

A High-Order Accurate Unstructured Newton-Krylov Solver 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 Newton-Krylov unstructured flow solver is developed for higher-order computation of the Euler equations using an upwind scheme. The Generalized Minimal Residual (GMRES) algorithm is used for solving the linear system arising from implicit time discretization of the governing equations. An Incomplete Lower-Upper factorization technique is employed as the preconditioning strategy, and an approximate first order Jacobian as the preconditioning matrix. A proper implementation of limiter for higher-order discretization is discussed and a new formula for higher-order limiter is introduced. A defect correction procedure is used for the start-up process before performing Newton iterations. All orders of accuracy show fast convergence characteristics demonstrating the robustness of the proposed approach.

I. Introduction

FOR structured flow solvers, application of higher-order algorithm has progressed considerably. It has been shown that, for practical levels of accuracy in aerodynamic problems, using a higher-order accurate method can be more efficient both in terms of solution time and memory usage and improve the quality of the solution compared to a second-order scheme.^{1,2} Considering the advantages of unstructured flow solvers for complex geometries due to flexibility and adaptation capability of unstructured grids, application of higher-order accurate methods for unstructured meshes would combine accuracy and robustness in numerical simulation of complex geometries. 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. Furthermore, any upwind scheme higher than first order often causes oscillations in the vicinity of sharp gradients and discontinuities; this could produce instability problem. A common solution to that is using limiters, which adversely affects convergence³. At the same time, proper limiter implementation for higher-order reconstruction is quite challenging and it can reduce the order of accuracy even for smooth regions.⁷ Consequently, accuracy and robustness become the key issue for the practical usage of higher-order unstructured solvers.

Newton-Krylov solvers⁸⁻¹⁴ are used extensively in CFD simulations because of their property of semi-quadratic convergence when starting from a good initial solution. Since the GMRES algorithm, among other Krylov techniques, only needs matrix vector products and these products can be computed by matrix free approach, matrix-free GMRES¹⁰ is a very practical technique for dealing with the complicated Jacobian matrices arising from higher-order discretization. This approach saves memory usage considerably and removes the problem of explicitly forming the higher-order Jacobian matrix. Results of an unstructured mesh solver for Poisson's equation¹³ clearly showed the possibility of reducing computational cost required for a given level of solution accuracy using higher-order methods and a non-preconditioned matrix free GMRES as a convergence acceleration technique. In case of problems with non-linear flux function, including CFD problems, the Jacobian is typically ill-conditioned and effective preconditioning is necessary for satisfactory GMRES convergence. Several authors have studied the effect of various preconditioning methods on convergence of matrix-free GMRES both for structured and unstructured meshes.^{8-10,12,15,16} Their research shows incomplete lower-upper (ILU) factorization of the approximate Jacobian is a very efficient preconditioning strategy for CFD problems. Delanaye et al.⁷ presented an ILU preconditioned matrix-free GMRES solver for Euler and Navier-Stokes equations on unstructured adaptive grids using quadratic

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

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

reconstruction. A totally matrix-free implicit method was introduced by Luo et al.¹⁷ 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. Recently a preconditioned matrix-free LUSGS-GMRES algorithm has been successfully implemented for higher order inviscid supersonic flow computations¹⁴. A first order approximate analytical Jacobian was used as a preconditioning matrix. 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.

Despite the simplicity of LU-SGS and its effectiveness in supersonic flow, LU-SGS-like other stationary preconditioners-suffers from stability condition making it inappropriate for effective preconditioning of Newton iterations in transonic flow. The objective of this research is to develop an efficient higher-order unstructured flow solver using Newton-GMRES technique and ILU preconditioning. The solution strategy is to reach a good initial solution state using an implicit defect correction procedure known as the start-up phase, and then moving to Newton iterations where infinite time step is taken to achieve the desirable super-linear convergence rate.

II. Governing Equations

Conservation of mass, momentum, and energy are the principal equations which govern the dynamics of all fluid flows. Neglecting dissipation, viscosity, and thermal conductivity in a flow reduce the fluid flow equations to Euler equations governing inviscid compressible flows. For many practical aerodynamic applications, Euler flow is a relatively accurate representation of the flow field and produces a very good prediction for lift and wave drag. Also, a robust Euler solver is an essential part of any Navier-Stokes solver. The 2D unsteady 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

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

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)):

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

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

where $\frac{\partial R}{\partial U}$ in Eq. (3-2) is the Jacobian matrix resulting from the residual linearization. Equation (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 to generate a k^{th} order reconstruction polynomial (up to the cubic polynomial) for each control volume using a proper reconstruction stencil. That enables us to compute all the flow variables in the interior and at the boundaries up to 4th order of accuracy. Flow quantities at flux integration points are computed to the desired accuracy and fluxes at control volume boundaries are calculated by Roe's flux difference-splitting formula¹⁸ 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 treatment is also employed.⁶

B. Linear System Solver

Iterative methods are the only viable option for inverting large sparse matrices. Among iterative linear solvers, Krylov-subspace family methods are the most common, and amongst these, the GMRES¹⁹ (Generalized Minimal Residual) algorithm has been developed mainly for non-symmetric systems such as those resulting from unstructured discretization. Furthermore the GMRES algorithm minimizes the residual of the linearized system implying that if the linearization of the non-linear system is accurate then GMRES provides the best update for solution at each iteration. 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 GMRES¹⁰, where matrix vector products are approximated by directional derivative formula, Eq. (4). ε_0 is a very small number, typically equal to the square root of machine accuracy.

$$\frac{\partial R}{\partial U} \cdot v \cong \frac{R(U + \varepsilon v) - R(U)}{\varepsilon} \quad (4-1)$$

$$\varepsilon = \frac{\varepsilon_0}{\|v\|_2} \quad (4-2)$$

C. Preconditioned GMRES

In the case of the 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 GMRES convergence, which is highly dependent on the condition number of the Jacobian matrix. Therefore using an effective preconditioner for GMRES becomes necessary for practical purposes. In principal, preconditioning produces a modified linear system which is relatively better conditioned than the original system and therefore that makes it easier to solve by an iterative process. Equation 5 shows the modified system using right-preconditioning.

$$AM^{-1}(MX) = b \quad (5)$$

M in (5) is an approximation to matrix A which has simpler structure and/or better condition number and consequently is less difficult to invert. If M is a good approximation to A, AM^{-1} becomes close to identity matrix, increasing the performance of the linear solver through eigenvalue clustering around unity. Unlike matrix A which need not be computed explicitly in the GMRES algorithm, we need to compute matrix M explicitly to build the preconditioning operator. Jacobian calculation even for a second-order flux is very expensive, therefore we include only the first neighbors in our Jacobian calculation (first-order Jacobian). For effective preconditioning, in addition to applying a good preconditioner matrix we need to employ a good preconditioning strategy. Stationary methods such as Gauss Seidel and SSOR are easy to implement and they are effective in damping high frequency errors.

However, they often have restrictive stability condition, reducing the benefits of Newton method especially for off-diagonal systems. At the same time due to their inherent formulation, they are relatively slow in damping low frequency errors, and therefore they need to be used together with a proper multigrid scheme in preconditioner. A more effective preconditioning strategy is incomplete lower-upper factorization with varying levels of fill (ILU-P). The fill-level in the factorized matrix determines the memory usage and accuracy of ILU decomposition; using larger fill-level often leads to more accurate factorization increasing the performance of preconditioning. However, there is a restriction in increasing fill-level in practice due to memory limitation, which would affect the accuracy of preconditioning. ILU-P factorization is proven to be a robust strategy (specifically ILU-2) for GMRES preconditioning^{12,15} and in general SSOR is no match for incomplete factorization even when the original matrix graph (ILU-0) has been used.⁸ Our experience shows for higher-order methods using the first order preconditioner matrix, ILU-4 provides the best efficiency in preconditioning (especially in transonic flow) with the number of non-zero elements in the factorized matrix about twice the number of non-zero elements in the original preconditioner.²⁰

D. Jacobian Matrix Approximation

In our case (cell centered, 2D, unstructured) each control volume has 3 neighbors, and consequently the first order Jacobian matrix has 4 non-zero blocks per row. 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

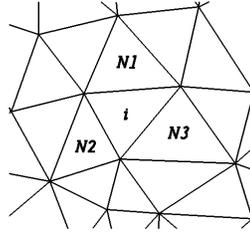


Figure 1. A typical control volume with its first neighbors

cell “ i ” with the direct neighbors of N_1 , N_2 and N_3 (Fig. 1), the residual can be written in the form of Eq. (6), where \hat{n} is an outward normal vector for each face and l is the face length.

$$R_i = \sum_{faces} F_i \hat{n} ds = F(U_i, U_{N_1})(\hat{n})_{i,N_1} + F(U_i, U_{N_2})(\hat{n})_{i,N_2} + F(U_i, U_{N_3})(\hat{n})_{i,N_3} \quad (6)$$

The next step is taking the derivative of the residual function with respect to the solution vector of U at control volume “ i ” and its neighbors. Equation (7-1) through Eq. (7-4) 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.

$$J(i, N_1) = \frac{\partial R_i}{\partial U_{N_1}} = \frac{\partial F(U_i, U_{N_1})}{\partial U_{N_1}} (\hat{n})_{i,N_1} \quad (7-1)$$

$$J(i, N_2) = \frac{\partial R_i}{\partial U_{N_2}} = \frac{\partial F(U_i, U_{N_2})}{\partial U_{N_2}} (\hat{n})_{i,N_2} \quad (7-2)$$

$$J(i, N_3) = \frac{\partial R_i}{\partial U_{N_3}} = \frac{\partial F(U_i, U_{N_3})}{\partial U_{N_3}} (\hat{n})_{i,N_3} \quad (7-3)$$

$$J(i,i) = \frac{\partial R_i}{\partial U_i} = \frac{\partial F(U_i, U_{N_1})}{\partial U_i} (\hat{n}l)_{i,N_1} + \frac{\partial F(U_i, U_{N_2})}{\partial U_i} (\hat{n}l)_{i,N_2} + \frac{\partial F(U_i, U_{N_3})}{\partial U_i} (\hat{n}l)_{i,N_3} \quad (7-4)$$

The fluxes at control volume faces are calculated based on Roe's flux formula,¹⁸ for example, for the face between cell "i" and its neighbor, cell "N_j", the flux function can be expressed by Eq. (8).

$$F(U_i, U_{N_j}) = \frac{1}{2}(F(U_i) + F(U_{N_j})) - \frac{1}{2} |\tilde{A}|_{(i,N_j)} (U_{N_j} - U_i) \quad (8-1)$$

$$|\tilde{A}| = \tilde{X}^{-1} |\tilde{\Lambda}| \tilde{X}, \quad |\tilde{\Lambda}| = \text{Diag} |\tilde{\lambda}| \quad (8-2)$$

\tilde{A} is the Jacobian matrix of the Euler flux function evaluated based on the Roe's average¹⁸ where \tilde{X} are the right eigenvectors and $\tilde{\lambda}$ are the eigenvalues of the Euler flux.²¹ Therefore the flux function derivative terms in Eq. (7-1) through Eq. (7-4) simply can be computed by ignoring changes in $|\tilde{A}|$ matrix. Equation (9-1) and Eq. (9-2) show examples of such derivatives.

$$\frac{\partial F(U_i, U_{N_j})}{\partial U_{N_j}} = \frac{1}{2} \left(\left(\frac{\partial F}{\partial U} \right)_{N_j} - |\tilde{A}|_{(i,N_j)} \right) \quad (9-1)$$

$$\frac{\partial F(U_i, U_{N_j})}{\partial U_i} = \frac{1}{2} \left(\left(\frac{\partial F}{\partial U} \right)_i + |\tilde{A}|_{(i,N_j)} \right) \quad (9-2)$$

If the change in solution is relatively small (start-up phase), this approximation works reasonably well, but for very large changes in solution especially for transonic flow keeping $|\tilde{A}|$ constant can adversely affect the quality of preconditioning matrix degrading the Newton-GMRES convergence rate. Since taking analytical derivative of $|\tilde{A}|$ is quite challenging and expensive, we have chosen to use finite difference perturbation (Eq.(10)) for Jacobian calculation. In Eq.(10), ε is the root square of machine accuracy and α_j is a vector which its jth element is one and the rest of the elements are zero; j is the variable index in the Euler solution vector.

$$\frac{\partial R_i}{\partial U_j} = \frac{R_i(U + \varepsilon \alpha_j) - R_i(U)}{\varepsilon} \quad (10)$$

Jacobian computation cost would be increased by about 70% using finite difference approach; but as we have only one Jacobian computation per GMRES outer iteration in Newton-GMRES phase and number of Newton iterations are generally small, still overall CPU-Time is not affected considerably. In fact as this approach leads to more accurate preconditioner matrix overall convergence rate is increased considerably by reducing number of outer iterations.

E. Monotonicity

Enforcing monotonicity is one of the main issues both for 2nd order and higher-order schemes. Limiters are often needed to suppress the oscillations around discontinuities and to avoid reconstructing non-physical solution (negative density) at gauss points located close to such locations. However limiters introduce some problems. First they hamper the convergence as their values oscillate across the shock, especially in the case of a non-differentiable limiter formulation such as Barth-Jespersion limiter.²² Even using a differentiable limiter would not guarantee good convergence behavior. Secondly, they manipulate the reconstruction polynomial through reducing the solution gradients reducing the accuracy of the reconstructed solution. These issues for higher-order methods are even more

complicated. In general, an ideal limiter is differentiable, does not have a large oscillations around discontinuities and it does act firmly in the shock region suppressing possible over /undershoots. Such a limiter also should not be active in smooth regions despite existence of non-monotone solutions due to higher-order reconstruction. In this research, Venkatakrishnan limiter²³ (semi-differentiable) which addresses most of the aforementioned issues has been employed with some modifications.

$$\Delta_+ = U_{max} - U_i, \Delta_- = U_G - U_i, U_{max} = \text{Max}(U_i, U_{Neighbors})$$

$$\phi = \frac{1}{\Delta_-} \left[\frac{(\Delta_+^2 + \varepsilon^2)\Delta_- + 2\Delta_-^2\Delta_+}{\Delta_+^2 + 2\Delta_-^2 + \Delta_- \Delta_+ + \varepsilon^2} \right] \text{ for } \Delta_- > 0 \quad (11)$$

In Eq. (11), U_G is the reconstructed value at the gauss point, and $\varepsilon^2 = (K\Delta x)^3$, where Δx is the local mesh length scale and can be defined as the diameter of the largest circle that may be inscribed into a local control volume. K is a constant, and some how determines the extent of monotonicity enforcement. A very large value of K essentially means no limiting and could make the solution process unstable. Normally increasing K up to some value would enhance convergence characteristics as long as divergence does not occur. In contrary small value for that constant slows or stops the convergence although it produces more monotonic solution. In this research, in order to achieve favorable convergence behavior $K = 10$ has been used and there was no need to freeze limiter during the solution process. Applying the same limiter value to both linear and higher-order part of the reconstruction polynomial, normally, would result in more diffusive and less accurate solution and to avoid that issue the methodology of Ref. 5 is followed. A limited higher-order reconstruction for a control volume can be cast in the form of Eq. (12).

$$P_{High-Order} = \text{MeanValue} + [(1-\sigma)\phi + \sigma][\text{Linear part}] + \sigma[\text{High-Order part}] \quad (12)$$

In Eq. (12) ϕ is the classic limiter for linear terms, and σ is a limiter for higher order terms. By setting σ equal to zero, the original limited linear reconstruction polynomial would be recovered. For regions with discontinuity we would like to switch from the higher-order to the linear polynomial to prevent any further oscillatory behavior of the reconstruction polynomial due to second and third derivatives. This is done by a discontinuity detector assigning zero to σ , therefore the remaining limited linear part would be used in the shock region. Switching σ between zero and one for the region that limiter fires, stalls the convergence. To overcome this problem, we have defined σ as a differentiable function of ϕ , in such a way that σ is nearly one for the region that $\phi \geq \phi_0$ and would quickly go toward zero for other values of ϕ . Equation (13) is used for calculation of σ as a function of ϕ , with $\phi_0 = 0.8$, and $S = 20$.

$$\sigma = \frac{1 - \tanh(S(\phi_0 - \phi))}{2} \quad (13)$$

For a strong transonic shock over an airfoil, employing large K (required for fast convergence) may cause noticeable over/undershoots in Mach or pressure profile along the chord²³. We have used a simple shock sensor to cure that problem. If the ratio r in (14) is in the order of one then we are in the vicinity of a shock, and if this ratio is far less than one we are in the smooth regions. Density or pressure is selected as a shock sensor variable in (14) since other flow variables (u and v) may experience relatively large changes in some smooth regions such as leading edge and stagnation point.

$$r = \frac{(U_{Max} - U_{Min})_{i,direct-neighbors}}{(U_{Max} - U_{Min})_{Domain}} \quad (14)$$

Following the same methodology for σ , another continuous step like function is defined to compute ϕ_s in such a way that if the ratio of r is larger than a top limit, then ϕ_s quickly goes toward zero and for other values of r it remains very close to 1.0.

$$\phi_s = [1 - \tanh((r - T_{limit})S)]/2 \quad (15)$$

Multiplying ϕ by ϕ_s would effectively suppress over/undershoots in the shock vicinity and helps the convergence as well. Our experience shows that $T_{limit} = 0.3$ is a reasonable value for most transonic cases.

F. Start up process and Newton-GMRES iteration

Convergence performance and stability of the Newton-GMRES technique, especially for compressible flows, are quite sensitive to the start-up process and initial guess. In other words, Newton-GMRES should be started from a relatively good initial condition. Otherwise, convergence stalls or diverges after a couple of iterations. This is due to the fact that linearization of the higher-order Euler flux is not accurate especially if Newton iteration (infinite time step) is performed in early stage of iterations. To reach a good initial guess, implicit time advance (Eq. (3-2)) should be started with small Δt , i.e. low CFL number, and CFL would be increased gradually. To make the start up process even smoother, especially for higher-order discretization, we perform several implicit iterations in the form of defect correction referred to as pre-iterations in this paper. In defect correction phase the right handside of Eq. (3.2) or residual of flux integral is evaluated to the desired order of accuracy (higher-order) while the flux Jacobian, $\frac{\partial R}{\partial U}$, is

computed based on first order discretization (i.e. approximate analytical Jacobian). With this approach, higher-order Jacobian computation which is very expensive and not accurate at this early stage is avoided. Furthermore the resultant linear system is easy to solve because left handside is constructed based on the first order discretization and can be effectively preconditioned by the same matrix, which is available explicitly. The linear solver for the start-up process still is GMRES with ILU preconditioning. ILU-1 is used for preconditioning in start-up phase. We have to do several implicit pre-iterations to reach a good initial state before switching to Newton-GMRES stage. 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. This time matrix vector products in GMRES technique are computed through directional derivatives (Eq. (4)) and are based on higher-order flux calculation. However completely solving the higher-order linear system at each Newton iteration does not necessarily accelerate overall convergence rate. For highly non-linear problems such as transonic flows, the linearized system is not an accurate representative of the original problem. As a result, completely solving the linear system does not necessarily improve the overall convergence rate. The linear system is solved up to some tolerance criteria which is chosen as a fraction (typically $10^{-1} - 10^{-2}$) of the flux integral on the right hand side. With this approach we do not achieve the semi-quadratic convergence rate of Newton method but we do reach convergence in less CPU-time. The same strategy is adopted in the pre-iteration phase as well. For preconditioning of Newton-GMRES phase, ILU-4 is employed with the first order Jacobian, computed by finite difference approach.

IV. Results

To study the convergence and robustness of the proposed Newton-GMRES solver with a higher order unstructured discretization, different test cases have been investigated. Here, we present one subsonic and one transonic test case which include most features of our solver performance characteristics. Test cases include subsonic $M = 0.63, \alpha = 2^\circ$ and transonic $M = 0.8, \alpha = 1.25^\circ$ flows over NACA 0012 airfoil. Five different meshes (O domain) from a coarse mesh to a relatively fine mesh have been used (Fig. 1 and Table 1). For the sake of pressure recovery, all meshes have proper refinement at leading and trailing edges. The far field is located at 25 chords and characteristic boundary conditions are implemented implicitly. Tolerance of solving the linear system for the start up part is 5×10^{-2} and for the Newton-GMRES part is 1×10^{-2} . For all parts and test cases a subspace of 30 has been set and no restart has been allowed. Consequently, on some occasions, the system is being under solved and the tolerance was not reached (especially in Newton-GMRES part in transonic flow). This would increase the number of outer iterations, but from overall performance point of view, we would keep the number of inner iterations inside each GMRES outer iterations relatively reasonable and limit the cost of each outer iteration. In fact that is useful especially for higher-order cases where Jacobian calculation based on directional derivatives becomes quite expensive both in terms of memory and operation. For the 4th-order transonic case, the subspace in Newton iteration has been reduced to 20 to decrease the cost of 4th-order Jacobian calculation, but after reaching the

tolerance of 1×10^{-10} a subspace of 30 is employed again. For all cases, initial condition is set equal to far field flow condition, and steady-state convergence is achieved when L2 norm of density residual is dropped below 10^{-12} .

G. Subsonic Flow

For all meshes, the solution starts with 30 pre-iterations in the start-up process to reach a good initial solution before switching to Newton-GMRES iterations. Starting CFL is 2.0 and it is increasing gradually to CFL=20. The first 15 pre-iterations are done with the first order of accuracy. The rest of 15 pre-iterations are performed in the defect correction form, and first order Jacobian is used both for constructing the left hand side of Eq. (3-2) and for preconditioning the same linear system. The right hand side of Eq. (3-2), flux integral, is evaluated up to the correct order of accuracy. The cost of each pre-iteration includes one Jacobian calculation (first-order), one flux evaluation, and one system solve using GMRES. Our numerical experiments for subsonic flow shows a reasonable starting point for Newton iteration could be easily achieved by relatively small number of pre-iterations and there is no need to decrease the residual by some order of magnitude. Only a rough physical solution over the airfoil is good enough for starting Newton iterations. After start-up, we switch to Newton-GMRES iteration with infinite CFL recovering the true Newton iteration. Table 2 shows convergence summary for 2nd, 3rd, and 4th-order discretizations in terms of total number of residual evaluations, total CPU-time, total work units (i.e. cost of one residual evaluation for the corresponding order of accuracy), number of Newton-GMRES iterations, and the cost of Newton phase in work units. For all meshes, solution has converged after a few Newton iterations. However, total work unit increases as larger mesh is used. The linear system arising from a larger mesh is more difficult to solve than a similar system arising from a coarser mesh. Consequently, more outer iterations needs to be done for reducing the residual of the corresponding non linear system. Notice that the total work unit has increased approximately by factor of two while the mesh size has increased by factor of 16. As we expected, total work rises with increasing order of accuracy, showing the fact that the complexity of the linear system rises with increasing the discretization order. Figure 2 compares convergence history for 3 discretization orders for mesh 3. In general, 3rd-order solution is about 1.3 to 1.5 times and 4th-order solution is about 3.5 to 5.0 times more expensive than 2nd-order solution. In Fig. 3, Mach contours for all orders of accuracy in leading edge region are shown for a coarse mesh (mesh 2). As it is clear, using a higher-order discretization improves the quality of solution in the flow field for a coarse grid. The CL and CD of the subsonic test case over mesh 3 have been tabulated in Table 3 which are in good agreement with an adaptively refined Cartesian mesh result.²⁴ The calculated drag coefficient for 4th-order discretization using mesh 5 (finest mesh) is 0.000308 which despite the potential flow theory is still far from zero. The main contribution to this drag comes from the entropy production at trailing edge of the airfoil. Figure 4 demonstrates the 4th-order entropy contours for the leading edge and trailing edge regions of the airfoil for the finest mesh. The entropy production in leading edge area is limited to stagnation point and is less than 0.007% of the entropy at the far field. At the trailing edge, 0.6% entropy production is observed although the mean entropy around trailing edge is close to 1.0. Figure 5 compares Mach profiles along the chord for the coarsest mesh with the 4th-order computed solution over the finest mesh. Larger flow acceleration over the leading edge is noticeable, in addition to smoothness of Mach profiles for higher-order cases despite coarseness of the mesh. The difference between 4th-order solutions over the finest and the coarsest mesh remains very small.

H. Transonic Flow

For transonic flows, it is more difficult to get fast convergence. This is because of mix subsonic/supersonic nature of the flow and the existence of discontinuities (shock) in solution. The methodology for handling of discontinuity can increase the complexity of the problem. For instance, using a limiter in upwind methods for higher-order schemes is very challenging. This is true especially for implicit schemes, where there could be a large change in solution update in each iteration and limiter values could have large oscillations. In the case of matrix-free Newton methods which matrix vector multiplication is computed through flux perturbation, any oscillatory behavior in limiter could severely degrade the solution convergence. All these facts amount to worsening the conditioning of the linearized systems, and increasing the difficulty in solving them. Flow is solved for all orders of accuracy over mesh 3. For 2nd and 3rd-order start up phases, pre-iterations in the form of defect correction continue until the residual of the non-linear system drops 1.5 order below the residual of the initial condition. In defect correction phase starting CFL number is 2 and it is increased gradually to 200 after 50 iterations. The CFL is not increased above the value of 200 as increasing CFL would not help that much where linearization is not accurate in the start up phase. 69 and 81 pre-iterations were needed to reduce the residual by 1.5 order for 2nd and 3rd-order respectively. In the Newton-GMRES phase, 2nd and 3rd-order cases are followed with infinite time step. The start up phase for the 4th-order case, includes 200 pre-iterations with similar CFL trend. Although residual dropping was not

achieved up to 1.5 order, the solution after 200 iterations was good enough for Newton-GMRES phase. In general, for transonic flow before switching to Newton iteration, the shock location and its strength needs to be captured relatively accurately otherwise Newton iterations would not decrease the residual of the non-linear system effectively. For the 4th-order case, using an infinite time step causes inaccurate linearization and limiter oscillation affecting a large reconstruction stencil. Since this leads to slow convergence, CFL=10,000 has been set for Newton-GMRES phase of the 4th-order. In the limiter, $K=1$ has been used for the 4th order case. Table 4 shows convergence summary for 2nd, 3rd, and 4th-order discretization on mesh 3, and as it was described the 4th-order convergence is considerably slower than the other orders of accuracy. The convergence history graph of the transonic case is shown in Fig. 6. Reduction in convergence slope for 3rd and 4th-order cases is because of decreasing the quality of the solution of the linear system in each Newton iteration. By increasing subspace size and/or allowing restart, it is possible to solve the system more accurately and reduce the number of Newton iterations. But that would increase the number of inner GMRES iterations per each outer iteration with the penalty of CPU-time as it was discussed in section F. Table 5 summarizes the CL and CD of the transonic case for all orders. Both lift and drag coefficients in transonic flow are mainly determined by the shock location and its strength, and to have a good prediction for these coefficients, it is crucial to capture the shocks accurately. Figure 7 displays the Mach contours for all orders of accuracy. The strong shock on upper surface and the weak shock at the lower surface are quite visible. The non-smooth contour lines close to the upper shock (especially at the region that mesh is coarse) is due to limiter firing. Figure 8 shows the limiter values for the 3rd-order transonic case. It is clear that limiter is not active except at the vicinity of the strong shock, and has not been fired for the weaker shock. The Mach profile along the chord is shown in Fig. 9. Both the location and strength of the shocks are in good agreement with AGARD data.²⁵ It appears that the 3rd-order discretization produces less noise in shock capturing, and this is probably because of the quadratic reconstruction characteristic. However, the cubic reconstruction associated to the 4th-order solution demonstrates some oscillations in the shock; this behavior is often expected in approximating a discontinuity by a higher-order polynomial.

V. Conclusion

An ILU preconditioned Newton-GMRES algorithm has been presented for higher-order computation of solution of 2D Euler equations. The Robustness and fast convergence of the approach have been demonstrated. Effect of the mesh size on convergence of the Newton-GMRES solver for a subsonic case (flow without discontinuity and limiter) has been studied. A new formulation for higher-order terms in limiter was introduced and implemented successfully. A detailed comparison between 2nd and 3rd and 4th-order discretizations both in terms of convergence and accuracy was presented for a subsonic and a transonic test case. Using an efficient start up method, good preconditioner, and effective preconditioning strategy are the key factors in robustness of any Newton-GMRES solver. We are currently working on increasing the efficiency of the preconditioning (especially for 4th-order discretization), and enhancing the robustness of the limiter for higher-order application.

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.
- ⁷Delanaye, 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.

⁸Venkatakrisnan, V., and Mavriplis, D., "Implicit Solvers for Unstructured Meshes," AIAA Conference Paper 91-537, 1991.

⁹Orkwis, P. D., "Comparison of Newton's and Quasi-Newton's Method Solvers for Navier-Stokes Equations," AIAA Journal, Vol.31, 1993, pp. 832-836.

¹⁰Barth, 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.

¹¹Ollivier-Gooch, C., "Toward Problem-Independent Multigrid Convergence Rates for Unstructured Mesh Methods," In 6th International Symposium on Computational Fluid Dynamics, 1995.

¹²Pueyo, A., and Zingg, D. W., "Improvement to a Newton-Krylov Solver for Aerodynamic Flows," AIAA Conference Paper 98-0619, 1998.

¹³Nejat, A., and Ollivier-Gooch C., "A High-Order Accurate Unstructured GMRES Solver for Poisson's Equation," CFD 2003 Conference Proceeding, 2003, pp. 344-349.

¹⁴Nejat, A. and Ollivier-Gooch. C., "A High-Order Accurate Unstructured GMRES Algorithm for Inviscid Compressible Flow," AIAA Conference Paper, 2005-5341, 2005.

¹⁵Blanco, 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.

¹⁶Manzano, 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.

¹⁷Luo, 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.

¹⁸Roe, P. L., "Approximate Riemann Solvers, Parameter vectors, and Difference Schemes," Journal of Computational Physics, Vol. 43, 1981, pp. 357-372.

¹⁹Saad, 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.

²⁰Nejat, A., and Ollivier-Gooch. C., "On Preconditioning of Newton-GMRES algorithm for a Higher-Order Accurate Unstructured Solver," 14th Annual Conference of CFD Society of Canada, 2006 Conference Proceeding, 2006 (to be appear).

²¹Rohde, A., "Eigenvalues and Eigenvectors of the Euler Equations in General Geometries," AIAA Conference paper, 2001-2609, 2001.

²²Barth T. J., and Jespersen, D. C., "The Design and application of Upwind Schemes on Unstructured Meshes," AIAA Paper 89-0366, 1989.

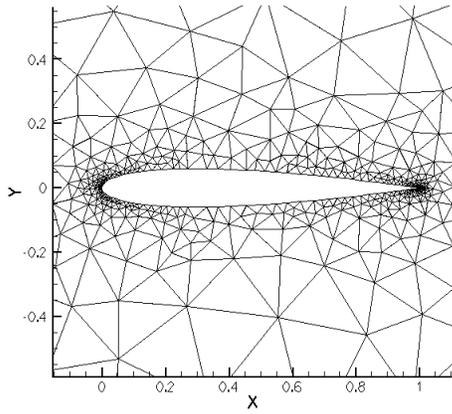
²³Venkatakrisnan, V., "On the Accuracy of Limiters and Convergence to Steady State Solutions," AIAA Conference paper, 93-0880, 1993.

²⁴De Zeeuw D. L., "A Quadtree-Based Adaptively-Refined Cartesian Grid Algorithm for Solution of the Euler Equations," PhD Thesis, Aerospace Engineering Department, University of Michigan, 1993.

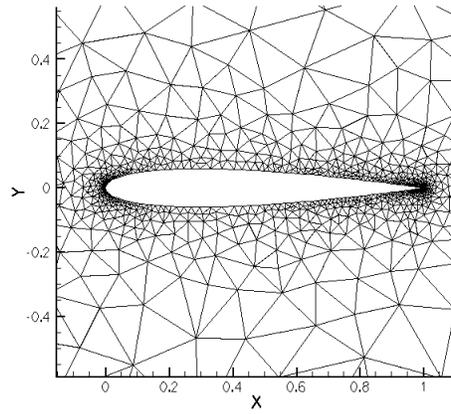
²⁵AGARD AR-211, Test Cases for Inviscid Flow Fields, AGARD, 1985.

Table 1. Mesh detail for NACA 0012 airfoil.

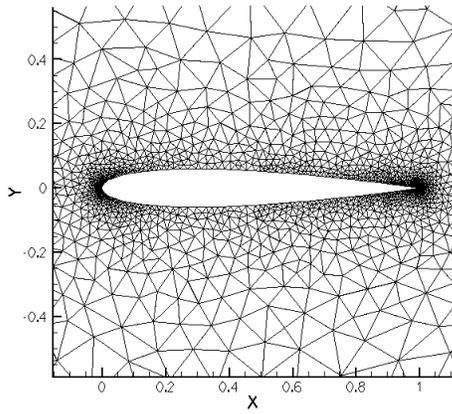
Mesh	No. of CVs along the chord (each side)	Total No. of Control Volumes
Mesh 1	61	1245
Mesh 2	101	2501
Mesh 3	127	4958
Mesh 4	198	9931
Mesh 5	260	19957



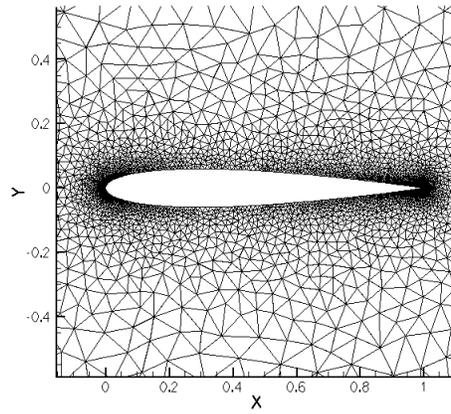
Mesh 1



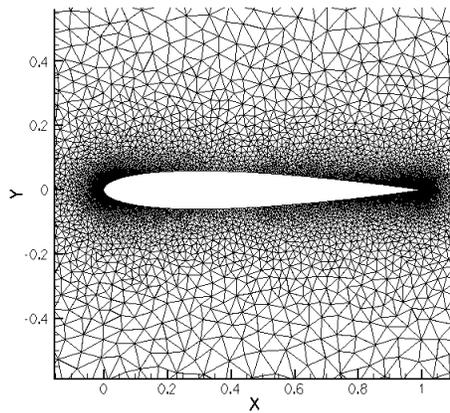
Mesh 2



Mesh 3



Mesh 4



Mesh 5

Figure 1. Meshes for NACA 0012 airfoil.

Table 2. Convergence summary for NACA 0012 airfoil, $M = 0.63, \alpha = 2^\circ$.

Test Case	Total No. of Residual Evaluations	Total –CPU Time (sec)	Total Work Unit	No. of Newton-GMRES Iteration	Newton-GMRES Work -Unit
Mesh1					
2 nd	108	5.9	235	3	102
3 rd	131	8.7	198	4	124
4 th	244	27.7	261	7	231
Mesh2					
2 nd	125	13.4	291	3	127
3 rd	136	17.6	212	4	128
4 th	283	61.6	298	8	261
Mesh3					
2 nd	126	26.1	322	3	129
3 rd	147	35.5	240	4	138
4 th	247	102.4	289	7	242
Mesh4					
2 nd	158	61.4	376	4	175
3 rd	158	73	271	4	159
4 th	318	231.8	377	9	324
Mesh5					
2 nd	254	164.8	558	7	327
3 rd	254	208.2	414	7	280
4 th	414	584.5	488	12	427

Table 3. CL and CD for NACA 0012 airfoil, Mesh 3, $M = 0.63, \alpha = 2^\circ$.

Mesh	CL	CD
2 nd /Mesh 3	0.324905	0.000401969
3 rd /Mesh 3	0.325242	0.000498239
4 th /Mesh 3	0.325317	0.000347570
Ref. 24 / Cartesian (adaptive) 10694 CV	0.32890	0.0004

Table 4. Convergence summary for NACA 0012 airfoil, Mesh 3, $M = 0.8, \alpha = 1.25^\circ$.

Test Case	Total No. of Residual Evaluations	Total –CPU Time (sec)	Total Work Unit	No. of Newton-GMRES Iteration	Newton-GMRES Work -Unit
2 nd /Mesh 3	197	65.6	279	4	91
3 rd /Mesh 3	241	106.7	281	5	119
4 th /Mesh 3	450	311.4	590	10	221

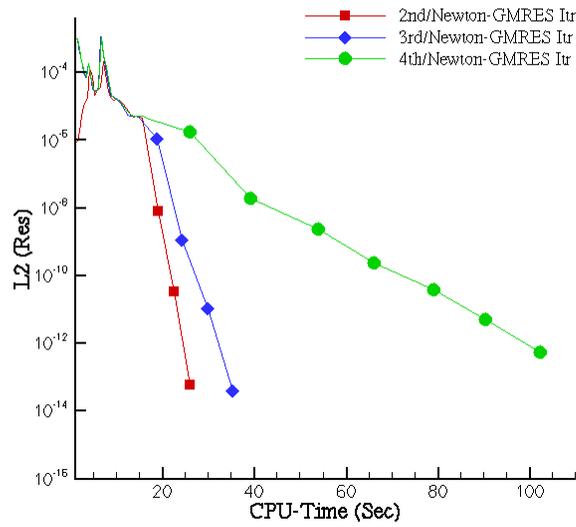


Figure 2. Convergence history in terms of CPU-Time for NACA 0012 airfoil (Mesh 3), $M = 0.63$, $\alpha = 2^\circ$.

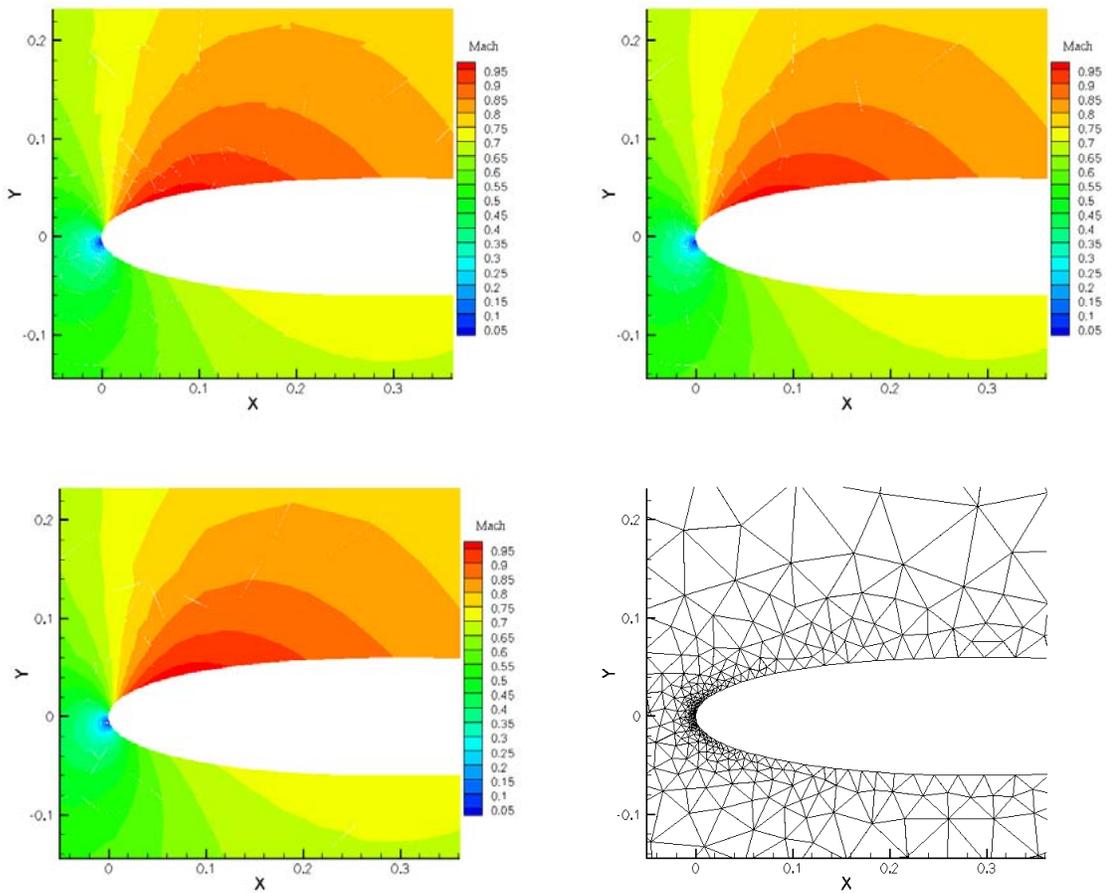


Figure 3. Close up of Mach contours and the mesh for 2nd, 3rd and 4th-orders of accuracy (left to right), NACA 0012 airfoil, Mesh 2, $M = 0.63$, $\alpha = 2^\circ$.

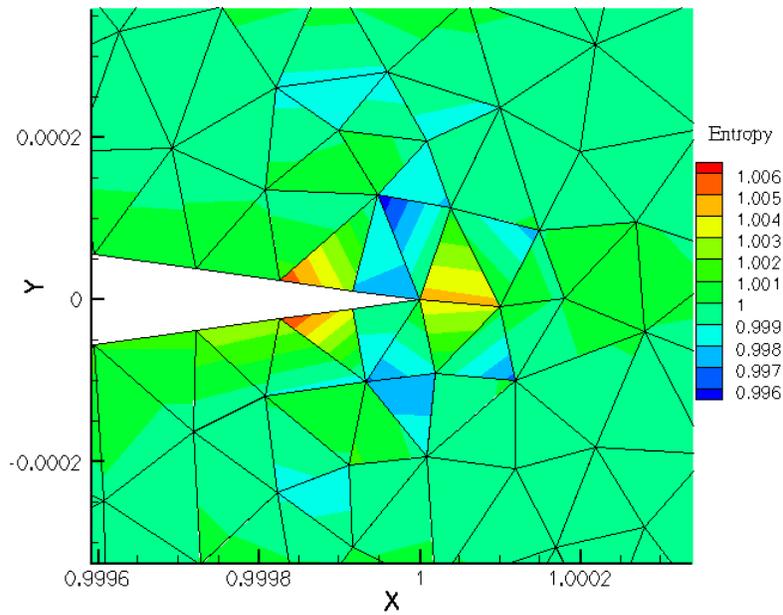
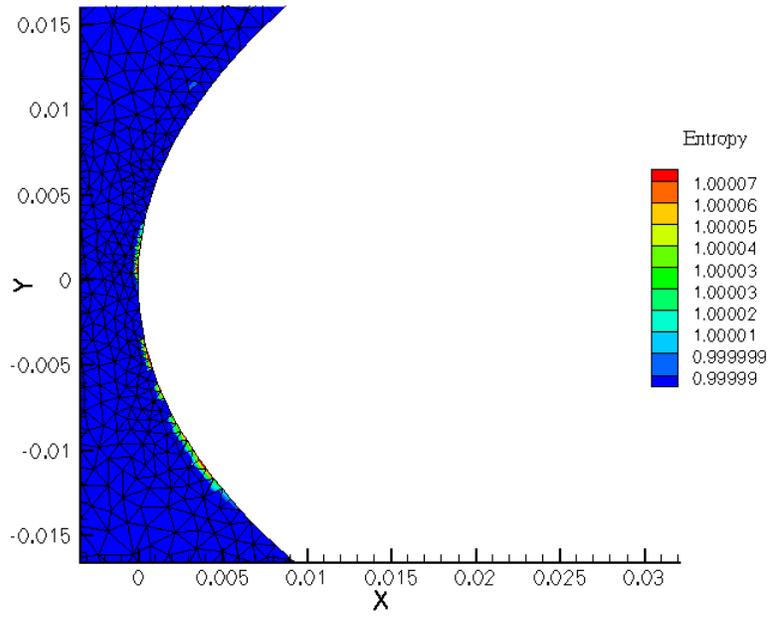


Figure 4. Entropy contours around the leading edge (above) and trailing edge (below) of the NACA 0012 airfoil, 4th-order solution, Mesh 5, $M = 0.63$, $\alpha = 2^\circ$.

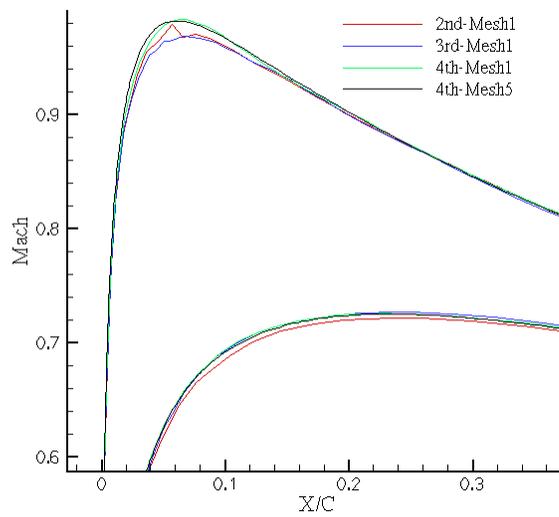
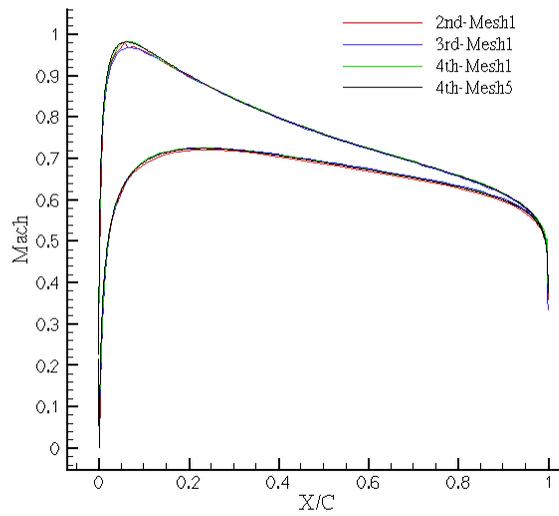


Figure 5. Mach profile along the chord, NACA 0012 airfoil, $M = 0.63$, $\alpha = 2^\circ$.

Table 5. CL and CD for NACA 0012 airfoil, Mesh 3, $M = 0.8$, $\alpha = 1.25^\circ$.

Mesh	CL	CD
2 nd /Mesh 3	0.337593	0.0220572
3 rd /Mesh 3	0.339392	0.0222634
4 th /Mesh 3	0.345111	0.0224720
Ref. 25 / Structured (192*39)	0.3474	0.0221

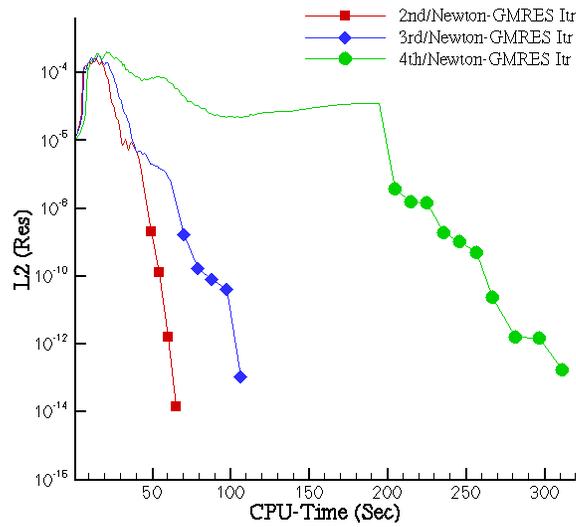


Figure 6. Convergence history in terms of CPU-Time for NACA 0012 airfoil, $M = 0.8$, $\alpha = 1.25^\circ$.

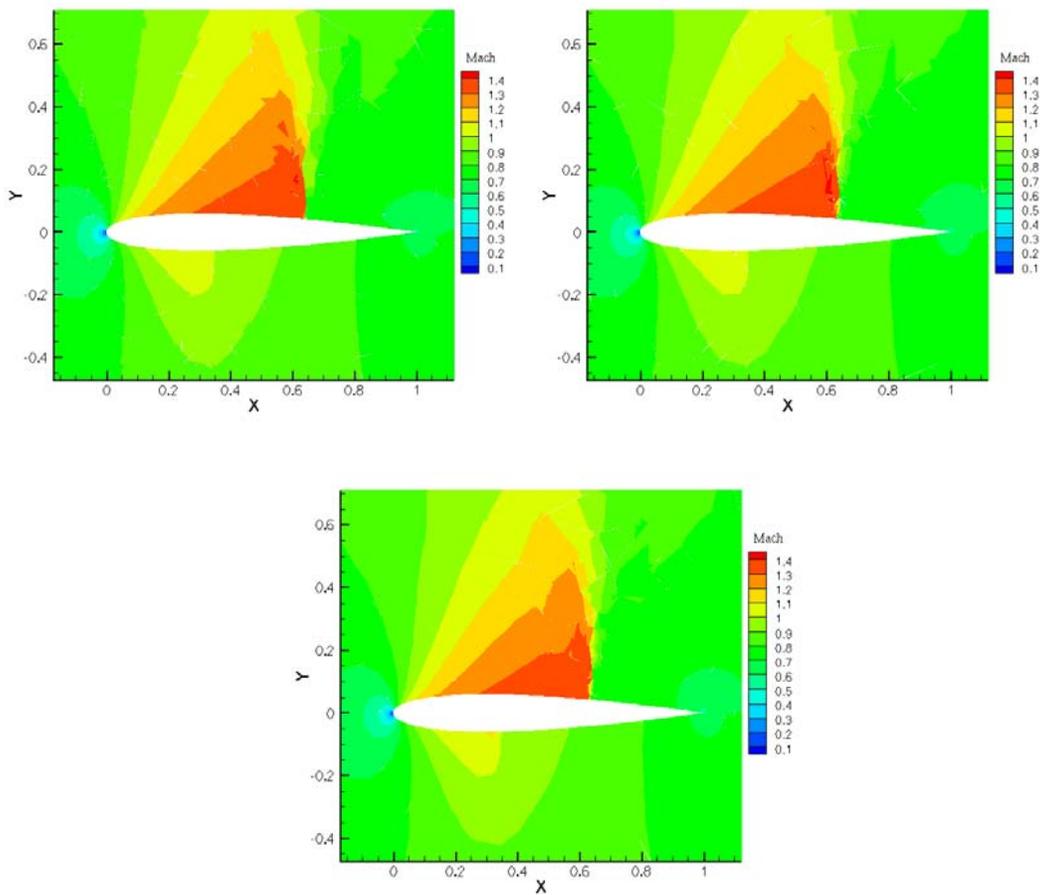


Figure 7. Mach contours for 2nd (top left), 3rd (top right), and 4th-order (below) for NACA 0012 airfoil, Mesh 3, $M = 0.8$, $\alpha = 1.25^\circ$.

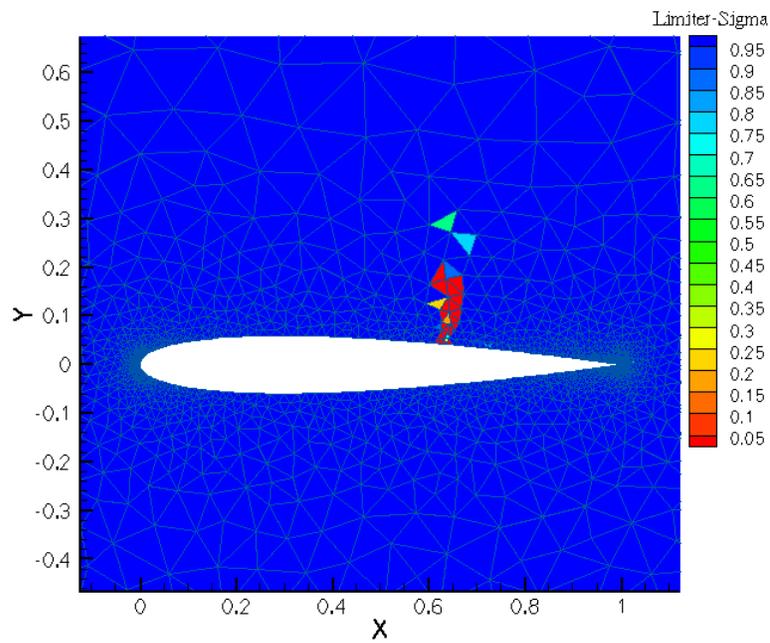
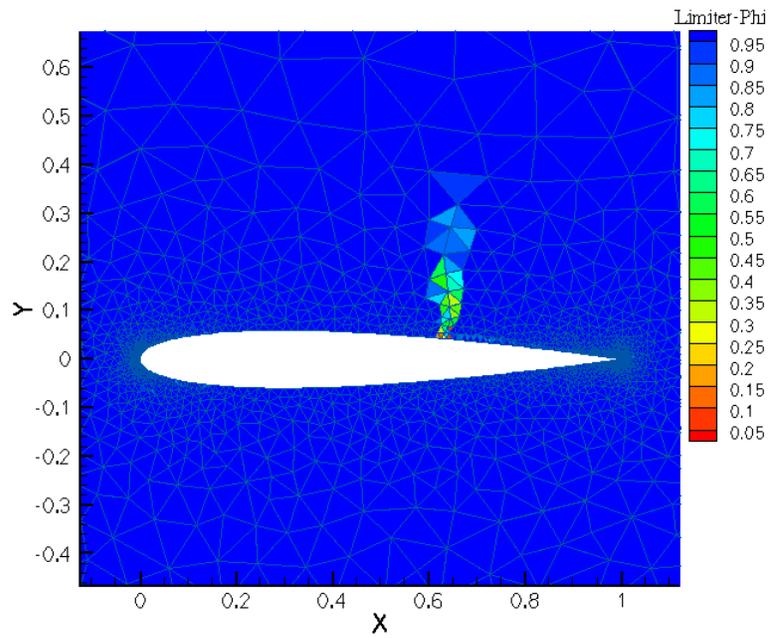


Figure 8. Limiter value for ϕ and σ , NACA 0012 airfoil (Mesh 3), 3rd-order solution, $M = 0.8$, $\alpha = 1.25^\circ$.

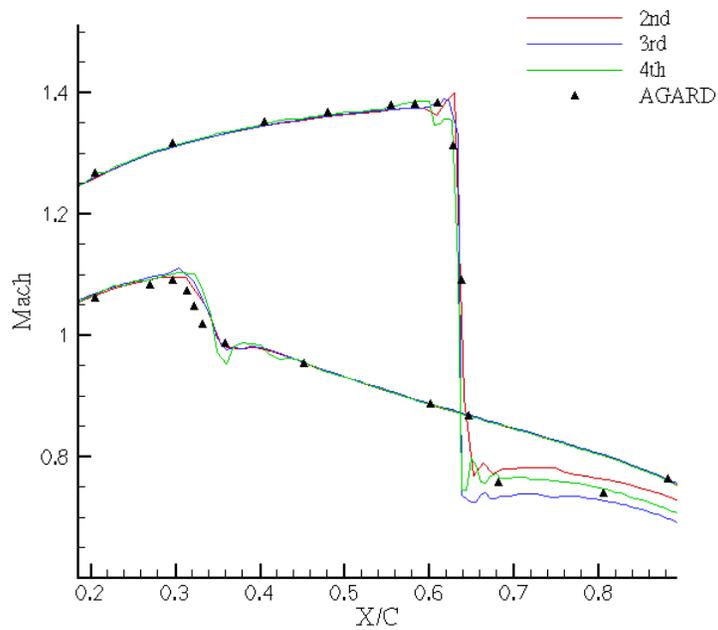
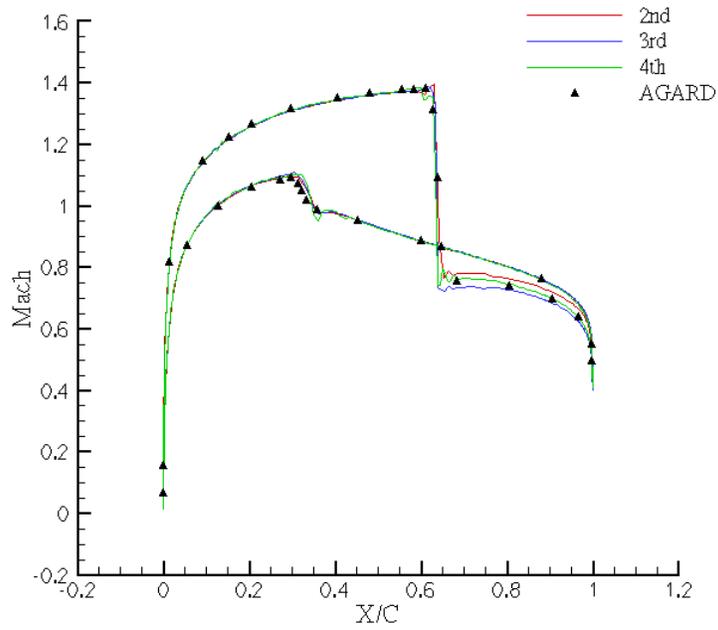


Figure 9. Mach profile along the chord, NACA 0012 airfoil, Mesh 3, $M = 0.8$, $\alpha = 1.25^\circ$.