Kuwait J.Sci., Vol.48 (3), July.2021, pp(1-13)

Approximate Factorization Method Using Alternating Cell Direction Implicit Method: Comparison of Convergence Characteristics Using Basic Model Equations

Ali Ruhşen ÇETE*
Dept. of Aerospace Engineering
Adana Alparslan Türkeş Science and Technology University, Turkey
**Corresponding author: arcete@gmail.com*

Abstract

In this paper, a fast implicit iteration scheme called the alternating cell directions implicit (ACDI) method is combined with the approximate factorization scheme. The use of fast implicit iteration methods with unstructured grids is hardly. The proposed method allows fast implicit formulations to be used in unstructured meshes, revealing the advantages of fast implicit schemes in unstructured meshes. Fast implicit schemes used in structured meshes have evolved considerably and are much more accurate and robust, and are faster than explicit schemes. It is a crucial novel development that such developed schemes can be applied to unstructured schemes. In steady incompressible potential flow, the convergence character of the scheme is compared with the Runge-Kutta order 4 (RK4), Laasonen, point Gauss–Seidel iteration, old version ACDI, and line Gauss–Seidel iteration methods. The scheme behaves like an approximation of the fully implicit method (Laasonen) up to an optimum pseudo-time-step size. This is a highly anticipated result because the approximate factorization method is an approach to a fully implicit formulation. The results of the numerical study are compared with other fast implicit methods (e.g., the point and line Gauss–Seidel methods), the RK4 method, which is an explicit scheme, and the Laasonen method, which is a fully implicit scheme. The study increased the accuracy of the ACDI method. Thus, the new ACDI method is faster in unstructured grids than other methods and can be used for any mesh construction.

Keywords: Alternating cell direction implicit (ACDI) method; approximate factorization method; fast implicit schemes; implicit formulation

1. Introduction

The numerical solution of partial differential equations (PDEs) is the basis of scientific computation applications. Solutions are determined using the discrete forms of the equations with finite cells instead of infinitesimal extents. The discretization approach of equations specifies the classification of solution methods.

One of the classifications is having an implicit or explicit formulation. A competition still exists between the two methods (explicit: (İnan 2017); implicit: (Al-Azemi 2015)). The explicit schemes are more common in unstructured grids ((Caughey *et al.* 1994, Dimitri 2007)). Implicit formulations are more difficult to program than explicit schemes and have a better convergence characteristic.

Being more convergent can be used as advantage if fast implicit methods, such as alternating directions implicit (ADI) method, also known as the Douglas method, are applied (Al-Azemi 2015). These kinds of fast implicit formulations are usually appropriate for structured grids (Dimitri 2007, Çete *et al.* 2006, Çete *et al.* 2007), which comprise ordered finite cells on the solution domain. Their formulations typically use tri-diagonal or penta-diagonal matrix solutions, and these kinds of matrices are relatively easier to solve. Fully implicit schemes require more computational time because they end up with mass matrices for the solution, and the efforts to obtain quicker results with these applications are

generally focused on faster solution methods of mass matrices (Venkatakrishnan 1995).

The alternating cell directions implicit (ACDI) method is a fast implicit scheme that can be used for structured and unstructured grids (Çete *et al.* 2017, Çete *et al.* 2006, Çete *et al.* 2007). This ability is not common for fast implicit schemes. (Dimitri 2007) stated that it is possible to generate grid directions with the edges of triangular elements and that these directions can be used for line implicitness. However, these sweep directions are not unique for an unstructured grid, and they are defined by a method selected by the programmer.

One of the most important studies on the line implicitness of unstructured grids is the study by (Venkatakrishnan 1995). None of the line implicit methods on unstructured meshes defined a unique, natural line. Thus, none of them became widespread in unstructured meshes. (Çete *et al.* 2006, Çete *et al.* 2007, Çete *et al.* 2017) studied the line implicit ACDI method. This study, however, uses the approximate factorization (AF) approach, which is a mathematical approach and is not line implicit.

The ACDI method is offered to combine the advantages of successful convergence characteristics of fast implicit methods with the easier meshing advantage of unstructured grids. The method is inspired from ADI methods by (Meyer 1977, Kellog 1969, Vries 1984, Al-Azemi 2015), but these fast implicit methods can only be used in structured grids. Because one of the objectives of this study is to increase the implicitness level of the ACDI method to the order of AF (Çete *et al.* 2006), the classical AF approach is briefly explained. The main aim of this study is to combine an AF scheme with the ACDI time iteration scheme to increase the implicitness level and accuracy of the previous ACDI studies by (Çete *et al.* 2006, Çete *et al.* 2007, Baş 2007, Çete *et al.* 2017).

Although the main study is conducted using cell-centered finite volume discretization, the AF approach used for this study is also explained for node-based finite difference discretization on structured grids for clearness, and it is aimed to serve a mathematically well-defined approach. The Laplace equation is selected as the model equation for the validation of the method in this study.

2. Numerical Methods

2.1 Approximate Factorization Using ACDI

The method using the cell-centered finite volume approach is explained. In this problem, using the finite volume approach is suitable because the success of the approach is compared with the previous ACDI method, which also uses cell-centered finite volumes. The unsteady heat equation, Equation (1), is used for comparison purposes. This equation can be applied to steady potential flow and heat transfer problems for a steady (converged) solution. Thus, in this study, the steady potential flow around the circle and the heat transfer problems in the wall of infinite length were studied:

$$\frac{\partial T}{\partial t} - \nabla^2 T = 0. \tag{1}$$

Integrating the above equation over the control volume cell shown in Fig. 1 results in the following:

$$\int_A \frac{\partial T}{\partial t} dA - \int_A \nabla^2 T dA = 0. \tag{2}$$

Applying Green's theorem for the second term of the left-hand side (LHS) and writing the first term in discrete form, results in the following:

$$A\frac{\Delta \overline{T}}{\Delta t} - \oint_C \frac{\partial T}{\partial n} ds = 0, \tag{3}$$

where $\overline{T}$ is the mean value of the dependent variable placed at the cell center such that

$$\overline{T} = \frac{1}{A}\int_A T dA.$$

If it is assumed that the fluxes at each cell edge are constant through the edge, where the cell has N edges as depicted in **Figure 1**, the expression can be written in the following form:

$$A\frac{\Delta \overline{T}}{\Delta t} - \sum_{i=1}^{N} \left(\frac{\partial T}{\partial n}\right)_i \Delta s_i = 0, \tag{4}$$

If Equation (4) is written in the following form:

$$A\frac{\Delta \overline{T}}{\Delta t} - \left[\left(\frac{\partial T}{\partial n}\right)_1 \Delta s_1 + \ \ldots \ + \left(\frac{\partial T}{\partial n}\right)_N \Delta s_N\right] = 0, \tag{5}$$

the flux terms are grouped such that the opposite edge fluxes take place in the same group, where T^{n+1} is the next time-step value of the dependent variable at each cell center to be calculated:

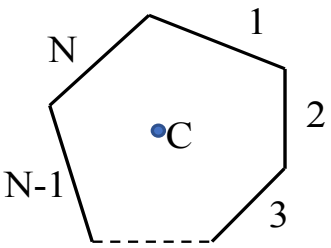

Fig. 1. Control volume with N edges.

$$A\frac{T^{n+1}-T^{n}}{\Delta t}- \left[\begin{array}{c} \left[\left(\frac{\partial T}{\partial n}\right)_1 \Delta s_1 + \left(\frac{\partial T}{\partial n}\right)_{1+\frac{N}{2}} \Delta s_{1+\frac{N}{2}}\right]+ \\ \left[\left(\frac{\partial T}{\partial n}\right)_2 \Delta s_2 + \left(\frac{\partial T}{\partial n}\right)_{2+\frac{N}{2}} \Delta s_{2+\frac{N}{2}}\right]+ \\ \cdots\cdots \\ +\left[\left(\frac{\partial T}{\partial n}\right)_{\frac{N}{2}} \Delta s_{\frac{N}{2}} + \left(\frac{\partial T}{\partial n}\right)_N \Delta s_N\right] \end{array}\right] = 0 \quad (6)$$

It is possible to rearrange Equation (6) such that T^n takes place at the right-hand side (RHS) of the equation because it is a known value at the n^{th} time step:

$$T^{n+1}- \frac{\Delta t}{A}\left[\begin{array}{c} \left[\left(\frac{\partial T}{\partial n}\right)_1 \Delta s_1 + \left(\frac{\partial T}{\partial n}\right)_{1+\frac{N}{2}} \Delta s_{1+\frac{N}{2}}\right]+ \\ \left[\left(\frac{\partial T}{\partial n}\right)_2 \Delta s_2 + \left(\frac{\partial T}{\partial n}\right)_{2+\frac{N}{2}} \Delta s_{2+\frac{N}{2}}\right]+ \\ \cdots\cdots \\ +\left[\left(\frac{\partial T}{\partial n}\right)_{\frac{N}{2}} \Delta s_{\frac{N}{2}} + \left(\frac{\partial T}{\partial n}\right)_N \Delta s_N\right] \end{array}\right] = T^n \quad (7)$$

The fluxes are calculated implicitly; thus, the LHS can be written in the operator form. For the sake of simplicity T^{n+1} is represented as T, and T^n is represented as the RHS from this point forward:

$$\left\{1-\frac{\Delta t}{A}\left[\begin{array}{c} \left[\left(\frac{\partial}{\partial n}\right)_1 \Delta s_1 + \left(\frac{\partial}{\partial n}\right)_{1+\frac{N}{2}} \Delta s_{1+\frac{N}{2}}\right]+ \\ \left[\left(\frac{\partial}{\partial n}\right)_2 \Delta s_2 + \left(\frac{\partial}{\partial n}\right)_{2+\frac{N}{2}} \Delta s_{2+\frac{N}{2}}\right]+ \\ \cdots\cdots \\ +\left[\left(\frac{\partial}{\partial n}\right)_{\frac{N}{2}} \Delta s_{\frac{N}{2}} + \left(\frac{\partial}{\partial n}\right)_N \Delta s_N\right] \end{array}\right]\right\} T = RHS. \quad (8)$$

The form given with Equation (8) is for the application of AF. The equation given above offers a form that is suitable for AF for any polygonal cells that have an even number of edges. It is also possible to use the same factorization for cells that have an odd number of edges by developing simple strategies to convert these cell edges to even-numbered edges. Possible strategies for odd numbered edges are discussed in the latter part of the study. Applying AF to the form is given below:

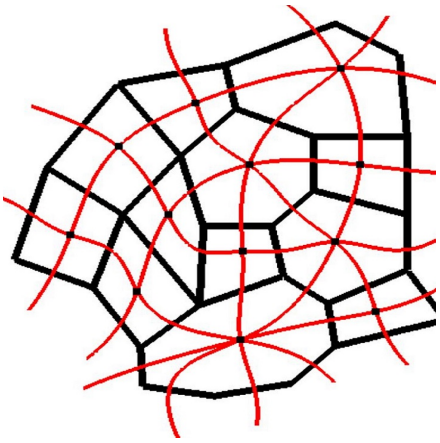

Fig. 2. Cell directions drawn, traveling opposing edges of cells.

$$\begin{array}{c} \left(1-\frac{\Delta t}{A}\left(\frac{\partial}{\partial n}\right)_1 \Delta s_1 + \frac{\Delta t}{A}\left(\frac{\partial}{\partial n}\right)_{1+\frac{N}{2}} \Delta s_{1+\frac{N}{2}}\right) \\ \left(1-\frac{\Delta t}{A}\left(\frac{\partial}{\partial n}\right)_2 \Delta s_2 + \frac{\Delta t}{A}\left(\frac{\partial}{\partial n}\right)_{2+\frac{N}{2}} \Delta s_{2+\frac{N}{2}}\right) \\ \cdots\cdots \\ \left(1-\frac{\Delta t}{A}\left(\frac{\partial}{\partial n}\right)_{\frac{N}{2}} \Delta s_{\frac{N}{2}} + \frac{\Delta t}{A}\left(\frac{\partial}{\partial n}\right)_N \Delta s_N\right) T = RHS. \end{array} \quad (9)$$

Grouping fluxes written for the opposite edges in the factorized equation allows the scheme to be applied using alternating cell directions that are shown on the even-numbered edged polygonal cells in **Figure 2**.

It is possible to write expression (9) as given in equation (10) for each factorized term via assigning semi-solutions that are declared as $T_1, T_2, \ldots, T_{N/2}$, such that

$$\begin{array}{c} \left(1-\frac{\Delta t}{A}\left(\frac{\partial}{\partial n}\right)_1 \Delta s_1 + \frac{\Delta t}{A}\left(\frac{\partial}{\partial n}\right)_{1+\frac{N}{2}} \Delta s_{1+\frac{N}{2}}\right) T_1 = RHS \\ \left(1-\frac{\Delta t}{A}\left(\frac{\partial}{\partial n}\right)_2 \Delta s_2 + \frac{\Delta t}{A}\left(\frac{\partial}{\partial n}\right)_{2+\frac{N}{2}} \Delta s_{2+\frac{N}{2}}\right) T_2 = RHS \\ \cdots\cdots \\ \left(1-\frac{\Delta t}{A}\left(\frac{\partial}{\partial n}\right)_{\frac{N}{2}} \Delta s_{\frac{N}{2}} + \frac{\Delta t}{A}\left(\frac{\partial}{\partial n}\right)_N \Delta s_N\right) T_{\frac{N}{2}} = RHS. \end{array} \quad (10)$$

Writing the operators as shown above and using semi-solutions is performed to obtain tri-diagonal matrices for equation sets that are written for the solution bands shown in **Figure 2.**

The value of T must be calculated after the semi-solutions are obtained. The operators of (10) can be named as follows:

$$\begin{aligned} X_i,\ i = 1,\ 2,\ 3\ldots,\ N/2,\\ X_2,\ X_3,\ \ldots,\ X_{N/2}T\ &= T_1,\\ X_1,\ X_3,\ \ldots,\ X_{N/2}T\ &= T_2,\\ &\ldots\\ X_1\ldots X_j\ldots X_{N/2}T\ &= T_i\ \ i\neq j,\\ &\ldots\\ X_1\ldots X_j\ldots X_{N/2-1}T\ &= T_{N/2}, \end{aligned} \tag{11}$$

and the expression $X_1.\ .X_j..X_{\frac{N}{2}}\ \ T{=}T_i,\ \ i{\neq}j$ is equal to the following form before the factorization:

$$\begin{aligned} &X_1\ldots X_j\ldots X_{N/2}T\ \cong\\ &\left\{\begin{array}{c} 1-\left[\frac{\Delta t}{A}\Delta s_1\left(\frac{\partial}{\partial n}\right)_1+\frac{\Delta t}{A}\Delta s_{1+\frac{N}{2}}\left(\frac{\partial}{\partial n}\right)_{1+\frac{N}{2}}\right]+\\ \ldots\\ \left[\frac{\Delta t}{A}\Delta s_i\left(\frac{\partial}{\partial n}\right)_i+\frac{\Delta t}{A}\Delta s_{i+\frac{N}{2}}\left(\frac{\partial}{\partial n}\right)_{i+\frac{N}{2}}\right]+\\ \ldots\\ \left[\frac{\Delta t}{A}\Delta s_{\frac{N}{2}}\left(\frac{\partial}{\partial n}\right)_{\frac{N}{2}}+\frac{\Delta t}{A}\ \Delta s_N\left(\frac{\partial}{\partial n}\right)_N\right] \end{array}\right\}T. \end{aligned} \tag{12}$$

If the expressions in (11) are summed using the assumption given in (12), the summation results in the following:

$$\frac{N}{2}T-\left(\frac{N}{2}-1\right)\sum_{i=1}^{N}\frac{\Delta t}{A}\Delta s_i\left(\frac{\partial T}{\partial n}\right)_i=\sum_{i=1}^{N}T_i. \tag{13}$$

Rearranging Equation (13) yields the following:

$$T+\left(\tfrac{N}{2}-1\right)\underbrace{\left(T-\sum_{i=1}^{N}\frac{\Delta t}{A}\Delta s_i\left(\frac{\partial T}{\partial n}\right)_i\right)}_{RHS}=\textstyle\sum_{i=1}^{N}T_i \tag{14}$$

Then, the value of T can be calculated using the simple expression below:

$$T=\sum_{i=1}^{N}T_i-\left(\frac{N}{2}-1\right)RHS. \tag{15}$$

2.2 Von Neumann Stability Analysis

The Von Neumann stability analysis is performed assuming Cartesian cells. A sample Cartesian cell and its neighbors are depicted in **Figure 3.** with the notation used for the analysis. The analysis is conducted using inverse distance weighting flux calculation with AF. Rewriting Equation (10) for the quadrilateral cells results in the following:

$$\begin{aligned} T_1-\tfrac{\Delta t}{A}\Delta s_1\left(\tfrac{\partial T_1}{\partial n}\right)_1-\tfrac{\Delta t}{A}\Delta s_3\left(\tfrac{\partial T_1}{\partial n}\right)_3&=RHS,\\ T_2-\tfrac{\Delta t}{A}\Delta s_2\left(\tfrac{\partial T_2}{\partial n}\right)_2-\tfrac{\Delta t}{A}\Delta s_4\left(\tfrac{\partial T_2}{\partial n}\right)_4&=RHS. \end{aligned} \tag{16}$$

The low order flux terms can be inserted in the following form:

$$\left(\frac{\partial T_1}{\partial n}\right)_1=w_{1_{N_1}}T_{1_{N_1}}\ w_{1C}\ T_{1C}, \tag{17}$$

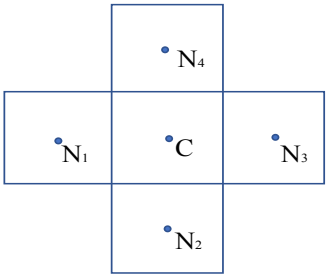

Fig. 3. Example a Cartesian cell and its neighbors.

where $w_{1_{N_1}}$ and w_{1C} are the weights used to calculate the flux term from the cell with the cell center N_1 illustrated in **Figure 3.**The weights for the Cartesian cells are all equal to $1/\Delta s$. Then, the flux terms can be inserted into Equation (17) in the following form:

$$\begin{aligned} \left(\tfrac{\partial T_1}{\partial n}\right)_1&=\tfrac{1}{\Delta s}\ (T_{1_{N_1}}-T_{1C}),\\ \left(\tfrac{\partial T_1}{\partial n}\right)_3&=\tfrac{1}{\Delta s}\ (T_{1_{N_3}}-T_{1C}),\\ \left(\tfrac{\partial T_2}{\partial n}\right)_2&=\tfrac{1}{\Delta s}\ (T_{2_{N_2}}-T_{2C}),\\ \left(\tfrac{\partial T_2}{\partial n}\right)_4&=\tfrac{1}{\Delta s}\ (T_{2_{N_4}}-T_{2C}). \end{aligned} \tag{18}$$

Inserting the flux terms into the equations given in (16) and using the spatial and temporal indices results in the following:

$$\begin{aligned} \left(1+2\tfrac{\Delta t}{A}\right)T_{1\ i,j}^{n+1}-\tfrac{\Delta t}{A}\ T_{1\ i-1,j}^{n+1}-\tfrac{\Delta t}{A}\ T_{1\ i+1,j}^{n+1}&=T_{i,j}^{n},\\ \left(1+2\tfrac{\Delta t}{A}\right)T_{2\ i,j}^{n+1}-\tfrac{\Delta t}{A}\ T_{2\ i,j-1}^{n+1}-\tfrac{\Delta t}{A}\ T_{2\ i,j+1}^{n+1}&=T_{i,j}^{n}. \end{aligned} \tag{19}$$

To determine the growth rate of error at each time iteration, $T_{i,j}^{n}=\xi^n e^{ikx}e^{iky}$ is inserted into the first expression given in (19):

$$\begin{aligned} &\left(1+2\tfrac{\Delta t}{A}\right)\xi^{n+1}e^{ikx}e^{iky}-\ \tfrac{\Delta t}{A}\ \xi^{n+1}e^{ik(x-\Delta x)}e^{iky}-\\ &\tfrac{\Delta t}{A}\ \xi^{n+1}e^{ik(x+\Delta x)}e^{iky}\ =\ \xi^n e^{ikx}e^{iky} \end{aligned} \tag{20}$$

Dividing both the RHS and LHS of (20) by $\xi^n e^{ikx}e^{iky}$ yields the following:

$$\left(1+2\frac{\Delta t}{A}\right)\xi-\ \frac{\Delta t}{A}\ \xi e^{-ik\Delta x}\ -\ \frac{\Delta t}{A}\ \xi e^{ik\Delta x}=\ 1. \tag{21}$$

Rearranging (21) gives the following:

$$\xi\left[\left(1+2\frac{\Delta t}{A}\right)-\ \frac{\Delta t}{A}(e^{-ik\Delta x}+\ e^{ik\Delta x})\right]=\ 1. \tag{22}$$

Thus, the growth factor is as follows:

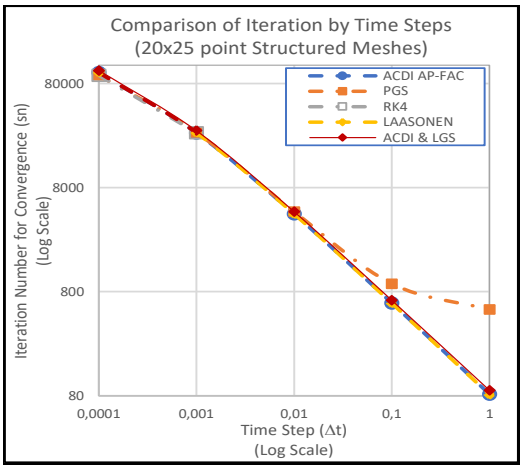

Fig. 4. Iteration numbers required for the convergence versus time-step size (s) using different methods for the 500-element structured grid (potential flow around cylinder).

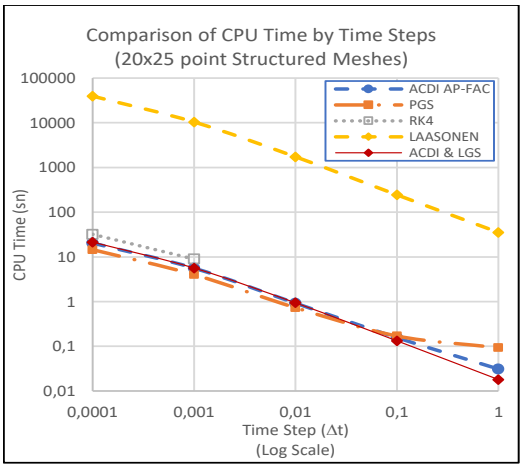

Fig. 5. CPU time required for convergence versus time-step size (s) using different methods for the 500-element structured grid (potential flow around cylinder).

$$\xi = \frac{1}{\left(1 + 2\frac{\Delta t}{A}\right) - 2\frac{\Delta t}{A}(cosk\Delta x)}. \tag{23}$$

Similarly, the growth factor for the second expression of (19) is the following:

$$\zeta = \frac{1}{\left(1 + 2\frac{\Delta t}{A}\right) - 2\frac{\Delta t}{A}(cosk\Delta y)}. \tag{24}$$

The stability criterion is $|\xi| \leq 1$, $|\zeta| \leq 1$ because

$$1 \leq cosk\Delta x \leq 1\,,\ -1 \leq cosk\Delta y \leq 1. \tag{25}$$

Both growth factors satisfy the stability criterion. This scheme is unconditionally stable. In addition, unconditional stability is at the full limit and can be conditionally stable in curved structured and unstructured grids in the Cartesian grid (i.e., in the most ideal case). The scheme can be converted to Crank-Nicolson AF with a small change, and the literature has indicated that Crank-Nicholson AF is much more stable than normal AF (Ballhaus *et al.* 1977).

3. Results and Discussion

3.1 Comparison of Convergence Characteristics Using Steady Laplace Equations

In this section, the method is applied to a steady problem. Incompressible potential flow around a cylinder is solved to observe the convergence behavior of the method for elliptical equations. The potential flow equation is solved using ACDI with AF, ACDI, line Gauss–Seidel (LGS), Runge-Kutta order 4 (RK4), Laasonen, and point Gauss–Seidel (PGS) iteration methods. The steady Laplace equation used for the solution of the potential equation is given below:

$$\frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial y^2} = 0, \tag{26}$$

where ψ represents the stream function in the equation given above. The velocity component in the x- and y-directions are the following:

$$u = \frac{\partial \psi}{\partial y} \ \ and \ \ v = -\frac{\partial \psi}{\partial x}. \tag{27}$$

The equation is solved using the pseudo-time step. The equation can be rewritten as an unsteady equation:

$$\frac{\partial \psi}{\partial t} - \frac{\partial^2 \psi}{\partial x^2} - \frac{\partial^2 \psi}{\partial y^2} = 0. \tag{28}$$

The solution converges to a steady state if the pseudo-time step is used, and the solutions before the convergence are meaningless in terms of unsteadiness. Potential flow around a two-dimensional (2D) cylinder with a diameter of 1 m is solved for the comparisons. The free stream velocity is taken as 1 m/s, and the analytic solution for the stream function is given below:

$$\psi\,(x, y) = y - \frac{0.25y}{x^2 + y^2}. \tag{29}$$

3.2 Comparison of Convergence Characteristics Using a Structured Grid

The iteration numbers and central processing unit (CPU) times of the methods to reach a previously

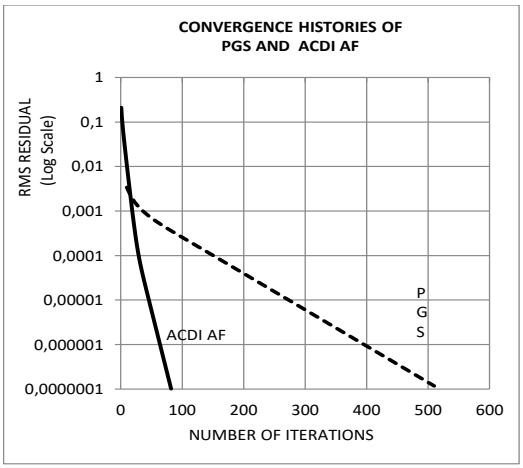

Fig. 6. Convergence histories of ACDI AF and PGS methods with Δt=1 with the 500-element structured grid (potential flow around cylinder).

defined root mean square residual are obtained using different time-step sizes for the ACDI-AF, AF, ACDI, LGS, PGS, Laasonen, and RK4 methods. The residual value to be reached for the convergence is selected as 10^{-7}, and the root mean square residual is defined as follows:

$$RMS\ Residual = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\psi_i^{n+1} - \psi_i^n)^2}, \quad (30)$$

where N represents the number of elements for cell-centered calculations and the number of nodes for node-based calculations, n is the time indices, and i is the element or node indices.

The Dirichlet boundary condition is used at the far field of the domain. Because the problem is symmetrical, half of the domain is used for the solution, and again, the Dirichlet boundary condition is used for the symmetry and wall faces, where $\psi = 0$. In **Figure 4**, the required iteration number using different time-step sizes is plotted for the simple ACDI-AF, AF, ACDI, PGS, Laasonen, and RK4 methods using a 500-element structured grid.

When compared with the other methods, the RK4 method has poor stability. The AF method results exactly match the ACDI-AF results, and the LGS method results match the ACDI method. This is a feature of the ACDI method. In structured grids, the ACDI method formulation is the same as the LGS formulation. It is the same for ACDI-AF and AF methods. The required time-step size for the convergence is larger than the other methods because small step sizes are required for

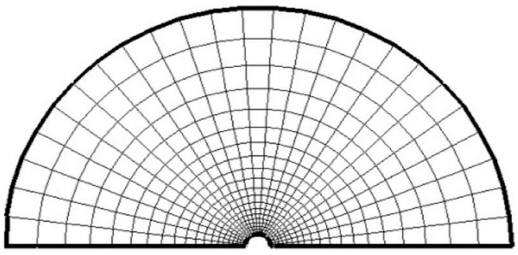

Fig. 7. A 500-element structured grid for comparing convergency.

stability. The required iteration number converges to a certain number with increased time-step sizes for both the PGS and Laasonen methods, and both of the methods remain stable with the use of relatively larger time-step sizes. The ACDI-AF method has an optimum value of time-step size that is different than the Laasonen and PGS methods. The PGS method leaves the trend before the ACDI AF and converges to a certain iteration number value, whereas the number of time steps required increases for the ACDI-AF method after the optimum time-step size value. Therefore, the ACDI AF is a better approximation of the fully implicit scheme than the PGS method until the optimum time-step size. The Laasonen and ACDI-AF methods have the same convergence behavior as expected.

In **Figure 5**, the CPU time versus the time-step size plots are given for the methods. The Laasonen method loses its advantage of fewer required iteration numbers. The method inverts a mass matrix for each iteration step, and this operation increases the requirement of the CPU time for convergence.

The convergence histories of the PGS and ACDI-AF methods with the minimum iteration numbers required are plotted in **Figure 6**. In **Figure 7** and **8**, a 20x25 structured grid and the stream function contours obtained using ACDI AF are given. The L0, L1, L2, and L∞ norm errors are calculated using the following formulation:

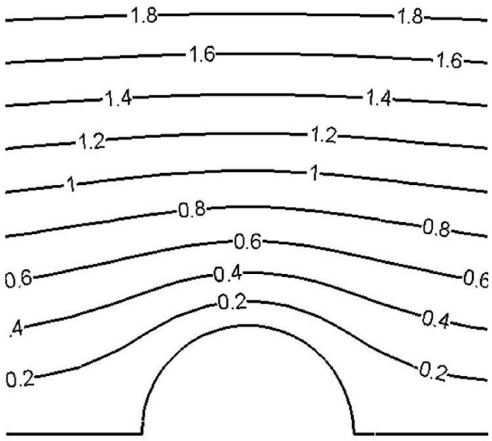

Fig. 8. Stream function contours using ACDI AF with 500-element structured grid (potential flow around cylinder).

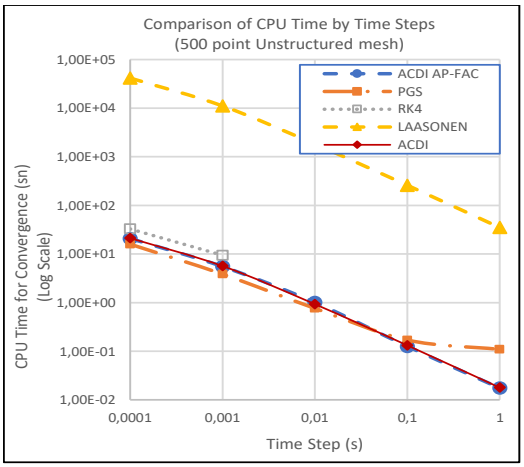

Fig. 10. CPU time for the convergence versus time step using different methods for the 500-element unstructured grid (potential flow around cylinder).

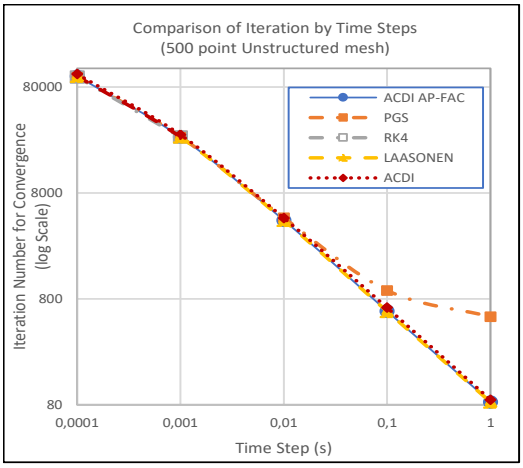

Fig. 9. Iteration numbers for the convergence versus time step using different methods for the 500-element unstructured grid (potential flow around cylinder).

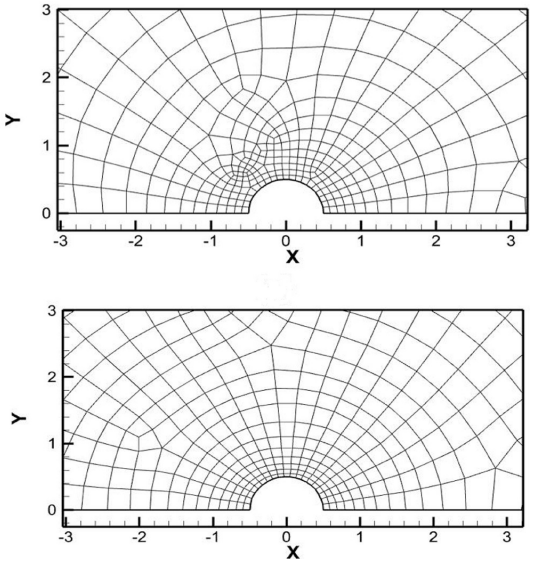

Fig. 11. 500-element unstructured and hybrid locally structured grid views.

$$
\begin{aligned}
L0\ Error &= \left| \psi_{\substack{Node \\ Analytical}} - \psi_{\substack{Node \\ Numerical}} \right| \\
L1\ Error &= \frac{\Sigma^{N} \left| \psi_{\substack{Node \\ Analytical}} - \psi_{\substack{Node \\ Numerical}} \right|}{N} \\
L2\ Error &= \sqrt{\frac{\Sigma^{N} \left(\psi_{\substack{Node \\ Analytical}} - \psi_{\substack{Node \\ Numerical}} \right)^{2}}{N}} \\
L3\ Error &= Max \left| \psi_{\substack{Node \\ Analytical}} - \psi_{\substack{Node \\ Numerical}} \right|
\end{aligned}
\tag{31}
$$

3.3 Comparison of Convergence Characteristics Using Unstructured and Hybrid Grids

The unstructured and hybrid (locally structured) grids with 500 elements used for calculations are presented in **Figure11**. The iteration number required versus the time-step size for the ACDI-AF, ACDI, Laasonen, PGS, and RK4 methods are given for the solution of the Laplace equation with the unstructured grid in **Figure 9**. The curves obtained for the methods have behavior similar to the structured grid case results. In (Δt = 1), the ACDI-AF method appears to provide much faster results than the PGS method.

The hybrid grid is an unstructured grid using a

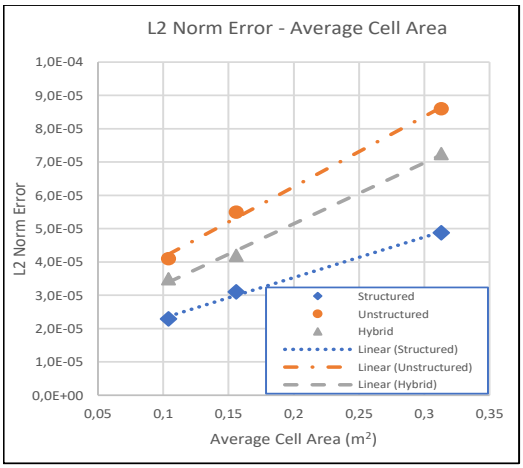

Fig. 12. In structured, unstructured and hybrid meshes, the variation of L2 norm errors of the ACDI-AF method according to the average cell area for $\Delta t = 0.001$ (potential flow around cylinder).

Table 1. L0 and L∞ errors of ACDI-AF methods for $\Delta t = 0.001$.

GRID	**Str 20x25**	**Str 30x35**	**Str 35x43**
L∞	0.0060	0.0044	0.0040
L0 (0,0.6)	0.0022	0.0010	0.0007
GRID	**Uns 500**	**Uns 1000**	**Uns 1500**
L∞	0.0140	0.0120	0.0130
L0 (0,0.6)	0.0030	0.0015	0.0009
GRID	**Hyb 500**	**Hyb 1000**	**Hyb 1500**
L∞	0.0600	0.0099	0.0090
L0 (0,0.6)	0.0026	0.0012	0.0008

structured grid in its partial regions. In **Figure 10**, the Laasonen method uses a greater CPU time per iteration, and the RK4 method requires relatively smaller time-step sizes; thus, both methods seem disadvantageous for obtaining convergent results faster.

The convergence of the ACDI method and CPU time characteristics are similar to those of ACDI AF. Another comparison approach addresses how the method changes its sensitivity in different grid types. To examine the accuracy of methods and the effect of solution mesh types on accuracy, solutions were obtained for structured, unstructured, and hybrid solution meshes with 500, 1000, and 1500 elements in the same solution area. Thus, solution meshes with an average cell area of approximately 0.1, 0.15, and 0.3 m^2 were in question. In **Figure 12**, **13**, and **14**, the variations of L0, L1, and L2 norm errors according to the average cell areas for structured, unstructured, and hybrid

Table 2. CPU time and iteration number required for the methods; 20x25 structured and 500 unstructured cells in potential flow around cylinder.

	ACDI AP-FAC	**PGS**	**RK4**	**LAASONEN**	**ACDI & LGS**
Δt	**Structured 20x25 CPU Time (s)**				
0.0001	20.32	14.50	32.00	39,438.42	21.38
0.001	5.59	4.00	8.92	10,388.59	5.63
0.01	0.92	0.73	-	1732.34	0.94
0.1	0.16	0.17	-	244.10	0.13
1	0.03	0.09	-	35.35	0.02
Δt	**Structured 20x25 Iteration Number**				
0.0001	101787	95929	95854	101780	106876
0.001	26812	27100	26962	26809	28153
0.01	4471	4696	-	4465	4695
0.1	623	955	-	621	662
1	83	539	-	82	90
Δt	**Unstructured 500 Point CPU Time (s)**				
0.0001	20.82	15.90	32.70	41,640.06	21.19
0.001	5.51	3.85	9.42	11,117.19	5.66
0.01	1.01	0.76	-	1822.81	0.93
0.1	0.13	0.17	-	259.11	0.13
1	0.02	0.11	-	35.07	0.02
Δt	**Unstructured 500 Point Iteration Number**				
0.0001	100924	99710	99638	100920	105970
0.001	26945	27120	26974	26939	28292
0.01	4418	4662	-	4420	4639
0.1	612	961	-	609	659
1	85	543	-	85	89

solution meshes are presented, respectively. Linear behavior in the mean cell area plot of the error is evidence of a second-order precision because, in equally spaced cells, the area is proportional to Δx^2, and the error that is proportional to Δx^2 represents the second-order precision. In **Figure12**, **13**, and **14**, L0, L1, and L2 norm errors are proportional to the average cell area. This indicates that ACDI AF presented in these charts has second-order precision. As expected, these graphics reveal that errors in the structured solution meshes are lower than errors in unstructured solution meshes. As expected, errors of hybrid mesh results are among the errors of structured and unstructured mesh results. One of the key features of ACDI methods is that it practically supports hybrid solution meshes. The different norm errors of ACDI-AF methods are listed in **Table 1**

3.4 Comparison of Convergence Characteristics of Methods Using Structured and Unstructured Grids

Table 2 presents the iteration numbers and CPU times of these studies for different step times. In **Figure 15** and **16**, the L2 norm error results of different methods are given for structured and unstructured meshes. These graphs demonstrate

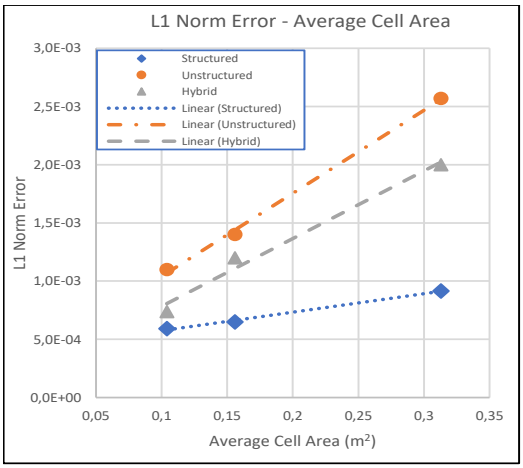

Fig. 13. In structured, unstructured, and hybrid meshes, the variation of L1 norm errors of the ACDI-AF method according to the average cell area for $\Delta t = 0.001$ (potential flow around cylinder).

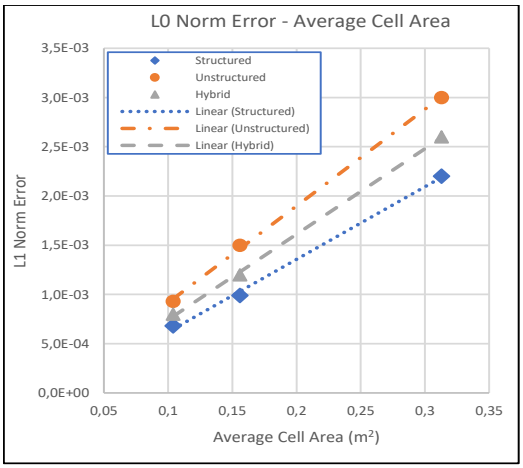

Fig. 14. In structured, unstructured, and hybrid meshes, the variation of L0 norm errors (on $x = 0$, $y = 0.6$) of the ACDI-AF method according to the average cell area for $\Delta t = 0.001$ (potential flow around cylinder).

that all methods are second-order accurate. The most accurate results in both structured and unstructured meshes appear to be the ACDI-AF and Laasonen methods. In the results, the Laasonen method is slightly more accurate. Naturally, the accuracy is slightly reduced in unstructured meshes, but the most reasonable method in unstructured mesh is ACDI AF. Laasonen exhibits the same performance but takes much more CPU time, and as the number of elements increases, it increases logarithmically.

At this stage, it is necessary to indicate where structured and unstructured solution meshes are used. The results reveal that structured solution meshes are a great advantage in many aspects. However, the structured mesh generation is often exceedingly difficult in complex geometries and is sometimes impossible. In this case, unstructured solution meshes that can be applied quite easily are used.

The Delaunay mesh production method (Rebay 1991) used in this study is a quite common and easy-to-use method. As a result, in cases where structured meshes are not used or are not available, unstructured meshes are used. In another case, this method improves performance in hybrid methods using structured meshes in sensitive areas. The ACDI methods are crucial in supporting a hybrid mesh structure automatically.

3.5 Investigation of Unsteady Performance of Methods Using the Heat Transfer Problem in the Infinite Wall

The ACDI-AF method is to increase the implicitness in the partial differential solution. Implicitness is most important in unsteady problems. Therefore, a problem that has simple and analytical results should be defined. The unsteady heat conduction problem on a rectangular plate is solved for the comparison. The infinite wall boundary condition is used at horizontal boundaries, and the constant temperature boundary condition is applied at the vertical walls. The representation of this 1D problem and the boundary conditions and sizes are given in **Figure 17**. The unsteady heat conduction equation is given below:

$$\frac{\partial T}{\partial \tau} - a\nabla^2 T = 0, \tag{32}$$

where $\boldsymbol{a}$ is the heat conduction coefficient. The analytical solution for $a = 1$ of this problem is provided below:

$$T(x, y, \tau) = \sum_{n=1}^{\infty} A_n e^{-\lambda_n \tau} \cos(\lambda_n x)$$

$$A_n = \frac{4\sin\lambda_n}{2\lambda_n + \sin 2\lambda_n} \tag{33}$$

$$\lambda_n \tan\lambda_n = 1$$

Since there is a Neuman boundary condition in Y = 0 and Y = 1, the 1-dimensional problem

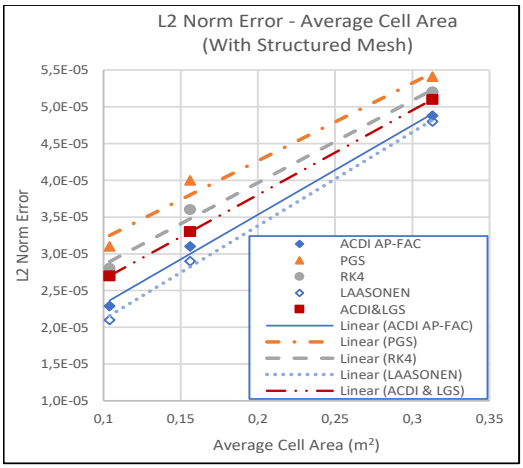

Fig. 15. In structured meshes, the variation of L2 norm errors of different methods according to the average cell area for $\Delta t = 0.001$ (potential flow around cylinder).

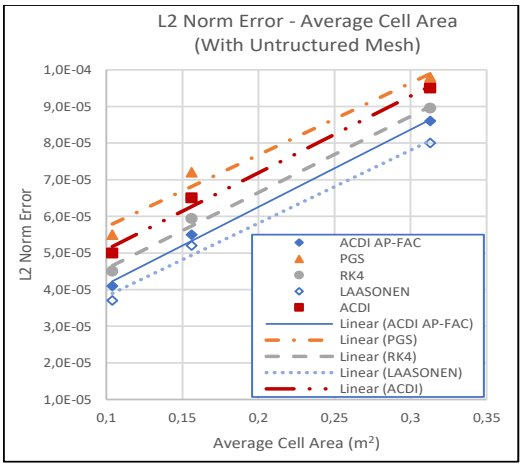

Fig. 16. In unstructured meshes, the variation of L2 norm errors of different methods according to the average cell area for $\Delta t = 0.001$ (potential flow around cylinder).

is actually solved in 2 dimensions. Naturally, temperature values do not change much during Y. It is also obvious that the problem converges to a situation where the temperature is the same everywhere. Consequently, only solutions at τ = 0.2 were compared to make the comparison clearer.

In **Figure 18**, in the solution on the center line of the 17x17 structured element mesh, the errors of the methods are shown comparatively at τ = 0.2. Similarly, in **Figure 19**, in the solution on the center line of the 297-element unstructured mesh, the errors of the methods are shown comparatively at τ = 0.2. In the structured mesh, ACDI AF provides the same results as the Laasonen method. In this case, the only difference between the two formulations is the factorization error, which is minute. The RK4 method is best in terms of precision, but it is an explicit formulation and is conditionally stable, operating at a very small step time (Δt). Even under these conditions (low Δt) ACDI AF is more effective than RK4 and results in the same accuracy but more quickly.

This also applies to unstructured mesh. Although the Laasonen and RK4 methods are applied easily on unstructured meshes, CPU time of these methods are greater than others. The Laasonen CPU time problem is impossible in many problems because the number of meshes increases logarithmically. The same situation becomes impossible in RK4 with the condition of stability, using a very small step time. The ACDI AF seems to be a more effective method,

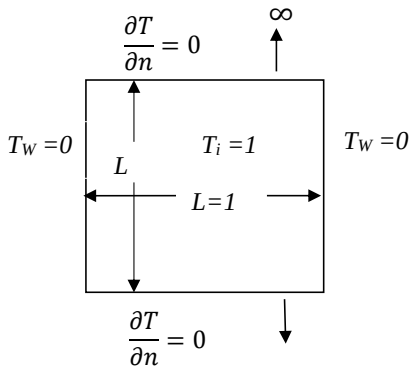

Fig. 17. Schematic representation of the problem of heat transfer in the infinite wall.

especially in unsteady problems.

In **Table 3**, the CPU times of the methods for 10,000 iterations are given. Since Lasonen method involves solving full matrix equations, its CPU time is considerably higher than the others. Therefore, Laasonen CPU time is proportional to the square of the number of mesh elements, although it is known in other methods that the CPU time is proportional to the number of meshes.

4. Conclusion

This study aims to improve the ACDI method by increasing the overall performance with the implicitness characteristic. The scheme was tested on steady and unsteady problems.

The ACDI methods have a basic fast implicit scheme and behave exactly like these schemes in

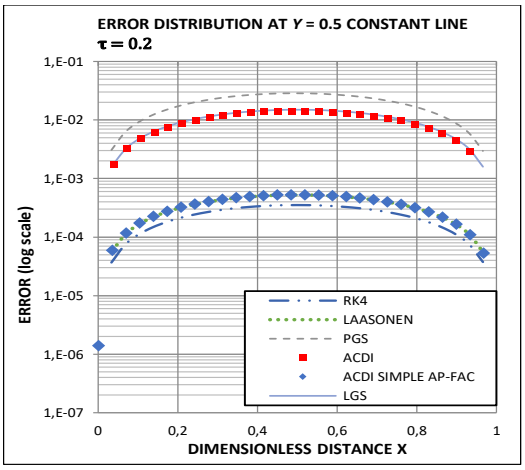

Fig. 18. Error distribution through $y = 0.5$ constant line at $\tau = 0.2$, 17x17 elements structured grid, $\Delta\tau = 0.0001$ (heat transfer in infinite wall).

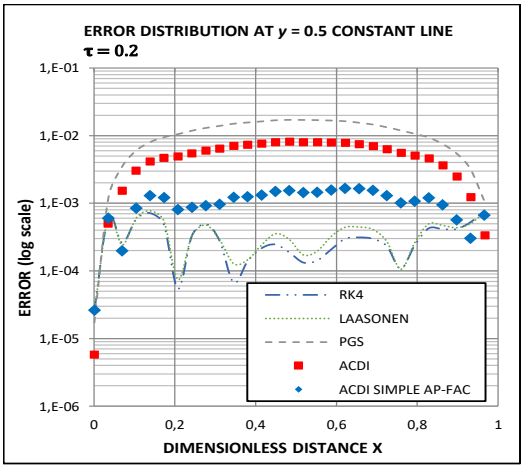

Fig. 19. Error distribution through $y = 0.5$ constant line at $\tau = 0.2$; 297-element unstructured grid, $\Delta\tau = 0.0001$ (heat transfer in the infinite wall).

structured meshes. The basic fast implicit scheme of the old ACDI method is the LGS method. The basic fast implicit scheme of the ACDI-AF method developed in this study is the AF method. In both schemes, it acts like the basic fast implicit schemes in structured meshes.

The ACDI-AF method is observed to behave like an approximation of the fully implicit method (Laasonen method), like a basic AF method. Moreover, the ACDI-AF method exhibits this feature in unstructured meshes.

The ACDI-AF method provides a serious improvement in hybrid meshes (locally structured), automatically detecting structured regions in the unstructured meshes. Thus, hybrid meshes exhibit higher performance. Fast implicit methods

Table 3. CPU times of 10,000 iterations for the methods; 17x17 structured and 297 unstructured elements of heat transfer in the infinite wall.

METHOD USED	CPU time (s) for 10,000 iterations for 17x17 str. mesh	CPU time (s) for 10,000 iterations for 297 pt. uns. mesh
RK4	3.520	3.832
LAASONEN	61.588	62.459
PGS	2.698	2.780
ACDI	3.069	3.052
ACDI AF	2.801	2.828
LGS	3.010	-

are more advantageous than explicit schemes in structured meshes. Unstructured meshes should be used because structured meshes cannot be applied in difficult geometries. This method provides a solution to the problem of not applying fast implicit schemes in unstructured meshes.

The ACDI AF always results in a lower CPU time than the PGS method. Thus, it is more advantageous than the PGS method. The accuracy of an advanced high-order explicit scheme, such as RK4, is slightly more accurate than ACDI AF but is slower and conditionally stable. Stability conditions are very weak; therefore, it is impractical. The theoretical stability analysis of the ACDI-AF method indicates that it is unconditionally stable in the Cartesian mesh structure. However, the stability criterion is at the full limit, which leads to the conclusion that it is conditionally stable when it moves away from the ideal Cartesian structure. This study could not be extended to these limit values. In numerical tests, the stability in structured meshes is preserved. In unstructured meshes, instability has occurred at very high step times. This issue has not been emphasized much because this problem can be solved with a formulation improvement (Ballhaus *et al.* 1977). In this study, only the elliptical (steady potential flow) and parabolic (unsteady heat conduction) PDE solution performance of ACDI AF was examined. Other related research has also investigated the performance of hyperbolic PDEs. As a result, the ACDI method can be applied as a high-order fast implicit scheme for all types of grids. Thus, complex PDEs, such as Navier-Stokes can be solved using ACDI schemes. The use of fast and robust implicit solvers in hybrid meshes is also one of the important research topics in computational fluid mechanics.

References

Abrashin, V.N., & Dzyuba I.A. (1994). An alternating direction method for solving multidimensional problems of mathematical physics in domain with curvilinear boundary Differential Equations, **30**:1082-1087.

Al-Azemi, F. (2015). An improvement of the Douglas scheme for the Black-Scholes equation. Kuwait Journal of Science, **42**, 3:105-119.

Ballhaus, W.F., Jameson, A., & Albert, J. (1977). Implicit approximate-factorization schemes for the efficient solution of steady transonic flow problems. NASA/Ames Research Center, January.

Bas, O. (2007). Development of an incompressible Navier-Stokes solver with alternating cell direction implicit method on structured and unstructured quadrilateral grids. MSc thesis, Middle East Technical University. Ankara, Turkey.

Caughey, D.A. & Hafez, M.M. (1994). Frontiers of computational fluid dynamics. John Willey & Sons Press, New York.

Cete, A.R., Bas¸, O., Mengi, S., Tuncer, I.H., & Kaynak, U. (2017). A novel alternating cell directions implicit method for the solution of incompressible Navier Stokes equations on unstructured grids. Journal of Applied Fluid Mechanics, **10**(6):1561-1570.

Cete, A.R. (2004). Alternating cell directions implicit method. Ph.D. thesis, ˙Istanbul Technical University, ˙Istanbul, Turkey.

Cete, A.R. & Kaynak, U. (2006). A new approximate factorization method suitable for structured and unstructured grids. AIAA Paper 2006-3789, 9th AIAA/ASME Joint Thermophysics and Heat Transfer Conference, San Francisco, California, USA.

Cete, A.R., Yukselen, M.A., & Kaynak, U. (2007). A unifying grid approach for solving potential flows applicable to structured and unstructured grid configurations. Computers & Fluids, textbf **37**(1):35-50.

Dimitri, J.M. (2007). Unstructured mesh discretization and solvers for computational aerodynamics. AIAA Computational Fluid Dynamics Conference, **18**, pp. 1-28.

Gibbons, A. (1985). Algorithmic graph theory. Cambridge University Press, New York, NY, USA.

Hassan, O., Morgan, K., & Peraire, J. (1989). An adaptive implicit/explicit finite element scheme for compressible high speed flows. AIAA Paper, **89**-0363.

Hoffman, K.A., & Chiang, S.J. (2000). Computational Fluid Dynamics Vol. 1, A Publication of Engineering Education, Kansas, USA.

Inan, B. (2017). Finite difference methods for the generalized Huxley and Burgers-Huxley equations. Kuwait Journal of Science, **44**, (3):20-27.

Kellog, R.B. (1969). A nonlinear alternating direction method. Mathematics of Computation, **9**:23-27.

Meyer, G.H. (1977). An alternating directions method for multi-dimensional parabolic free surface problems. International Journal for Numerical Methods in Engineering, **11**:741-752.

Rebay, S. (1991). Efficient unstructured mesh generation by means of Delaunay triangulation and Bowyer-Watson algorithm Journal of Computational Physics, **106**:125 138.

Venkatakrishnan, V. (1995). Implicit schemes and parallel computing in unstructured grid CFD. NASA ICASE,

Lecture Notes Prepared for 26th Computational Fluid Dynamics Lecture Series Program of VKI, Belgium.

Vries, H.B.D. (1984) A comparative study of ADI splitting methods for parabolic equations in two space dimensions. Journal of Computational and Applied Mathematics, **10**:179-193.

Submitted: 06/02/2020
Revised: 13/08/2020
Accepted: 15/09/2020
DOI: 10.48129/kjs.v48i3.9137