Computational Fluid Dynamics

UUM 510E

Project 3

By

Muhammet Berati Kılıç

511251184

Course given by:

Mehmet Şahin

Aeronautics and Astronautics Engineering / Graduate School

This document was prepared using a modified LaTeX template originally developed by Javad Ibrahimli.

CONTENTS

1. PROBLEM STATEMENT

This project considers the two dimensional Laplace equation around a 2D cylinder with a domain $[1, 2] \times [0, 2\pi]$. The governing equation is given by:

$$\nabla^2 u = 0 \tag{1.1}$$

The analytical solution is:

$$u(r, \theta) = U_\infty \left(r - \frac{R}{r} \right) \sin(\theta)$$

The boundary conditions are the Dirichlet boundary conditions at:

$$u(1, \theta) = 0 \quad \text{and} \quad u(2, \theta) = \left(2 - \frac{1}{2} \right) \sin(\theta)$$

so

$$U_\infty = 1 \quad \text{and} \quad R = 1$$

The problem is solved using the vertex based finite volume method (FVM) with quadrilateral elements. Uniform meshes of 81×41, 161×81, and 321×161 are used. The numerical solution is compared with the analytical solution, and the error is also compared with the finite difference method (FDM) and Galerkin finite element method (FEM) results obtained in project 1 and project 2, respectively.

2. METHODOLOGY

2.1. Finite Volume Formulation

The governing equation, the Laplace equation, can be given in strong form as given in equation 1.1. In the finite volume method the equation is not discretized first in differential form. Instead, it's discretized after a control volume is defined around each unknown (around vertices) and the equation is integrated over control volumes. This integration can be expressed by strong integral form given by:

$$\iint_{CV} \nabla^2 u \, dxdy = 0 \tag{2.1}$$

To apply Green theorem, Laplace equation can be given as:

$$\nabla^2 u = \nabla \cdot (\nabla u)$$

Green theorem converts the area integral into a boundary flux integral given by:

$$\iint_{CV} \nabla \cdot (\nabla u) \, dxdy = \oint_{\partial CV} \mathbf{n} \cdot \nabla u \, dl = 0 \tag{2.2}$$

Using the normal derivative definition,

$$\mathbf{n} \cdot \nabla u = \frac{\partial u}{\partial n}$$

equation 2.2 can be written as:

$$\oint \frac{\partial u}{\partial n} \, dl = 0 \tag{2.3}$$

This equation represents the strong integral form of the governing equation and provides the mathematical basis for the finite volume discretization. Equation is written as a sum of edge fluxes over a control volume that doesn't overlap. If any inner edge is shared by two control volumes, the normal directions are opposite. Therefore, the flux leaving one control volume enters the neighbor control volume with the opposite sign. This is the conservative property of the finite volume method [1].

In this study, vertex based finite volume method is used. In this method, the unknown values are stored at the mesh vertices, nodes, and a control volume is defined around each vertex. The construction of control volumes and discretization equations are given in the following sections.

2.2. Mesh Generation

The computational mesh is generated uniformly in the radial and angular directions within the domain intervals. The radial and angular mesh spacings are:

$$\Delta r = \frac{1}{N_r - 1} \qquad \Delta\theta = \frac{2\pi}{N_\theta}$$

where N_r is the number of radial nodes for each angle and N_θ is the number of angular nodes for each radial layer. For node $P_{i,j}$, radial and angular coordinates can be written as $r_i = 1 + (i-1)\Delta r$ where $i = 1, 2, \ldots, N_r$ and $\theta_j = (j-1)\Delta\theta$ where $j = 1, 2, \ldots, N_\theta$.

The total number of mesh points is defined as $N_{\text{nodes}} = N_r N_\theta$. However, only the internal radial layers are solved as unknowns because the inner and outer radial boundaries are defined by Dirichlet boundary conditions. Therefore, the number of unknowns in the linear system is defined as $N_{\text{unknown}} = (N_r - 2)N_\theta$.

2.3. Control Volume Generation

The physical domain is represented in computational domain by $1 \leq r \leq 2$ and $0 \leq \theta < 2\pi$. The mesh is uniform in both the radial and angular directions.

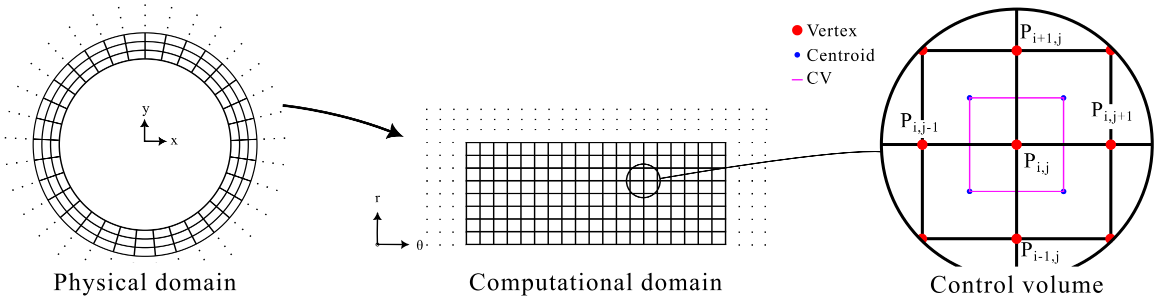


Figure 2.1: Schematic representation of the control volume constructed around a vertex.

As shown in figure 2.1, a control volume is constructed around a vertex $P_{i,j}$ by using the mid locations between neighboring nodes in the radial and angular directions. The corner points of the control volume coincide with the centroids of the surrounding quadrilateral regions formed by neighboring nodes. Therefore, control volumes have domain given by:

$$r_{i-\frac{1}{2}} \leq r \leq r_{i+\frac{1}{2}} \qquad \theta_{j-\frac{1}{2}} \leq \theta \leq \theta_{j+\frac{1}{2}}$$

Since the mesh is uniform, vertices are surrounded by a quadrilateral control volume in the computational domain. The inner and outer radial boundaries are defined by Dirichlet boundary conditions, so control volumes are formed only for the unknown nodes, internal vertices.

2.4. Numerical Discretization

The finite volume discretization is obtained from the flux balance over each control volume. For a control volume with M edges, the strong integral form given by equation 2.3 is approximated by the summation of the flux contributions over all control volume edges as:

$$\oint \frac{\partial u}{\partial n}\, dl \cong \sum_{m=1}^{M} \left(\frac{\partial u}{\partial n}\right)_m \Delta l_m = 0 \tag{2.4}$$

Here, Δl_m is the length of the m'th control volume edge, and $(\partial u/\partial n)_m$ is the normal derivative evaluated at that edge.

The basic idea of the flux approximation can first be shown on a uniform cartesian control volume, as illustrated in figure 2.2.

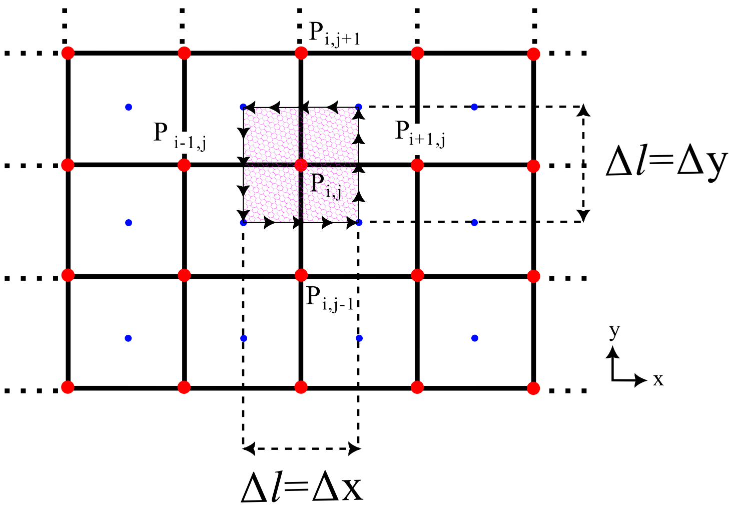


Figure 2.2: Control volume on a uniform cartesian grid.

For the cartesian control volume, the selected node is denoted by $P_{i,j}$ and the neighboring nodes are $P_{i-1,j}$, $P_{i+1,j}$, $P_{i,j-1}$, and $P_{i,j+1}$. The normal derivative at each edge is approximated by using the selected node and the neighboring node in the normal direction of that edge. Therefore, by using equation 2.4 flux contributions, represented by F, in the x direction can be written as:

$$F_{i-1,j} = \frac{u_{i-1,j} - u_{i,j}}{\Delta x}\Delta y \quad \text{and} \quad F_{i+1,j} = \frac{u_{i+1,j} - u_{i,j}}{\Delta x}\Delta y$$

Similarly, flux contributions in the y direction can be written as:

$$F_{i,j-1} = \frac{u_{i,j-1} - u_{i,j}}{\Delta y}\Delta x \quad \text{and} \quad F_{i,j+1} = \frac{u_{i,j+1} - u_{i,j}}{\Delta y}\Delta x$$

So, discrete flux balance over the cartesian control volume is:

$$\sum_{m=1}^{4}\left(\frac{\partial u}{\partial n}\right)_m \Delta l_m = F_{i-1,j} + F_{i,j-1} + F_{i+1,j} + F_{i,j+1} = 0 \tag{2.5}$$

Since grid is uniform cartesian grid, spatial displacements in x and y directions are equal and represented by Δl. Thus, equation 2.5 can be rewritten as:

$$\sum_{m=1}^{4}\left(\frac{\partial u}{\partial n}\right)_m \Delta l_m = u_{i-1,j} + u_{i,j-1} + u_{i+1,j} + u_{i,j+1} - 4u_{i,j} = 0$$

The same flux approximation method is used for the control volume in the computational domain. However, the geometric quantities are different from the cartesian grid because the radial edge lengths depend on the radial location. The control volume used in the current discretization is shown in figure 2.3.

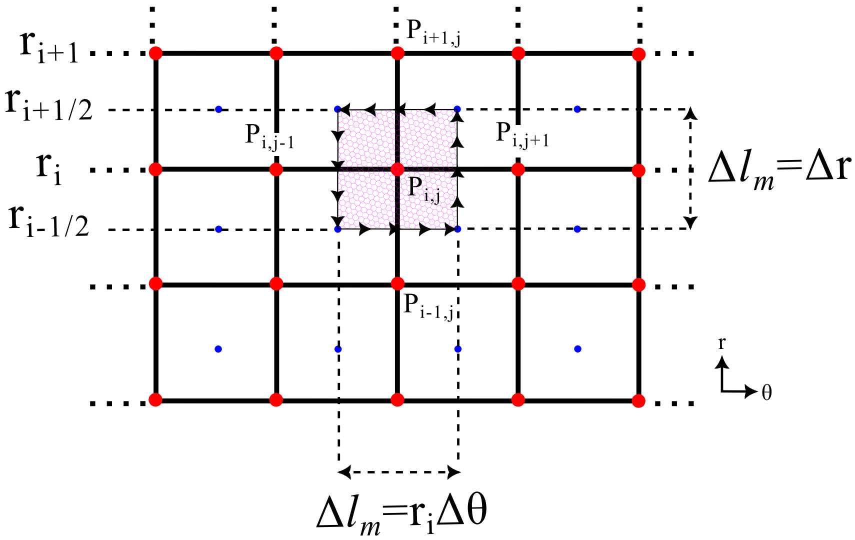


Figure 2.3: Control volume around a vertex in the computational domain.

For a node located at $P_{i,j}$, the radial edges of the control volume are located at:

$$r_{i-\frac{1}{2}} = r_i - \frac{\Delta r}{2} \quad \text{and} \quad r_{i+\frac{1}{2}} = r_i + \frac{\Delta r}{2}$$

The lengths of these radial edges can be calculated as $r_{i-\frac{1}{2}}\Delta\theta$, and $r_{i+\frac{1}{2}}\Delta\theta$. Since the distance between two neighboring radial layers is Δr, the radial flux contributions can be written as:

$$F_{i-1,j} = \frac{u_{i-1,j} - u_{i,j}}{\Delta r}\left(r_{i-\frac{1}{2}}\Delta\theta\right) \quad \text{and} \quad F_{i+1,j} = \frac{u_{i+1,j} - u_{i,j}}{\Delta r}\left(r_{i+\frac{1}{2}}\Delta\theta\right)$$

For the angular direction, the control volume edge length is Δr. The physical distance between two neighboring angular nodes is $r_i\Delta\theta$. Therefore, the angular flux contributions can be written as:

$$F_{i,j-1} = \frac{u_{i,j-1} - u_{i,j}}{r_i\Delta\theta}\Delta r \quad \text{and} \quad F_{i,j+1} = \frac{u_{i,j+1} - u_{i,j}}{r_i\Delta\theta}\Delta r$$

So, discrete flux balance over control volume can be written as:

$$\sum_{m=1}^{4}\left(\frac{\partial u}{\partial n}\right)_m \Delta l_m = F_{i-1,j} + F_{i,j-1} + F_{i+1,j} + F_{i,j+1} = 0 \tag{2.6}$$

For simple notation, the coefficients are defined as:

$$a_{i-1,j} = \frac{r_{i-\frac{1}{2}}\Delta\theta}{\Delta r} \qquad a_{i+1,j} = \frac{r_{i+\frac{1}{2}}\Delta\theta}{\Delta r} \qquad a_{i,j-1} = \frac{\Delta r}{r_i\Delta\theta} \qquad a_{i,j+1} = \frac{\Delta r}{r_i\Delta\theta}$$

So algebraic equation 2.6 can be rewritten as:

$$\begin{aligned}\sum_{m=1}^{4}\left(\frac{\partial u}{\partial n}\right)_m \Delta l_m = a_{i-1,j}u_{i-1,j} + a_{i,j-1}u_{i,j-1} + a_{i+1,j}u_{i+1,j} + a_{i,j+1}u_{i,j+1} \\ -u_{i,j}\left(a_{i-1,j} + a_{i+1,j} + a_{i,j-1} + a_{i,j+1}\right) = 0\end{aligned}$$

If $a_{i,j}$ defined as sum of other coefficients, algebraic equation becomes:

$$a_{i-1,j}u_{i-1,j} + a_{i+1,j}u_{i+1,j} + a_{i,j-1}u_{i,j-1} + a_{i,j+1}u_{i,j+1} - a_{i,j}u_{i,j} = 0 \tag{2.7}$$

This equation is written for each internal node of the computational domain.

2.5. Boundary Conditions and Solution of the Linear System

After the finite volume equations are written, they form the following linear system:

$$A\overrightarrow{x} = \overrightarrow{b}$$

where A is the highly sparse coefficient matrix, $\overrightarrow{x}$ is the unknown vector, and $\overrightarrow{b}$ is the right hand side vector.

The algebraic equation obtained from the finite volume discretization is written only for the internal nodes. Since the inner and outer radial boundaries are defined by Dirichlet boundary conditions, these boundary nodes are not included in the unknown vector. Therefore, the unknowns are defined only for $i = 2, 3, \ldots, N_r - 1$.

Thus, for internal node $P_{i,j}$, the corresponding row number in the linear system can be defined as:

$$k = (i-2)N_\theta + j$$

The Dirichlet boundary conditions are given as:

$$u_{1,j} = 0 \quad \text{and} \quad u_{N_r,j} = 1.5 \sin\theta_j$$

Since the boundary nodes are not solved as unknowns, their known values are inserted into the finite volume equations only when an internal control volume is adjacent to a radial boundary. For the first internal radial layer, $i = 2$, the neighboring node in the inner radial direction is on the inner boundary. Since $u_{1,j} = 0$, this boundary contribution is zero and does not change the right hand side vector. For the last internal radial layer, $i = N_r - 1$, the neighboring node in the outer radial direction is on the outer boundary. Since $u_{N_r,j} = 1.5 \sin\theta_j$, the outer boundary contribution is moved to the right hand side as:

$$b_k = b_k - a_{i+1,j} u_{N_r,j}.$$

This imposes the outer Dirichlet boundary condition without adding the outer boundary node to the unknown vector.

No additional node is stored at $\theta = 2\pi$. Instead, the angular neighbor indices are connected at that point due to $\theta = 2\pi = 0$. For $j = 1$, the $j-1$ neighbor is taken as N_θ, and for $j = N_\theta$, the $j+1$ neighbor is taken as 1. In this way, the first and last angular nodes are connected to each other during the finite volume assembly.

Since each internal node is connected only to neighboring nodes, the coefficient matrix contains many zero entries. Therefore, the matrix is stored in sparse form. In MATLAB sparse matrix notation, this structure can be described by three arrays which store the row indices, column indices, and numerical values of the nonzero entries, respectively [2]. After all finite volume coefficients are written in this form, the sparse coefficient matrix A is formed without storing the zero entries.

The final sparse linear system is solved using MATLAB's direct solver [3] (command: `x = A\b`).

The solution vector $\overrightarrow{x}$ contains only the internal nodal values. After the system is solved, these values are placed into the numerical solution matrix U according to their radial and angular positions. Then, the boundary values are directly inserted as defined. As a result, the complete numerical solution field is obtained over the entire 2D cylindrical domain.

2.6. Error and Convergence Definition

The numerical accuracy is investigated by comparing the computed solution with the analytical solution at the same grid points. The error is evaluated by the normalized L_2 error which is equivalent to the root mean square error and given by:

$$Error = \frac{\| u_{i,j} - u_{\text{analytic}} \|_2}{\sqrt{i_{\max} j_{\max}}} \tag{2.8}$$

Here, $u_{i,j}$ denotes the numerical solution, u_{analytic} denotes the analytical solution, and $i_{\max}$ and $j_{\max}$ represent the total numbers of grid points in the radial and angular directions, respectively.

The spatial convergence rate is determined by comparing the error values between meshes of increasing density. Since the discretization involves both radial and angular grid spacings, the convergence rate is evaluated in terms of Δr and $\Delta\theta$.

For the radial direction, convergence rate is given by:

$$p_{12_{\Delta r}} = \frac{\log \left(E_{\text{old}} / E_{\text{new}} \right)}{\log \left(\Delta r_{\text{old}} / \Delta r_{\text{new}} \right)} \tag{2.9}$$

For the angular direction, convergence rate is given by:

$$p_{12_{\Delta \theta}} = \frac{\log \left(E_{\text{old}} / E_{\text{new}} \right)}{\log \left(\Delta \theta_{\text{old}} / \Delta \theta_{\text{new}} \right)} \tag{2.10}$$

Where the p values represents convergence rate for consecutive meshes.

3. RESULTS AND DISCUSSION

3.1. Numerical Error Results

The vertex based finite volume solution was computed on three uniform meshes. The number of nodes and the computed error values are listed in Table 3.1. The error is calculated using equation 2.8.

Table 3.1: Numerical error values obtained by the vertex based finite volume method.

Mesh	Nodes	Error
81×41	3321	4.027×10^{-5}
161×81	13041	1.036×10^{-5}
321×161	51681	2.629×10^{-6}

The error decreases consistently as the mesh is refined as shown in table 3.1. This behavior indicates that the vertex based finite volume method yields better systematically with mesh refinement.

3.2. Comparison With the FDM and FEM

The finite volume error obtained in this study is compared with the FDM error obtained in project 1 and the Galerkin FEM error obtained in project 2. The comparison is given in table 3.2 for the same mesh densities.

Table 3.2: Comparison of FVM, FDM and FEM errors.

Mesh	FVM Error	FDM Error	FEM Error
81×41	4.027×10^{-5}	4.027×10^{-5}	5.906×10^{-7}
161×81	1.036×10^{-5}	1.036×10^{-5}	1.481×10^{-7}
321×161	2.629×10^{-6}	2.629×10^{-6}	3.709×10^{-8}

As shown in Table 3.2, the finite volume method gives the same error values as the finite difference method for all mesh densities. This is an expected behavior for uniform cartesian grids [4]. A similar behavior is also observed in this study because the mesh is uniform in the computational domain and the polar grid lines are orthogonal. In the first project, laplace equation is discretized directly in its polar differential form. In the finite volume method, the same equation is first integrated over the control volume as given in equation 2.1 and then written as a flux balance through the control volume edges as given in equation 2.4. Therefore, the FVM equation gives the area integrated form of the FDM equation. When the FVM equation is divided by the polar control volume area, the same coefficients of the second order central polar FDM stencil can be obtained as in the first project. So, FVM equation used in this project is simply same as area times FDM equation used in project 1. While this operation only scales the corresponding equation of the linear system, solution is not effected. That means, both methods yield the same numerical solution for this problem.

On the other hand, the FEM errors are smaller than both the FDM and FVM errors for all mesh levels. This behavior was also observed in Project 2. One possible reason for this difference is that the Q1 Galerkin FEM formulation has a larger interaction region, from the surrounding quadrilateral elements through the global assembly process, than the five point flux balance used in the FVM solution.

Figures 3.1 and 3.2 show the variation of the error with respect to Δr and $\Delta\theta$ in logarithmic scale. In both figures, the FDM and FVM lines overlap because their error values are the same for the same mesh levels. The FEM curve remains below both FDM and FVM curves as expected.

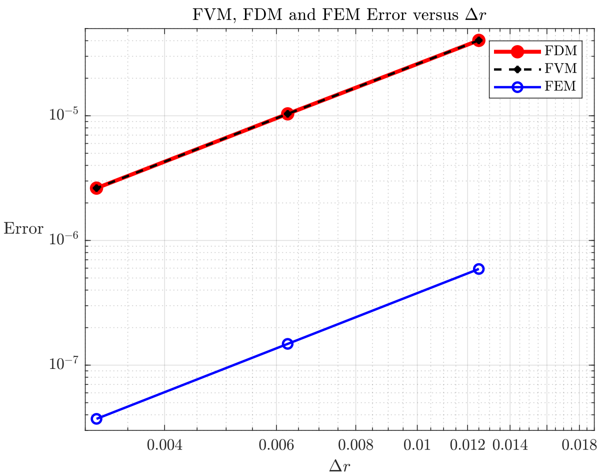


Figure 3.1: Comparison of FVM, FDM and FEM errors with respect to radial mesh spacing.

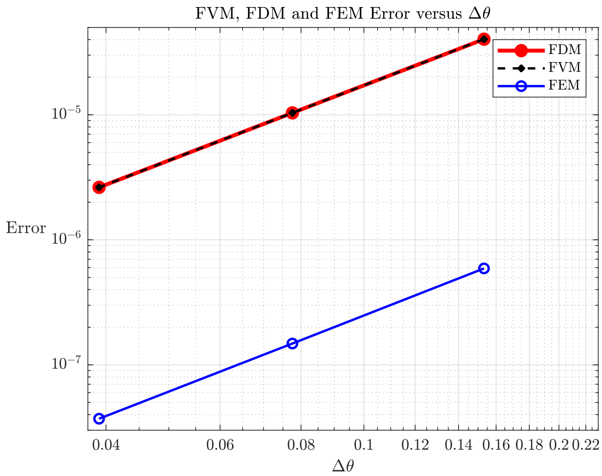


Figure 3.2: Comparison of FVM, FDM and FEM errors with respect to angular mesh spacing.

3.3. Convergence

The observed convergence rates were calculated using equations 2.9 and 2.10. The obtained rates are listed in Table 3.3.

Table 3.3: Observed spatial convergence rates.

Spacing	Mesh 1 - Mesh 2 (p_{12})	Mesh 2 - Mesh 3 (p_{23})
Δr	1.958	1.979
$\Delta\theta$	1.993	1.997

As shown in Table 3.3, the observed convergence rates are close to two in both coordinate directions. This indicates that the implemented finite volume formulation yields an approximately second order convergence trend in this study. The small deviations from exactly second order behavior can be attributed to the use of finite mesh levels.

Figure 3.3 illustrates the convergence behavior in logarithmic form. In a logarithmic convergence plot, the slope of the error curve corresponds to the observed convergence rate. Since the plotted data follow an almost straight line with slopes close to two, the figure is consistent with the approximately second order convergence behavior obtained from the finite volume solution.

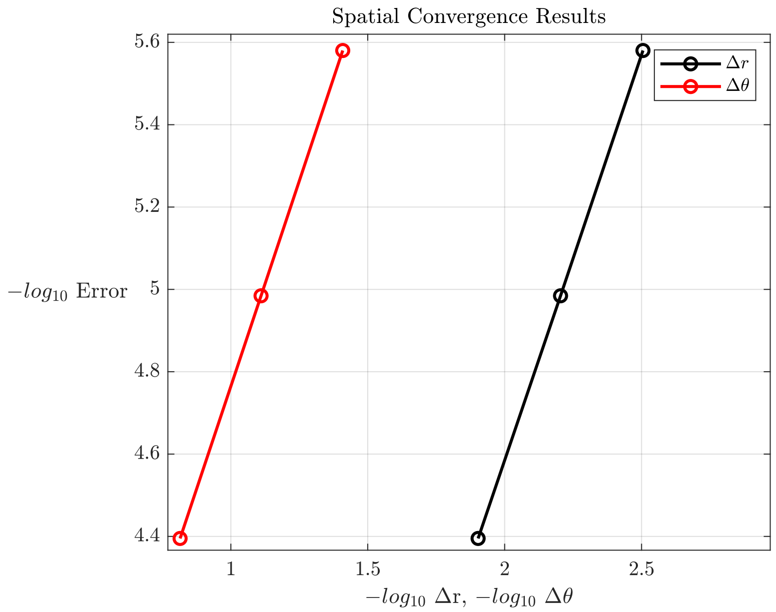


Figure 3.3: Spatial convergence results for the finite volume solution.

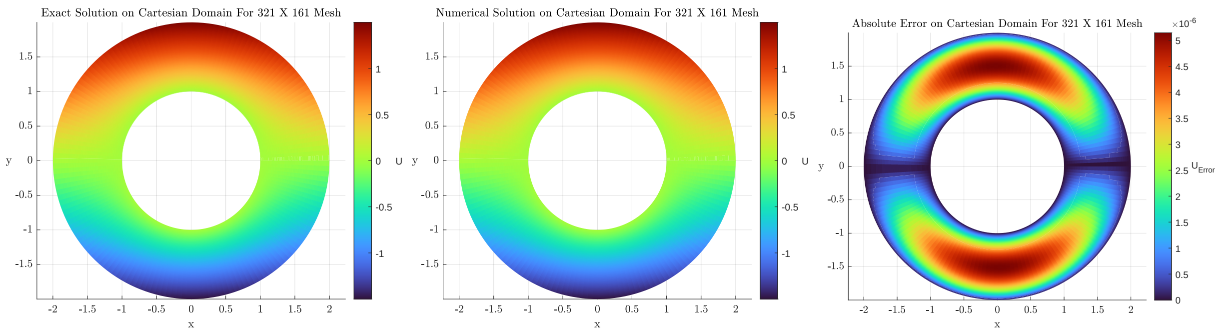
3.4. Visualization

The analytical solution, numerical FVM solution, and absolute error distribution of the 321×161 mesh are displayed in the cartesian plane in figure 3.4. The mesh is generated in polar coordinates, but the results are represented on the physical cartesian domain by using the coordinate transformation:

$$x = r\cos\theta \qquad y = r\sin\theta$$

This representation places the computed finite volume solution directly on the 2D cylindrical domain of the problem. Therefore, the spatial variation of the analytical solution, numerical FVM solution, and absolute error can be examined in the actual geometry.

The absolute error plot is useful because it shows the spatial distribution of the difference between the numerical and analytical solutions. Even when the analytical and numerical solution fields seem visually same, the error plot makes the local differences more visible.



(a) Analytical solution. (b) Numerical FVM solution. (c) Absolute error.

Figure 3.4: Analytical solution, numerical FVM solution, and absolute error distribution represented on the cartesian domain for the 321×161 mesh.

CONCLUSION

In this study, the two dimensional Laplace equation on a 2D cylindrical domain was solved using the vertex based finite volume method. The governing equation was integrated over each control volume and written as a discrete flux balance through the control volume edges. The resulting algebraic equations were assembled into a sparse linear system, and the Dirichlet boundary conditions were imposed directly on the algebraic equations.

The numerical results show that the error decreases systematically as the mesh is refined. The observed convergence rates are close to two in both radial and angular directions, indicating an approximately second order convergence behavior.

The comparison with the finite difference method shows that the implemented FVM gives the same error values as the FDM for all tested mesh resolutions. This is expected because both methods produce an equivalent five point discretization on the polar grid used in both studies. The cartesian visualizations also show that the numerical FVM solution agrees well with the analytical solution on the 2D cylindrical domain. Overall, the implemented finite volume method gives an accurate, conservative, and consistent solution for the given problem.

BIBLIOGRAPHY

[1] J. Ferziger, M. Perić, R. Street, Computational Methods for Fluid Dynamics, 2020, pp. 81-83. doi:10.1007/978-3-319-99693-6.

[2] The MathWorks, Inc., sparse: Create sparse matrix, https://www.mathworks.com/help/matlab/ref/sparse.html, accessed: 7 April 2026.

[3] The MathWorks, Inc., mldivide: Solve systems of linear equations Ax = B for x, https://www.mathworks.com/help/matlab/ref/double.mldivide.html, accessed: 7 April 2026.

[4] M. Şahin, UUT514E Computational Fluid Dynamics Course Notes, Unpublished lecture notes, Department of Aeronautical and Astronautical Engineering, Istanbul Technical University, pp. 38-41, file: CFD-1-merged.pdf (Spring 2010).