INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
*Int. J. Numer. Meth. Fluids* 2008; **58**:743–750
Published online 17 March 2008 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/fld.1766

# A multigrid procedure for Cartesian ghost-cell methods

L. A. Catalano*,†,‡, A. Dadone, V. S. E. Daloiso and D. Scardigno

*Dipartimento di Ingegneria Meccanica e Gestionale & Centro di Eccellenza in Meccanica Computazionale, Politecnico di Bari, via Re David 200, I-70125 Bari, Italy*

SUMMARY

This paper proposes a multigrid technique for Cartesian grid flow solvers. A recently developed ghost body-cell method for inviscid flows is combined with a nested-level local refinement procedure, which employs multigrid to accelerate convergence to steady state. Different from standard multigrid applications for body-fitted grids, a fictitious residual needs to be defined in the ghost cells to perform a correct residual collection and thus to avoid possible stalling of the multigrid procedure. The efficiency of the proposed local refinement multigrid Cartesian method is demonstrated for the case of the inviscid subsonic flow past a circular body. Copyright © 2008 John Wiley & Sons, Ltd.

Received 27 March 2007; Revised 21 December 2007; Accepted 27 December 2007

KEY WORDS: Cartesian methods; multigrid; ghost cells; residual collection; inviscid flow; boundary conditions

## 1. INTRODUCTION

The continuous growing power of computers is encouraging engineers to rely on computational fluid dynamics (CFD) for the design of new aerodynamic shapes. Numerical simulations allow the analysis of flow phenomena without resorting to expensive and complex experimental measurements. On the other hand, while simple geometries discretized by regular grids are efficiently handled by codes and hardware, flows with complex geometries discretized on body-fitted grids are still challenging problems for computers of today. The grid generation process, whether for multiblock structured grids or unstructured grids, is often laborious and is not always automated as desired. Moreover, the conceptual design process often requires detailed body-fitted grids over

*Correspondence to: L. A. Catalano, Dipartimento di Ingegneria Meccanica e Gestionale, Sezione Macchine ed Energetica, Italy.
†E-mail: catalano@poliba.it
‡Professor.

Contract/grant sponsor: MUR
Contract/grant sponsor: CIRA

a wide class of different geometric topologies. Grid generation for complex topologies often takes weeks or even longer and this becomes a major impediment to using CFD in design. Further complications may occur due to moving boundaries or in problems that require continuing regeneration or deformation of grids.

One approach to develop a simplified alternative to complex grid generation involves the use of Cartesian grid methods. In these methods the body does not coincide with grid surface or grid volume edges. On the contrary, the body is immersed in the grid: the cumbersome task of generating a body-fitted grid is substituted by the complications associated with the implementation of the surface boundary conditions. Furthermore, since the body surfaces are not aligned with the grid, it is nearly impossible to adequately resolve high Reynolds number boundary layers with Cartesian mesh methods. For these reasons, a useful application of Cartesian grid methods can be obtained only in combination with an accurate representation of the boundary conditions near the solid walls and with the choice of non-uniform computational grids, which can be obtained either by coarsening a fine grid (as done in Reference [1]) or by refining a coarse grid (as proposed in Reference [2]). Much greater advantages can result from the use of a nested-level multigrid procedure.

The aim of this paper is to develop an accurate and efficient Cartesian method for inviscid steady flows. The space discretization consists of the ghost body-cell method (GBCM) for Cartesian grids proposed in [1]. The crux of this method is the so-called curvature-corrected symmetry technique (CCST) described in Reference [3], where an accurate near-body flow model is proposed as an alternative to extrapolating pressure at the wall. Reference [4] proposes its application to unstructured grids, whereas Reference [1] extends the CCST to non-body-fitted Cartesian grids. The accuracy of the resulting GBCM procedure has been already verified *versus* inviscid flow test cases, including the flow past circular bodies and airfoils; see [1] for details. Rather than using coarsening and stretching procedures, as done in Reference [1], this paper proposes to couple the GBCM with an adaptive local grid refinement technique, similar to the approach proposed in Reference [2]. However, different from [2], the locally refined grid is considered here as the superposition of nested incomplete grid levels, which can include refined and unrefined cells. This approach naturally adapts to the implementation of a multigrid technique, which allows one to solve the flow equations on all cells of each level, so as to accelerate convergence to steady state. However, employing a Cartesian grid does not allow a standard coarsening, in particular for what concerns the residual collection, which requires a special treatment for the near-body regions.

This paper aims at answering the above requirements: after describing the basic concepts of the GBCM, the paper proposes a local refinement strategy based on both geometrical and fluid-dynamic parameters; then, we propose how to perform the residual collection near-solid walls, where an incomplete set of refined cells exist. A simple test case, namely the inviscid flow past a circular body, is proposed to demonstrate the efficiency of the local refinement multigrid technique for Cartesian solvers.

## 2. GHOST BODY-CELL METHOD

The GBCM is an innovative numerical technique to enforce non-penetration conditions at solid walls in Cartesian grid computations. The approach uses only rectangular cells without having to define *cut cells* in the vicinity of the solid boundaries.



DOI: 10.1002/fld

The GBCM is described by the following steps [1]:

1. The body is defined by a dense, ordered sequence of $(x, y)$ coordinate pairs. The definition of the surface is completely independent of the mesh.
2. A Cartesian grid is selected and the computational flow field is subdivided into finite-volume cells.
3. The cell centers are discriminated between those located at the interior of the body and those located externally. The internal cell centers are flagged with $\lambda=0$, while the external ones are flagged with $\lambda=1$.
4. The row of cell centers closest to the body and located at the interior of the body is singled out. These cell centers are indicated by circles in Figure 1, which presents a part of the cell-center net and the curved body profile.
5. A second row of cell centers located at the interior of the body and adjacent to the first row is singled out. These cell centers are indicated by triangles in Figure 1.
6. For each of the interior cell centers of the two rows described above (ghost-cell centers), a corresponding symmetric point is determined at a location exterior to the body and reflected symmetrically with respect to the body surface. These symmetric points are indicated by dark squares in Figure 1. As an example, the point B and the interior cell center A in Figure 1 are symmetric with respect to the body.
7. The four cell centers surrounding each symmetric point are determined. As an example, points C, D, E, F are the cell centers surrounding point B in Figure 1. Note that one or even two of these four cell centers may be located inside the body.
8. The value of the conserved variables $\mathbf{q}$ at the point B are determined by a bilinear interpolation:

$$\mathbf{q}_B = k_1(x-x_C)(y-y_C)+k_2(x-x_C)+k_3(y-y_C)+k_4 \tag{1}$$

where the constants $k_i$ $(i=1,2,3,4)$ are determined by the values of the conserved variables $\mathbf{q}$ at the cell centers C, D, E, F. The bilinear interpolation is also used in situations when

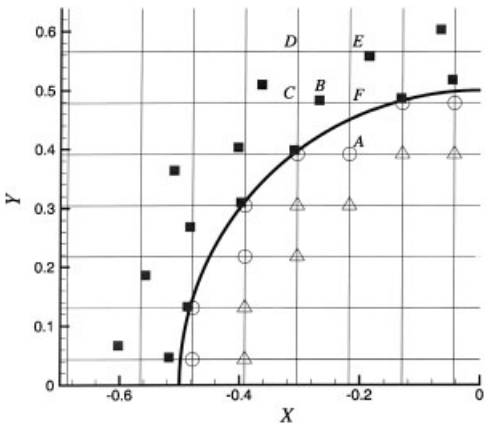


Figure 1. Cartesian grid: cell-center net, interior cell centers close to the body (open symbols) and symmetrically exterior points (solid symbols).



DOI: 10.1002/fld

one or two of the cell centers surrounding point B are located inside the body surface. The internal values are determined from the CCST boundary conditions (described next).

9. The values of the physical variables at the cell centers belonging to the previously defined two rows of internal cell centers are determined according to the CCST boundary conditions. As an example, the physical variables at the cell center A are obtained by enforcing the normal momentum equation and the non-penetration boundary condition as follows:

$$p_A = p_B - \rho_B \frac{\tilde{u}_B^2}{R} \Delta n \tag{2}$$

$$\rho_A = \rho_B \left( \frac{p_A}{p_B} \right)^{(1/\gamma)} \tag{3}$$

$$\tilde{u}_A^2 = \tilde{u}_B^2 + \frac{2\gamma}{\gamma - 1} \left( \frac{p_B}{\rho_B} - \frac{p_A}{\rho_A} \right) \tag{4}$$

$$\tilde{v}_A = -\tilde{v}_B \tag{5}$$

In Equations (2)–(5), $p$ is the pressure, $\rho$ is the density, $\tilde{u}$ and $\tilde{v}$ are the velocity components tangential and normal to the body, respectively, $\Delta n$ is the distance between the cell centers A and B, while $R$ is the signed local radius of curvature of the body.

10. The time-dependent computation of all conserved variables at cell centers located externally to the body now proceeds without any other boundary condition. Each cell center utilizes the fluxes at surrounding cell edges without further approximation. Cell centers located near the boundary utilize the interior (ghost) cells.

## 3. LOCAL REFINEMENT

Standard local refinement procedures need to be slightly modified when combined with Cartesian solvers. Here, a very coarse, uniformly spaced, Cartesian grid, with cell sizes comparable to the body dimensions, is adopted as the coarsest grid. The choice of such a coarse mesh allows to avoid stretching. However, the body cannot be seen at this level: thus, recurrent applications of local refinement only based on a geometric criterion (e.g. the distance from the body) are needed to properly *define* the body, namely before starting the flow computations. Then, local refinement can be based, as usual, on properly defined fluid-dynamic criteria, such as flow gradients or undivided differences. However, two difficulties usually occur: (i) fluid-dynamic refinement criteria generate irregular clusters of refined cells and (ii) a cumbersome interpolation of the flow variables of the coarser-level unrefined cells is needed to compute the flux through the boundaries between two nested levels. For these motivations, the local refinement procedure proposed in this paper is structured as follows:

1. choose a suitable refinement criterion, based on the distance from the body surface and/or on the flow gradients;
2. select all cells satisfying the refinement criterion (cells flagged with $m$ in Figure 2);



DOI: 10.1002/fld

| | | | | | | |
|---|---|---|---|---|---|---|
| | 2 | | | 2 | 2 | |
| 2 | m | 2 | 2 | m | m | 2 |
| 2 | m | m | m | m | 1 | 2 |
| 2 | m | m | m | 1 | m | 2 |
| | 2 | m | m | 1 | m | 2 |
| | | 2 | 2 | 2 | 2 | |

Figure 2. Local refinement strategy.

3. select all cells (flagged with 1 in Figure 2) having more than one (or more than two, as done in Figure 2 and in the proposed numerical test) neighboring cells which have been already selected for the refinement; repeat this step until no more cells are selected; this avoids irregularities in the finer, locally refined, level;
4. select (if not already selected) all cells, neighboring cells *m* and 1 (flagged with 2 in Figure 2); indeed, the computation of the flux through the finer-grid interfaces close to the refinement boundary requires knowledge of the flow field also beyond the refinement boundary; the required flow values could be either computed every time they are needed, with the consequent coding complexity, or, more efficiently, interpolated and stored in the ghost cells resulting from the refinement of the cells flagged with 2. Following this approach, which requires a slightly greater storage, the interpolation of the flow variables from the coarser to the finer level is performed only once at each iteration (before starting the flux computations at the finer level) for all refined cells; moreover, no particular treatment is required for the interfaces close to the refinement boundary;
5. refine all selected cells.

As usual, a difference of only one level is allowed for contiguous cells of the composite grid.

## 4. RESIDUAL COLLECTION FOR THE GHOST CELLS

One of the steps characterizing a multigrid cycle is the residual collection: on a cell-center grid it consists of summing up all residuals (namely all flux balances) in the cells originated from the same coarser cell. However, when employing a Cartesian method, this set of cells (four in 2D, eight in 3D, for an isotropic refinement) can be incomplete, since one or more cells could exist only as ghost cells. The flow solution in the ghost cells is computed by means of the GBCM, and thus it is updated only according to the update of the flow solution at the internal surrounding cells. Since no residuals are computed in the ghost cells, a special treatment is required to obtain a proper residual collection, and thus to avoid possible stalling of the multigrid procedure. The approach proposed in this paper consists of defining and computing a fictitious residual $\mathscr{R}_{\mathrm{gc}}$ in each ghost cell. The residual $\mathscr{R}_{\mathrm{gc}}$ is not used for updating the flow solution in the ghost cells, but



DOI: 10.1002/fld

only for the residual collection; it is computed as follows:

1. store the fine-grid solution at iteration $n$: $\mathbf{q}^n$;
2. apply the GBCM to obtain the ghost-cell solution: $\mathbf{q}^n_{\mathrm{gc}}$;
3. update the fine-grid solution and compute the ghost-cell solution: $\mathbf{q}^{n+1}_{\mathrm{gc}}$;
4. compute the ghost-cell residual $\mathscr{R}_{\mathrm{gc}}$ as

$$\mathscr{R}_{\mathrm{gc}} = \frac{\mathbf{q}^{n+1}_{\mathrm{gc}} - \mathbf{q}^n_{\mathrm{gc}}}{\Delta t} \tag{6}$$

   and use $\mathscr{R}_{\mathrm{gc}}$ in the residual collection;
5. restore the ghost-cell value $\mathbf{q}^n$.

All other multigrid operators can be applied straightforwardly.

## 5. RESULTS

The efficiency of the proposed local refinement multigrid Cartesian method is demonstrated for the case of the inviscid subsonic flow ($M_\infty = 0.4$) past a circular body with radius $r = 0.5$, centered in the point (0; 0). The computational domain ranges from $-22.4$ to 22.4 in both coordinate directions, with a coarsest grid composed of $14 \times 14$ uniformly spaced quadrilateral cells ($\Delta x = \Delta y = 3.2$). A local refinement based on a geometric criterion, namely on the distance from wall (the threshold distance is reduced at each finer level), is firstly performed; the resulting composite grid allows one to define, even not accurately, the surface, and thus to perform some iterations of the flow solver. At this stage, a fluid-dynamic criterion, namely the pressure gradient, is employed to further refine the grid. This procedure corresponds to the nested iteration of the full multigrid cycle [5].

At the end of the nested iteration, the finest mesh size $\Delta x = \Delta y = 0.0125$ has been reached. Figure 3 shows the Mach number contours computed after a residual drop of about five orders of magnitude, which can be considered sufficient for convergence. The symmetry of the Mach number contours demonstrates the good space accuracy obtained by extending the CCST method to Cartesian grids, as already proven in Reference [1].

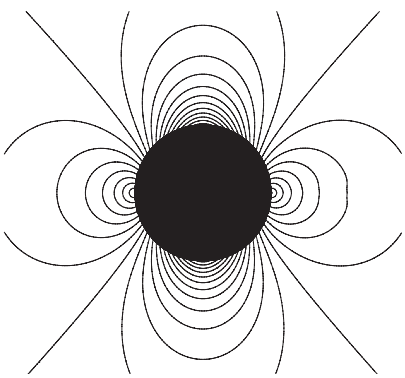

Figure 3. Mach number contours.



DOI: 10.1002/fld

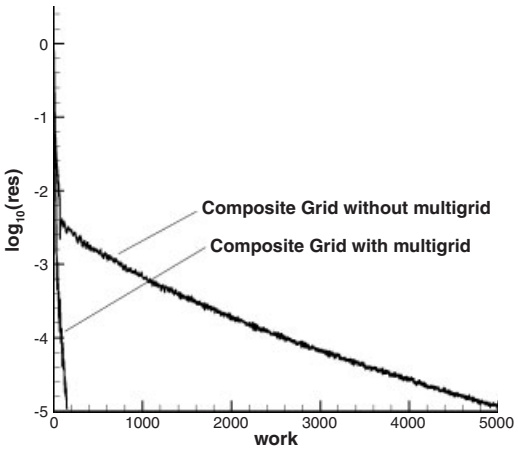


Figure 4. Convergence histories.

Figure 4 presents the convergence histories obtained on the same composite grid by using the proposed multigrid method with fictitious ghost-cell residuals and by computing the flow solution only on the unrefined cells, namely without the aid of multigrid. One work unit is defined as $work = 10^{-7} w_{sf}$, $w_{sf}$ being the central processing unit (CPU) time required by one flux computation on a single interface. Figure 4 clearly demonstrates that the proposed multigrid strategy with fictitious ghost-cell residuals requires a CPU time about 30 times lower than that required by computing the same flow on the same grid, without multigrid. This efficiency gain is almost independent of the Mach number, as demonstrated by further computations with lower values of the Mach number (down to $M = 0.1$), not reported here.

## 6. CONCLUSIONS

The paper has proposed a local refinement multigrid technique that can be effectively combined with Cartesian grid flow solvers. The entire procedure consists of (i) a GBCM, able to accurately simulate the presence of solid walls; (ii) a local grid refinement strategy adapted for the application to Cartesian grid flow solvers; and (iii) the definition of a fictitious residual in all ghost cells involved in the residual collection. The efficiency of the proposed local refinement multigrid Cartesian method has been demonstrated *versus* the inviscid flow past a circular body: the proposed multigrid method with fictitious ghost-cell residuals has required a CPU time about 30 times smaller than that required by computing the flow solution only on the unrefined cells of the same, locally refined, grid (composite grid), namely without the aid of multigrid.

ACKNOWLEDGEMENTS

This work has been supported by MUR, under research programme PRIN 2005, and by CIRA, under ACADEMIA Project (supported by MUR).



DOI: 10.1002/fld

## REFERENCES

1. Dadone A, Grossman B. Ghost-cell method for inviscid two-dimensional flows on Cartesian grids. *AIAA Journal* 2004; **42**(12):2499–2507.
2. Aftosmis MJ. *Solution Adaptive Cartesian Grid Methods for Aerodynamic Flows with Complex Geometries*. VKI Lecture Series 1997-02, von Karman Institute for Fluid Dynamics, Belgium, 1997.
3. Dadone A, Grossman B. Surface boundary conditions for the numerical solution of the Euler equations. *AIAA Journal* 1994; **32**(2):285–293.
4. Wang ZJ, Sun Y. Curvature-based wall boundary condition for the Euler equations on unstructured grids. *AIAA Journal* 2003; **41**(1):27–33.
5. Brandt A. Multi-level adaptive solutions to boundary-value problems. *Mathematics of Computation* 1977; **31**: 333–390.



DOI: 10.1002/fld