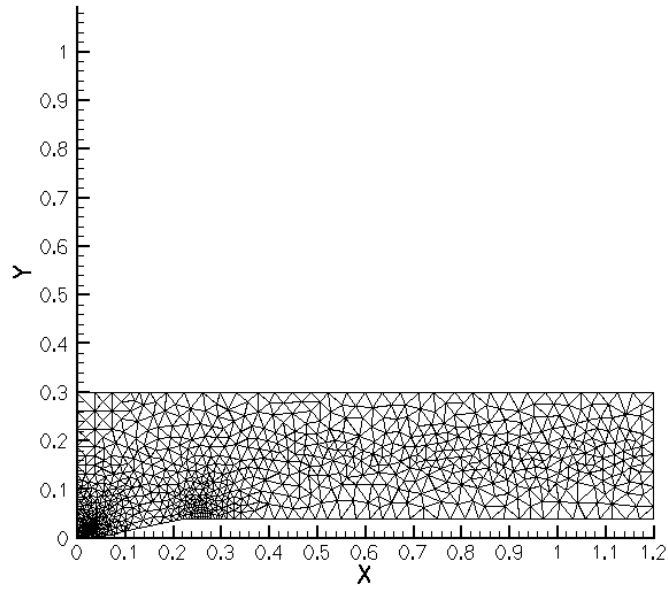


Figure 2. Domain and mesh for the supersonic duct.

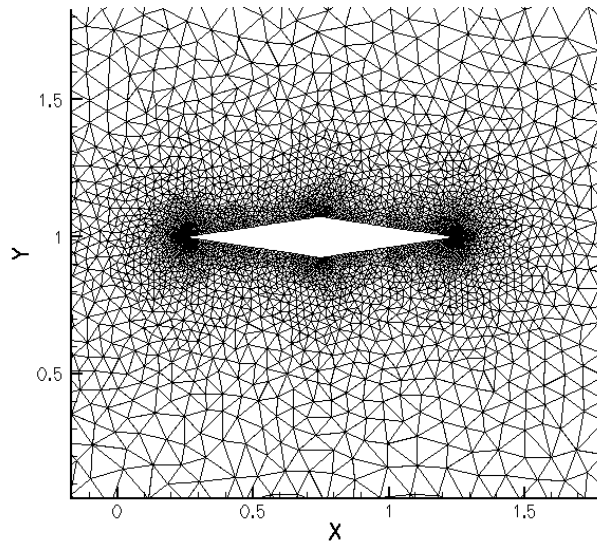


Figure 3. Mesh around the 15% diamond airfoil.

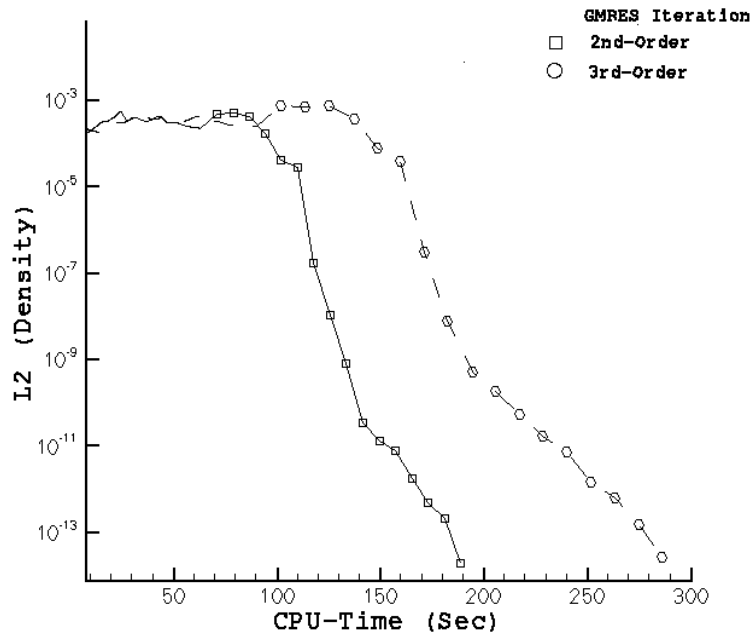


Figure 4. Convergence history in terms of CPU-Time, supersonic duct.

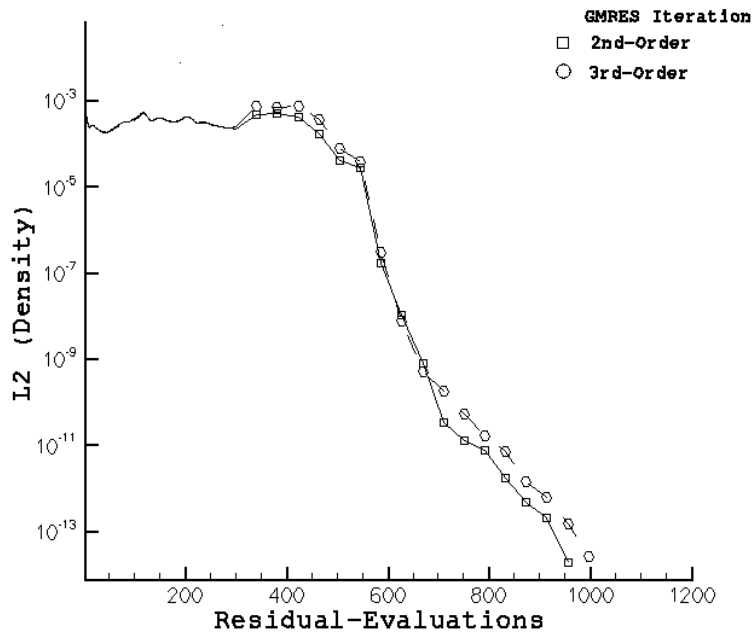


Figure 5. Convergence history in terms of Residual evaluations, supersonic duct.

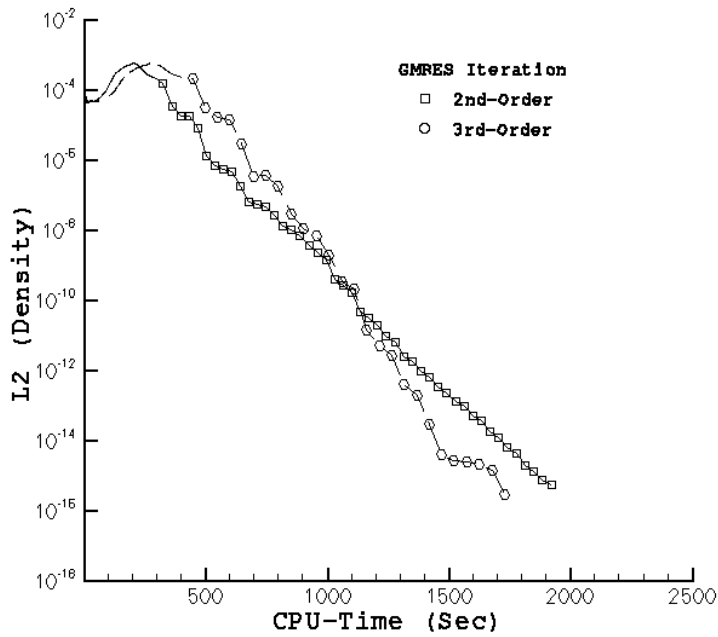


Figure 6. Convergence history in terms of CPU-Time, diamond airfoil.

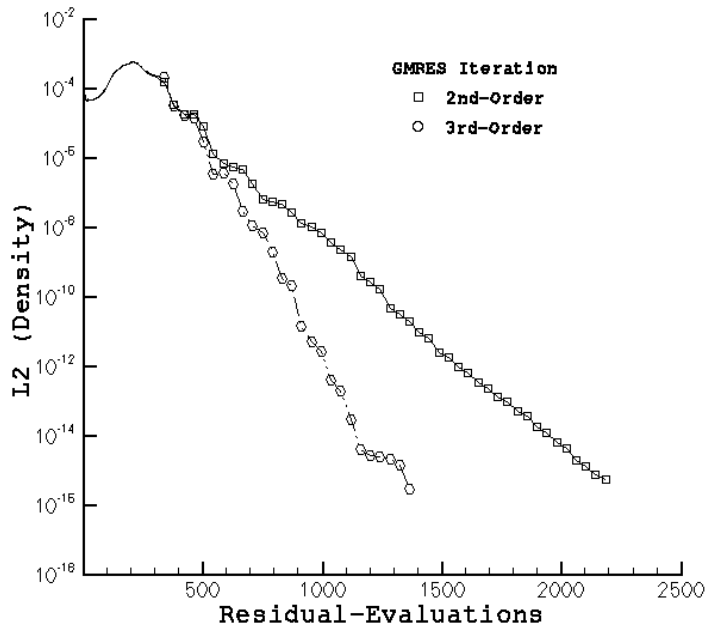


Figure 7. Convergence history in terms of Residual evaluations, diamond airfoil.

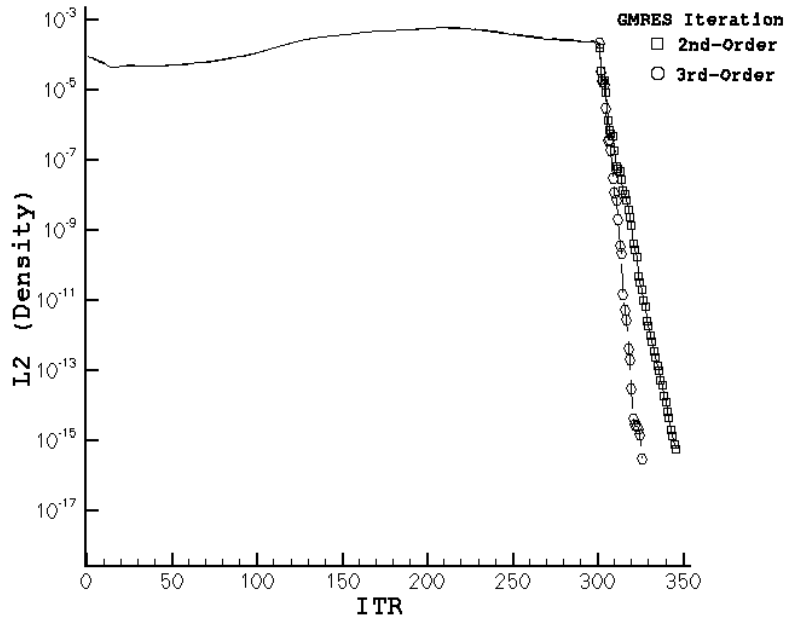


Figure 8. Convergence history in terms of iteration number, diamond airfoil.

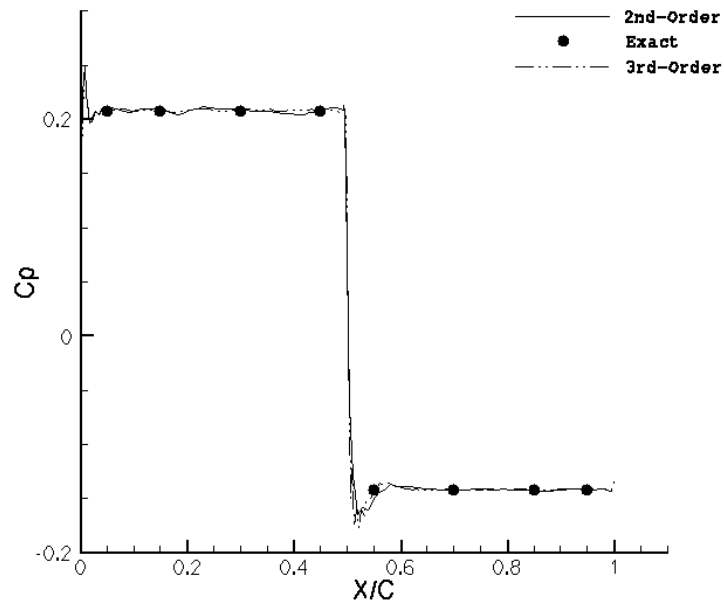


Figure 9. Pressure coefficient along the chord, diamond airfoil.

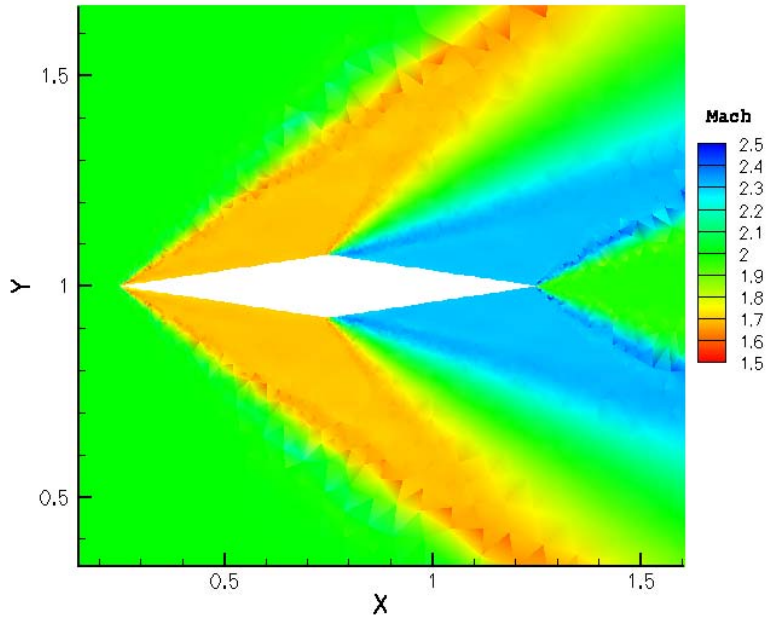


Figure 10. Mach contours, 2nd order discretization, diamond airfoil.

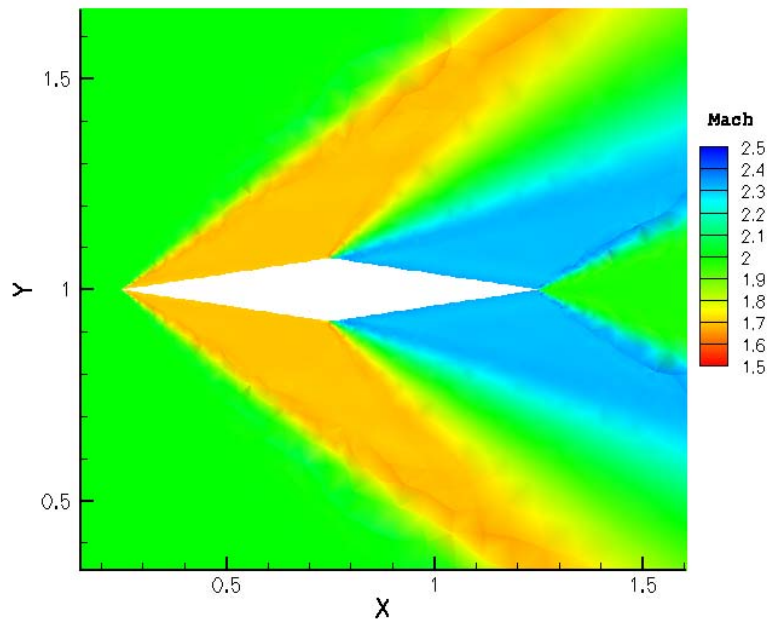


Figure 11. Mach contours, 3rd order discretization, diamond airfoil.