AIAA JOURNAL
Vol. 33, No. 5, May 1995

Fully Implicit Solution of Time-Dependent Partial Differential Equations in Arbitrary Multizone Domains

Moshe Rosenfeld*
Tel-Aviv University, Tel-Aviv 69978, Israel
and
Yuval Yassour†
Technion—Israel Institute of Technology, Haifa 32000, Israel

The alternating direction multizone implicit method is extended to generalized coordinate systems with complex grid-topology cases. The alternating direction multizone implicit method is a novel approach for solving implicitly systems of time-dependent partial differential equations in multizone domains. The method combines alternate direction implicit (or approximate factorization) methods and domain decomposition techniques to yield an efficient time-accurate solution method. Fully implicit solution is possible by defining a separate set of zone for each stage of the alternate direction implicit method. A flexible and simple data structure is developed, so that the alternating direction multizone implicit method serves as a driver that manages data transfers between permanent zones, where the problem is defined and stored, and temporary (sweep) zones, where the calculations are actually performed. The discretization is done only at the level of the sweep zones. Consequently, the alternating direction multizone implicit method can be applied to many existing solvers (that use alternate direction implicit or approximate factorization techniques) without anticipating major programming efforts. Several test cases demonstrate the capabilities of the alternating direction multizone implicit method to solve implicitly problems with complex geometry.

Introduction

THERE are several reasons the decomposition of a computational domain into subdomains (zones) may be required. Geometric complexity is probably the most common reason; it may not be possible to generate a single-zone structured grid that covers the entire domain with adequate control on the mesh spacing. Still other reasons may lead to domain decomposition, such as in the implementation of parallel algorithms or in the case of large problems that do not fit into the core memory.

Although domain decomposition methods simplify the discretization of complex problems, they suffer from limitations, especially if the stiffness of the problem (such as in the case of viscous flows) invokes the use of implicit schemes. Two of the most severe limitations are the fictitious numerical boundary conditions needed for solving the governing equations in each zone and the nonconservation of fluxes between the zones. Both problems arise because of the creation of fictitious boundaries in the interface between the zones. These boundaries are referred to as zonal boundaries. Nevertheless, the zonal boundary points are regular interior field points, where the governing equations should be satisfied as in any other point of the domain.

In most previous "implicit" solution methods, the zonal boundary conditions were treated explicitly by variants of the Schwarz[1] procedure. Typically, the boundary conditions at the zonal boundaries were first guessed, and the equations in the zone were advanced one iteration (or time step) implicitly, using existing single-zone solvers. In the next iteration, the zonal boundary conditions were updated based on the solution obtained from the neighboring zones. This technique works reasonably well for steady problems. In the case of time-dependent calculations, however, the explicit treatment of the zonal boundary conditions may degrade the temporal accuracy of the solution, unless expensive subiterations are performed at each time step. The subiterations are needed to update the explicit zonal boundary conditions but increase significantly the computational time. Time-dependent calculations are anyway computationally intensive; the additional overhead required for the multizone solutions might be prohibitive. Moreover, because the zones are treated as separate entities, difficulties might be encountered in the conservation of fluxes between the zones, unless special flux-conserving approximations are employed.

A scheme is fully implicit only if the points on the zonal boundaries are solved implicitly as well. Such fully implicit solutions of multizone problems are very scarce. Rogers[2] used subiterations in a multizone calculation of the time-dependent incompressible Navier–Stokes equations using a line relaxation method. The solution was advanced in time by the artificial compressibility method, which requires anyway subiterations at each time step. Consequently, the zonal boundaries were updated implicitly, at the expense of increased computational requirements. Lei et al.[3] devised a fully implicit solution method for solving the Navier–Stokes equations using multizones. The whose set of equations for all of the zones, including for the points on the zonal boundaries, was solved simultaneously. This necessarily implies the solution of a very large number of algebraic equations; the computational resources might be prohibitive, especially if time-dependent problems are considered.

The alternating direction implicit (ADI) method, also known as the approximate factorization method, is a very common method of solving implicitly and efficiently partial differential equations (PDE), especially in the context of the Navier–Stokes equations.[4–6] Yet, existing multizone methods employed the ADI technique to each zone separately. Recently, the present authors[7] suggested combining domain decomposition and ADI methods into an efficient, fully implicit solution method of PDE using multizones. The method is known as the alternating direction multizone implicit (ADMZI) method. The layout of the method was described in a previous paper,[7] that demonstrated the concepts but applied them to very simple problems. In particular, issues concerning the treatment of complex mesh-topology and the underlying data structures were not addressed.

In the present paper, we extend the ADMZI method to more complex cases, devise a flexible but simple data structure, and demonstrate the capabilities of the method to solve problems with complex geometry. The principles of the ADMZI method will be reviewed

Presented in part as Paper 94-0067 at the AIAA 32nd Aerospace Sciences Meeting, Reno, NV, Jan. 10–13, 1994; received April 12, 1994; revision received Aug. 30, 1994; accepted for publication Sept. 6, 1994.

*Lecturer, Department of Fluid Mechanics and Heat Transfer. Member AIAA.
†Research Associate, Faculty of Aerospace Engineering.

and extended in the next section, followed by several implementation notes. In the Results section, these notions will be demonstrated by solving several test cases.

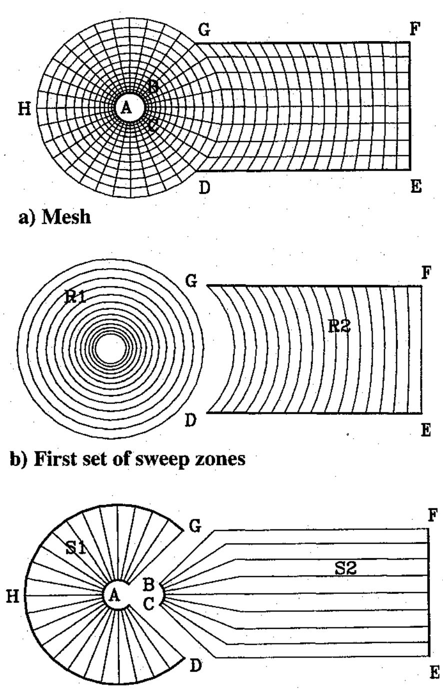
Principles of the Alternating Direction Multizone Implicit Method

In the ADI (or approximate factorization) method, the discretized PDE are approximately solved by representing the original discrete equations in a factored form and solving for each factor implicitly. Each factor requires the solution of the discrete equations along one family of coordinate lines. In the present paper, the simultaneous solution of the equations along a coordinate line is referred to as a sweep, whereas all of the sweeps belonging to a single factor of the ADI method are referred to as a stage.

In standard multizone calculations, one or two edges of the sweep lines might be on zonal boundaries. On the zonal boundaries, the numerical boundary conditions are lagged in time and, thus, a fully implicit solution of the entire domain cannot be obtained in a simple and efficient way. These difficulties can be avoided if the boundaries at the edges of the sweep lines are real, i.e., boundary conditions are specified on them. If standard techniques are used, this condition cannot be generally met for multizones, because the same set of zones is used for executing all of the stages of the ADI method.

The key idea of the ADMZI method is the use of a different set of zones for each stage of the ADI method. Abandoning the requirement of using a single set of zones for all of the stages allows constructing each zone to have real boundaries at the edges of the sweep lines. This procedure eliminates zonal boundaries and allows the fully implicit solutions of the discrete equations in multizone domains.

To demonstrate the general idea and to establish the terminology to be used throughout the paper, a simple example is given in Fig. 1. The two-dimensional domain over a circular cylinder is composed of two zones. An O grid is generated around the circular cylinder, and an H grid is attached to the right side of the O grid, creating the zonal boundary DG. The first stage of the ADMZI method is performed for the O-sweep lines of zone R1, followed by the vertical sweep lines of zones R2, see Fig. 1b. The thin lines show the sweep lines, whereas the thick lines represent the boundaries at the edges of the sweep lines. In zone R2, physical boundary conditions are specified at the edges of the sweep lines, whereas periodic conditions are given for the O-sweep lines of zone R1. In either case, the equations along all of the sweep lines can be solved implicitly.



a) Mesh

b) First set of sweep zones

c) Second set of sweep zones

Fig. 1 Domain decomposition required for the ADMZI method, case 1.

Only when first stage is completed for all of the R zones is the second ADI stage performed. For executing it, a different set of zones should be used to satisfy the condition that physical (not numerical) boundaries should be specified at the edges of the sweep lines, Fig. 1c. Two zones are needed for the second stage as well; zone S1 consists of the radial sweep lines between the cylinder and the outer boundary GHD of the O grid. Zone S2 consists of continuous sweep lines that are composed of the radial lines emanating from the circular cylinder (CDGB) and the horizontal coordinate lines of the H grid (GDEF).

Thus, in the ADMZI method a different set of zones is created for each stage of the ADI method. These zones will be referred to as sweep zones. All of the sweep zones have identical properties, no matter which stage they belong to or how complex is the geometry of the problem. The main properties are summarized as follows for two-dimensional cases.

1) Each sweep zone is a logically rectangular region, and the boundaries are always at the edges of the zones. No holes or other irregular points exist inside the field. The global topology of the mesh has no relevance to the sweep zones, where only one-dimensional sweep lines exist.

2) Real boundary conditions are specified at the edges of the sweep lines, i.e., no conditions should be guessed or lagged in time.

3) The sweep lines should be continuous, imposing certain restrictions on the mesh. The continuity of the mesh lines allows the conservation of fluxes throughout the solution domain, including across the zonal boundaries.

4) The discretization and the solution of the algebraic equations are performed only at the level of the sweep zones. It is a straightforward task to construct the discrete equations because all of the internal points are regular field points; boundaries exist only at the edges of the sweep zones. Thus, conditional statements are mostly avoided. The setup and solution of the equations are the principal CPU-time consumers and, therefore, the simplicity offered by the structure of the sweep zones may reduce the overall CPU-time requirements.

5) The form of the PDE is relevant only in the sweep zones, i.e., when the discretization is carried out. The mechanisms of constructing the sweep zones and transferring data to and from them are independent of the equations being solved or the discretization procedure being employed.

6) The sweep zones are created as needed, i.e., no permanent storage should be allocated for them. The problem itself (e.g., the geometry, the type of the PDE, the boundary conditions) is defined in a separate and unrelated set of zones (called memory zones, see following section).

Thus, the ADMZI method is essentially a driver for handling the data structures involved in the solution of PDE is multizones using ADI methods. Provided such a driver is developed, many existing solution methods can be plugged into it to obtain, with minimal programming efforts, implicit solutions of multizone domains. The main interest of the present study is the development of the ADMZI driver for complex geometrical situations.

Implementation Considerations

The ADMZI method is a driver that manages the data structures associated with the zones where the problem is defined and stored and the sets of sweep zones where the calculations are performed. One of the contributions of the present paper is the development of a systematic method for the manipulation of the different type of zones and the data transfers between them.

For solving PDE using the ADMZI method, the domain of calculation is first decomposed into an arbitrary number of topologically rectangular zones, labeled in the present work memory zones. The memory zones are selected to define the problem in the most convenient way. They have no direct relation to the sweep zones, and therefore, the requirements imposed on the sweep zones do not hold for the memory zones. The memory zones are the only permanent zones that are stored in the ADMZI method. The sweep zones are temporary zones that are created as needed. Each sweep zone is constructed by copying the relevant sections from the contributing memory zones into a temporary work area.

The boundaries parallel to the sweep lines may be composed of physical as zonal boundaries. For example, the boundary BGF of zone S2 (Fig. 1) consists of the zonal boundary BG and the physical boundary GF. The boundary sweep lines are made identical to any other sweep line by including the physical boundary points and imposing the boundary conditions on the physical segments of the boundary (instead of the governing differential equations). To simplify the discretization near the zonal boundaries, where data from adjacent zones is needed, the sweep zones include a margin of several (usually 1–2) mesh points wide. The margins are filled with data copied from the relevant memory zones. Yet, no unknowns are solved for the margin points and, thus, the margins have no relation to the overlapping-regions approach used in many existing multizone domains.

The problem introduced in Fig. 1 is used to illustrate these notions. The memory zones, M1 and M2, coincide in this case with the sweep zones of the first stage, R1 and R2 (for simplicity, the margins are not shown). To construct S1, the data included in the section CDHGBAC of M1 (Fig. 1a) are copied into S1. To generate S2, first the segment CDGB from M1 is copied, followed by M2 that is copied entirely to the right of the previous segment.

To define how each sweep zone (including the margins) is constructed from the memory zones, the logically rectangular segments of the memory zones should be specified, along with the segments of the sweep zone where these regions should be mapped to. It uniquely defines the construction of each sweep zone, as well as the inverse mapping from the sweep zones back to the memory zones. The overhead in CPU time for constructing the sweep zones is usually only a small fraction of the total CPU time required to setup and solve the equations.

The coefficients of the discrete equations should be calculated only for the sweep zones. This is a simple task, because the boundaries are always at the edges of the sweep zones and all of the internal points are regular field points. At the boundaries parallel to the sweep lines, the use of margins allows the computation of the coefficients (including the right-hand side) of the points next to or on the zonal boundaries exactly in the same manner as for any other internal point. Consequently, the coding required for calculating the coefficients is straightforward. First, the coefficients are calculated for all of the points, including for the boundary points that are parallel to the sweep lines (assuming they are all zonal boundaries), without using any conditional statements. Finally, the boundary conditions are imposed on the physical segments of the boundaries.

The steps required to advance the solution one time-step (or iteration) using the ADMZI method are summarized as follows.

1) Construct a sweep zone by assembling it from the appropriate memory zones, i.e., copy into the sweep zone the relevant data (coordinate system, boundary conditions, dependent variables, etc.) from the contributing memory zones.

2) Discretize the problem in the sweep zone.

3) Perform the sweeps, i.e., solve simultaneously the discrete equations along the sweep lines.

4) Copy of the solution back to the appropriate memory zones.

5) Repeat steps 1–4 for all of the sweep zones of the first stage.

6) Repeat steps 1–5 for the rest of the ADI stages (one or two more stages for two- and three-dimensional cases, respectively).

7) Update the solution at the new time level.

Note that step 2 is the only one where the form of the PDE and the coordinate system are taken into account. This module can be plugged in from existing solvers, e.g., incompressible or compressible flow solvers. The rest of the ADMZI method is generic and applies to any other PDE solver that uses ADI (or approximation factorization) method.

Results

This section presents several numerical test cases. The aim of the calculations is to demonstrate the capabilities of the ADMZI method to solve implicitly PDE using multizones. As was explained in the preceding section, the form of the equations is irrelevant to the ADMZI method itself, which serves as a driver. Therefore, to make things simple and put emphasize on the ADMZI driver itself, the two-dimensional, time-dependent heat conduction equation $\partial T/\partial t = \nabla^2 T + S$ is solved; T is the temperature, S is a source term, and t is the time. Nevertheless, the results apply equally well to more complex equations, as far as the ADMZI method itself is concerned.

The equation is discretized in time using a three-time-level scheme to ensure second-order temporal accuracy in generalized coordinate systems with mixed derivatives. A second-order accurate finite volume flux-conserving scheme is used for the spatial discretization with the variable T located at the center of the cell. The solution is advanced in time by the Douglas–Gunn[8] ADI scheme in delta form. These test cases are presented to demonstrate the type of problems the ADMZI method can handle. In all of the cases, a time-accurate solution is obtained, but only the steady solution is presented.

Case 1

The example given previously (Fig. 1) is solved in this case. The outer circular boundary is at a distance of 3 (nondimensionalized by the diameter of the cylinder). The boundary EF is at a distance of 9 from the center of the circular cylinder. A temperature of $T = 0$ is specified on the circular cylinder, whereas the outer boundary is kept at a temperature that varies linearly between $T = 0$ and 60 on the lower and upper boundaries, respectively. No source term is specified ($S = 0$). A mesh of 13×13 and 17×9 points is used in the memory zones M1 and M2. The steady solution is presented in Fig. 2. As expected, a smooth solution is obtained, including across the zonal boundaries. It is, of course, not surprising, bearing in mind that in the ADMZI method the zonal boundaries are regular field points.

Case 2

In the second case, the diffusion equation is solved for the region shown in Fig. 3a, where a possible mesh is also given. The three bodies represent a mechanical device placed in the middle of a chanel with (nondimensional) length and height of 30.5 and 10, respectively. The mushroom-like body is composed of a half-circular cylinder of radius 3 and a base with a width of 2 and height of 1.25. The small half-circular cylinder body, placed at a distance of 1.25 from the base of the mushroom-like body, has a diameter of 2. The annular cylinder has an outer radius of 3 and a thickness of 1. To obtain adequate control on the distribution of the mesh points, an O grid is generated around the whole mechanical device, whereas the other parts of the mesh are of H type.

The memory zones are shown in Fig. 3b. It should be recalled that the memory zones do not have to comply with the requirements imposed on the sweep zones. They are selected to define the problem in the most convenient way. For example, the memory zones may include holes, such as the holes shown by the blank regions in the memory zones M2 and M3. In the memory zones M2, the circular-arc-like body is treated as a hole, (i.e., it is ignored during the construction of the sweep zones), whereas in M3 the base of the mushroom-like body is treated as a hole.

The two sets of sweep zones are shown in Figs. 3c and 3d. For clarity, zones S3–S6 are displaced from their original positions inside S2. The construction of the sweep zones is more complex, but still all of the zones are created so that boundary conditions could be specified at the edges of the sweep lines. For example, the sweep lines of S3 start on the upper part of the mushroom-like body, encircle the half-circular cylinder body, and end up at the lower part of the mushroom-like body. Also, the sweep zone S2 (with periodic boundary conditions) consists of contributions from three memory

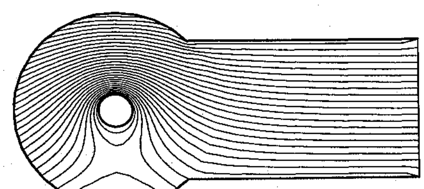
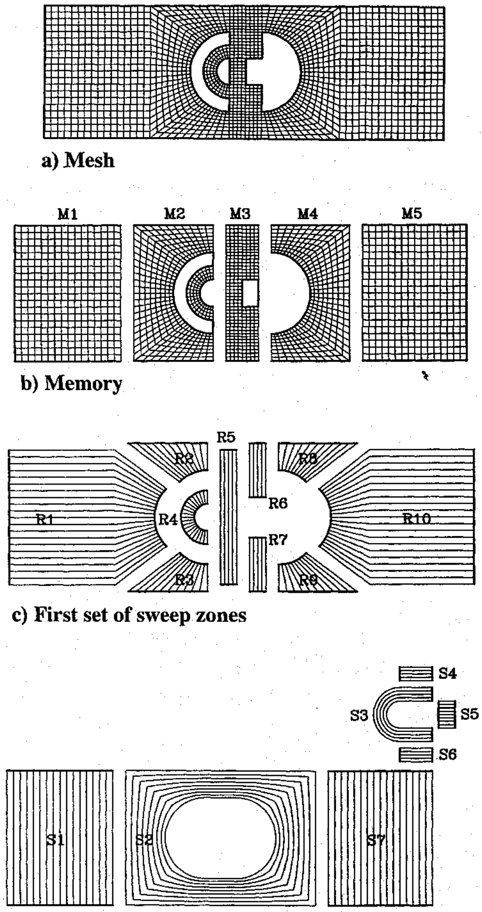


Fig. 2 Sample solution of case 1.



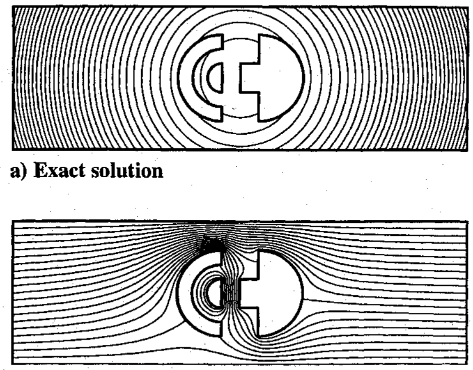
a) Mesh

b) Memory

c) First set of sweep zones

d) Second set of sweep zones

Fig. 3 Domain decomposition of case 22, zones S3–S6 are displaced from their original position inside S2.



a) Exact solution

b) Sample solution

Fig. 4 Sample solutions of case 2.

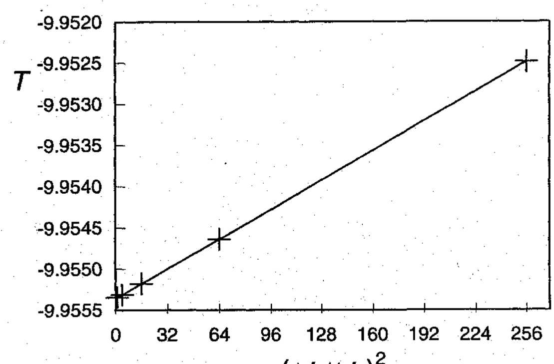
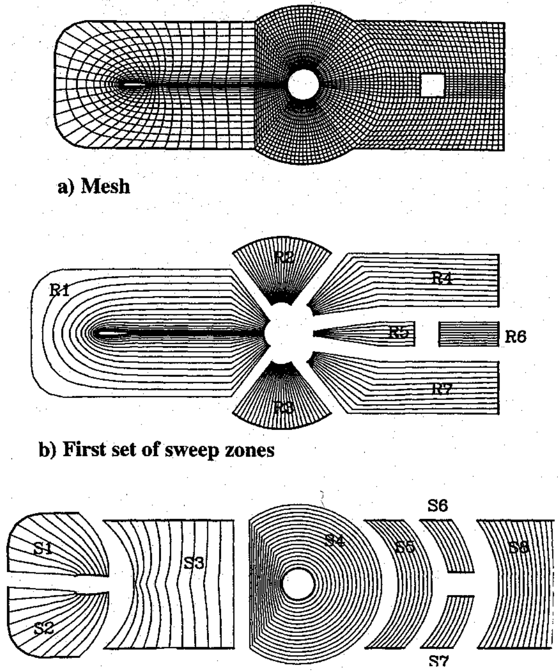


Fig. 5 Time-step refinement study.



a) Mesh

b) First set of sweep zones

c) Second set of sweep zones

Fig. 6 Domain decomposition of case 3.

zones M2, M3, and M4. Although a large number of sweep zones are needed for solving the equations implicitly (a total of 17 zones in both directions), all the sweep zones have identical properties, as listed earlier.

Two sample solutions were obtained. In the first case an analytical solution $T = r^2$ was specified (r is the radial distance) and the source term was adjusted accordingly ($S = 4$). Figure 4a reveals that the analytic solution is restored accurately (the maximal error was less than 0.01%). Yet, it should be realized, that the spatial accuracy is not determined by the ADMZI method, which just serves as a driver, but by the discretization method used in the sweep zones. The discretization method can be replaced (to enhance spatial accuracy, for example), without changing any other aspect of the ADMZI method. This property permits plugging existing single-zone solvers into the ADMZI method and obtaining fully implicit solutions in multizone domains.

In the second solutions, the forward circular-arc body, the half-circular cylinder, and the mushroom-like body were assigned a uniform temperature of $T = 20$, 0, and 50 deg, respectively. The boundary condition on the outer boundaries varied linearly between the upper ($T = 100$ deg) and lower boundaries ($T = 0$ deg). The solution is given in Fig. 4b for a mesh with a total number of over 68,000 points. As expected, a smooth solution is obtained in the entire domain.

One of the great advantages of the ADMZI method is the preservation of the temporal accuracy of the scheme, due to the implicit treatment of the zonal boundaries. Figure 5 shows the results of a time-step refinement study for a point that is in the middle of the zonal boundary between M4 and M5 ($\Delta t_0 = 0.025$ is the coarsest time step). As expected, second-order accuracy in time is maintained, contrary to many other multizone methods, where the explicit zonal boundaries may degrade the temporal accuracy (unless expensive subiterations are performed.[7]

Case 3

In this test case, the problem is solved in the domain shown in Fig. 6a. Three bodies (a NACA 0012 airfoil, a circular cylinder, and a square cylinder) are placed along a horizontal line. The airfoil and the circular cylinder have a (nondimensional) length of 1, whereas the square cylinder is 0.75 wide. The distances between the airfoil and the circular and square cylinders are 4 and 8, respectively. The upstream and downstream boundaries are at a distance of 2 and 11.5

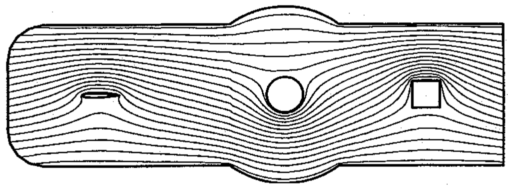


Fig. 7 Sample solution of case 3.

from the leading edge of the airfoil. The outer boundaries are at a distance of 2 from the symmetry line; the circular section of the outer boundary has a radius of 2.5.

A possible mesh is also shown in Fig. 6a. It consists of a C grid around the airfoil, an O grid around the circular cylinder, and an H grid over the square cylinder. Obviously, the distribution of the mesh points is not optimized, but for the purposes of demonstrating the capabilities of the ADMZI method, it is adequate. In this example, three grid topologies are used to show the complexity of problems the ADMZI method can handle without any difficulties. Nevertheless, the global topology of the grid is irrelevant in the case of the ADMZI method, where in each sweep zone one-dimensional sweeps are performed. The two sets of sweep zones are shown in Figs. 6b and 6c. For performing the two stages of the ADMZI method, is sweep zones are required. It should be noticed that the sweep lines in zone R1 start on the upper upstream part of the circular cylinder, go around the airfoil, and end up at the lower part of the circular cylinder.

A sample solution is given in Fig. 7, demonstrating again that using the ADMZI method, a fully implicit solution can be obtained for complex configurations. The airfoil, circular cylinder and the square cylinder are kept at $T = 20, 60$, and 20 deg, respectively. The temperature of the outer boundary varies linearly between $T = 0$ and 80 deg on the lower and upper boundaries. The small oscillations in the solution near the zonal boundary that separates between the O and C grids are a result of discontinuity in the meshsize and in the slope of the sweep lines, Fig. 6.

Concluding Remarks

The alternating direction multizone implicit (ADMZI) method introduced a novel approach for solving implicitly and efficiently time-dependent PDE in complex structured multizone domains. Different sets of sweep zones are generated for each stage of the ADI solution method, contrary to available multizone methods where only one set of zones is employed. The sweep zones are generated so that boundary conditions are always specified at the edges of the sweep lines. All of the sweep zones have identical properties. In the two-dimensional case, they are logically rectangular regions, and physical boundary conditions are specified at the edges of the sweep zones. No zonal boundaries exist along the sweep lines, allowing the fully implicit solution of the discrete equations. Moreover, flux-conserving schemes are easy to implement because zonal boundaries are avoided and the coordinate lines are continuous.

In the present study the ADMZI method was extended to complex grid-topology cases, and the aspects concerning the functioning of the ADMZI driver were addressed. A systematic technique is devised for facilitating the data transfers between the memory and sweep zones and the mapping of the solution back to the memory zones. Only the memory zones are stored permanently, whereas the sweep zones are assembled as needed from the relevant contributions of the memory zones.

The ADMZI approach can be used in a wide range of applications that require the solution of PDE in complex regions. The method can be applied to existing single-zone solvers (based on ADI or other approximate factorization techniques) without major modifications, because the ADMZI method is essentially a driver for solving PDE. Therefore, the form of the PDE or the spatial discretization procedure are relevant only when performing the sweeps inside the sweep zones. The ADMZI method does not intervene in the aspects of the solution; its role is to construct the sweep zones (e.g., copy the data from the contributing memory zones) and copy the solution back to the memory zones. Thus, although in the present paper the ADMZI method was demonstrated for a simple PDE (the unsteady diffusion equation), the results apply to more complex equations, as far as the ADMZI method itself is concerned. Moreover, we anticipate the many available compressible and incompressible viscous flow solvers (that use approximate factorization technique) can be plugged in with only minor modifications.

Although the ADMZI method was elaborated for two-dimensional cases only, the same methodology can be used to extend it to three-dimensional cases. The addition of the third direction may require the definition of three sets of zones, one set for each stage of the ADI method. Otherwise, no changes are required; each zone is a simple hexahedron and the sweep lines start and end on physical boundaries.

Acknowledgments

This work was supported in part by a grant from the Defense Ministry of Israel. The technical monitor was A. Kuritzki.

References

[1]Schwarz, H. A., "Über einige Abbildungsaufgaben," *Journal Riene und Angewandte Mathematik*, Vol. 70, 1869, pp. 105–109.

[2]Rogers, S. E., "On the Use of Implicit Line-Relaxation and Multi-Zonal Computations," AIAA Paper 91-1611, June 1991.

[3]Lai, Y. G., Jingo, Y., and Przekwas, A. J., "An Implicit Multi-Domain Approach for the Solution of Navier–Stokes Equations in Body-Fitted-Coordinate Grids," AIAA Paper 93-0541, Jan. 1993.

[4]Beam, R, and Warming, R. F., "An Implicit Factored Scheme for the Compressible Navier–Stokes Equations," *AIAA Journal*, Vol. 16, No. 4, 1978, pp. 393–402.

[5]Briley, W. R., and McDonald, H., "Solution of the Multidimensional Compressible Navier–Stokes Equations by a Generalized Implicit Method," *Journal of Computational Physics*, Vol. 24, 1977, pp. 372–397.

[6]Rosenfeld, M., Kwak, D., and Vinokur, M., "A Fractional-Step Solution Method for the Unsteady Incompressible Navier–Stokes Equations in Generalized Coordinate Systems," *Journal of Computational Physics*, Vol. 94, May 1991, pp. 102–137.

[7]Rosenfeld, M., and Yassour, Y., "The Alternative Direction Multi-zone Implicit Method," *Journal of Computational Physics*, Vol. 110, Feb. 1994, pp. 212–220.

[8]Douglas, J., and Gunn, J. E., "A General Formulation of Alternating Direction Methods," *Numerische Mathematik*, Vol. 6, Dec. 1964, pp. 428–453.