Iranian Journal of Science & Technology, Transaction B, Engineering, Vol. 31, No. B1, pp 1-12
Printed in The Islamic Republic of Iran, 2007

A COMPARISON OF ADAPTIVE-GRID REDISTRIBUTION AND EMBEDDING WITH A COMBINATION OF THESE TWO METHODS*

M. AMERI** AND A.A. OLOOMI

Dept. of Mechanical Engineering, University of Shahid Bahounar, Kerman, Iran
Email: ameri_mm@mail.uk.ac.ir

Abstract– Several approaches have been employed for grid adaptation. The most widely used ones utilize adaptive-grid redistribution and adaptive-grid embedding. However, the combination of these two main methods (r-h) is also possible. This work compares redistribution and embedding with a combination of these two methods for steady transonic airfoil flows. The Euler equations are integrated into a steady state by an explicit, finite volume, Ni's Lax-Wendroff scheme. Comparison with other numerical solutions is employed in order to evaluate the accuracy and efficiency of the techniques. The combination of adaptive-grid redistribution and embedding is somewhat more complex than the adaptive embedding method, but the results indicate that for two-dimensional invisid flows, when high accuracy is required, their combination is more efficient.

Keywords– Redistribution, embedding, adaptive-grid method, Euler equations

1. INTRODUCTION

One of the difficult problems facing the computational fluid dynamics is the lack of priori information concerning the gradients in the dependent variables to be calculated. Without this information the grid used in many numerical solutions is usually wasteful, and does not satisfactorily resolve those important regions where gradients exist. Adaptation is the process by which the computational mesh changes in response to an evolving solution. Certain regions of a computational domain will always have complex flow features such as shocks, expansions and boundary layers, while other regions will have smooth and relatively uniform flow. The complex flow regions are often regions with high gradients and larger numerical errors. The basic idea behind grid adaptation is to increase the number of grid points in regions of high gradients, and reduce the number of grid points where the flow is smooth, thus increasing the accuracy and speed of convergence.

Therefore, the algorithms of grid generation and the solution must be linked. For this reason, adaptive methods are much more complicated than fixed-grid methods. Nevertheless, the saving of memory and time for many flows are so large that this complexity is permissible. Adaptive methods are suitable for flows with different scales, and as the scale differences are higher, the saving is larger.

Adaptive schemes may be placed into one of three basic divisions. The divisions are r-methods, in which a fixed number of nodes are redistributed, h-methods, in which mesh is automatically refined, and p-methods, in which the local polynomial degree is increased. Among the adaptive-grid methods, redistribution and embedding techniques have been the center of attention of researchers, hence the most essential works have been done on them.

*Received by the editors September 5, 2005; final revised form January 2, 2007.
**Corresponding author

The numbers of grid points are fixed in the redistribution schemes but the nodes move from regions of small error or gradient to regions of large error or gradient. The main idea is that as the physical solution develops, the grid points are moved towards regions of large variations and therefore are concentrated in these regions. The redistribution method has two main advantages. First, the computational grid can often be aligned with interesting flow features, and second, it is relatively easy to implement the method into an existing code. The main disadvantages are that the cells tend to become too skewed or too irregular, and it is difficult to know, in advance, how many nodes are needed.

In the grid-embedding methods, the cells are locally divided in the regions of large errors or gradients. Indeed, the nodes are added in these regions. Hence, the equations are solved in a composite grid, a fixed global grid and adaptively embedded patches in the special regions. Embedded adaptive-grid method reduces, remarkably, the number of necessary nodes, memory and computational time. The main disadvantages are that computer time and memory increase with refinement, and that the coding and data structure are relatively difficult. In addition, some generated nodes do not connect to all neighbor nodes, and thus special works are needed to solve the flowfield. Adaptive-grid embedding reduces nodes and computational time, but maximum attainable accuracy will be equal to globally fine grid corresponding to the finest level of embedded grid.

Dannenhoffer [1] compared these two methods by solving steady inviscid transonic flows. The results have shown that redistribution schemes are easier to implement than the embedding schemes, but increasing accuracy in redistribution depends on initial nodes and, if high accuracy is desired, the embedding method is probably the better approach. If modest increase in the accuracy is adequate then redistribution is probably a better approach, however if high accuracy is desired, then embedding is the recommended approach.

Researchers have also used simultaneous or a combination of adaptive techniques. These combinations are often adaptive h-p finite element methods which exploit both h- and p- refinement according to the situation that is faced [2-4]. Kallinderis and Baron [5] have used grid-embedding associated with equation adaptation in order to solve two-dimensional laminar flows.

The combination of adaptive-grid redistribution and embedding (r-h) has been described in [6]. It was used to solve transonic and supersonic inviscid flows in channels. The resulting flowfield has shown an increased shock and expansion waves resolution in comparison with the uniform fine grid and embedding method alone. However, the evaluation of the advantages of this combination has apparently been insufficient. The accuracy of this method has only been evaluated qualitatively, not precisely with the two main adaptive-grid methods of redistribution and embedding.

This paper focuses on the use of the combination of adaptive-grid redistribution and embedding (r-h) in solving the steady transonic flows around NACA0012 airfoil. These relatively simple, two-dimensional examples of transonic flow past an airfoil illustrate the need for a solution-adaptive grid procedure. Even if only an inviscid Euler solution is required, the computational grid must resolve the high flow acceleration region around the leading edge, as well as the high streamwise gradients through the shocks. Results are obtained and compared qualitatively and quantitatively with globally fine grid, adaptive-grid redistribution and adaptive-grid embedding. Comparisons with other numerical simulations are employed in order to evaluate the efficiency and accuracy of this method.

2. GOVERNING EQUATIONS AND NUMERICAL METHOD

The invisid compressible fluid flow is governed by the Euler equations. They represent conservation of mass, momentum and energy. These equations for unsteady two-dimensional flows can be written as

$$\frac{\partial U}{\partial t} = -\left[\frac{\partial F}{\partial x} + \frac{\partial G}{\partial y}\right] \tag{1}$$

The state and flux vectors are

$$U = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ e \end{bmatrix} \quad F = \begin{bmatrix} \rho u \\ \rho u^2 + P \\ \rho uv \\ \rho u h_0 \end{bmatrix} G = \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + P \\ \rho v h_0 \end{bmatrix} \tag{2}$$

and for a perfect gas

$$h_0 = \frac{e+p}{\rho} = \frac{\gamma}{\gamma - 1}\frac{P}{\rho} + \frac{1}{2}\left(u^2 + v^2\right) \tag{3}$$

where ρ, *u* and *v*, *e*, *p*, h_0 and γ are density, velocity component in the *x* and *y* direction, total energy per unit volume, static pressure, total enthalpy and specific heat ratio respectively.

The unsteady Euler equations are integrated into a steady state by an explicit, finite volume, Lax-Wendroff type time-marching that was developed by Ni [7]. Local time steps are used to accelerate convergence to a steady state solution. In addition, Ni's multiple-grid acceleration technique is also used to couple the solution on various embedded grids and to accelerate the overall convergence rate.

3. ADAPTIVE-GRID METHODS

Since most of the adaptive-grid algorithms in this paper are the same as those used in [6], only a summary will be given here. There are various techniques for redistribution and embedding, but there is a little difference in results.

a) Adaptive-grid redistribution method

The aim of adapting the grid is to decrease the solution error due to the finite interval grid (truncation error) and obtain uniform error throughout the flow field. Ideally, we should like to express error explicitly and adjust the grid intervals in order to minimize the maximum error. Unfortunately, it is rarely possible to obtain the error. Thus, to decrease it indirectly, we will minimize the maximum value of the product of grid intervals and a quantity. We will call this quantity the weight function *W*, and it is assumed that in a uniform grid, the truncation error will be high where *W* is high and vice-versa.

By reducing the grid interval in one place, the grid size will be increased somewhere else. So the minimization will be accomplished by setting this product to a constant value. This procedure is known as equidistribution.

If W_i is the weight function in the interval *(i, i+1)*, the equidistribution equation is:

$$W_i\left(S_{i+1} - S_i\right) = C \tag{4}$$

where *S* is the distance between nodes i and a special point along the grid line. Since *W* is a function of *S*, Eq. (4) is non-linear and must be solved iteratively. This one-dimensional equation can be applied in each direction.

One of the main disadvantages of a one-dimensional adaptation method is that the cells tend to become too skewed and therefore, the accuracy of the results is reduced. To avoid excessive skewness, grid-points distribution must somehow be affected by their near neighbors. To do this, the direct method has been employed. This method is based on a tension and torsion spring analogy suggested by Nakahashi and Diewert [8]. In this scheme, the grid-points are imagined to be suspended by tension and torsion springs and optimally redistributed by minimizing the energy of the springs system. Tension springs connect the neighboring nodes along grid lines. Satisfying the equidistribution Eq. (4) is analogous to obtaining equilibrium in a system of tension springs. In a stretched spring, tension is equal to a spring constant multiplied by a spring extension that is analogous to the weight function W multiplied by node spacing. In addition, torsion springs connect nodes to their corresponding nodes on the previously adapted grid lines which are now fixed. Further, torsion springs control inclinations of grid lines and prevent excessive grid skewness.

b) Adaptive grid-embedding method

As mentioned earlier, in grid-embedding methods, cells are locally divided in the regions of large error or gradient. To do this, the algorithm must sense large gradient or error regions and automatically divide cells in these regions. The process is repeated several times and therefore local embedded grids become finer and finer in order to resolve special regions adequately.

Although subdividing a quadrilateral by quadsection is natural and the simplest way, for some flowfield features with a strong one-dimensional nature such as shock wave or boundary layers, bisection (directional subdividing) is more efficient to save computational time and storage [5, 9, 10]. In this work, directional subdividing from the beginning of the adaptation procedure and throughout the flow field is used without any restriction.

One of the basic steps in the embedding adaptive technique is to set the adaptation parameter and threshold to detect the existence and track the evolution of special features of the flowfields such as shock waves. The first difference of density is the criterion used in this paper to calculate the adaptation parameter. The procedure is as follows:

1) The absolute values of the density differences of the two opposite walls will be used as the adaptation parameter

$$R_1 = |\rho_2 + \rho_3 - \rho_1 - \rho_4| \text{ and } R_2 = |\rho_3 + \rho_4 - \rho_2 - \rho_1| \tag{5}$$

where *1–4* indicate the nodes surrounding each cell.

2) The average, R_{ave}, and standard deviation, R_{sd}, are used to calculate the threshold, R_{th} [5].

$$R_{th} = R_{ave} + \alpha R_{sd} \tag{6}$$

3) The threshold is compared with the adaption parameter in each cell and if the adaption parameter is bigger then the cell will be divided.

The value of parameter α is chosen empirically. Too large or too small values of α may cause deficient or extra cells to be refined respectively.

Hierarchical quadtree grid generation offers an efficient method for the spatial discretization of arbitrary shaped two-dimensional domains. It consists of recursive algebraic splitting of sub-domains into quadrants, leading to an ordered hierarchical data structure with regard to the storage of mesh information. With some modifications to the quadtree structure, the approach proves highly flexible and has been adopted for the adaptive grid-embedding procedure.

A consequence of grid embedding is the internal boundaries between cells with different levels. An interface is distinguished by an abrupt change in the cell size. The grid lines may continue across the interface or be cut off by the interface. In the latter, cells contain extra nodes at the midside called hanging nodes. In this paper, the hanging nodes are removed by the transition of local connections to surrounding nodes such that triangular and quadrilateral cells are produced. This method is simple and conservative.

c) Combination of redistribution and embedding methods

Using an initial structured grid makes it possible to combine these two adaptation approaches. This means that before the grid becomes unstructured, we can employ a redistribution technique and then use an embedding method. The following steps describe the solution algorithm:

1) Initially, the solution is obtained on the relative coarse grid to allow rapid convergence and the flow details to appear.
2) Redistribution method is employed.
3) Solution is marched to steady state on the new grid.
4) Steps 2 and 3 are repeated several specified times.
5) Grid-embedding method is used.
6) Solution is marched to steady state on the new grid.

Steps 5 and 6 are repeated for the desired number of adaptation.

4. NUMERICAL RESULTS

To evaluate the accuracy and efficiency of the adaptive-grid methods, two examples are presented. The test cases consist of steady inviscid transonic flows around the NACA0012 airfoil. The coarse and globally fine grid is o-type with the far field boundary placed 12 chord lengths away from the airfoil.

The final convergence criterion is $\Delta(\rho v)_{max}<10^{-5}$. When the convergence criterion reaches to 5×10^{-5}, redistribution is performed and it is iterated three times. However, in the adaptation-embedded grid, when the convergence criterion reaches 10^{-5}, the adaptation is done and the embedding is iterated twice. In each of the following cases, six grid configurations are shown. These are:

1) Relatively coarse initial grid.
2) Grid in which only redistribution is employed.
3) Grid in which the combination of redistribution and embedding is used and the embedding process is employed only once.
4) Grid in which the combination of redistribution and embedding is used and the embedding process is employed twice.
5) Grid in which only embedding is employed twice.
6) Globally fine grid corresponding to the finest level of embedded grid.

The first case consists of a free-stream Mach number $M_\infty=0.8$ and 1.25^o angle of attack. The initial grid is 80×17 and Fig. 1 shows the generated grids and the Mach number contours. The flow solution on the initial grid shows much smeared shock waves. The initial grid was intentionally chosen to be rather coarse in the leading edge region. This leads to the production of a large amount of spurious entropy in the solution. After redistribution, the quality of the grid and the solution are improved. This flowfield is then subjected to adaptive grid-embedding twice. After every adaptation, the resulting flowfield shows an increased shock and expansion waves resolution, and a comparison of the Mach number contours indicates that the accuracy of a combination of these two methods in special regions is better than the globally fine grid and embedding method alone. Figure 2 compares the pressure coefficient distribution on the airfoil for the final embedded grid, a combination of adaptive-grids, and for the globally fine grid. Table 1 shows the number of nodes, the computational time, the obtained lift and drag coefficients, the range of values obtained in [11], and the value given in [12]. A comparison of the references values indicates that the combination of adaptive-grids shows better lift and drag coefficients.

Besides, the combination of adaptive-grids solution took 5.5 times less computational time and 2.75 times fewer nodes than globally fine grids containing 320×65 nodes. However, the computational time for the embedding method alone is naturally 84% of the combination of adaptive-grids.

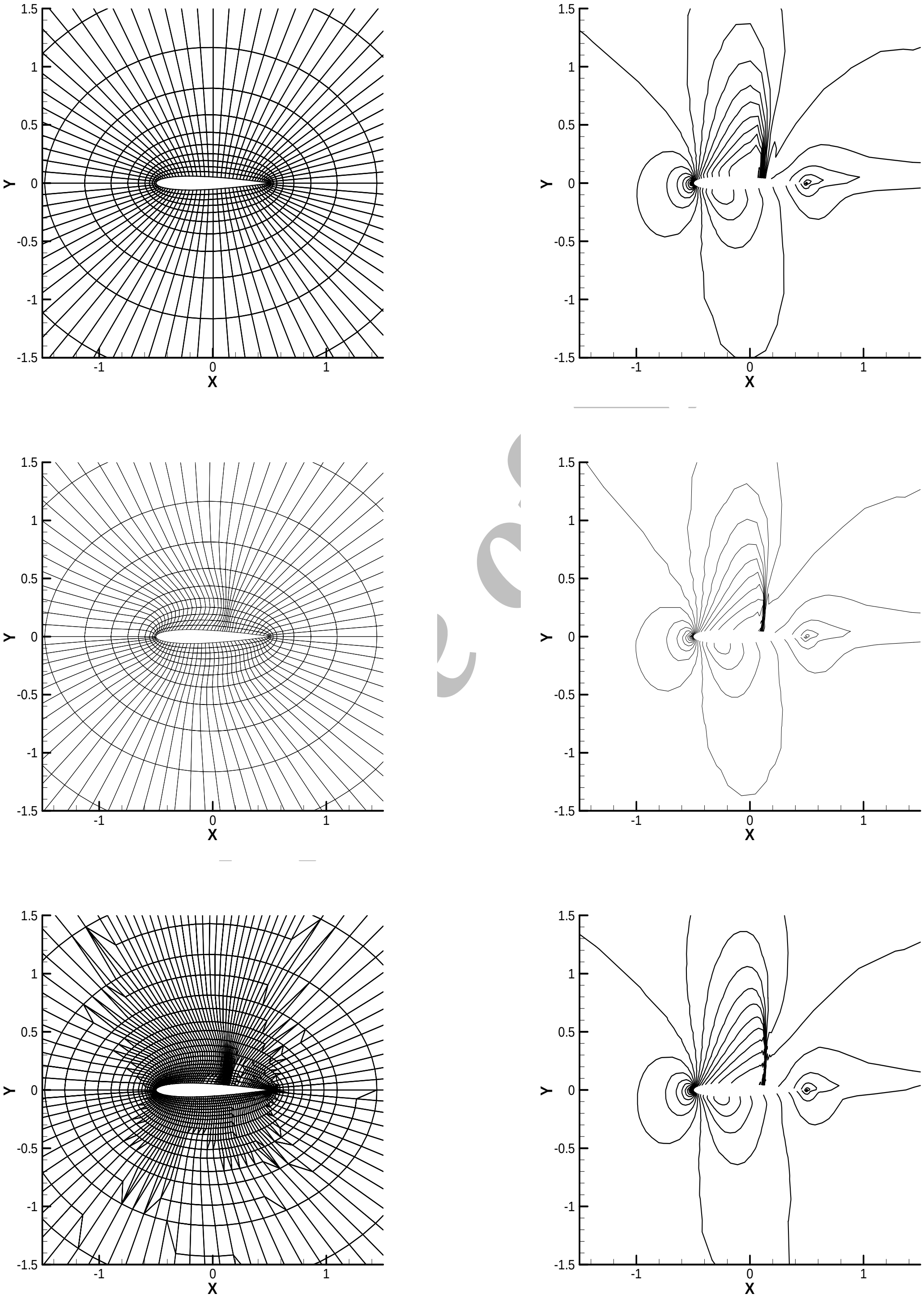

Figure 1 Continued.

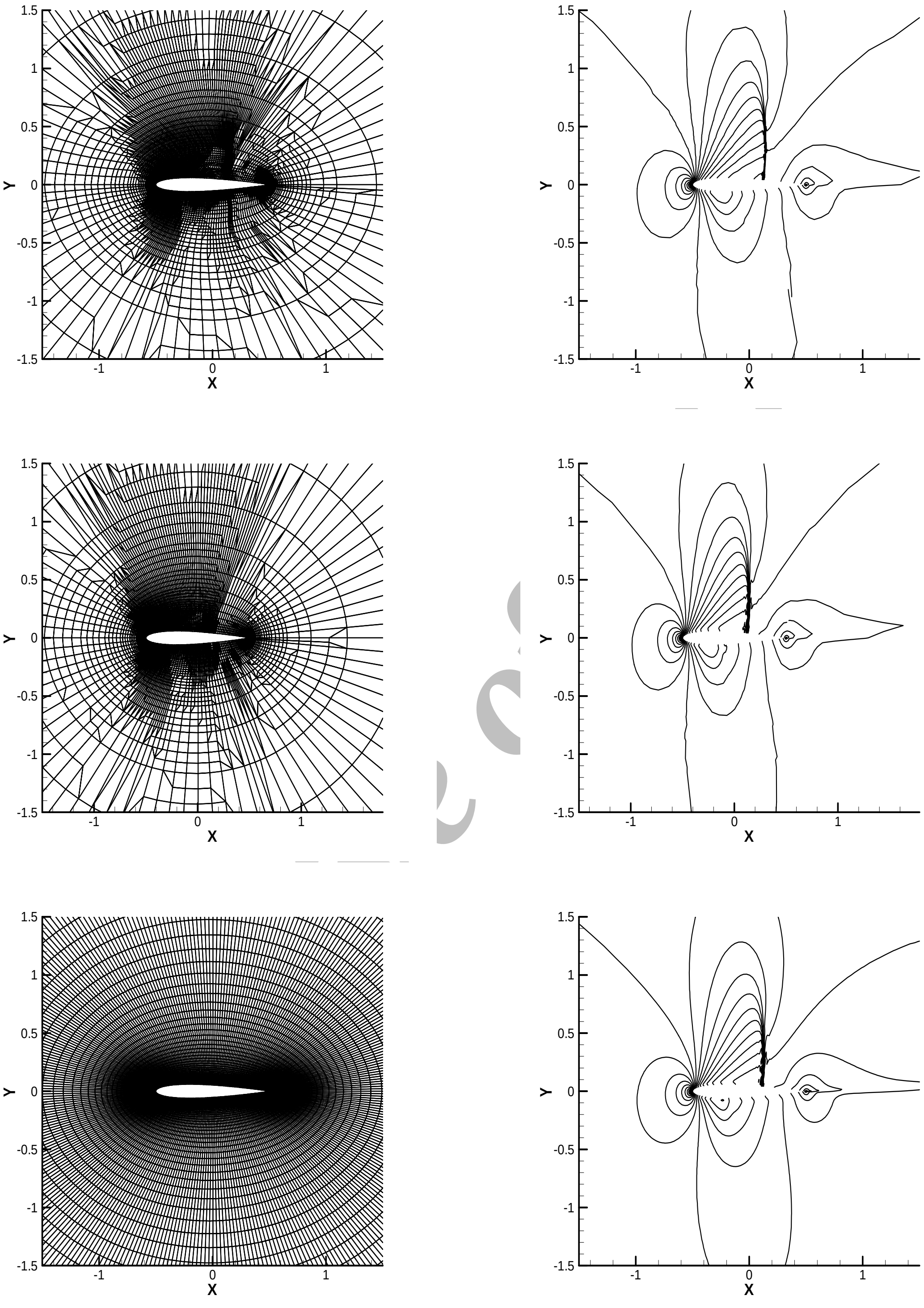

Fig. 1. Comparison of computational grids and the Mach number contours;
NACA0012, M_∞=0.8 and α=1.25^o

The minor purpose of this test case is the verification of accuracy of the numerical method. The obtained C_L, C_D and shock position (x/c=0.13 in [12]) for combination, embedding and fine grid confirm that the flowfield was solved with adequate accuracy.

The second case consists of a free-stream Mach number M_∞=0.85 and 1.0^o angle of attack. The initial grid and results are the same as the previous case. Figure 3 shows the generated grids and the Mach number contours and Fig. 4 compares the pressure coefficient distribution on the airfoil. Table 2 shows the number of nodes, the computational time, the obtained lift and drag coefficients, the range of values obtained in [13], and the value given in [12].

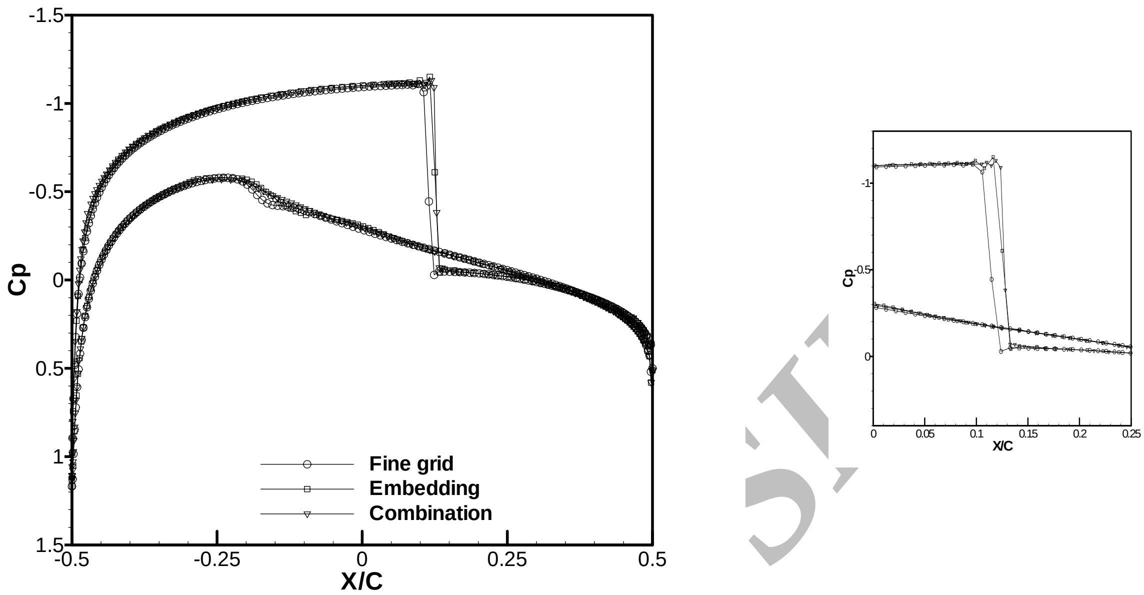

Fig. 2. Comparison of pressure coefficient distribution on the airfoil; NACA0012, M_∞=0.8 , α=1.25°

Table 1. Number of nodes, Computational time, C_L and C_D coefficients;
NACA0012, M_∞=0.8 and α=1.25°

Adaptation technique	Nodes	Time	C_L	C_D
Embedding	7289	15.1%	0.3589	0.0235
Combination with only once embedding process	3062	9%	0.3800	0.0219
Combination with twice embedding process	7562	18%	0.3623	0.0233
Fine grid	20800	100%	0.3514	0.0215
Ref. 11			0.3463-0.3736	0.0221-0.0237
Ref. 12			0.3654	0.0232

Again, the results show that the accuracy due to the combination of adaptive-grid redistribution and embedding is much more than the globally fine grid and embedding method alone. As compared with the globally fine grid solution, the combination of adaptive-grids reached 2.5 times less computational time and nodes, but the computational time for the embedding method alone is 80% of the combination of adaptive-grids solution.

Table 2. Number of nodes, Computational time, C_L and C_D coefficients;
NACA0012, M_∞=0.85 and α=1.0°

Adaptation technique	Nodes	Time	C_L	C_D
Embedding	7655	32%	0.375	0.0593
Combination with only once embedding process	3071	17%	0.408	0.0560
Combination with twice embedding process	7579	40%	0.383	0.0586
Fine grid	20800	100%	0.3713	0.060
Ref. 13			0.36-0.39	0.056-0.059
Ref. 12			0.3861	0.0582

To evaluate the relation between accuracy and computational work (normalized CPU time) for different adaptation methods, the second case was solved for globally fine grid, redistribution, embedding and the combination of adaptive-grids on a sequence of successively finer grids. For globally fine grid and redistribution, the solutions were obtained on 80×17, 160×33, and 320×65 grids, whereas this case was solved for embedding and the combination of adaptive-grid on 80×17 and 160×33 grids. The embedding process for an 80×17 grid was employed twice, but for 160×33, the grid was employed only once. Table 3 and Fig. 5 show the computational work required for each solution as a function of the lift coefficient error. The solutions were compared with the result of ref [12].

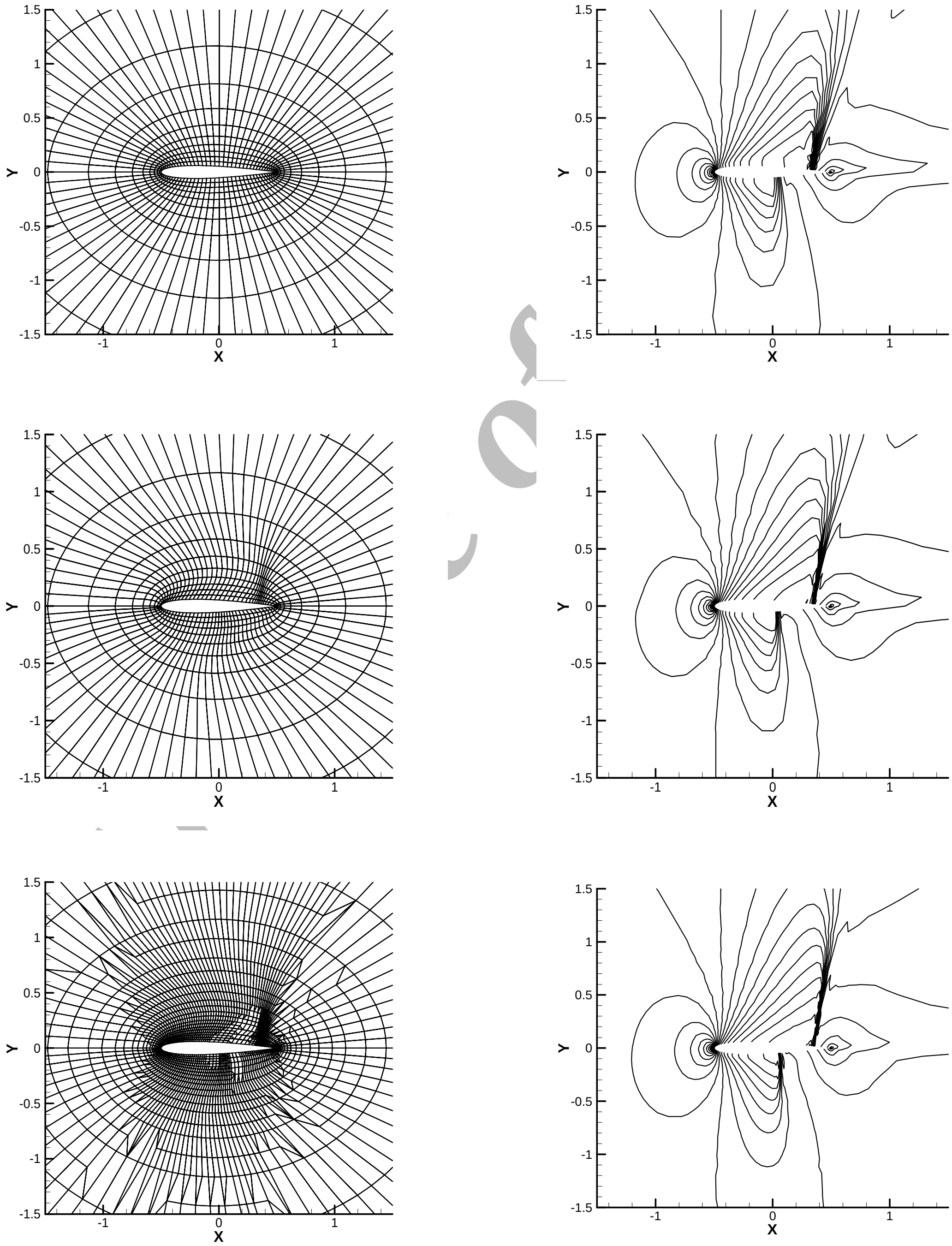

Figure 3 Continued.

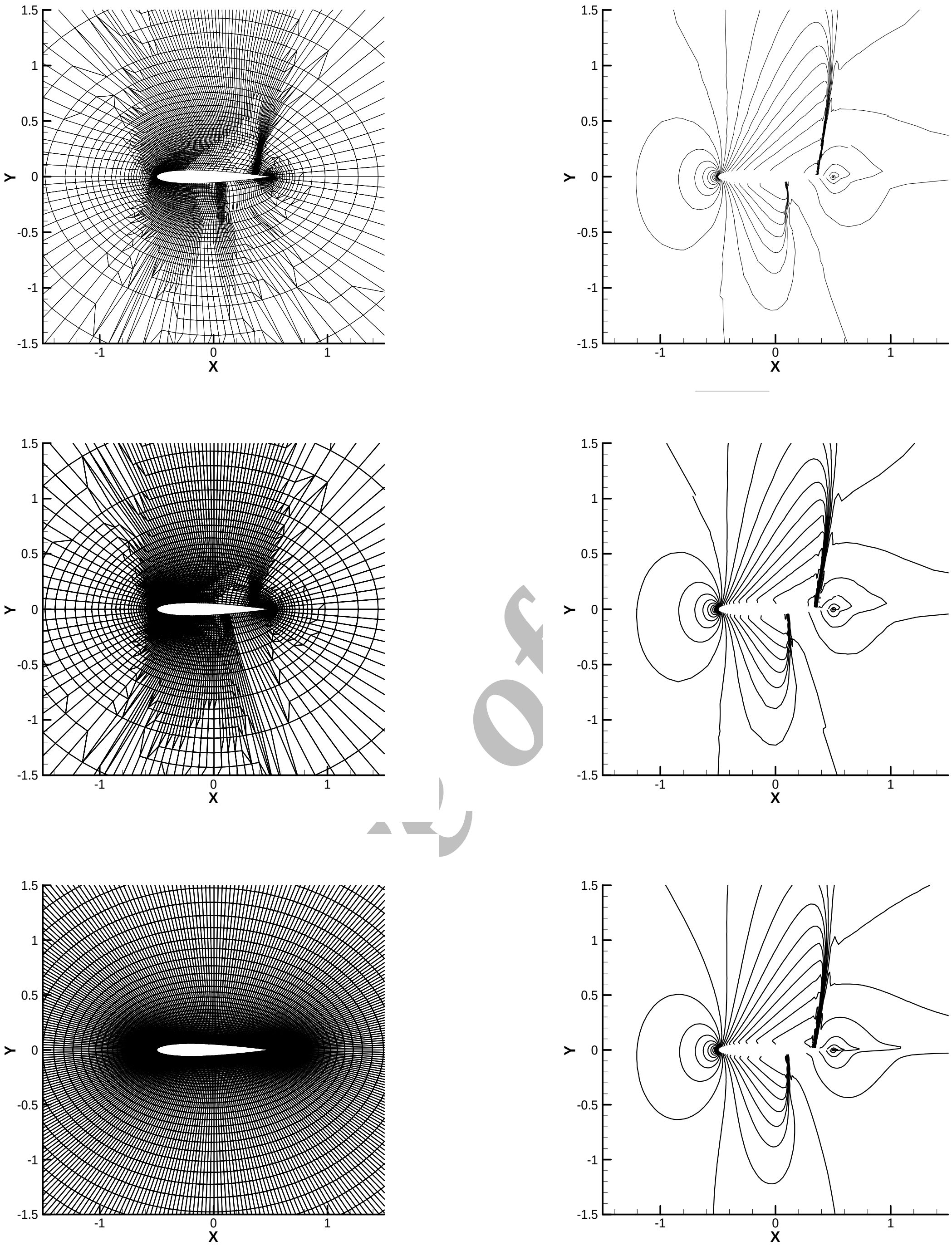

Fig. 3. Comparison of computational grids and the Mach number contours;
NACA0012, M_{∞}=0.85 and α=1.0^{o}

The results in the table and figure show that at high accuracy, the embedding method requires much less computing time in comparison with the redistribution method, and as accuracy increases, this difference becomes greater. This result is similar to Dannenhoffer's results [1] as was expected. However, the table and figure also indicate that at high accuracy, the combination of adaptive-grid redistribution and embedding is even better than the embedding method alone. In fact, the combination of adaptive-grids significantly reduces the computational time and the number of nodes for the same accuracy with adaptive-grid embedding.

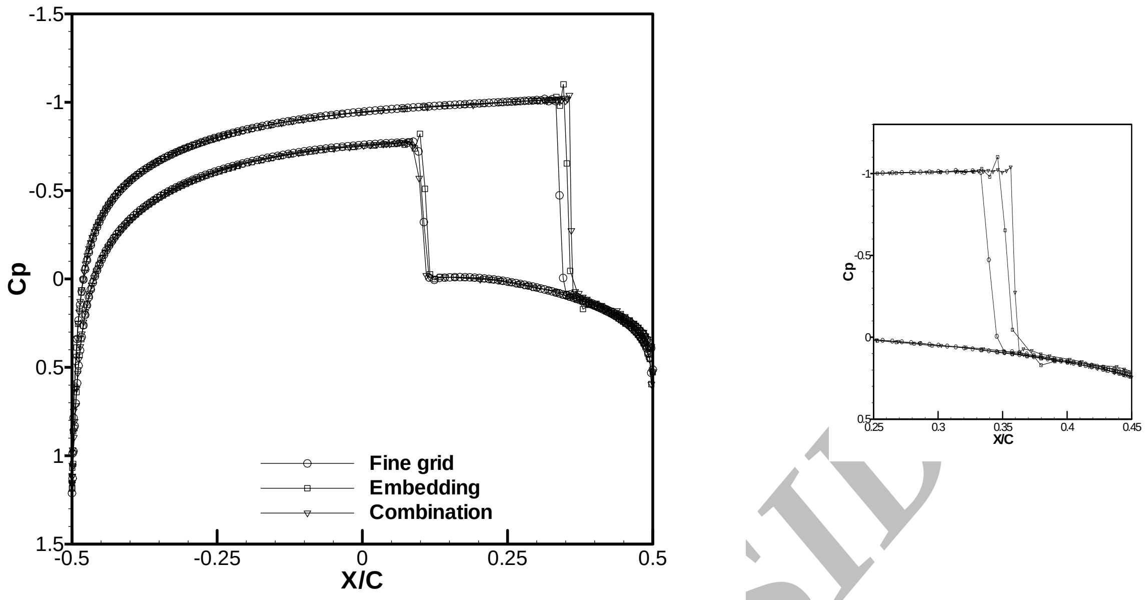

Fig. 4. Comparison of pressure coefficient distribution on the airfoil; NACA0012, M_∞=0.85, α=1.0

Table 3. Comparison of adaptation methods; NACA0012 airfoil, M_∞=0.85 and α=1.0°

Adaptation technique	Nodes	Work (normalized CPU time)	C_L error
None	1360	1.0	0.0773
	5280	5.12	0.0335
	20800	23.24	0.0148
Redistribution	1360	1.25	0.0684
	5280	6.23	0.0308
	20800	27.84	0.0120
Embedding	3043	2.87	0.0301
	7655	7.38	0.0111
	10796	10.4	0.0070
Combination	3071	3.94	0.0219
	7579	9.3	0.0031
	10771	13.1	0.0023

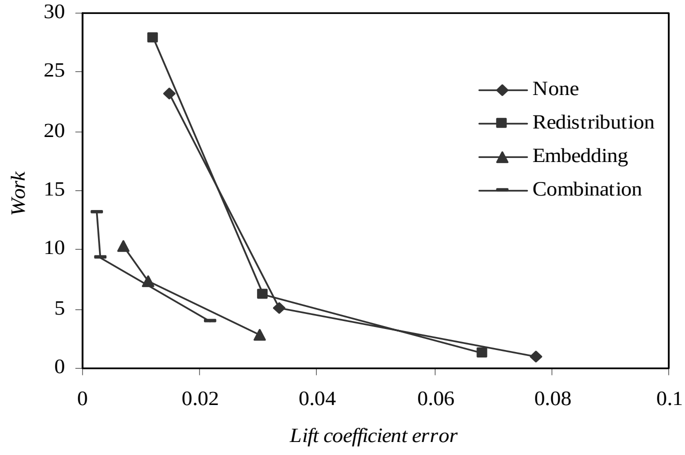

Fig. 5. Comparison of adaptation methods; NACA0012 airfoil, M_∞=0.85 and α=1.0

5. CONCLUSIONS

It can be concluded that for two-dimensional invisid flows when high accuracy is required, the combination is more efficient than grid redistribution and grid embedding alone. Although, the redistribution method is easier to implement, at high accuracy, the preference of embedding has been cleared. The combination method is even better than embedding for the same accuracy. It reduces the number of nodes and computational time further. Since the redistribution is relatively easy, we can easily add it to the invisid codes which use the adaptive-grid embedding. However, more work is needed for comprehensive comparison. This should include unsteady, viscous and three-dimensional flows.

REFERENCES

1. Dannenhofer, J. F. (1991). A comparison of adaptive-grid redistribution and embedding for steady transonic flows. *International Journal for Numerical Methods in Engineering*, *32*, 653-663.
2. Babuska, I. & Guo, B. (1988). The h-p version of the finite element method for domains with curved boundaries. *SIAM Journal on Numerical Analysis*, *25*, 837-861.
3. Zienkiewicz, O. C. & Zhu, J. Z. (1991). Adaptivity and mesh generation. *International Journal for Numerical Methods in Engineering*, *32*, 783-810.
4. Oden, J. T., Rachowicz, W. & Kennon, S. R. (1991). Numerical analysis of three-dimensional compressible Navier-stokes equations using an adaptive *h-p* finite element method. *AIAA Paper* 91-0119.
5. Kallinderis, Y. & Baron, J. R. (1989). Adaptation methods for a new Navier-stokes algorithm. *AIAA Journal*, *27*, 37-43.
6. Ameri, M. & Shirani, E. (2004). Combination of adaptive-grid redistribution and embedding methods. *Communications in Numerical Methods in Engineering*, *20*, 229-240.
7. Ni, R. H. (1982). A multiple grid scheme for solving the Euler equations. *AIAA Journal*, *20*, 1565-1571.
8. Nakahashi, K. & Diewert, G. S. (1986). Three-dimensional adaptive grid method. *AIAA Journal*, *24*, 948-954.
9. Davis, L. D. & Dannenhofer, J. F. (1994). Three-dimensional adaptive grid-embedding Euler technique. *AIAA Journal*, *32*, 1167-1174.
10. Ameri, M. & Shirani, E. (2003). Effects of directional subdividing on adaptive grid-embedding. *International Journal of Engineering Islamic Republic of Iran*, *16*, 71-78.
11. Viviand, H. (1985). Test cases for invisid flow field methods. *AGARD Fluid dynamics panel*, AGARD Advisory Report AR-211.
12. Jameson, A. & Martinelli, L. (1998). Mesh refinement and modeling errors in flow simulation. *AIAA Journal*, *36*, 676-686.
13. Derviex, A., Van Leer, B., Periaux, J. & Rizzi, A. (1989). Numerical simulation of compressible Euler flows. *26*, *Notes on Numerical Fluid Mechanics*, Vieweg.