# On Preconditioning of Newton-GMRES algorithm for a Higher-Order Accurate Unstructured Solver

Amir Nejat and Carl Ollivier-Gooch

*Department of Mechanical Engineering, University of British Columbia*
*Vancouver, BC, V6T 1Z4, Canada*

Email: *nejat@mech.ubc.ca*

## ABSTRACT

A preconditioned Newton-GMRES (Generalized Minimal Residual) algorithm has been used for unstructured higher-order computation of inviscid subsonic, transonic, and supersonic flows. The Incomplete Lower-Upper factorization (ILU-P), and LU decomposition have been used for preconditioning of the GMRES linear solver and their performance is compared for various orders of accuracy and speed regimes. The effect of fill-level in ILU-P on both efficiency of the preconditioning and memory requirement is examined. The importance of accurate preconditioning for efficient convergence of the higher-order unstructured solver is demonstrated.

## 1. INTRODUCTION

Application of higher-order algorithms has progressed considerably for structured meshes. Studies show that achieving the desired accuracy in practical aerodynamic flow using the higher-order algorithm not only is possible but also has some advantages [1]. Upwind methods having higher than second-order accuracy for solving compressible aerodynamic flows on unstructured meshes have been introduced more than a decade ago [2]. However, applying these methods efficiently still remains a challenge due to slow convergence.

Implicit time advance (Backward Euler) is a fairly common and efficient method for steady state solution of fluid flow equations (Eq. 2). In this technique, temporal discretization of fluid flow equation leads to a very large sparse linear system which needs to be solved at several time steps till steady state convergence. Newton-Krylov iterative methods for solving these large linear systems have been used in CFD since the late 80's [3] and are considered an attractive approach due to their property of semi-quadratic convergence when starting from a good initial guess. Since the GMRES [4] algorithm, among other Krylov techniques, only needs matrix vector products and these products can be computed by matrix free approach, matrix-free GMRES [5] is a very practical approach for dealing with the complicated Jacobian matrices arising from higher-order discretization. Matrix-free approach removes the issue of explicitly forming the higher-order Jacobian matrix and reduces memory usage noticeably. However convergence of the GMRES algorithm (as a linear solver) is highly dependent on the conditioning of the linear system (i.e. Jacobian matrix). Using higher-order discretization introduces more off-diagonal entries in Jacobian and makes the linear system even more difficult to solve. In the case of inviscid compressible flow, with non-linear flux function and possible discontinuities in the solution, applying a good preconditioner for the GMRES algorithm becomes a necessity. Several authors have studied the effect of various preconditioning methods on convergence of matrix-free GMRES [5,6,7]. Their studies suggest that incomplete lower-upper (ILU-P) factorization of the approximate Jacobian matrix is a very efficient preconditioning strategy for variety of Newton-Krylov solvers. P in ILU-P is the fill-level in the factorized matrix. P equal to zero means no fill is permitted during ILU decomposition. In ILU-0 the factorized matrix and the original preconditioner matrix built from direct neighbors have the same graph (i.e. same location for non zero-elements). Choosing P larger than zero would allow some additional fill-in in factorized matrix which normally increases the accuracy of factorization and quality of preconditioning. However, increasing the fill-level would be at the expense of memory usage and increasing the preconditioning cost.

In this research, we compare the convergence behavior of various fill level "P" in ILU-P preconditioning and LU preconditioning for inviscid compressible flows with second, third and fourth-order discretizations. The effect of different fill level, P, is examined in terms of CPU-Time and memory usage. The sensitivity of preconditioning would be discussed for different type of flow speeds. Finally, we summarize benefits and limitations of our approach.

## 2. 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:

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

In (1) $U$ is the solution vector (unknowns) and $F$ is the flux vector of Euler equations.

## 3. ALGORITHM DESCRIPTION

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

$$(\frac{dU_i}{dt} + R_i) = 0 \qquad (2\text{-}1)$$

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

where $\frac{\partial R}{\partial U}$ in Eq. 2-2 is the Jacobian matrix resulting from the residual linearization in time. Eq. 2-2 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 until the residual $R_i$ converges to machine zero.

### 3.1 Spatial Discretization

For spatial discretization, a cell-centered higher-order accurate least-square unstructured reconstruction scheme [8] has been used in the interior of the domain in order to compute flow quantities at control volume boundaries up to the desired accuracy. Fluxes at control volume boundaries are calculated by Roe's flux difference-splitting formula [9] using reconstructed flow quantities. Having computed the fluxes, we integrate along each control volume boundary using Gauss quadrature integration technique with the proper number of points to get the required order of accuracy for the flux integral. In transonic flow and in the presence of excessive oscillation close to discontinuity regions, using an appropriate limiter is inevitable. Venkatakrishnan limiter [10] because of its superior convergence properties has been used. Higher-order terms in reconstructed polynomials are dropped if limiter is fired and only limited linear terms are kept [7].

### 3.2 Linear System Solver

Because the linear system (matrix) is asymmetric both in fill and values, GMRES is an appropriate choice for solving Eq. 2.2. The linear system arising from high-order discretization has four-five times as many non-zero entries as a second-order scheme. Considering the difficulty in computing entries of the higher-order Jacobian analytically (even for the second order), we use matrix-free implementation of GMRES [5], where matrix vector products are approximated by directional derivative formula, Eq. 3.

$$\frac{\partial R}{\partial U}.v \cong \frac{R(U+\varepsilon v) - R(U)}{\varepsilon} \qquad (3\text{-}1)$$

$$\varepsilon = \frac{\varepsilon_0}{\|v\|_2} \qquad (3\text{-}2)$$

$\varepsilon_0$ is a very small number, typically equal to square root of machine accuracy.

## 4. PRECONDITIONING

To enhance the convergence performance of GMRES solver for complicated linear systems, it is necessary to apply preconditioning. 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 4 shows the modified system using right-preconditioning.

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

$M$ in (4) 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 second-order flux is very expensive, therefore we include only the first neighbors in our Jacobian calculation (first-order Jacobian). In our case (cell center 2D unstructured) each control volume has 3 neighbors, and consequently the first order Jacibian matrix has 4 non-zero blocks per row [11]. The first-order Jacobian is easy to compute, it is better conditioned than the higher-order Jacobian (much easier to invert), and it has the same graph of the unstructured mesh, saving considerable amount of memory. 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 SOR 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. Other type of preconditioning techniques are incomplete lower-upper factorization methods (ILU). They are proven to be a robust strategy (specifically ILU-2) for GMRES preconditioning [5,6,7], and in general SOR is no match for incomplete factorization even when the original matrix graph (ILU-0) has been used [3]. As it was mentioned in the introduction section, 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.

## 5. START UP PROCESS

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. 2-2) should be started with small $\Delta t$, 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. 2.2 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 [11]. 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. We have to do several implicit pre-iterations to reach a good initial guess before switching to Newton-GMRES stage. At this stage the $\frac{I}{\Delta t}$ term in Eq. 2-2 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. 3) and it is 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.

## 6. RESULTS

To assess the effect of fill-level, P, in ILU-P preconditioning, we would employ P values from zero to four, in comparison with complete LU factorization for three different types of flow field and three orders of accuracy. However in the start up process, as our left hand side has constructed based on first order discretization and relatively low CFL has been used, it is not necessary to use a large fill level. Therefore for ILU-P ($P \geq 1$) and LU, ILU-1 was used as the preconditioning technique in the strat-up process. In the case of ILU-0 preconditioning, ILU-0 was used in the start-up process for preconditioning of the linear system. Tolerance of solving the linear system for the start up part is $5\times10^{-2}$ and for the Newton-GMRES part is $1\times10^{-2}$. 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 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 $10^{-12}$. Test cases include subsonic, $M = 0.63,\ \alpha = 2^{o}$, and transonic, $M = 0.85,\ \alpha = 1^{o}$, flows over NACA 0012 airfoil. The domain is a circle with radius of 25 chords, meshed with 5354 triangles (Fig. 1). A supersonic test case also has been studied over a diamond airfoil with 15% thickness at $M = 2.0,\ \alpha = 0^{o}$ on a mesh of 6163 triangles (Fig. 2).

## 6.1 Subsonic Flow

Solution starts with 50 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. In the defect correction phase, first order Jacobian is used both for constructing the left hand side of Eq. 2-2 and for preconditioning the same linear system. The right hand side of Eq. 2-2, 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. Obviously, this is not the most efficient start-up process, and freezing the Jacobian for multiple iterations could reduce the start up time. For the subsonic test case Jacobian has been frozen for 5,3, and 1 iterations, in 2$^{nd}$ , 3$^{rd}$, and 4$^{th}$ order cases respectively. The update from a first order linear system also has been over relaxed by factors of 1.25, 1.1, and 1.05 for the mentioned orders of accuracy. After start-up, we switch to Newton-GMRES iteration, with infinite CFL, recovering the true Newton iteration. Table. 1 shows convergence history for 2$^{nd}$ , 3$^{rd}$, and 4$^{th}$ order discretization in terms of total number of residual evaluations, total CPU-time, total work units (i.e. cost of one residual evaluation for corresponding order of accuracy), number of Newton-GMRES iterations and the cost of Newton phase in work units. For all orders of accuracy, ILU-P=2,3&4 preconditioning provide a reasonable convergence performance. However, ILU-P=0&1 have been less successful, especially for ILU-0 where, the number of Newton iterations are considerably higher. For 4$^{th}$ order discretization, accurately inverting the preconditioner seems to be more beneficial than for the other two orders of accuracy and total CPU-time can be improved up to 12% when ILU-4 is used (compare to ILU-2). LU preconditioning, for all cases, provides the minimum number of outer iterations, but as each LU decomposition is more expensive than ILU-P factorization, total CPU-time has increased up to 15% with respect to ILU-4. Increasing the fill-level has its own penalty which is increasing the cost of decomposition and memory usage. Decomposition cost normally balances itself through more accurate and effective preconditioning, and as a result the number of inner iterations inside GMRES necessary to satisfy convergence tolerance is decreased; but memory usage is elevated almost linearly by fill-level. Table 2 shows the number of non-zero elements in the factorized matrix. For example ILU-2 needs 50% and ILU4 needs a little bit more than 100% extra memory for storing the decomposed matrix in comparison with the original matrix graph. In the LU case, it is quite clear because of the huge memory requirements, its practical application would not be feasible. Figure 3 displays convergence history for three orders of accuracy preconditioned by ILU-4. Residuals are dropped sharply after starting Newton iterations, although the increase in number of Newton iterations is clear for higher-order cases as the complexity of the system rises. Overall, for the subsonic case using a large fill-level reduced CPU-time, but even P=2 produces satisfactory results.

## 6.2 Transonic Flow

For transonic flows, it is relatively 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, especially for implicit higher-order schemes, is very challenging [7]. All these facts amount to worsening the

conditioning of the linearized system, demanding more effective preconditioning.
In the start up phase, 150 pre-iterations have been performed for 2nd and 3rd order cases, starting from CFL=2 and reaching CFL=200 after 100 iterations. The same phase for the 4th order cases, takes 200 pre-iterations with similar CFL trend. No Jacobian freezing or update relaxation are considered in defect correction phase this time. In the Newton-GMRES phase, 2nd and 3rd order cases are continued with infinite time step. However in the 4th order cases using an infinite time step causes inaccurate linearization and limiter oscillation, affecting a large reconstruction stencil. This leads to slow convergence, therefore CFL=10,000 has been set for Newton-GMRES phase of 4th order. Table 3 summarizes convergence history for transonic flow. Like subsonic flow, ILU-P=0&1 appear to be improper preconditioners. ILU-P=2,3&4 are working more or less properly for 2nd and 3rd orders, but in the 4th order cases ILU-4 demonstrates a great improvement over ILU-2&3, 40% and 25% respectively in CPU-time. Not surprisingly, ILU-4 has been shown for all cases (looking at outer iteration number), best approximates the complete LU factorization. Therefore it makes the preconditioning more accurate. Secondly and even more important, ILU-4 factorization has a larger bandwidth, consequently providing more stability in the preconditioning, in the presence of limiter oscillations and perturbation of the linearized system. These factors make ILU-4 more effective than other preconditioners. Figure 4 shows convergence history in terms of CPU-time for 4th order cases, where performance of ILU-1,2,3&4 are compared. In Fig. 5, convergence history of different orders of accuracy has been demonstrated (preconditioned by ILU-4).

## 6.3 Supersonic Flow

Pure supersonic flow, is the easiest one for preconditioning when no limiting is needed as is the case for our diamond airfoil. The high Mach number helps to damp low frequency errors quite effectively as soon as the main structure of shocks and expansion fans is formed over the airfoil. It is clear that having a good initial guess is the key for Newton-GMRES convergence, otherwise due to highly non-linear and discontinuous regions in the supersonic flow field convergence stalls shortly after starting point. Our defect correction scheme is very effective in finding such a solution before switching to Newton-iteration. Only 30 pre-iterations with increasing CFL from 1 to 50 were enough to get a decent starting point. Infinite CFL number is taken in Newton phase, and full convergence were achieved rapidly for all orders of accuracy. Table 4 has summarized the convergence history for all orders and preconditioners. This time all ILU-P preconditioners including ILU-0 perform well. However, ILU-0 takes 25%-35% more CPU-time. Very small number of Newton iterations (three) in 2nd and 3rd order cases show that both linearization and solving of the resultant system were done almost accurately. In the case of 4th order, again the complexity of the linear system results in more outer (Newton) iterations. Fig. 6 displays convergence history of three different orders of accuracy preconditioned by ILU-2. Notice that the residual of the non-linear system still is large right before switching to Newton iteration, but by that point the main structure of the flow field has been formed, and a relatively large residual drop after first Newton iteration is observed.

## 7. CONCLUDING REMARKS

An efficient ILU-preconditioned Newton-GMRES unstructured solver has been employed for computation of subsonic, transonic and supersonic aerodynamic flows in 2nd , 3rd and 4th orders of accuracy. Effect of different fill-levels in incomplete factorizations have been studied in detail compare to LU factorization. Number of Newton iterations for ILU-4 and LU for all cases were almost the same, showing the equivalence of the quality of these two preconditioning techniques given the pre-set tolerance and first order preconditioner matrix. ILU-2 in most cases seem to be a promising preconditioner as it was reported by other references. Using ILU-3&4 slightly improves CPU-time for some of the cases at the expense of increasing memory usage. In the case of 4th order transonic flow, using ILU-4 accelerates convergence rate dramatically due to its accuracy and stability in preconditioning. Therefore for tough higher-order cases, increasing the fill-level could be quite beneficial if the memory penalty is bearable.

### ACKNOWLEDGEMENTS

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

## References

1-De Rango, S., and Zingg D. W., “Aerodynamic Computations Using a Higher-Order Algorithm,” AIAA Conference Paper 99-0167, 1999.

2-Barth, 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.

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

4-Saad, 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.

5-Barth, 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.

6-Blanco, 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.

7-Delanaye, M., Geuzaine Ph. and Essers, J. A., "Compressible Flows on Unstructured Adaptive Grids" AIAA Conference Paper, 97-2120, 1997.

8- Ollivier-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, pp. 729-752, 2002.

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

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

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

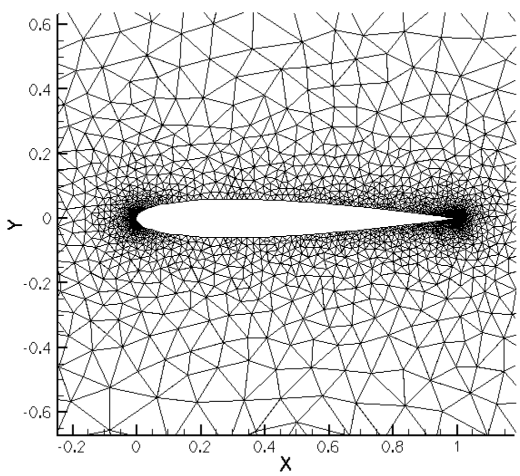


Fig. (1) NACA 0012 airfoil, 5354 control volumes.

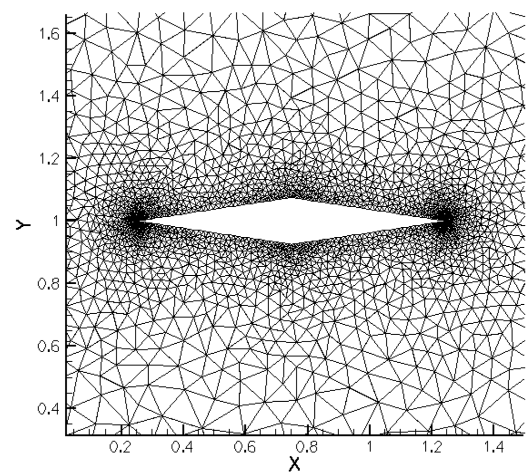


Fig. (2) 15% diamond airfoil, 6163 control volumes.

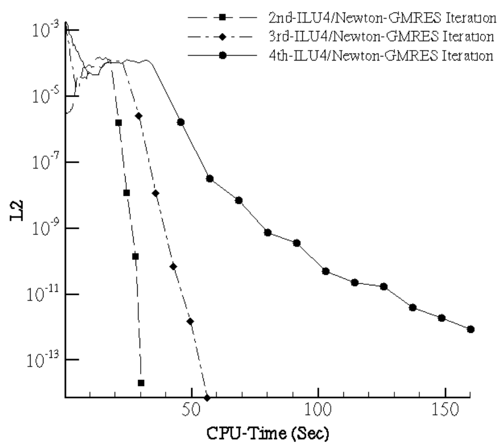


Fig. (3) Convergence history for subsonic flow.

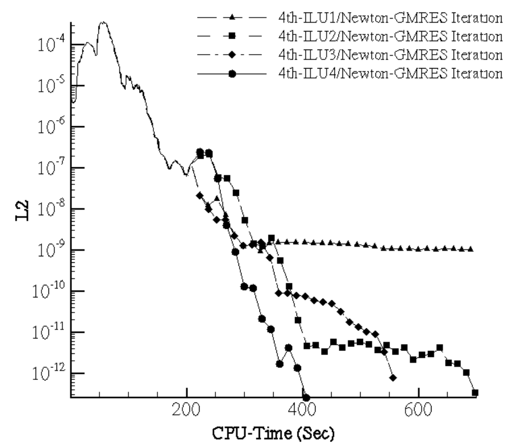


Fig. (4) Effect of fill-level on convergence of transonic flow.

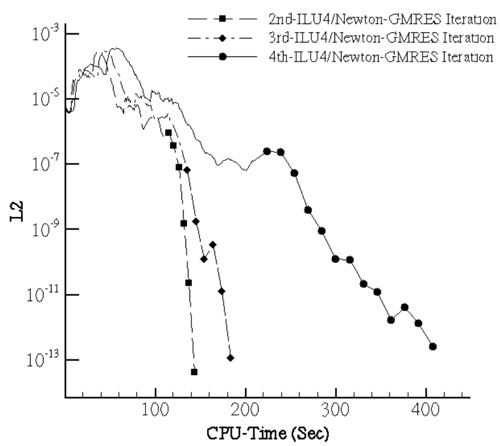


Fig. (5) Convergence history for transonic flow.

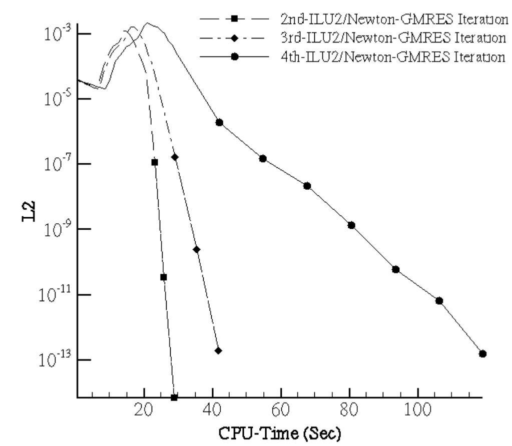


Fig. (6) Convergence history for supersonic flow.

| Preconditioning | Total No. of Residual Evaluations | Total –CPU Time (sec) | Total Work Unit | No. of Newton-GMRES Iteration | Newton-GMRES Work -Unit |
|---|---|---|---|---|---|
| $2^{nd}$/ILU-0 | 818 | 102.5 | 1220 | 24 | 971 |
| $2^{nd}$/ILU-1 | 242 | 39 | 469 | 6 | 249 |
| $2^{nd}$/ILU-2 | 178 | 32.3 | 398 | 4 | 171.8 |
| $2^{nd}$/ILU-3 | 168 | 31.7 | 378 | 4 | 159 |
| $2^{nd}$/ILU-4 | 153 | 30.4 | 371 | 4 | 148 |
| $2^{nd}$/LU | 143 | 42.5 | 524 | 4 | 300 |
| | | | | | |
| $3^{rd}$/ILU-0 | 658 | 130.3 | 874 | 9 | 705 |
| $3^{rd}$/ILU-1 | 274 | 61.8 | 418 | 7 | 264 |
| $3^{rd}$/ILU-2 | 210 | 51.5 | 338 | 5 | 188 |
| $3^{rd}$/ILU-3 | 210 | 52 | 345 | 5 | 192 |
| $3^{rd}$/ILU-4 | 210 | 56.3 | 370 | 5 | 220 |
| $3^{rd}$/LU | 178 | 59.2 | 400 | 4 | 248 |
| | | | | | |
| $4^{th}$/ILU-0 | 882 | 327 | 1009 | 26 | 895 |
| $4^{th}$/ILU-1 | 562 | 213.5 | 657 | 16 | 551 |
| $4^{th}$/ILU-2 | 466 | 182.3 | 556 | 13 | 450 |
| $4^{th}$/ILU-3 | 434 | 171.4 | 522 | 12 | 416 |
| $4^{th}$/ILU-4 | 402 | 160.4 | 494 | 11 | 387 |
| $4^{th}$/LU | 370 | 184 | 559 | 10 | 454 |

Table 1. Convergence summary for subsonic flow.

| Preconditioning | No of. non-zero elements factorized matrix | Ratio of non-zero elements Factorized Mat. / Original Mat. |
|---|---|---|
| ILU-0 | 338528 | 1 |
| ILU-1 | 420544 | 1.25 |
| ILU-2 | 505632 | 1.5 |
| ILU-3 | 601312 | 1.8 |
| ILU-4 | 717664 | 2.1 |
| LU | 2.4326e8 | 7.2 |

Table 2. Number of non-zero elements in factorized matrix.

| Preconditioning | Total No. of Residual Evaluations | Total –CPU Time (sec) | Total Work Unit | No. of Newton-GMRES Iteration | Newton-GMRES Work -Unit |
|---|---|---|---|---|---|
| 2nd/ILU-0 | Fails Full Converge. | ------ | ------ | ------ | ------ |
| 2nd/ILU-1 | 502 | 170 | 726 | 11 | 617 |
| 2nd/ILU-2 | 342 | 143 | 610 | 6 | 143 |
| 2nd/ILU-3 | 342 | 143.1 | 611 | 6 | 144 |
| 2nd/ILU-4 | 340 | 143.4 | 613 | 6 | 146 |
| 2nd/LU | 338 | 163 | 697 | 6 | 231 |
| | | | | | |
| 3rd/ILU-0 | Fails Full Converge. | ------ | ------ | ------ | ------ |
| 3rd/ILU-1 | 470 | 221.4 | 583 | 10 | 247 |
| 3rd/ILU-2 | 374 | 193 | 512 | 7 | 174 |
| 3rd/ILU-3 | 342 | 184.6 | 488 | 6 | 150 |
| 3rd/ILU-4 | 342 | 183.5 | 485 | 6 | 148 |
| 3rd/LU | 342 | 205.5 | 542 | 6 | 205 |
| | | | | | |
| 4th/ILU-0 | Fails Full Converge. | ------ | ------ | ------ | ------ |
| 4th/ILU-1 | Fails Full Converge. | ------ | ------ | ------ | ------ |
| 4th/ILU-2 | 1224 | 698 | 1303 | 32 | 911 |
| 4th/ILU-3 | 936 | 557 | 1014 | 23 | 635 |
| 4th/ILU-4 | 616 | 407 | 764 | 13 | 373 |
| 4th/LU | 616 | 450 | 844 | 13 | 453 |

Table 3. Convergence summary for transonic flow.

| Preconditioning | Total No. of Residual Evaluations | Total –CPU Time (sec) | Total Work Unit | No. of Newton-GMRES Iteration | Newton-GMRES Work -Unit |
|---|---|---|---|---|---|
| 2nd/ILU-0 | 158 | 38 | 395 | 4 | 160 |
| 2nd/ILU-1 | 109 | 30.5 | 318 | 3 | 103 |
| 2nd/ILU-2 | 92 | 29 | 302 | 3 | 87 |
| 2nd/ILU-3 | 85 | 28.4 | 296 | 3 | 81 |
| 2nd/ILU-4 | 81 | 28.3 | 295 | 3 | 80 |
| 2nd/LU | 76 | 41.4 | 431 | 3 | 215 |
| | | | | | |
| 3rd/ILU-0 | 222 | 63 | 382 | 6 | 227 |
| 3rd/ILU-1 | 158 | 47.7 | 289 | 4 | 151 |
| 3rd/ILU-2 | 126 | 41.9 | 253 | 3 | 115 |
| 3rd/ILU-3 | 126 | 42 | 254.8 | 3 | 117 |
| 3rd/ILU-4 | 126 | 42.4 | 255.4 | 3 | 117.4 |
| 3rd/LU | 125 | 56.7 | 344 | 3 | 205 |
| | | | | | |
| 4th/ILU-0 | 350 | 159.1 | 417.7 | 10 | 333.5 |
| 4th/ILU-1 | 254 | 119.2 | 316.9 | 7 | 239.2 |
| 4th/ILU-2 | 254 | 119.1 | 316.1 | 7 | 238.6 |
| 4th/ILU-3 | 254 | 120.4 | 318.6 | 7 | 239.6 |
| 4th/ILU-4 | 254 | 120.3 | 319.2 | 7 | 242 |
| 4th/LU | 254 | 153 | 416 | 7 | 337 |

Table 4. Convergence summary for supersonic flow.