Alternating Cell Direction Implicit Method using Approximate Factorization on Hybrid Grids

Ali Ruhşen Çete¹,* and Oğuz Kaan Onay²

¹ Adana Alparslan Türkeş Science and Technology University, Adana, Turkey
² Middle East Technical University, Ankara, Turkey

Received 5 February 2020; Accepted (in revised version) 2 May 2021

Abstract. In this study, a novel fast-implicit iteration scheme called the alternating cell direction implicit (ACDI) method is combined with the approximate factorization scheme. This application aims to offer a mathematically well-defined version of the ACDI method and to increase the accuracy of the iteration scheme used for the numerical solutions of partial differential equations. The ACDI method is a fast-implicit method that can be used for unstructured grids. The use of fast implicit iteration methods with unstructured grids is not common in the literature. The new ACDI method has been applied to the unsteady diffusion equation to determine its convergence and time-dependent solution ability and character. The numerical tests are conducted for different grid types, such as structured, unstructured quadrilateral, and hybrid polygonal grids. Second, the ACDI was applied to the unsteady advection-diffusion equation to understand the time-dependent and progression capabilities of the presented method. Third, a full potential equation solution is created to understand the complex flow solving ability of the presented method. The results of the numerical study are compared with other fast implicit methods, such as the point Gauss–Seidel (PGS) and line Gauss–Seidel (LGS) methods and the fourth-order Runge-Kutta (RK4) method, which is an explicit scheme, and the Laasonen method, which is a fully implicit scheme. The study increased the abilities of the ACDI method. Due to the new ACDI method, the approximate factorization method, which is used only in structural grids that are known to be advantageous, can be applied to any mesh structure.

AMS subject classifications: 68Q01, 65Y04, 65Y20, 35D99, 74G15

Key words: Approximate factorization, ACDI method, fast implicit schemes.

1 Introduction

The numerical solution to partial differential equations is the basis of scientific computation applications. Solutions are found using discrete forms of equations with finite
instead of infinitesimal cells. The discretization approaches of equations specify the classification of the solution methods. One of the classifications is having an implicit or explicit formulation. Implicit formulations result in linear sets of equations in matrix form, whereas explicit formulations directly use the previous iteration or time-step values to obtain the current iteration or time-step value. Explicit schemes are more common for unstructured grids [1, 2].

Implicit formulations are approaches that are more difficult to program than explicit schemes and have better convergence. Being more convergent can be an advantage if fast implicit methods, such as the alternating direction implicit (ADI) method, are applied. These kinds of fast implicit formulations are usually appropriate for structured grids [1, 2] that are composed of ordered finite cells on the solution domain. Their formulations generally use tri-diagonal or pentadiagonal matrix solutions, and these kinds of matrices are relatively easier to solve. Fully implicit schemes require more computational time because they result in 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 [3].

The alternating cell direction implicit (ACDI) method is a fast-implicit scheme that can be used for both structured and unstructured grids [4, 5]. This ability is not common for fast implicit schemes. Dimitri [2] 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 line implicitness of unstructured grids is on the usage of Hamiltonian Tours [6, 7]. Venkatakrishnan [3] stated that many Hamiltonian Tours might exist and that there might be none for a two-dimensional (2D) unstructured mesh.

The alternating cell direction implicit (ACDI) method is a fast-implicit scheme that can be used for both structured and unstructured grids [4, 5]. This ability is not common for fast implicit schemes. Dimitri [2] 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 line implicitness of unstructured grids is on the usage of Hamiltonian Tours [6, 7]. Venkatakrishnan [3] stated that many Hamiltonian Tours might exist and that there might be none for a two-dimensional (2D) unstructured mesh.

The ACDI method is offered to combine the advantages of the successful convergence of fast implicit methods with the easier meshing advantage of unstructured grids. The method is inspired by ADI methods by Meyer [8], Kellog [9], and Vries [10], but its algorithm structure is similar to that of the line Gauss–Seidel (LGS) iteration method, which is another fast-implicit method for the existence of quadrilateral elements. Because one of the objectives of this study is to increase the implicitness of the ACDI method to the order of approximate factorization [11, 12], the classical approximate factorization approach is explained briefly after the point Gauss–Seidel (PGS), LGS, ACDI, and full implicit Laasonen methods.

The main aim of this study is to combine an approximate factorization scheme with the ACDI [11, 12] time iteration scheme to increase the implicitness and accuracy of the previous ACDI studies. Although the main study is conducted on using cell-centered finite volume discretization, the approximate factorization 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 as a mathematically well-defined approach. The unsteady diffusion, unsteady advection-diffusion, and transonic full potential equations are selected as the model equations for the validation of the method.
Two different approximate factorization methods are adapted to the ACDI method. One of these factorizations uses standard implicit discretization (ACDI-AF), whereas the other uses the Crank–Nicolson discretization (ACDI-CN-AF). The Crank–Nicholson discretization averages the new and old values of the unknown in the central cell; thus, the scheme is more accurate and stable.

Çete et al. [5] previously demonstrated that the ACDI is appropriate for structured, unstructured quadrilateral, and a combination of triangular and quadrilateral quad-dominant unstructured grids. The approach is generalized for all kinds of polygonal cells with this study. The data structure of the previous programming approach was re-configured to obtain a flexible data structure that suits different numbers of edges in the cells. In addition, the ACDI study was moved to an advanced point in Onay’s master thesis [12], applying the ACDI method to the approximate factorization method. In this publication, part of this master thesis is presented.

2 Numerical method

2.1 Approximate factorization using the alternating cell direction implicit method

The method using the cell-centered finite volume approach is explained. Using the finite-volume approach is more suitable because the success of the approach is compared with the previous ACDI method, which also uses cell-centered finite volumes:

$$\frac{\partial T}{\partial t} - \nabla^2 T = 0. \quad (2.1)$$

Integrating the equation over the control volume cell results in the following:

$$\int_A \frac{\partial T}{\partial t} - \int_A \nabla^2 T dA = 0. \quad (2.2)$$

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

$$A \frac{\Delta T}{\Delta t} - \int_C \frac{\partial T}{\partial n} ds = 0, \quad (2.3)$$

where $T$ is the mean value of the dependent variable that is placed at the cell center such that the following is obtained:

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

If it is assumed that the fluxes at each cell edge are constant through the edge, where the cell has $N$ edges as illustrated in Fig. 1, which can be written in the form as shown:

$$A \frac{\Delta T}{\Delta t} - \sum_{i=1}^{N} \left( \frac{\partial T}{\partial n} \right)_i \Delta s_i = 0. \quad (2.4)$$
Eq. (2.4) can be written in the following form:

\[
A \frac{\Delta T}{\Delta t} \left[ \frac{\partial T}{\partial n} \right]_1 \Delta s_1 + \left( \frac{\partial T}{\partial n} \right)_2 \Delta s_2 + \cdots + \left( \frac{\partial T}{\partial n} \right)_N \Delta s_N = 0, \tag{2.5}
\]

where the flux terms are grouped such that opposite edge fluxes occur 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:

\[
A \frac{T^{n+1} - T^n}{\Delta t} = \left[ \left( \frac{\partial T}{\partial n} \right)_1 \Delta s_1 + \left( \frac{\partial T}{\partial n} \right)_{1+\frac{N}{2}} \Delta s_{1+\frac{N}{2}} \right] - \left[ \left( \frac{\partial T}{\partial n} \right)_2 \Delta s_2 + \left( \frac{\partial T}{\partial n} \right)_{2+\frac{N}{2}} \Delta s_{2+\frac{N}{2}} \right] - \cdots - \left[ \left( \frac{\partial T}{\partial n} \right)_N \Delta s_N + \left( \frac{\partial T}{\partial n} \right)_{N+\frac{N}{2}} \Delta s_{N+\frac{N}{2}} \right] = 0. \tag{2.6}
\]

It is possible to rearrange (2.6) such that \(T^n\) occurs at the right-hand side (RHS) of the equation because it is a known value at the \(n^{th}\) time step:

\[
T^{n+1} = \left[ \frac{\Delta t}{A} \frac{\partial T}{\partial n} \right]_1 \Delta s_1 + \frac{\Delta t}{A} \left( \frac{\partial T}{\partial n} \right)_{1+\frac{N}{2}} \Delta s_{1+\frac{N}{2}} - \left[ \frac{\Delta t}{A} \frac{\partial T}{\partial n} \right]_2 \Delta s_2 + \frac{\Delta t}{A} \left( \frac{\partial T}{\partial n} \right)_{2+\frac{N}{2}} \Delta s_{2+\frac{N}{2}} - \cdots - \left[ \frac{\Delta t}{A} \frac{\partial T}{\partial n} \right]_N \Delta s_N + \frac{\Delta t}{A} \left( \frac{\partial T}{\partial n} \right)_{N+\frac{N}{2}} \Delta s_{N+\frac{N}{2}} = T^n. \tag{2.7}
\]

The fluxes are calculated implicitly; thus, it is possible to write the LHS in operator form. For simplicity, \(T^{n+1}\) is represented as \(T\), and \(T^n\) is represented as the RHS from this point forward:

\[
\left( I - \left[ \frac{\Delta t}{A} \frac{\partial}{\partial n} \right]_1 \Delta s_1 + \left[ \frac{\Delta t}{A} \frac{\partial}{\partial n} \right]_{1+\frac{N}{2}} \Delta s_{1+\frac{N}{2}} \right) T = \text{RHS}. \tag{2.8}
\]

The form in Eq. (2.8) is for the application of ACDI-AF. The derivations for the usage of ACDI-AF and ACDI-CN-AF are very similar; thus, only the derivation of usage of ACDI-AF is given (Crank–Nicolson discretization averages the new and old values of the unknown in the central cell). Eq. (2.8) offers a form that is suitable for approximate factorization for any kind of 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
Figure 1: Control volume with $N$ edges.

Figure 2: Cell directions drawn traveling on opposing edges of cells.

strategies for odd-numbered edges are discussed later in this study. Applying a simple ACDI-AF to the form is shown below:

\[
T = \text{RHS.} \quad (2.9)
\]

Grouping fluxes written for 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 Fig. 2.

Eq. (2.9) can be written as given in Eq. (2.10) for each factorized term by assigning
semi-solutions that are declared as $T_1, T_2, \ldots, T_{N/2}$, such that the following is obtained:

$$
\begin{align*}
\begin{bmatrix}
I - \Delta t A \Delta s_1 \left( \frac{\partial}{\partial n} \right)_1 - \Delta t A \Delta s_{1+\frac{N}{2}} \left( \frac{\partial}{\partial n} \right)_{1+\frac{N}{2}} \\
I - \Delta t A \Delta s_2 \left( \frac{\partial}{\partial n} \right)_2 - \Delta t A \Delta s_{2+\frac{N}{2}} \left( \frac{\partial}{\partial n} \right)_{2+\frac{N}{2}} \\
\vdots \\
I - \Delta t A \Delta s_{N/2} \left( \frac{\partial}{\partial n} \right)_{N/2} - \Delta t A \Delta s_N \left( \frac{\partial}{\partial n} \right)_N
\end{bmatrix} T_1 &= \text{RHS} \\
\begin{bmatrix}
I - \Delta t A \Delta s_1 \left( \frac{\partial}{\partial n} \right)_1 - \Delta t A \Delta s_{1+\frac{N}{2}} \left( \frac{\partial}{\partial n} \right)_{1+\frac{N}{2}} \\
I - \Delta t A \Delta s_2 \left( \frac{\partial}{\partial n} \right)_2 - \Delta t A \Delta s_{2+\frac{N}{2}} \left( \frac{\partial}{\partial n} \right)_{2+\frac{N}{2}} \\
\vdots \\
I - \Delta t A \Delta s_{N/2} \left( \frac{\partial}{\partial n} \right)_{N/2} - \Delta t A \Delta s_N \left( \frac{\partial}{\partial n} \right)_N
\end{bmatrix} T_2 &= \text{RHS} \\
\vdots \\
\begin{bmatrix}
I - \Delta t A \Delta s_{N/2} \left( \frac{\partial}{\partial n} \right)_{N/2} - \Delta t A \Delta s_N \left( \frac{\partial}{\partial n} \right)_N
\end{bmatrix} T_{N/2} &= \text{RHS}
\end{align*}
$$

(2.10)

Writing the operators as shown above and using semi-solutions is performed to obtain tridiagonal matrices for the equation sets written over the solution bands shown in Fig. 2. The value of $T$ must be calculated after the semi-solutions are obtained. If the operators of Eq. (2.10) are $X_i, i=1, \ldots, N/2$, then the following is obtained:

$$
\begin{cases}
X_2 X_3 \cdots X_{N/2} T = T_1, \\
X_1 X_3 \cdots X_{N/2} T = T_2, \\
\vdots \\
X_1 X_2 \cdots X_{i-1} X_{i+1} \cdots X_{N/2} T = T_i, \quad i \neq j, \\
\vdots \\
X_1 X_2 \cdots X_{N-1} T = T_{N/2}.
\end{cases}
$$

(2.11)

If it is assumed that the expression $X_1 X_2 \cdots X_{i-1} X_{i+1} \cdots X_{N/2} T = T_i, \quad i \neq j$ is equal to the form before the factorization, then the following is found:

$$
X_1 \cdots X_{i-1} X_i X_{i+1} \cdots X_{N/2} T \\
\cong \left[ 1 - \left[ \Delta t A \Delta s_i \left( \frac{\partial}{\partial n} \right)_i + \Delta t A \Delta s_{i+\frac{N}{2}} \left( \frac{\partial}{\partial n} \right)_{i+\frac{N}{2}} \right] \right] \\
\quad + \left[ \Delta t A \Delta s_i \left( \frac{\partial}{\partial n} \right)_i + \Delta t A \Delta s_{i+\frac{N}{2}} \left( \frac{\partial}{\partial n} \right)_{i+\frac{N}{2}} \right] T_i \quad \text{(2.12)}
$$

If the expressions of (2.11) are summed using the assumption given in (2.12), the following summation is obtained:

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

(2.13)
Rearranging (2.13) yields the following:

$$T \left( N - 1 \right) \left( T - \sum_{i=1}^{N} \frac{\Delta t}{A} \Delta s_i \left( \frac{\partial T}{\partial n} \right)_i \right) = \sum_{i=1}^{N} T_i. \tag{2.14}$$

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

$$T = \sum_{i=1}^{N} T_i - \left( N - 1 \right) \text{RHS}. \tag{2.15}$$

2.2 Von Neumann stability analysis

The Von Neumann stability analysis is performed assuming Cartesian cells. A sample Cartesian cell and its neighbors are shown in Fig. 3 with the notation used for the analysis. The analysis is conducted using the inverse distance weighting flux calculation with ACDI-AF. Rewriting the equations in (2.10) for quadrilateral cells, the following is obtained:

$$T_1 - \Delta t \frac{\partial T_1}{\partial n} \left( T_{N_1} - T_{C} \right)_1 = \text{RHS}, \tag{2.16a}$$

$$T_2 - \Delta t \frac{\partial T_2}{\partial n} \left( T_{N_2} - T_{C} \right)_2 = \text{RHS}. \tag{2.16b}$$

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

$$\left( \frac{\partial T_1}{\partial n} \right)_1 = w_{1N_1} T_{1N_1} - w_{1C} T_{1C}, \tag{2.17}$$

where $w_{1N_1}$ and $w_{1C}$ are the weights used to calculate the flux term from the cell with the cell center $N_1$ shown in Fig. 3. The weights for Cartesian cells are all equal to $1/\Delta s$. The flux terms can be inserted into Eq. (2.17) in the below form:

$$\left( \frac{\partial T_1}{\partial n} \right)_1 = \frac{1}{\Delta s} \left( T_{N_1} - T_{C} \right), \quad \left( \frac{\partial T_1}{\partial n} \right)_3 = \frac{1}{\Delta s} \left( T_{N_3} - T_{C} \right), \tag{2.18a}$$

$$\left( \frac{\partial T_2}{\partial n} \right)_2 = \frac{1}{\Delta s} \left( T_{N_2} - T_{C} \right), \quad \left( \frac{\partial T_2}{\partial n} \right)_4 = \frac{1}{\Delta s} \left( T_{N_4} - T_{C} \right). \tag{2.18b}$$

Inserting the flux terms into the equations given in (2.16) and using spatial and temporal indices results in the following:

$$\left( 1 + 2 \frac{\Delta t}{A} \right) T_{i+1,j}^{n+1} - \frac{\Delta t}{A} T_{i-1,j}^{n+1} - \frac{\Delta t}{A} T_{i+1,j}^{n+1} = T_{i,j}^{n}, \tag{2.19a}$$

$$\left( 1 + 2 \frac{\Delta t}{A} \right) T_{2i,j}^{n+1} - \frac{\Delta t}{A} T_{2i,j-1}^{n+1} - \frac{\Delta t}{A} T_{2i,j+1}^{n+1} = T_{i,j}^{n}. \tag{2.19b}$$
To determine the growth rate of the error at each time iteration, \( T_{ij}^{n} = \zeta^{n} e^{ikx} e^{iky} \) is inserted into the first expression given in (2.19):

\[
\left(1 + 2\frac{\Delta t}{A}\right) b^{n+1} e^{ikx} e^{iky} - \frac{\Delta t}{A} b^{n+1} e^{i(k(x-\Delta x))} e^{iky} - \frac{\Delta t}{A} b^{n+1} e^{i(k(x+\Delta x))} e^{iky} = \zeta^{n} e^{ikx} e^{iky}.
\] (2.20)

Dividing both the RHS and LHS of Eq. (2.20) by \( \zeta^{n} e^{ikx} e^{iky} \) yields the following:

\[
\left(1 + 2\frac{\Delta t}{A}\right) \zeta - \frac{\Delta t}{A} \zeta e^{-ik\Delta x} - \frac{\Delta t}{A} \zeta e^{ik\Delta x} = 1.
\] (2.21)

Rearranging (2.21) results in the following:

\[
\zeta \left[ \left(1 + 2\frac{\Delta t}{A}\right) - \frac{\Delta t}{A} \left(e^{-ik\Delta x} + e^{ik\Delta x}\right) \right] = 1.
\] (2.22)

Then, the growth factor is as follows:

\[
\zeta = \frac{1}{\left(1 + 2\frac{\Delta t}{A}\right) - 2\frac{\Delta t}{A} (\text{Cos}k\Delta x)}.
\] (2.23)

Similarly, the growth factor for the second expression of (2.19) is as follows:

\[
\zeta = \frac{1}{\left(1 + 2\frac{\Delta t}{A}\right) - 2\frac{\Delta t}{A} (\text{Cos}k\Delta y)}.
\] (2.24)

The stability criterion is \(|\zeta| \leq 1, |\zeta| \leq 1\), due to the following:

\[-1 \leq \cos k\Delta x \leq 1 - 1 \leq \cos k\Delta y \leq 1.\] (2.25)

Both growth factors satisfy the stability criterion. This scheme is unconditionally stable on homogeneous Cartesian grids (formulation written on Cartesian meshes). However, this equation is equal to 1 in the limit (i.e., at this point, it is predicted that unconditional stability can change in inhomogeneous and unstructured grids).
2.3 Odd number of edges

The ACDI scheme is appropriate for elements that have an even number of edges, as stated previously. The solution bands, which are called cell directions, are generated by traveling on the opposing edges of the cells, as depicted in Fig. 2. Such generation is impossible for cells that have an odd number of edges. Two different strategies to overcome this problem are explained in this part. One of these methods is splitting one of the edges of the cell into two and converting it into a cell that has an even number of edges. This approach is only possible if the split edge takes place on the boundary of the domain. If the cell is inside the domain, and one of the edges is split into two, the neighbor cell that has an even number of edges is converted into a cell that has an odd number of edges, which is not desired.

Fig. 4 illustrates that splitting the edge at the boundary causes very sharp turns of the cell directions, which is also an undesirable situation. Although the new element has poor quality and the cell directions of the element have very sharp turns, the strategy is still useful and is used for some solutions presented later in this study. A cell that has an odd number of edges may occur inside the domain. In this situation, one of the nodes of the cell is considered a zero-length edge, and the cell direction ends inside the domain at that edge. The edge has two nodes that have the same coordinates. No boundary condition application is required for this edge because the flux is zero at this edge.

An example cell that is converted into a quadrilateral element with a zero-length edge and the related cell directions are shown in Fig. 5. The existence of a zero-length edge breaks the cell direction, as shown in Fig. 5. Having more triangular elements inside the domain means being less implicit. The selection of the location of the zero-length edge is also random; thus, the cell directions generated for the grid are no longer unique. One of the aims of this study is to produce a mathematically well-defined version of previous ACDI studies. No randomness of the solution exists for even-numbered edges over the domain. The sequence of solution bands does not change the numerical results. However, having an odd number of edges inside the domain damages this property.

Figure 4: Split edge of a triangular element at the boundary and cell directions.
3 Solution to the problem

3.1 Validation of the numerical approach using the unsteady diffusion equation

First, the unsteady heat conduction problem on a rectangular plate is solved for comparison. The infinite wall boundary condition is used at the horizontal boundaries, and the constant temperature boundary condition is applied at the vertical walls. The representation of this 1D problem is given in Fig. 6. The unsteady diffusion equation is given below:

\[
\frac{1}{\alpha} \frac{\partial T}{\partial t} - \nabla^2 T = 0. \tag{3.1}
\]

In (3.1), \(\alpha\) is the thermal diffusivity of the material. Dimensionless time for the unsteady diffusion equation is as follows:

\[
\tau = \frac{\alpha t}{k}. \tag{3.2}
\]

The dimensionless distance is the following:

\[
X = \frac{x}{L}. \tag{3.3}
\]

Nondimensionalization of the temperature is calculated as given below:

\[
\theta(x,y,t) = \frac{T(x,y,t) - T_\infty}{T_i - T_\infty}. \tag{3.4}
\]

Thus, the nondimensional form of the equation is as follows:

\[
\frac{\partial^2 \theta}{\partial X^2} + \frac{\partial^2 \theta}{\partial Y^2} = \frac{\partial \theta}{\partial \tau}. \tag{3.5}
\]
The analytical solution of the problem is presented:

$$\theta(x,y,t) = \sum_{n=1}^{\infty} A_n e^{-\lambda_n \tau} \cos(\lambda_n X), \quad A_n = \frac{4\sin \lambda_n}{2\lambda_n + \sin 2\lambda_n}, \quad \lambda_n \tan \lambda_n = Bi,$$  \hspace{1cm} (3.6)

where Bi is the nondimensional heat transfer coefficient Biot number:

$$Bi = \frac{hL}{k}.$$  \hspace{1cm} (3.7)

It is taken as infinity for the given numerical comparisons. Nondimensional temperature contours at $\tau = 0.1, 0.2,$ and $0.3$ were obtained using the analytical formulation given in Fig. 7.

Figure 6: Heat conduction problem on a rectangular plate.

Figure 7: Nondimensional temperature $\theta$ contours at (a) $\tau = 0.1$, (b) $\tau = 0.2$, and (c) $\tau = 0.3$ using an analytical formulation.
3.2 Validation of the proposed approximate factorization

The problem explained in the previous section is solved using ACDI-AF and ACDI-CN-AF for $20 \times 20$ and $40 \times 40$ structured grids. Because it is not appropriate to use the classical approximate factorization methods with unstructured grids, only structured grids are used for this comparison. The analytical solution is used to calculate the time-accurate mean errors of the time integration schemes. The mean error is defined as the average of the absolute differences of numerical and analytical solutions at nodes for the whole solution domain

$$MEAN\ ERROR = \left( \sum_{i=1}^{N} |\theta_{i,ANALYTICAL} - \theta_{i,NUMERICAL}| \right) \frac{1}{N},$$ (3.8)

where $N$ represents the total number of nodes if the solution is node based or represents the total number of cells if the solution is cell-center based. A node-based finite difference approach is used for the solutions. Second-order central differences are used for the spatial derivative terms. The mean error histories of the solutions for up to $\tau = 0.5$ are plotted and compared (Figs. 8 and 9).

Although the ACDI-AF method has a higher order of error for both grids, it yields results comparable to ACDI-CN-AF discretization. Having a higher order of error with the ACDI-AF method is an expected result. Explicit terms exist in the spatial Crank-Nicolson discretization, and these terms increase the order of the method.

![Figure 8](image)

**Figure 8**: Mean error histories of the numerical solutions for ACDI-AF and ACDI-CN-AF methods using $20 \times 20$ structured cells with time step size $\Delta \tau = 0.0001$.

3.3 Validation of the proposed approximate factorization with the alternating cell direction implicit method

First, the time-accurate results of ACDI-CN-AF and simple ACDI-AF are compared with previous methods (i.e., ACDI, PGS, LGS, RK4, and Laasonen) for structured cells. Then, a comparison is conducted for quadrilateral unstructured cells. The analytic results of the
problem are also calculated at nondimensional time: $\tau = 0.1$, $0.2$, and $0.3$. These analytic results are used to plot the error distributions through the constant line at $Y = 0.5$.

### 3.4 Comparison of different methods using structured grids

The analytical results at $\tau = 0.1$, $0.2$, and $0.3$ are listed in Fig. 10. The absolute error distributions through the constant line at $Y=0.5$ are plotted for comparison for the ACDI-CN-AF, ACDI-AF, ACDI, PGS, LGS, RK4, and Laasonen methods for structured cells. The error at the nodes is calculated using the following formulation:

$$
\text{NODE ERROR} = |\theta_{\text{NODE, ANALYTICAL}} - \theta_{\text{NODE, NUMERICAL}}|.
$$

Error distributions at $\tau = 0.1$, $0.2$, and $0.3$ are given in Figs. 11, 12, and 13, respectively, where the results were obtained using the methods RK4 and ACDI-CN-AF. The error distributions of these two methods are almost equal.
In addition, the error distributions of the ACDI-AF and full implicit Laasonen methods are almost the same. The explicit node terms added to the Crank–Nicolson scheme may disturb the implicitness of the ACDI-CN-AF scheme while increasing its accuracy. The numerical tests show that ACDI-CN-AF is stable for up to $\Delta \tau = 0.00127$, whereas the RK4 method is stable for up to $\Delta \tau = 0.00087$. The nondimensional diffusion number,
Table 1: Computer processing unit time required for the methods with $20 \times 20$ structured cells.

<table>
<thead>
<tr>
<th>METHOD</th>
<th>CPU TIME (s) FOR 1000 ITERATIONS</th>
</tr>
</thead>
<tbody>
<tr>
<td>RK4</td>
<td>5.788</td>
</tr>
<tr>
<td>LAASONEN</td>
<td>270.588</td>
</tr>
<tr>
<td>PGS</td>
<td>3.698</td>
</tr>
<tr>
<td>ACDI</td>
<td>4.492</td>
</tr>
<tr>
<td>ACDI-CN-AF</td>
<td>4.682</td>
</tr>
<tr>
<td>ACDI-AF</td>
<td>4.152</td>
</tr>
<tr>
<td>LGS</td>
<td>4.140</td>
</tr>
</tbody>
</table>

which is the order of diffusivity of a property in a domain, is defined below:

$$v = \alpha \frac{\Delta t}{\Delta x^2}.$$ (3.10)

Thus, the RK4 method is stable for up to $v \approx 0.41$, and ACDI-CN-AF is stable for up to $v \approx 0.59$. The reconstruction of the variables at the nodes using inverse distance weighting may harm the stability behavior of both approaches. These node values are used explicitly for the flux calculations. Moreover, ACDI-CN-AF behaves as a high order explicit scheme with a slightly better stability character. Figs. 11, 12, and 13 reveal that the orders of the errors of the full implicit Laasonen method and ACDI-AF are almost equal to each other. Thus, ACDI-AF is an appropriate approximation of the fully implicit scheme for a diffusion problem solution with structured cells. The numerical tests indicate that the simple ACDI-AF is unconditionally stable for structured cells. The computer processing unit (CPU) times of the solutions were also obtained during the calculations.

In addition, ACDI-CN-AF is distinguishably faster than the RK4 method. The CPU time results are listed in Table 1 for $20 \times 20$ structured grids. Moreover, RK4 has four steps for the integration application, which means four times the reconstruction of the variables. Additionally, ACDI-CN-AF does not deal with the whole discretized equation through cell directions and handles only half of the operations, which is another reason for being faster than the RK4 method. The absolute error at the selected point ($X = 0.5$, $Y = 0.5$) is given in Fig. 14 for different time-step sizes. The order of the error of ACDI-CN-AF does not increase with the increasing time-step size and loses stability after a certain point. The stability of ACDI-AF is not lost at up to $\Delta \tau = 0.1$, and it provides meaningful results with only two iteration steps at $\tau = 0.2$, whereas the accuracy becomes poorer with the increasing time-step size. Both the time-accurate error distribution plots and order of accuracy plots prove that ACDI-CN-AF presents the character of a high order explicit scheme, whereas ACDI-AF displays the character of a fast-implicit scheme.

3.5 Comparison of different methods using unstructured grids

The error distributions through the constant line $Y = 0.5$ are given in Figs. 15, 16, and 17 at dimensionless time $\tau = 0.1, 0.2$, and 0.3, respectively. A 297-element unstructured grid
Order of accuracy at $(X = 0.5, Y = 0.5)$ vs. time steps

Figure 14: Variation of the absolute error at point $(X = 0.5, Y = 0.5)$ with the change in time-step size for the ACDI-CN-AF and ACDI-AF solutions at $\tau = 0.2$ with a $20 \times 20$ structured grid.

Figure 15: Error distribution: constant line $Y = 0.5$ at $\tau = 0.1$ on a 297-element unstructured grid, $\Delta \tau = 0.0001$.

Figure 16: Error distribution: constant line $Y = 0.5$ at $\tau = 0.2$ on a 297-element unstructured grid, $\Delta \tau = 0.0001$.

was used for the comparisons (Fig. 19). The time step was selected as $\Delta \tau = 0.0001$ because all the integration schemes are stable for this time step. The order of accuracy is, again, almost the same for the RK4 and ACDI-CN-AF methods, whereas the stability behavior is better for the ACDI-CN-AF method. Numerical tests reveal that ACDI-CN-AF is stable at up to $\eta \approx 0.273$ and that the RK4 method is stable at up to $\eta \approx 0.187$ for the 297-element unstructured grid.
Table 2 indicates that the ACDI-CN-AF method is faster than the RK4 method if the CPU time per iteration is considered. The order of the error distributions was quite close for the ACDI-AF and Laasonen methods for structured cells. The order of the error, again, is comparable for these two methods, but the full implicit approximation of the ACDI-AF method is slightly disturbed for unstructured quadrilateral cells. The order of the error is lower than the previous ACDI study for both new ACDI methods.

Nondimensional $\theta$ contours at $\tau = 0.1$, $\tau = 0.2$, and $\tau = 0.3$ are given in Fig. 22 for the 297-element unstructured grid using ACDI-CN-AF. The absolute error at the selected point ($X = 0.5$, $Y = 0.5$) is given in Fig. 18 for different time-step sizes. The behaviors of the schemes are similar to the structured grid case. The order of the absolute error does not increase with the increasing step size for the ACDI-CN-AF solution, but the method has poor stability similar to an explicit scheme.

The order of error increases with the increasing step size if the ACDI-AF method is used. The method does not lose its stability at up to $\Delta \tau = 0.1$, where the numerical results are calculated at $\tau = 0.2$. 
Figure 19: A 297-element unstructured grid for time accurate comparisons.

![Figure 19](image)

Figure 20: Nondimensional temperature $\theta$ contours at (a) $\tau = 0.1$, (b) $\tau = 0.2$, and (c) $\tau = 0.3$ for 297 elements on the ACDI-AF solution.

![Figure 20](image)

Table 2: Computer processing unit (CPU) time required for the methods for 297 unstructured cells.

<table>
<thead>
<tr>
<th>METHOD</th>
<th>CPU TIME (s) FOR 1000 ITERATIONS</th>
</tr>
</thead>
<tbody>
<tr>
<td>RK4</td>
<td>3.8322</td>
</tr>
<tr>
<td>LAASONEN</td>
<td>62.459</td>
</tr>
<tr>
<td>PGS</td>
<td>2.668</td>
</tr>
<tr>
<td>ACDI</td>
<td>3.052</td>
</tr>
<tr>
<td>ACDI-CN-AF</td>
<td>3.124</td>
</tr>
<tr>
<td>ACDI-AF</td>
<td>2.828</td>
</tr>
</tbody>
</table>

3.6 Comparison of different grid types

The grids generated for the comparison are given in Fig. 21. The trapezoidal rule flux calculation is used to obtain the numerical results. The resulting temperature contours and error distribution contours were plotted at $t = 0.1$ using $\Delta t = 0.0001$ time-steps. The
average errors of the numerical solutions for up to $\tau = 0.5$ were also compared to observe the behavior of the method for different cell types. In addition, ACDI-AF was used for the comparison.

Nondimensional temperature contours are given in Fig. 22 for triangular, quadrilateral, and hybrid polygonal grids at $\tau = 0.1$ nondimensional time. The absolute error distributions obtained using the analytical solution are also represented. The nondimensional temperature contours and absolute error contours for the structured grid are given in Fig. 23 separately. The results for the structured grid are not given with the other grids because the error is on the order of $10^{-4}$ for the structured grid, whereas the error is on the order of $10^{-2}$ for the other grids. The error distributions are not comparable, and it is impossible to represent the structured grid absolute error with the unstructured grid absolute error distributions using the same legend.

Comparing the results of the unstructured grid solutions, the quadrilateral grid has the lowest error, and the triangular grid has the highest error. The hybrid polygonal grid does not generate a high order of error in the regions where hexagonal elements exist. The high error regions occur over the hexagonal elements. The cell directions for this grid are given in Fig. 24. The edges of the hexagonal elements are split into two to obtain elements with an even number of edges. This application results in sharp turns in the
Figure 22: (a) Nondimensional theta contours and (b) error contours for the triangular grid, (c) nondimensional theta contours and (d) error contours for the quadrilateral grid, and (e) nondimensional theta contours and (f) error contours for the hybrid polygonal grid on the ACDI-AF solution.

cell directions, low-quality elements, and an increased order of error. Having the highest error for triangular cells was also expected. The cell directions are broken inside the domain, and this situation blocks the information flow, as stated previously. The implicitness of the scheme is harmed with broken cell directions, and the scheme becomes a hybrid of the approximate factorization method and PGS scheme. The mean error histories of different grid solutions are obtained for up to $\tau = 0.5$ using time steps $\Delta \tau = 0.0001$. The comparison of the mean error histories was tested. The lowest error was obtained with the structured grid, and the highest error was obtained with the triangular grid at up to $\tau = 0.15$. Having an even number of edges seems to be an advantage before approaching the steady state. Having regularly distributed cells becomes more important
for more accurate results when approaching a steady state. The arrangement of the structured cells favors the gradient directions; thus, it is an expected result to have a relatively lower order of errors with structured grids.

### 3.7 Time-accurate comparisons using the advection-diffusion equation

The ACDI-AF method was also tested for time-accurate solutions to the well-known advection-diffusion equation. Trials on the diffusion equation demonstrate that using the Crank–Nicolson discretization results in a scheme that behaves like a high order explicit scheme with a relatively better stability behavior, but this thesis aims to represent a fast-implicit scheme that is also useful for unstructured grids.

Thus, the ACDI-AF method was tested for the advection-diffusion equation instead of the Crank–Nicolson version. If the advective terms are added to the diffusion equation, the resulting partial differential equation is as follows:

\[
\frac{\partial c}{\partial t} - D_x \frac{\partial^2 c}{\partial x^2} - D_y \frac{\partial^2 c}{\partial y^2} + u \frac{\partial c}{\partial x} + v \frac{\partial c}{\partial y} = 0, \tag{3.11}
\]
where \( D_x \) and \( D_y \) represent the diffusion coefficients in the \( x \) and \( y \) directions, and \( u \) and \( v \) represent the advection coefficients in the \( x \) and \( y \) directions, respectively. The dependent variable \( c \) denotes the mass concentration for the model equation. The model problem solved for the time integration comparisons does not include advection in the \( y \)-direction; thus, the equation reduces to the following:

\[
\frac{\partial c}{\partial t} - D_x \frac{\partial^2 c}{\partial x^2} - D_y \frac{\partial^2 c}{\partial y^2} + u \frac{\partial c}{\partial x} = 0. \quad (3.12)
\]

Integrating over the control volume with the assumption of equal diffusion coefficients in the \( x \)- and \( y \)-directions and applying Green’s theorem results in the following [15]:

\[
A \frac{\Delta c}{\Delta t} = D \oint_C \frac{\partial c}{\partial n} \, ds + \oint_C uc_n \, ds = 0. \quad (3.13)
\]

In the equation given above, \( n_x \) is the \( x \) component of the normal vector of the cell edge. If the dependent variable and its derivatives are assumed to be constant over the cell edge, the discretized equation is as follows:

\[
A \frac{\Delta c}{\Delta t} - D \sum_{i=1}^{N} \left( \frac{\partial c}{\partial n} \right)_i \Delta s_i + \sum_{i=1}^{N} uc_n \Delta s_i = 0. \quad (3.14)
\]

If the approximate factorization is applied for the diffusive terms, and the advective terms are explicitly considered, the factorized equation for the quadrilateral cells is as follows:

\[
\left( I - D \frac{\Delta t}{A} \Delta s_1 \left( \frac{\partial}{\partial n} \right)_1 \right) \left( I - D \frac{\Delta t}{A} \Delta s_3 \left( \frac{\partial}{\partial n} \right)_3 \right) \left( I - D \frac{\Delta t}{A} \Delta s_2 \left( \frac{\partial}{\partial n} \right)_2 \right) \left( I - D \frac{\Delta t}{A} \Delta s_4 \left( \frac{\partial}{\partial n} \right)_4 \right) c^{n+1} = c^n - \left( \frac{\Delta t}{A} \sum_{i=1}^{N} uc_n \Delta s_i \right)^n. \quad (3.15)
\]

### 3.8 \( k \)-exact least squares reconstruction

The dependent variable should also be calculated at the edge centers and nodes of the cells. The node values required for the trapezoidal rule flux calculation were calculated using inverse distance weighting for the diffusion equation comparisons, but reconstruction at the nodes and edge centers with inverse distance weighting results in fluctuations for the solution to the advection-diffusion equation [16,17]. In addition, low-order reconstruction is not appropriate for the full potential solver, which is explained later.

Neel [18] stated that the \( k \)-exact least squares method is one of the most suitable reconstruction schemes for full potential solvers for no oscillatory results with linear distributions. Thus, the \( k \)-exact least squares method is embedded in the ACDI-AF code for no oscillatory results for the advection-diffusion equation and transonic full potential
solutions. An appropriate stencil is selected for the reconstruction of the variable on the desired point. The variables and coordinates of the points of the stencil are used to generate a $k^{th}$ order polynomial for the reconstruction. A polynomial of $k^{th}$ order provides a $k+1$ order of accuracy.

The explanation of the reconstruction application of the least square is adapted from Neel [18] for reconstructing variables at nodes using the adjacent cell-center values. If the first-order polynomial for the reconstruction in a 2D domain is in the following form, then the linear equations are equal to the number of the cells in the stencil for the reconstruction:

$$P = K_O + K_1 x + K_2 y.$$  \hspace{1cm} (3.16)

This situation results in an over-constrained set of equations. All adjacent cells are used for the calculation of the variable on the node, as shown in Fig. 25.

Weighted reconstruction is used for such situations. Weights are generally set using the inverse distance weighting method. The sum of the squared errors at each cell center of the reconstruction is as follows:

$$error = \sum_{i=1}^{N} w_i (K_O + K_1 x + K_2 y - c_i)^2.$$  \hspace{1cm} (3.17)

Setting the derivative of the error with respect to the constant terms equal to 0 results in three linear equations with three unknowns:

$$\frac{\partial e}{\partial K_0} = 2 \sum_{i=1}^{N} w_i (K_O + K_1 x + K_2 y - c_i) = 0,$$  \hspace{1cm} (3.18a)

$$\frac{\partial e}{\partial K_1} = 2 \sum_{i=1}^{N} w_i (K_O + K_1 x + K_2 y - c_i) x_i = 0,$$  \hspace{1cm} (3.18b)

$$\frac{\partial e}{\partial K_2} = 2 \sum_{i=1}^{N} w_i (K_O + K_1 x + K_2 y - c_i) y_i = 0.$$  \hspace{1cm} (3.18c)
3.9 Problem specifications for advection-diffusion equation comparisons

The diffusion and advection problem of an instantaneous point source is solved for the comparisons. The analytic solution for comparisons assumes an infinite domain; thus, the Dirichlet boundary condition is used as the analytic solution at the boundaries to avoid dealing with an extremely large domain with too many cells or complicated types of boundary conditions because the aim of the comparisons is simply to compare the performance of different iteration schemes. The time-accurate analytic solution of the problem is as follows [19]:

\[
c(x,y,t) = \frac{M}{L} \frac{1}{4\pi t \sqrt{D_x D_y}} \exp \left[-\frac{(x-x_0-ut)^2}{4D_x t} - \frac{(y-y_0)^2}{4D_y t}\right],
\]  

(3.19)

where \( M/L \) is the mass released instantaneously, and the unit for this term is kilograms per meter (kg/m) because the problem is 2D. The diffusion and advection coefficients are taken as unity for the numerical comparisons. The analytic solutions are plotted in Fig. 26. The concentration distribution of a 1kg/m mass that is released from point \((0,0)\) is given after 5s, 10s, and 15s.

![Figure 26](image)

Figure 26: Contours of mass concentration at (a) \( t = 5 \)s, (b) 10s, and (c) 15s using an analytical formulation.
3.10 Comparison with the advection-diffusion equation using structured grids

A $16 \times 32$ cell structured grid was used for the comparisons. The Laasonen method was selected as a fully implicit solution method, and the RK4 method was selected as a fully explicit method. The numerical calculations were calculated using two different time steps, which are $\Delta t = 0.01s$ and $\Delta t = 0.25s$. The nondimensional Péclet (Pe) number for the case with unity diffusion and advection terms is $Pe = 1.25$. The Pe number can be defined as the rate of the advective behavior to the diffusive behavior and is calculated as follows:

$$Pe = \frac{u \Delta x}{D}. \quad (3.20)$$

The numerical schemes require up-winding methods for a stable solution, where $Pe \geq 2$ [20]. Thus, no up-winding applications were required for the numerical experiments of this section of the study.

The Courant number is dimensionless and defines the rate of transport per each time

![Figure 27: Comparison of concentration distributions obtained using the analytical and numerical solutions with ACDI-AF for the $16 \times 32$ structured cells and $\Delta t = 0.01s$.](image1)

![Figure 28: Comparison of concentration distributions obtained using the analytical and numerical solutions with the Laasonen method for the $16 \times 32$ structured cells and $\Delta t = 0.01s$.](image2)
step, and it is calculated using the following formula:

$$
\alpha = \frac{u \Delta t}{\Delta x}. 
$$

(3.21)

The Courant number for the $16 \times 32$ structured grid is $\alpha = 8 \times 10^{-3}$ if $\Delta t = 0.01s$ and $\alpha = 0.2$ if $\Delta t = 0.25s$. In Figs. 27, 28, and 29, the evolution of the mass concentrations is provided for the simple ACDI-AF, Laasonen, and RK4 methods, including comparisons with the analytical solution at the constant line $y = 0$ for $\Delta t = 0.01s$. All of these methods provide slightly more diffusive results than the analytic solution, and the solutions of the methods are quite close to each other for a relatively low Courant number.

The solutions at the constant line $y = 0$ are given in Figs. 30, 31, and 32 for $\Delta t = 0.25s$. The results are relatively less diffusive than they should be, and the accuracy of the RK4 method is better than that of the ACDI-AF and Laasonen methods. The behavior of the ACDI-AF method is close to the full implicit Laasonen method. The increased time steps decrease the accuracy of both. Moreover, the decreased time-step size makes the schemes
behave more diffusively.

3.11 Comparison with the advection-diffusion equation using unstructured grids

A 654-element unstructured grid for the comparison of the advection-diffusion equation solutions using different methods is given in Fig. 39. The numerical solutions were obtained using the ACDI-AF, Laasonen, and RK4 methods with the time steps $\Delta t = 0.01s$ and $\Delta t = 0.25s$. The minimum Courant number is $\alpha = 5 \times 10^{-3}$, and the maximum Courant number is $\alpha = 0.0268$ inside the domain where $\Delta t = 0.01s$. The numerical solutions at the constant line $y = 0$ were plotted and compared with the analytical solutions at $t = 5, 10,$ and $15s$ for all three methods, and the results are given in Figs. 33, 34, and 35.

The solutions are quite close to each other, but this time, the numerical solutions are less diffusive than they should be. The numerical results obtained using the time step $\Delta t = 0.25s$ is illustrated in Figs. 36, 37, and 38, including the comparisons with the an-
analitical solutions. Both the Laasonen and ACDI-AF methods have relatively less diffusive numerical results than the exact solution, and the accuracy of the numerical solu-
Figure 36: Comparison of concentration distributions obtained using the analytical and numerical solutions with the ACDI-AF method for the 624-element unstructured cells and $\Delta t = 0.25$ s.

Figure 37: Comparison of concentration distributions obtained using the analytical and numerical solutions with the Laasonen method for the 624-element unstructured cells and $\Delta t = 0.25$ s.

Figure 38: Unstable results of the Runge-Kutta order 4 method for the 624-element unstructured cells and $\Delta t = 0.25$ s.

The minimum Courant number inside the domain is
\( \alpha = 0.1273 \), and the maximum Courant number is \( \alpha = 0.6702 \). The RK4 method does not result in a stable solution. The mass concentration plots at the constant line \( y = 0 \) indicate that very smooth results are obtained for both stable solutions of the Laasonen and ACDI-AF methods, even with unstructured cells.

Additionally, the linear distribution of the mass concentration contours can be observed in Fig. 40. The linear distribution of the variables at the nodes is provided by the \( k \)-exact least squares reconstruction. The contours of the mass concentration of the ACDI-AF method show that the regions where relatively smaller cells occur are less diffusive than the larger cell regions inside the domain. This situation supports the idea that solutions with a lower Courant number behave more diffusively than solutions with a relatively higher Courant number. The unstable behavior of the RK4 method for the test case is illustrated in Figs. 40(b). The instability provides meaningless results throughout the domain at \( t = 15s \).

### 3.12 Transonic full potential solver using the alternating cell direction implicit method with approximate factorization

A full potential flow solver was generated using the ACDI-AF method. The full potential equation is a scalar and nonlinear equation that includes the assumptions of irrotational, isentropic, and inviscid flow. The full potential equation is usually used for subsonic and transonic flow problems because the accuracy of the solutions decreases for strong shock cases where the isentropic assumption through the shock is no longer valid. Holst [21] stated that the full potential solution provides relatively good approximations of Euler solutions for the cases in which the Mach number does not exceed the value of 1.3.

The unsteady and conservative form of the 2D equation is as follows:

\[
\frac{\partial \rho}{\partial t} + \frac{\partial (\rho u)}{\partial x} + \frac{\partial (\rho v)}{\partial x} = 0, \quad (3.22a)
\]

\[
u = \frac{\partial \phi}{\partial x} \quad \text{and} \quad v = \frac{\partial \phi}{\partial y}, \quad (3.22b)
\]
where \( u \) is the velocity component in the \( x \)-direction, \( v \) is the velocity component in the \( y \)-direction, \( \varphi \) is the velocity potential, and \( \rho \) is the density. Then, the steady form of the equation is as follows:

\[
\frac{\partial}{\partial x} \left( \rho \frac{\partial \varphi}{\partial x} \right) + \frac{\partial}{\partial y} \left( \rho \frac{\partial \varphi}{\partial y} \right) = 0. \tag{3.23}
\]

The relation between the pressure and density is as follows:

\[
\frac{p}{p_\infty} = \left( \frac{\rho}{\rho_\infty} \right)^Y, \tag{3.24}
\]

where \( P \) is the pressure, and \( Y \) is the specific heat ratio. The calculation of the density using the free stream velocity is shown below:

\[
\rho^* = \left[ 1 + \frac{Y-1}{2} Ma^2 (1-q^*^2) \right]^{\frac{1}{Y-1}}, \tag{3.25}
\]

where \( \rho^* \) and \( q^* \) are the nondimensional density and nondimensional free-stream velocity calculated using the following equations:

\[
\rho^* = \frac{\rho}{\rho_\infty} \quad \text{and} \quad q^* = \frac{u^2 + v^2}{q_\infty}. \tag{3.26}
\]
Using artificial time for the iterations and integrating the conservative form of the full potential equation results in the following:

\[ \int_A \frac{\partial \phi}{\partial t} dA - \int_C \rho \frac{\partial \phi}{\partial n} ds = 0. \] (3.27)

Using Crank–Nicolson discretization where \( \theta = 0.5 \), the following is obtained:

\[ A \frac{\Delta \phi}{\Delta t} - \left( \sum_{i=1}^{N} \rho \left( \frac{\partial \phi}{\partial n} \right)_i \Delta s_i \right)^{n+1} + (1 - \theta) \left( \sum_{i=1}^{N} \rho \left( \frac{\partial \phi}{\partial n} \right)_i \Delta s_i \right)^n = 0. \] (3.28)

As observed, the equation is nonlinear because of the relation between \( \rho \) and \( \phi \). This problem is solved by using the constant density value at a specific time on the cell edge of the interest. The following is the factorized form of the equation for quadrilateral cells:

\[ \left[ I - \left( \rho \left( \frac{\partial}{\partial n} \right)_1 \Delta s_1 \right) - \left( \rho \left( \frac{\partial}{\partial n} \right)_3 \Delta s_3 \right) \right] \left[ I - \left( \rho \left( \frac{\partial}{\partial n} \right)_2 \Delta s_2 \right) - \left( \rho \left( \frac{\partial}{\partial n} \right)_4 \Delta s_4 \right) \right] \Delta \phi
= \phi^n - (1 - \theta) \left( \int_{i=1}^{N} \rho \left( \frac{\partial \phi}{\partial n} \right)_i \Delta s_i \right)^n = 0. \] (3.29)

The trapezoidal rule is used for flux calculations; however, using inverse distance weighting to calculate the node values results in fluctuations in the solution to the full potential equation. Thus, the \( k \)-exact least squares method was used for the reconstruction of the variables at the nodes and centers of the edges.

The behavior of the equation is elliptic for subsonic regions, parabolic for sonic lines, and hyperbolic for supersonic regions; thus, density biasing is required for the supersonic region. A well-known artificial viscosity method was used for the stability in the supersonic regions of the solution. The density term is replaced with the biased term given below [22]:

\[ \tilde{\rho} = \rho - \mu \frac{\partial \rho}{\partial s}. \] (3.30)

The derivative term in the equation above is as follows:

\[ \frac{\partial \rho}{\partial s} = \frac{u}{q} \frac{\partial \rho}{\partial x} \Delta x + \frac{v}{q} \frac{\partial \rho}{\partial y} \Delta y. \] (3.31)

In addition, \( \mu \) is the switching function:

\[ \mu = \max \left[ 0, 1 - \frac{M_c^2}{M^2 k} \right], \] (3.32)

where \( M \) is the local Mach number; \( M_c \) is the cut-off Mach number, which is about 0.98; and \( k \) is a programmer-defined constant between 1 and 3.
### 3.13 Problem specifications for the full potential equation solution

The full potential solution to the geometries with a blunt leading edge is a more challenging problem than that for slender bodies; thus, subcritical and supercritical cases for a 2D cylinder were selected as a benchmark problem. The pressure coefficients for the subcritical case were compared with the analytical results adapted from the study by Saied and Alireza [23], and the pressure coefficients obtained for the supercritical case were compared with the full potential solutions by Djojodihardjo and Widodo [24]. In the case of the existence of the angle of attack, the satisfaction of the Kutta condition is required, but only the axis-symmetric flow was considered for the current study.

The flow around a 2D cylinder with a diameter of 1 m is solved using two different grids for two different free-stream Mach numbers. The free-stream Mach numbers were selected at 0.25 and 0.7. The first grid is a structured c-grid comprising 1500 elements. The second grid is an unstructured grid comprising 4589 quadrilateral elements. The grids are displayed in Figs. 41(a) and (b). Fifty elements were used around the semicylinder for both grids.

![Figure 41: (a) Structured grid and (b) unstructured grid used for the full potential equation solution.](image)

### 3.14 Full potential equation solution using the alternating cell direction implicit method with Crank-Nicholson approximate factorization

The plots of the Mach contours obtained for $M_{\infty} = 0.25$ are illustrated in Fig. 42 for the structured grid and in Fig. 43 for the unstructured grid, including the grid around the cylinder. The effect of the $k$-exact least squares reconstruction is depicted on the solution domain of the quadrilateral unstructured grid. The linear distribution of the variables results in highly smoothed contours. The numerically obtained pressure coefficients over the surface of the cylinder were compared with an analytical solution shown in Fig. 44 [24]. The results of the structured and unstructured grid solutions are in good agreement, whereas the numerical pressure distribution results are slightly more diffusive than they should be.

Mach contours for the supercritical case where $M_{\infty} = 0.7$ are given in Fig. 45 for the structured grid and Fig. 46 for the unstructured grid. In addition, the pressure coefficient distributions are compared with the full potential solution by Djojodihardjo and
Figure 42: Left: Mach contours around the cylinder; right: the structured grid $\text{Ma}_\infty = 0.25$ (ACDI-CN-AF solution).

Figure 43: Left: Mach contours around the cylinder; right: the unstructured grid $\text{Ma}_\infty = 0.25$ (ACDI-CN-AF solution).

Widodo [24] in Fig. 47. In Fig. 47, the location of the weak shock is closer to that of the solution taken from the literature. In addition, both shocks captured using the current study are not steep enough, but the rest of the pressure coefficient curves are in good agreement. Soulis [22] stated that the selection of the switching function $k$ seriously affects accuracy. The selection of this parameter or using a relatively coarse mesh over the surface of the cylinder might be the reason for the flat, weak shock.

The artificial viscosity term for density biasing is first-order accurate. Based on the Mach contours and pressure coefficient distributions for the supercritical solution, the agreement of the pressure coefficient comparisons begins to break up beginning with the sonic line. The effect of the artificial viscosity term increases with the increase in the Mach number. Thus, the solution becomes closer to a first-order accurate solution when approaching a weak shock region.

4 Summarizing the results

In this study, the development of the ACDI method that enables the use of simple fast implicit schemes in unstructured meshes and the application and testing of the approximate
factorization scheme to this method are presented. In this publication, the developed ACDI-AF and ACDI-CN-AF methods were applied to some problems and compared with other numerical methods. The results of these studies are summarized below.

1. The unsteady heat conduction problem on a rectangular plate is solved for comparison. The performances of the methods according to various meshes and mesh
numbers were investigated comparatively. These methods are Laasonen, PGS, LGS, ACDI, RK4, ACDI-AF and ACDI-CN-AF.

2. ACDI-AF, Laasonen and RK4 methods were also tested for time-accurate solutions to the well-known advection-diffusion equation. The time accuracy and stability of the methods were investigated.

3. A full potential flow solver was generated using ACDI-AF method for testing flow solver performance. The results of ACDI-AF with numerically well-known solutions were tested and compared in structured and unstructured meshes. Thus, the performance of the method in more complex solver was tested.

5 Conclusions

In this study, ACDI-AF and ACDI-CN-AF methods developed from ACDI scheme were developed. The main purpose of this article is to present and test this method which was developed. For this purpose, a series of comparative applications were made and presented. Here, the results of these studies are examined.

Firstly, as expected, both ACDI-AF and ACDI-CN-AF methods provide better time-accurate results than the previous ACDI study [5]. On the other hand, ACDI-AF and ACDI-CN-AF methods were observed to behave like a fully implicit scheme as it is an approach to a fully implicit scheme. It is observed that the order of accuracy increases in these methods in both structured and unstructured meshes. In addition, these methods appear to compete with high-order explicit schemes (RK4) in terms of CPU time. It is seen that these methods usually give a little faster solution for the same accuracy order than high-order explicit methods. Lastly, these methods are much more stable than high-order explicit methods. Explicit methods are generally conditionally stable and may not converge in some solutions. ACDI-AF method is unconditionally stable according to the
stability analysis (Section 2) of the Cartesian uniform meshes. In this analysis, this unconditional stability state is at the limit. Therefore, this unconditional stability is slightly impaired in unstructured meshes, but it was numerically demonstrated that ACDI-CN-AF method maintains its unconditional stability state.

As a result, in this study, ACDI method improves its convergence, accuracy and time accuracy by approximate factorization method. In addition, it was investigated whether the solution could solve complex PDE systems such as Navier-Stokes equation. It was shown that ACDI can work successfully with approximate factorization by applying the unstable advection-diffusion equation and full potential flow equation solutions.

Future studies may include further development of the scheme, application directly to Navier-Stokes solver, and three-dimensional applications etc.

Acknowledgements

We thank Dr. Oğuz Uzol and Dr. Nilay Uzol for their support of this study.

References


