

# An Investigation of Directional-Coarsening And Line-Implicit Smoothing Applied to Agglomeration Multigrid

J. V. Lassaline \*

*Ryerson University*

*350 Victoria St, Toronto, ON, M5B 2K3, Canada*

D. W. Zingg †

*University of Toronto Institute for Aerospace Studies*

*4925 Dufferin St, Toronto, ON, M3H 5T6, Canada*

The convergence rates of explicit agglomeration multigrid algorithms for turbulent flow problems are significantly worse than for inviscid flow problems. Two techniques for reducing numerical stiffness are investigated and compared. Directional coarsening, controlled by the parameter  $\beta$ , alters the agglomeration algorithm to generate coarse grids that reduce localized cell stretching. Line-implicit smoothing, controlled by the parameter  $\alpha$ , adds implicit terms to the preconditioner along one-dimensional lines constructed on the mesh. Optimal values of  $\beta$  and  $\alpha$  are presented for structured and hybrid unstructured meshes. A factor of two improvement in asymptotic convergence rates has been demonstrated. Substantial improvements in CPU time can be obtained for a five-order residual reduction.

## Introduction

COMPUTATIONAL fluid dynamics has become a useful design and evaluation tool in the aerospace industry. For CFD tools to remain valuable it is necessary to provide algorithms that yield solutions of sufficient accuracy in an efficient manner. We define an efficient algorithm as one that strikes a balance between CPU and memory requirements as well as an empirical measure of ease of use. It is the goal of this research to provide an efficient multigrid turbulent flow solver that provides accurate solutions through a minimal set of parameters.

While explicit multigrid methods are attractive in terms of memory costs, convergence rates of turbulent flow problems lag significantly behind those of inviscid flow problems.<sup>1,2,3,4,5</sup> Near theoretical convergence rates for inviscid flow problems using explicit multigrid methods have been demonstrated in the past.<sup>6,7,8</sup> To efficiently resolve high Reynolds number turbulent flow problems, high aspect ratio grids are required. Unfortunately these grids introduce numerical stiffness that severely reduces the convergence rate of turbulent flow problems. Although a fully implicit solver would better handle the presence of highly stretched cells, the memory usage of a fully implicit solver grows rapidly with an increasing number of nodes, particularly in three-dimensions. We intend to strike a balance be-

tween these two approaches, beginning with an explicit multigrid algorithm and adding features to reduce sensitivity to numerical stiffness.

Generating suitable grids for a multigrid sequence can be a significant task. This research employs a heuristic directional agglomeration algorithm to provide nested coarse grids suitable for use with a finite-volume multigrid algorithm.<sup>9,10,11</sup> The algorithm works with any combination of structured, unstructured, or hybrid grids. In addition to automatic grid generation, the agglomeration algorithm attempts to alleviate numerical stiffness through directional coarsening.<sup>12,3,7,4,13</sup> Closely coupled to the directional coarsening algorithm, a line-implicit smoothing algorithm is employed to further reduce stiffness at a slight increase in memory footprint.<sup>3,7,4,13</sup>

This paper covers a detailed quantitative investigation of the directional-coarsening and line-implicit smoothing algorithms as applied to laminar and turbulent flows. Substantial improvements in convergence rates measured in terms of CPU time are demonstrated while minimizing memory requirements.

## Numerical Algorithm

### Spatial Discretization

Our algorithm employs a cell-vertex finite-volume spatial discretization of the Favre-averaged Navier-Stokes equations. Mean turbulence is modelled with the Spalart-Allmaras turbulence model.<sup>14</sup> The spatial discretization is applied to the centroidal-median dual constructed from a given source grid. The semi-discrete form of the equations can be written as follows

\*Assistant Professor, Member AIAA jvl@ryerson.ca

†Professor, Senior Member AIAA <http://goldfinger.utias.utoronto.ca/~dwz>

Copyright © 2003 by J. V. Lassaline and D. W. Zingg. Published by the American Institute of Aeronautics and Astronautics, Inc. with permission.

$$\frac{d\mathbf{Q}}{dt} + \mathcal{R}(\mathbf{Q}) = 0 \quad (1)$$

where the residual operator  $\mathcal{R}(\mathbf{Q})$  is formed primarily as face-based operations on the piece-wise constant solution vector  $\mathbf{Q}$ . Inviscid and viscous contributions to the residual operator are formed by the surface integral of the flux contributions over the boundary of each cell. Full viscous flux terms are formed using the next-to-nearest cell neighbour. A blend of first- and third-order artificial dissipation is applied through a combination of Laplacian and biharmonic operators.<sup>15</sup>

The  $i$ th block of the residual vector  $\mathbf{R} = \mathcal{R}(\mathbf{Q})$  for a given cell  $i$  with area/volume  $\Omega_i$  is

$$\mathbf{R}_i = \frac{1}{\Omega_i} \sum_{ik}^{\text{edges}} (\mathbf{F}_{ik} + \mathbf{G}_{ik} + \mathbf{D}_{ik}) - \mathbf{S}_i \quad (2)$$

where the inviscid flux  $\mathbf{F}_{ik}$ , viscous flux  $\mathbf{G}_{ik}$ , and artificial dissipation  $\mathbf{D}_{ik}$  for each face  $ik$  are formed as

$$\mathbf{F}_{ik} = \frac{1}{2} (F(\mathbf{Q}_i, \mathbf{N}_{ik}) + F(\mathbf{Q}_k, \mathbf{N}_{ik})) \quad (3)$$

$$\mathbf{G}_{ik} = G(\mathbf{Q}_i, \mathbf{Q}_k, \nabla \mathbf{Q}_i, \nabla \mathbf{Q}_k, \mathbf{N}_{ik}) \quad (4)$$

$$\mathbf{D}_{ik} = -\frac{1}{2} |A(\mathbf{Q}_i, \mathbf{Q}_k, \mathbf{N}_{ik})| (e_{ik}^{(2)}(\mathbf{Q}_k - \mathbf{Q}_i) - e_{ik}^{(4)}(\mathbf{L}_k - \mathbf{L}_i)) \quad (5)$$

$$\mathbf{L}_i = \sum_{ik}^{\text{edges}} (\mathbf{Q}_k - \mathbf{Q}_i). \quad (6)$$

For a given face between cell  $i$  and  $k$ , the unit normal is scaled by the length/area of the face to form  $\mathbf{N}_{ik}$ . Source terms arising from the turbulence model are added as cell-based vector  $\mathbf{S}_i$ .

Boundary conditions at solid walls are adiabatic and no-slip for viscous flows. Boundary conditions based upon Riemann invariants are applied at far-field boundaries.

### Relaxation Scheme

As we are interested in steady-state solutions, our goal is to solve

$$\mathcal{R}(\mathbf{Q}) = 0 \quad (7)$$

To further improve convergence we employ preconditioning of the form

$$\mathbf{P}^{-1} \mathcal{R}(\mathbf{Q}) = 0 \quad (8)$$

Applying an explicit multi-stage relaxation method, we time-march our flow problem from to  $t_n$  to  $t_{n+1}$  using an  $m$ -stage scheme

$$\mathbf{Q}^{(0)} = \mathbf{Q}_n \quad (9)$$

...

$$\mathbf{Q}^{(j)} = \mathbf{Q}^{(0)} - \alpha_j \mathbf{P}^{-1} \mathcal{R}(\mathbf{Q}^{(0)}, \dots, \mathbf{Q}^{(j-1)}) \quad (10)$$

...

$$\mathbf{Q}_{n+1} = \mathbf{Q}^{(m)} \quad (11)$$

where the  $i$ th block of the residual vector  $\mathbf{R}^{(j)} = \mathcal{R}(\mathbf{Q}^{(0)}, \dots, \mathbf{Q}^{(j-1)})$  is defined as

$$\mathbf{R}_i^{(j)} = \frac{1}{\Omega_i} \sum_{ik}^{\text{edges}} \left( \mathbf{F}_{ik}^{(j)} + \mathbf{G}_{ik}^{(0)} + \sum_l^{j-1} \Gamma_{j,l} \mathbf{D}_{ik}^{(l)} \right) - \mathbf{S}_i^{(0)} \quad (12)$$

$$\sum_{l=0}^{j-1} \Gamma_{j,l} \equiv 1 \quad (13)$$

$$\Gamma_{j,l} = \begin{cases} \gamma_j & \text{if } l = j-1 \\ (1 - \gamma_j) \Gamma_{j-1,l} & \text{if } l \neq j-1 \end{cases} \quad (14)$$

Block-Jacobi preconditioning is applied by forming the preconditioner  $\mathbf{P}$  from the diagonal blocks of the Jacobian  $\frac{\partial \mathbf{R}}{\partial \mathbf{Q}}$

$$\mathbf{P} = \left[ \frac{\partial \mathbf{R}_i}{\partial \mathbf{Q}_i} \right] \quad (15)$$

The effect of cell stretching upon convergence rates can be demonstrated by examining the application of the relaxation scheme to a Fourier analysis of the residual operator. Fig. 1 illustrates the amplification factor of the preconditioned residual operator as a function of Fourier angle  $(\theta_x, \theta_y)$  for a uniform grid spacing versus cell stretching in the flow direction. This five-stage relaxation method efficiently reduces the high-frequency Fourier error modes that appear in the upper-left, upper-right and lower-right quadrant of the uniformly spaced mesh. However as the cell spacing in the  $y$  direction is reduced the scheme becomes much less effective in the lower-right quadrant.

Line-implicit smoothing is applied by constructing lines normal to cell stretching in the grid. By reordering the cells as they occur in the implicit lines and adding implicit terms between each consecutive cell on a line, the following tridiagonal preconditioner is obtained

$$\mathbf{P} = \left[ \frac{\partial \mathbf{R}_i}{\partial \mathbf{Q}_{i-1}}, \frac{\partial \mathbf{R}_i}{\partial \mathbf{Q}_i}, \frac{\partial \mathbf{R}_i}{\partial \mathbf{Q}_{i+1}} \right] \quad (16)$$

The increase in the bandwidth of the preconditioner is limited to regions of high stiffness.

### Multigrid

Multigrid methods seek to further eliminate low-frequency errors by recasting the system of equations onto a sequence of coarser grids. Thus our coarse grids employ the same relaxation scheme as above but solve a forced version of the fine grid equations. Our coarse grid relaxation scheme in Eq. 10 at stage  $i$  becomes

$$\mathbf{Q}^{(i)} = \mathbf{Q}^{(0)} - \alpha_i \mathbf{P}^{-1} (\mathcal{R}(\mathbf{Q}^{(0)}, \dots, \mathbf{Q}^{(i-1)}) + I_f^c \mathcal{R}(Q_f) - \mathcal{R}(\mathbf{Q}^{(0)})) \quad (17)$$

where  $I_f^c \mathcal{R}(Q_f)$  is the restricted residual of the finer grid in the grid sequence. Corrections are interpolated

back to the finest grid through prolongation, using either simple injection or linear interpolation.

On a fine grid, the relaxation scheme is optimized to quickly relax errors in the high frequency domain. In reference to the amplification factor plot in Fig. 1, relaxation on a coarse grid can be viewed as mapping the lower-left quadrant ( $\theta_x, \theta_y \in [0, \pi/2]$ ) to the frequency domain  $\theta_x, \theta_y \in [0, \pi]$ . Directional coarsening in the  $y$  direction would map the lower quadrants to  $\theta_x, \theta_y \in [0, \pi]$ . Thus it is also possible to tailor our relaxation scheme through selective agglomeration.

### Agglomeration

To generate a coarse grid sequence we use agglomeration to selectively fuse fine grid cells to form a new coarse grid. A sequence of nested coarse grids can be produced by successively agglomerating the previous grid in the sequence. A sample grid sequence, produced by agglomerating in the default manner, appears in Fig. 2.

The agglomeration algorithm begins with the boundary cells and advances in a front, fusing all available fine grid cells. A pseudo-code algorithm can be found in Algorithm 1. If all neighbours of a given seed cell are fused, an isotropic distribution of coarse cells is produced with a coarsening ratio of approximately 4 : 1 in two-dimensions and 8 : 1 in three-dimensions. To force coarsening in a direction normal to local cell stretching a weight  $\omega_i$  is assigned to each face and coarsening is only applied across faces for which  $\omega_i > \beta\omega_{max}$ , where  $\omega_{max}$  is the maximum face weight about the seed cell and  $\beta$  is typically between 0 and 1.<sup>13</sup>

Mavriplis<sup>13</sup> has recommended a face weight proportional to the magnitude of the face normal  $\mathbf{N}_{ik}$ . This produces grids that are directionally coarsened in areas of cell stretching and isotropically coarsened in regions of uniform cell shape. As a further enhancement, a face weight proportional to the largest eigenvalue of the Jacobian of the residual operator would produce a coarse grid that should alleviate stiffness more directly. As this would require a reasonable snapshot of the solution, this weighting scheme is employed by regenerating the coarse grid sequence between the outer cycles of a full-multigrid sequence.

### Implicit Line Construction

To construct one-dimensional lines of cells for use with line-implicit smoothing, the same face-based weights as used in directional coarsening are examined. A pseudo-code algorithm appears in Algorithm 2. In regions of local cell stretching implicit lines are constructed normal to cell stretching, while in regions of uniform cell shape, lines containing only one cell are produced, and the preconditioner in Eq. 16 reduces to Eq. 15.

---

### Algorithm 1 Agglomeration method

---

```

Form a priority queue of cells (weighting boundary
cells higher than non-boundary cells.)
while priority queue is not empty do
  Pop a seed cell from the priority queue.
  Form new coarse cell from seed cell.
  for all neighbouring cells of seed do
    if neighbour matches criteria and not already
    fused then
      Add the neighbour cell to coarse cell
    end if
  end for
end while

```

---



---

### Algorithm 2 Implicit line construction

---

```

Form a priority queue of cells sorted by  $\omega_{max}/\omega_{min}$ .
while priority queue is not empty do
  Pop a seed cell from the priority queue.
  while  $\omega_{max}/\omega_{min}$  of seed cell is greater than  $\alpha$ 
  do
    Locate neighbour with largest  $\omega$  in priority
    queue
    Remove neighbour from priority queue and
    make neighbour new seed
  end while
end while

```

---

## Results

The following results were obtained with our research code HURRICANE. Figs. 4, 5 and 6 illustrate convergence histories for various strategies applied to laminar flow over a finite flat plate at  $M = 0.2$  and  $Re = 1 \times 10^5$ . Five-level W-cycle full multigrid was used for all cases with identical solver parameters. A structured grid (Fig. 3) with an off-wall spacing of  $5 \times 10^{-6}$  and a maximum cell aspect-ratio of 26500 was selected to illustrate numerical stiffness. Isotropic agglomeration, in which all available neighbours are fused with a seed cell, is equivalent to a directional coarsening coefficient  $\beta = 0.0$ . Fig. 4 demonstrates a factor of 2 improvement in CPU time requirements using  $\beta = 0.25$ . Holding  $\beta = 0.0$  while varying the line-implicit smoothing coefficient  $\alpha$  demonstrates a fairly uniform improvement in Fig. 5 with optimal  $\alpha$  in the 3 to 4 range.

It should be noted that the directional coarsening algorithm produces a coarsening ratio approaching 2 : 1 in anisotropic regions of the grid rather than the ideal ratio of 4 : 1 produced by removing every other line in both the streamwise and normal directions. Fig. 6 illustrates the convergence history obtained by permutations of coarsening in the streamwise (“x-dir”) and normal (“y-dir”) direction. To obtain a coarsening ratio of 4 : 1 the streamwise and normal direction algorithms were repeated twice. The history labelled “xy-dir” is equivalent to removing every other line in both directions. It should be expected that the “xy-

dir” algorithm is superior to the isotropic algorithm as the latter does not preserve the structured nature of the control-volumes on the coarse grids. It is interesting to note that the directional-coarsening algorithm utilizing  $\beta = 0.25$  converges in less time than the conventional structured grid coarsening algorithms.

The performance of the algorithm for turbulent flow was tested by solving for the flow about an RAE 2822 airfoil at  $M = 0.754$ ,  $\alpha = 2.57^\circ$  and  $Re = 6.2 \times 10^6$ . A structured  $209 \times 29$  C-mesh with off-wall spacing of  $1 \times 10^{-5}$  and maximum cell aspect-ratio of 3300 was used. Fig. 7 illustrates the convergence history for isotropic agglomeration versus various directional-coarsening  $\beta$  coefficients. Five-level W-cycle full multigrid was used for all cases with identical solver parameters. A significant improvement is demonstrated for  $\beta = 0.25$ , particularly in reducing the cost of the full multigrid cycle, with a factor of approximately 3 improvement at 5 orders of residual norm reduction. Fig. 8 illustrates the convergence history for various line-implicit smoothing coefficients. The line-implicit algorithm demonstrates a relative improvement in convergence rates. As the line-construction coefficient  $\alpha$  is reduced, the number of implicit terms in the preconditioner increases. Fig. 10 illustrates the construction of implicit lines with  $\alpha \in [1, 4]$  for this structured mesh. For a moderate  $\alpha$  in the range of 3 to 4, the increase in memory costs in solving the tridiagonal preconditioner is generally on the order of 10 – 15%. Fig. 9 demonstrates a comparison of the two methods applied separately and in conjunction. Further improvement upon the  $\beta = 0.25$  case is noted with the addition of the line-implicit smoother. Table 1 lists the relative time required to obtain aerodynamic coefficients  $C_l$  and  $C_d$  within 0.1% of fully converged values, normalized relative to the isotropic coarsening strategy. The relative performance improvement offered by the combined scheme yields converged coefficients in as little as 30% of the CPU time required for the default scheme.

The same turbulent flow problem was repeated for a hybrid unstructured mesh produced by wrapping a triangulated mesh about the inner layer of the previous structured C-mesh. With respect to grid anisotropy, cell stretching in the hybrid mesh was limited to the region nearest the body and along the wake. The original C-mesh contained additional regions of cell stretching normal to the direction of the flow emanating from the airfoil trailing edge toward the outer boundary. Fig. 11 illustrates the convergence history for a variety of directional-coarsening  $\beta$  coefficients relative to the isotropic coarsening strategy. Five-level W-cycle full multigrid was used for all cases with identical solver parameters. The most significant improvement was demonstrated for  $\beta = 0.5$ , again reducing the cost of the full multigrid cycle. The convergence history for various line-implicit smoothing coefficients is shown in

Fig. 12, with all parameters demonstrating similar performance. Fig. 14 illustrates the construction of implicit lines with  $\alpha \in [1, 4]$  for this hybrid mesh. In comparison to the structured C-mesh, there are significantly less line segments produced for similar values of  $\alpha$ . The addition of line-implicit smoothing to the  $\beta = 0.5$  directional-coarsening strategy demonstrates an increase in the cost relative to the  $\beta$ -only strategy as illustrated in Fig. 13. Although the line-implicit strategy reduces the number of multigrid cycles, it adds negligible improvement to the best directional coarsening strategy in terms of CPU time.

## Conclusions

We have demonstrated an algorithm for improving convergence rates and reducing CPU time requirements for explicit multigrid algorithms without significantly increasing memory requirements. Utilizing agglomeration with directional-coarsening and line-implicit smoothing, we have demonstrated an automated coarse grid generation algorithm that produces superior grid sequences for use with explicit multigrid. Optimal values of the directional-coarsening parameter  $\beta$  and the line-implicit smoothing parameter  $\alpha$  appear to be consistent for different flow problems. A factor of two improvement in asymptotic convergence rates has been demonstrated. Substantial improvements in CPU time can be obtained for a five-order residual reduction.

## References

- <sup>1</sup>Pierce, N. A., Giles, M. B., Jameson, A., and Martinelli, L., “Accelerating Three-Dimensional Navier-Stokes Calculations,” AIAA Paper 97-1953, AIAA, 1997.
- <sup>2</sup>Allmaras, S. R., “Analysis of a Local Matrix Preconditioner for the 2-D Navier-Stokes Equations,” AIAA Paper 93-3330-CP, AIAA, 1993.
- <sup>3</sup>Mavriplis, D. J., “Directional Coarsening and Smoothing for Anisotropic Navier-Stokes Problems,” *Elect. Trans. Numer. Anal.*, Vol. 6, 1997, pp. 182–197.
- <sup>4</sup>Mavriplis, D. J., “On Convergence Acceleration Techniques for Unstructured Meshes,” AIAA Paper 98-2966, AIAA, 1998.
- <sup>5</sup>Mavriplis, D. J. and Venkatakrishnan, V., “A Unified Multigrid Solver for the Navier-Stokes Equations on Mixed Element Meshes,” AIAA Paper 95-1666, AIAA, 1995.
- <sup>6</sup>Wesseling, P., *An Introduction to Multigrid Methods*, John Wiley and Sons, 1991.
- <sup>7</sup>Mavriplis, D. J., “Multigrid Strategies for Viscous Flow Solvers on Anisotropic Unstructured Meshes,” *J. Comp. Phys.*, Vol. 145, No. 1, 1998, pp. 141–165.
- <sup>8</sup>Lassaline, J. V., *A Navier-Stokes Equation Solver Using Agglomerated Multigrid Featuring Directional Coarsening and Line-Implicit Smoothing*, Ph.D. thesis, University of Toronto, 2003.
- <sup>9</sup>Lallemand, M.-H. and Dervieux, A., “A Multigrid Finite Element Method for Solving the Two-Dimensional Euler Equations,” *Proceedings of the Third Copper Mountain Conference*, April 1987, pp. 337–363.
- <sup>10</sup>Lallemand, M.-H., Steve, H., and Dervieux, A., “Unstructured Multigriding by Volume Agglomeration - Current Status,” *Comput. Fluids*, Vol. 21, No. 3, 1992, pp. 397–433.
- <sup>11</sup>Smith, W. A., “Multigrid Solution of Transonic Flow on Unstructured Grids,” *Recent Advances and Applications in*

| Coeff. | Strategy                  | Relative CPU Time |
|--------|---------------------------|-------------------|
| $C_l$  | Isotropic                 | 100%              |
|        | $\beta = 0.25$            | 39%               |
|        | $\alpha = 1$              | 57%               |
|        | $\alpha = 1 \beta = 0.25$ | 34%               |
| $C_d$  | Isotropic                 | 100%              |
|        | $\beta = 0.25$            | 37%               |
|        | $\alpha = 1$              | 55%               |
|        | $\alpha = 1 \beta = 0.25$ | 30%               |

**Table 1** Relative CPU time required to obtain aerodynamic coefficients within 0.1% of converged values for turbulent flow about an RAE 2822 airfoil at  $M = 0.754$ ,  $\alpha = 2.57^\circ$  and  $Re = 6.2 \times 10^6$ .

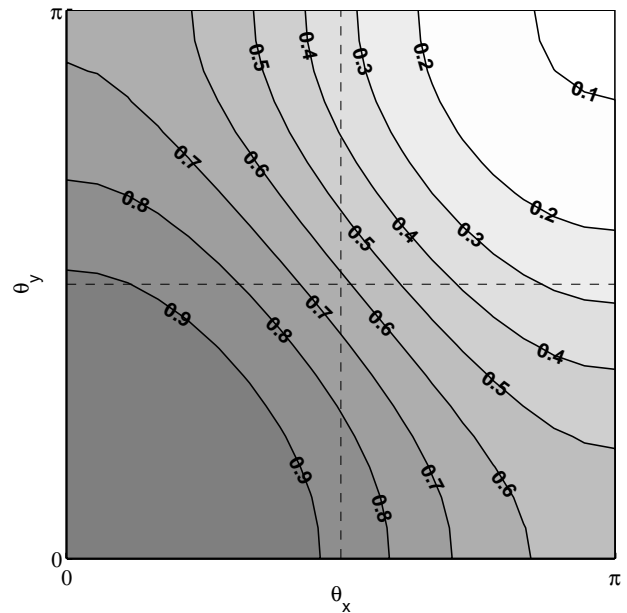
*Computational Fluid Dynamics : Presented at the Winter Annual Meeting of the American Society of Mechanical Engineers, Fluids Engineering Division, Vol. 103, ASME, 1990, pp. 145–152.*

<sup>12</sup>Morano, E., Mavriplis, D. J., and Venkatakrishnan, V., “Coarsening strategies for unstructured multigrid techniques with application to anisotropic problems,” *SIAM J. Sci. Comput.*, Vol. 20, No. 2, 1998, pp. 393–415.

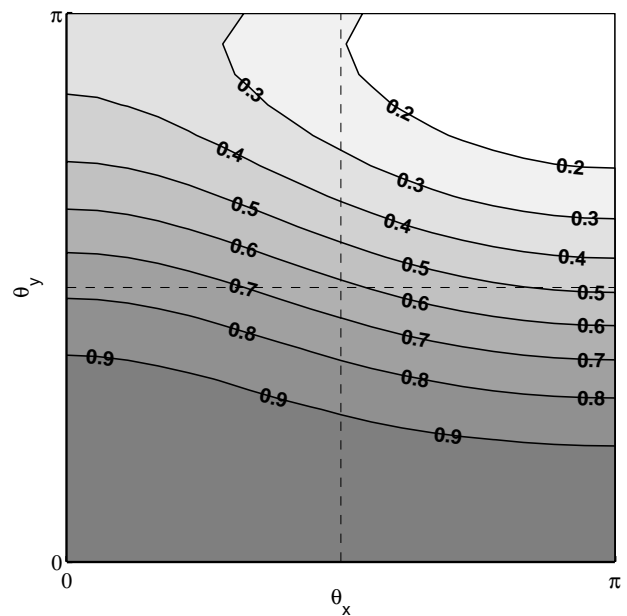
<sup>13</sup>Mavriplis, D. J., “Directional Agglomeration Multigrid Techniques for High Reynolds Number Viscous Flow Solvers,” *AIAA J.*, Vol. 37, No. 10, 1999, pp. 1222–1230.

<sup>14</sup>Spalart, P. R. and Allmaras, S. R., “A One-Equation Turbulence Model for Aerodynamic Flows,” *La Recherche Aéronautique*, No. 1, 1994, pp. 5–21.

<sup>15</sup>Jameson, A., Schmidt, W., and Turkel, E., “Numerical Solution of the Euler Equations by Finite-Volume Methods Using Runge-Kutta Schemes,” AIAA Paper 81-1259, AIAA, 1981.



a)  $dy/dx = 1$



b)  $dy/dx = 1 \times 10^{-6}$

**Fig. 1** Amplification factor  $|\sigma|$  versus Fourier angle  $(\theta_x, \theta_y)$  for rectangular control-volume  $dx \times dy$  at  $M = 0.5$ ,  $Re = 1 \times 10^6$  with flow aligned along  $x$  axis. Five stage explicit relaxation method with  $\alpha = [\frac{1}{5} \frac{1}{4} \frac{1}{3} \frac{1}{2} 1]$  and  $\gamma = [1 \ 0 \ 0.56 \ 0 \ 0.44]$ .

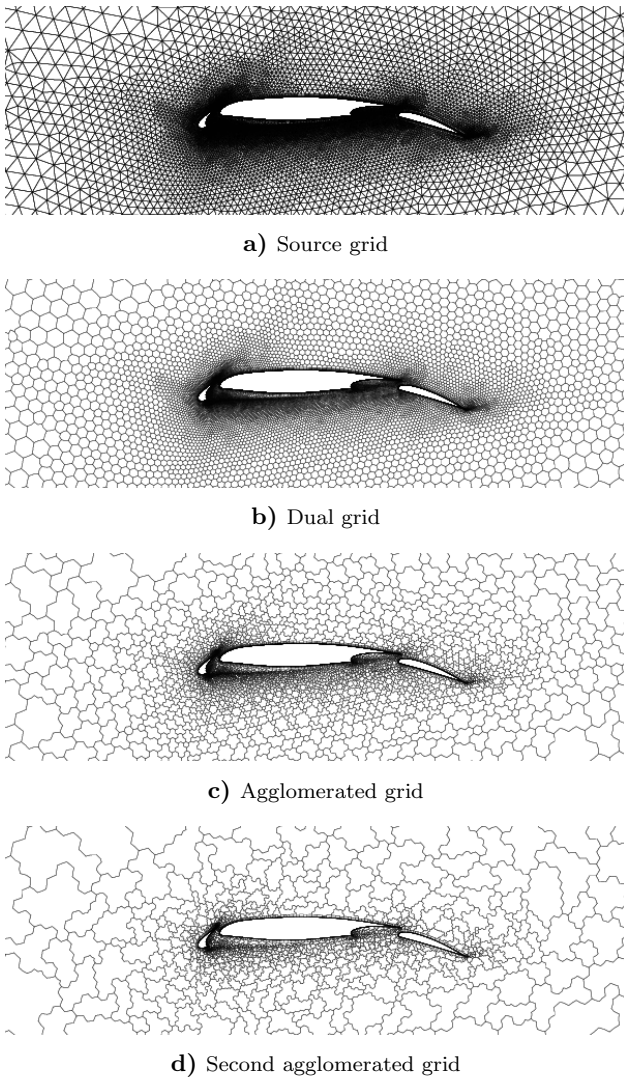


Fig. 2 A sample source grid with generated dual and agglomerated grid.

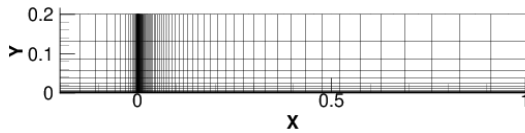


Fig. 3 A rectilinear mesh with maximum cell aspect ratio of 26500 and off-wall spacing of  $5 \times 10^{-6}$  about a flat plate aligned with the  $x$  axis from  $x \in [0, 1]$ .

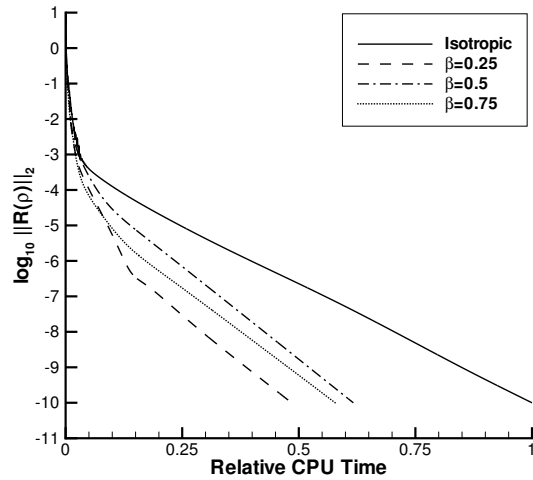


Fig. 4 Convergence history for various directional coarsening  $\beta$  coefficients for laminar flow over a flat plate at  $M = 0.2$  and  $Re = 1 \times 10^5$ . Computed on a rectilinear mesh with cell aspect ratio on the order of 26500.

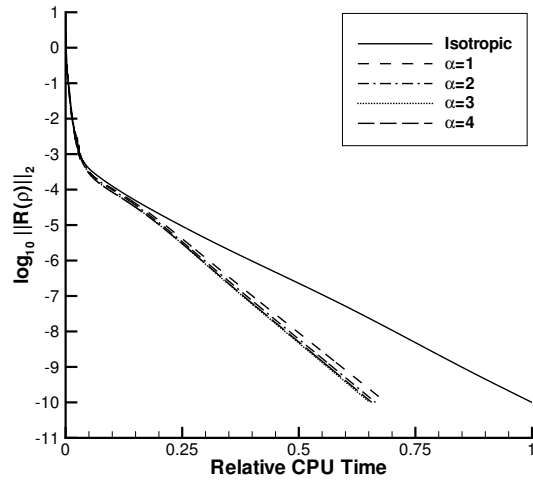


Fig. 5 Convergence history for various line-implicit smoothing  $\alpha$  coefficients for laminar flow over a flat plate at  $M = 0.2$  and  $Re = 1 \times 10^5$ . Computed on a rectilinear mesh with cell aspect ratio on the order of 26500.

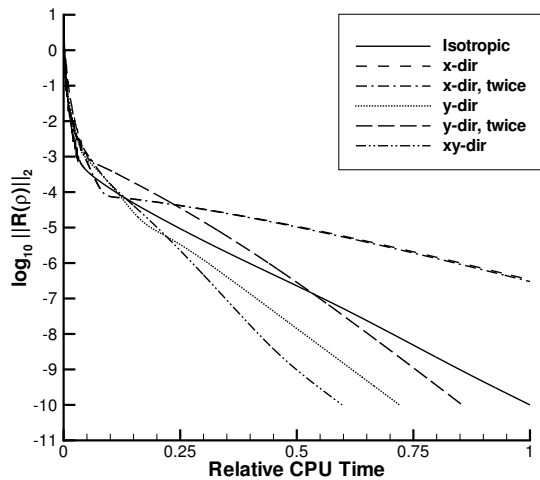


Fig. 6 Convergence history for explicit directional coarsening for laminar flow over a flat plate at  $M = 0.2$  and  $Re = 1 \times 10^5$ . Computed on a rectilinear mesh with cell aspect ratio on the order of 26500.

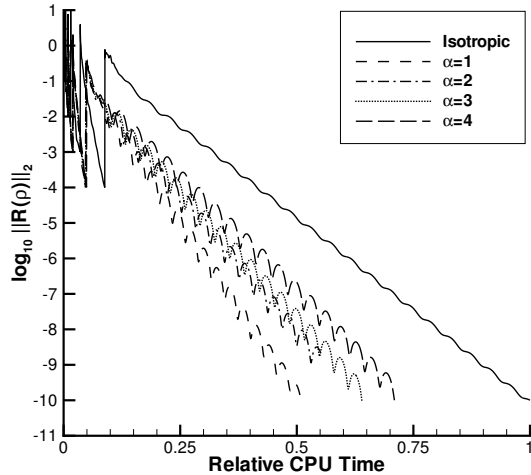


Fig. 8 Convergence history for various line-implicit smoothing  $\alpha$  coefficients for flow about an RAE 2822 airfoil at  $M = 0.754$ ,  $\alpha = 2.57^\circ$  and  $Re = 6.2 \times 10^6$ . Computed on a structured C-mesh with cell aspect ratio on the order of 3300.

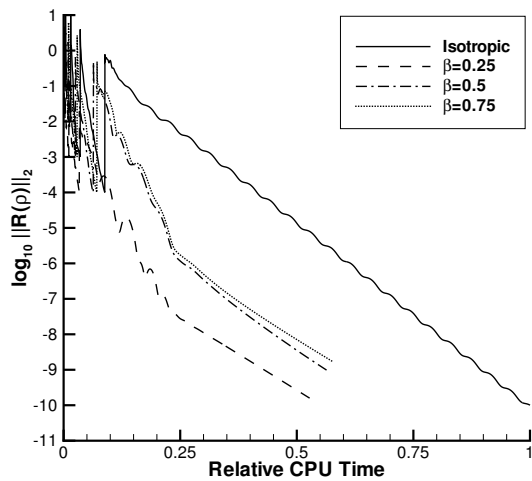


Fig. 7 Convergence history for various directional coarsening  $\beta$  coefficients for flow about an RAE 2822 airfoil at  $M = 0.754$ ,  $\alpha = 2.57^\circ$  and  $Re = 6.2 \times 10^6$ . Computed on a structured C-mesh with cell aspect ratio on the order of 3300.

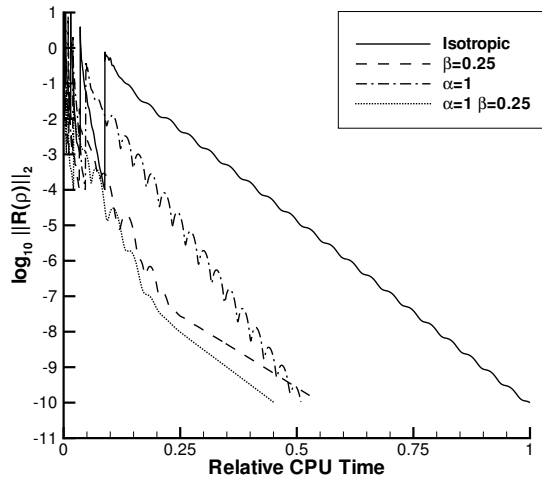


Fig. 9 Convergence history for combined directional coarsening and line-implicit smoothing for flow about an RAE 2822 airfoil at  $M = 0.754$ ,  $\alpha = 2.57^\circ$  and  $Re = 6.2 \times 10^6$ . Computed on a structured C-mesh with cell aspect ratio on the order of 3300.

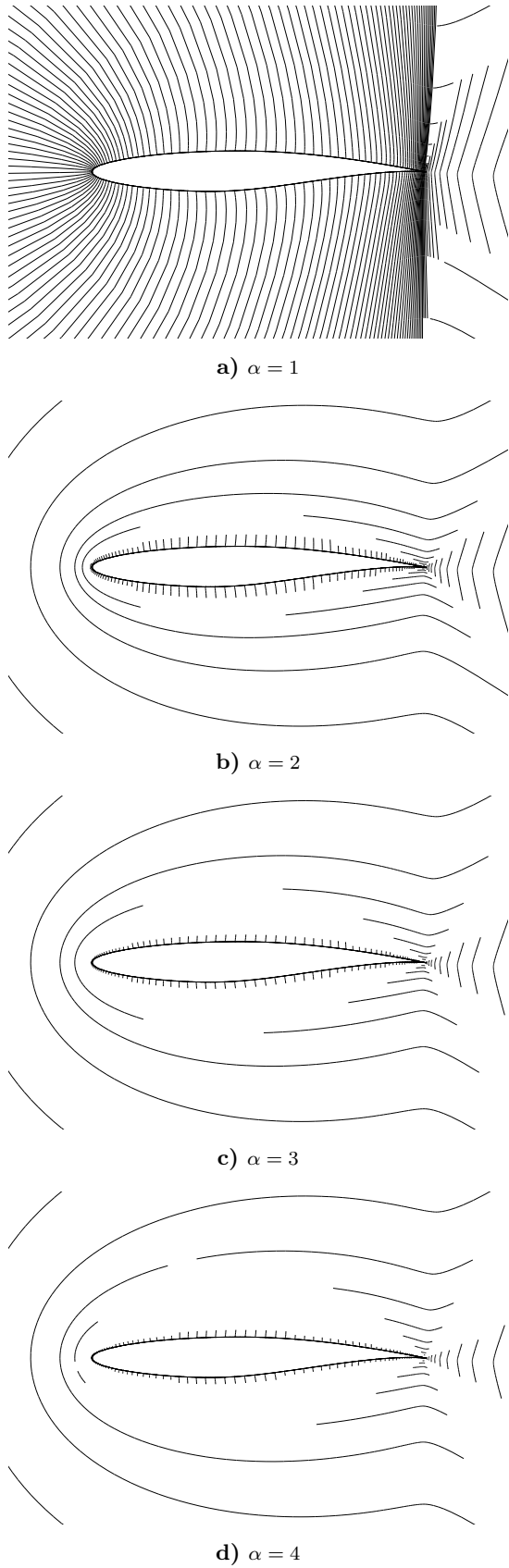


Fig. 10 Implicit line construction for a structured mesh about an RAE 2822 airfoil for  $\alpha \in [1 : 4]$ .

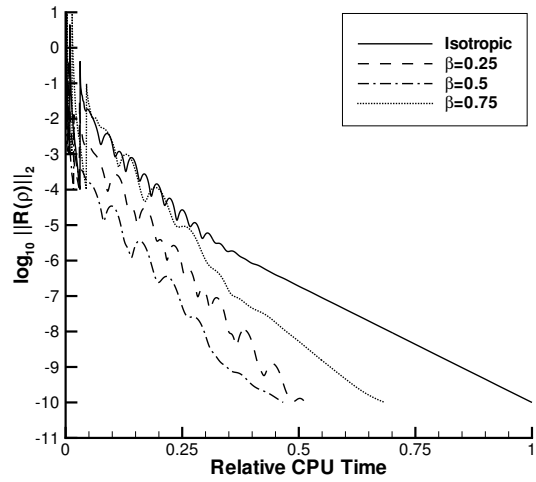


Fig. 11 Convergence history for various directional coarsening  $\beta$  coefficients for flow about an RAE 2822 airfoil at  $M = 0.754$ ,  $\alpha = 2.57^\circ$  and  $Re = 6.2 \times 10^6$ . Computed on a hybrid unstructured mesh with cell aspect ratio on the order of 3300.

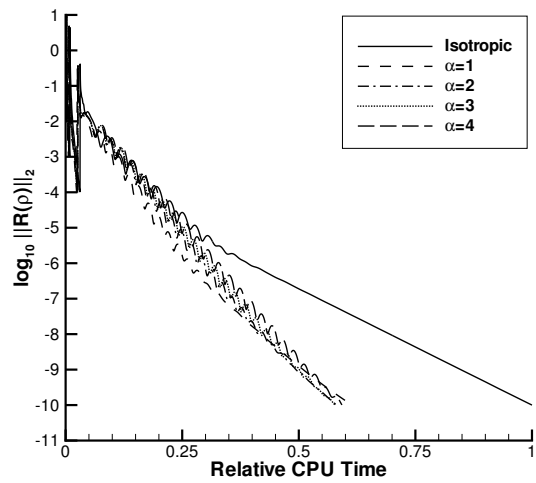


Fig. 12 Convergence history for various line-implicit smoothing  $\alpha$  coefficients for flow about an RAE 2822 airfoil at  $M = 0.754$ ,  $\alpha = 2.57^\circ$  and  $Re = 6.2 \times 10^6$ . Computed on a hybrid unstructured mesh with cell aspect ratio on the order of 3300.



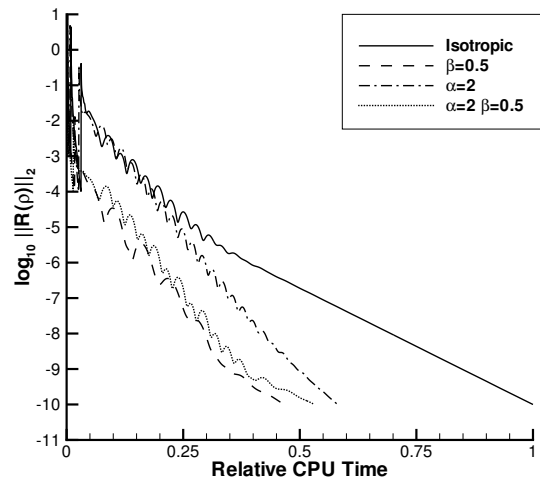
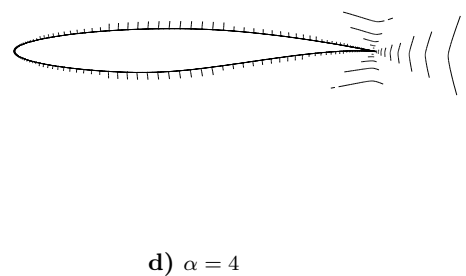
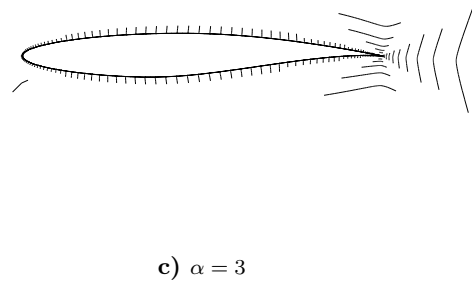
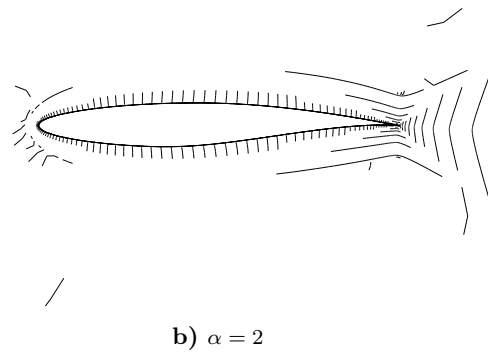
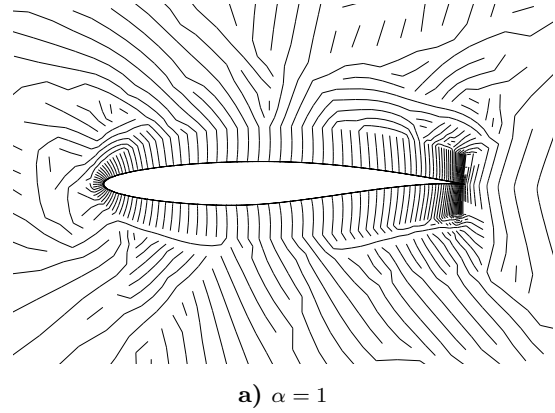


Fig. 13 Convergence history for combined directional coarsening and line-implicit smoothing for flow about an RAE 2822 airfoil at  $M = 0.754$ ,  $\alpha = 2.57^\circ$  and  $Re = 6.2 \times 10^6$ . Computed on a hybrid unstructured mesh with cell aspect ratio on the order of 3300.

Fig. 14 Implicit line construction for an unstructured hybrid mesh about an RAE 2822 airfoil for  $\alpha \in [1 : 4]$ .