İTÜ

Computational Fluid Dynamics

UUM 510E

Project 1

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 cylinder with a domain $[0, 2\pi] \times [1, 2]$ is given by:

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = \frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial u}{\partial r}\right) + \frac{1}{r^2}\frac{\partial^2 u}{\partial \theta^2} = 0 \tag{1.1}$$

The analytical solution is:

$$u(r, \theta) = U_\infty (r - R/r)\sin(\theta) \tag{1.2}$$

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

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

The problem is considered on uniform $81 \times 41,\ 161 \times 81,$ and 321×161 meshes.

2. METHODOLOGY

2.1. Governing Equation in Polar Coordinates

Since the geometry of the problem is circular, it is more convenient to work with equation 1.1 in polar coordinates. The polar form of the Laplace equation is given by:

$$\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial u}{\partial r}\right)+\frac{1}{r^2}\frac{\partial^2 u}{\partial \theta^2}=0 \tag{2.1}$$

Expanding equation 2.1 gives:

$$\frac{\partial^2 u}{\partial r^2}+\frac{1}{r}\frac{\partial u}{\partial r}+\frac{1}{r^2}\frac{\partial^2 u}{\partial \theta^2}=0 \tag{2.2}$$

2.2. Finite Difference Discretization

To discretize equation 2.2, second-order central finite difference approximations are derived from Taylor series expansions. Taylor expansions in the radial direction are written as:

$$u(r+\Delta r,\theta)=u(r,\theta)+\Delta r\,\frac{\partial u}{\partial r}+\frac{\Delta r^2}{2}\,\frac{\partial^2 u}{\partial r^2}+\frac{\Delta r^3}{6}\,\frac{\partial^3 u}{\partial r^3}+O(\Delta r^4) \tag{2.3}$$

and

$$u(r-\Delta r,\theta)=u(r,\theta)-\Delta r\,\frac{\partial u}{\partial r}+\frac{\Delta r^2}{2}\,\frac{\partial^2 u}{\partial r^2}-\frac{\Delta r^3}{6}\,\frac{\partial^3 u}{\partial r^3}+O(\Delta r^4) \tag{2.4}$$

Subtracting equation 2.4 from equation 2.3 gives the second-order central approximation for the first derivative:

$$\frac{\partial u}{\partial r}=\frac{u(r+\Delta r,\theta)-u(r-\Delta r,\theta)}{2\Delta r}+O(\Delta r^2) \tag{2.5}$$

Adding equation 2.3 and equation 2.4 gives the second-order central approximation for the second derivative:

$$\frac{\partial^2 u}{\partial r^2} = \frac{u(r+\Delta r,\theta) - 2u(r,\theta) + u(r-\Delta r,\theta)}{\Delta r^2} + O(\Delta r^2) \tag{2.6}$$

Similarly second-order central approximation for the second derivative in angular direction given by:

$$\frac{\partial^2 u}{\partial \theta^2} = \frac{u(r,\theta+\Delta\theta) - 2u(r,\theta) + u(r,\theta-\Delta\theta)}{\Delta \theta^2} + O(\Delta \theta^2) \tag{2.7}$$

Substituting equations 2.5, 2.6, and 2.7 into the governing equation 2.2 yields:

$$\frac{u(r+\Delta r,\theta) - 2u(r,\theta) + u(r-\Delta r,\theta)}{\Delta r^2} + \frac{1}{r}\frac{u(r+\Delta r,\theta) - u(r-\Delta r,\theta)}{2\Delta r}$$

$$+ \frac{1}{r^2}\frac{u(r,\theta+\Delta\theta) - 2u(r,\theta) + u(r,\theta-\Delta\theta)}{\Delta \theta^2} = 0 \tag{2.8}$$

For the discrete formulation, the following notation is used:

$$u(r_i,\theta_j) = u_{i,j}$$

so that

$$u(r_i+\Delta r,\theta_j) = u_{i+1,j}\ , \quad u(r_i-\Delta r,\theta_j) = u_{i-1,j},$$

and

$$u(r_i,\theta_j+\Delta\theta) = u_{i,j+1}\ , \quad u(r_i,\theta_j-\Delta\theta) = u_{i,j-1}$$

So equation 2.8 becomes:

$$\frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{\Delta r^2} + \frac{1}{r_i}\frac{u_{i+1,j} - u_{i-1,j}}{2\Delta r} + \frac{1}{r_i^2}\frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{\Delta \theta^2} = 0 \tag{2.9}$$

Rearranging equation 2.9 gives the finite difference form used at each interior grid points:

$$u_{i-1,j}\left(\frac{1}{\Delta r^2} - \frac{1}{2r_i\Delta r}\right) + u_{i,j}\left(-\frac{2}{\Delta r^2} - \frac{2}{r_i^2\Delta\theta^2}\right) + u_{i+1,j}\left(\frac{1}{\Delta r^2} + \frac{1}{2r_i\Delta r}\right)$$

$$+ u_{i,j-1}\left(\frac{1}{r_i^2\Delta\theta^2}\right) + u_{i,j+1}\left(\frac{1}{r_i^2\Delta\theta^2}\right) = 0 \tag{2.10}$$

Since all derivative approximations are obtained by second-order central finite differences, the discretization is second-order accurate.

2.3. Computational Grid

The computational domain is discretized by a uniform mesh in polar coordinates, consistent with the circular geometry of the problem. Numerical grid is constructed in the radial and angular directions using constant spacings in both coordinates. The radial and angular grid spacings are given by:

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

where N_r denotes the number of grid points in the radial direction and N_θ denotes the number of grid points in the angular direction.

The radial boundaries at $r = 1$ and $r = 2$ are included in the mesh as boundary nodes. In the angular direction, the points where $\theta = 2\pi$ is not stored as an additional grid points since they corresponds to the same physical location as $\theta = 0$. Thus, first and last angular locations are treated as connected in the numerical solution.

2.4. Numerical Solution Procedure and Boundary Treatment

The finite difference equation given in 2.10 is applied only at the interior nodes as:

$$i = 2, 3, \ldots, N_r - 1, \qquad j = 1, 2, \ldots, N_\theta$$

For simplicity, the coefficients in equation 2.10 are defined as:

$$a_i = \frac{1}{\Delta r^2} - \frac{1}{2r_i\Delta r}, \quad b_i = -\frac{2}{\Delta r^2} - \frac{2}{r_i^2\Delta\theta^2}, \quad c_i = \frac{1}{\Delta r^2} + \frac{1}{2r_i\Delta r}, \quad d_i = \frac{1}{r_i^2\Delta\theta^2}$$

These coefficients are evaluated locally at each radial position r_i. Therefore, the discrete equation at an interior node can be written as:

$$a_i\, u_{i-1,j} + b_i\, u_{i,j} + c_i\, u_{i+1,j} + d_i\, u_{i,j-1} + d_i\, u_{i,j+1} = 0 \tag{2.11}$$

Equation 2.11 represents one algebraic equation for each interior grid point. In the global matrix system, this equation becomes one row of the coefficient matrix. The coefficient of $u_{i,j}$ is placed on the diagonal entry of that row. The coefficients of $u_{i-1,j}$, $u_{i+1,j}$, $u_{i,j-1}$, and $u_{i,j+1}$ are placed on off-diagonal entries.

The radial boundary conditions are Dirichlet boundary conditions:

$$u_{1,j} = u(1, \theta_j) = 0, \qquad u_{N_r,j} = u(2, \theta_j) = \left(2 - \frac{1}{2}\right) \sin \theta_j$$

Because these values are known, they are not included among the unknowns of the system. The unknowns consist only of the interior radial layers, as $i = 2, 3, \ldots, N_r - 1$.

In the angular direction, the solution is periodic because $\theta = 0$ and $\theta = 2\pi$ represent the same physical location. Therefore, for $j = 1$, the left angular neighbor is taken from $j = N_\theta$, and for $j = N_\theta$, the right angular neighbor is taken from $j = 1$. This periodic treatment does not introduce an additional known boundary value. It only determines which angular unknown is used as the neighboring point in the algebraic system.

The radial boundary conditions have effect at the first and last interior radial layers. For $i = 2$, the term containing $u_{1,j}$ is known from the inner boundary condition. Since $u_{1,j} = 0$, this term does not add any value to the right hand side. For $i = N_r - 1$, the term containing $u_{N_r,j}$ is known from the outer boundary condition and is transferred to the right hand side:

$$a_i\, u_{i-1,j} + b_i\, u_{i,j} + d_i\, u_{i,j-1} + d_i\, u_{i,j+1} = -\, c_i\, u_{N_r,j} \qquad i = N_r - 1$$

In this manner, all known radial boundary values are treated explicitly, while only the interior nodal values remain unknown.

After the boundary treatment, the algebraic equations at all interior nodes are assembled into a single linear system of the form:

$$\mathbf{A}\,\overrightarrow{x} = \overrightarrow{b} \tag{2.12}$$

Here, $\mathbf{A}$ is the coefficient matrix, $\overrightarrow{x}$ is the vector of unknown interior nodal values, and $\overrightarrow{b}$ is the right hand side vector containing the known boundary terms.

Since only the interior radial layers are unknown, the total number of unknowns is:

$$N_{\text{unk}} = (N_r - 2)N_\theta$$

These unknowns are arranged into a one dimensional vector by ordering the angular values at each interior radial layer one after another:

$$\overrightarrow{x} = \begin{bmatrix} u_{2,1} & u_{2,2} & \cdots & u_{2,N_\theta} & u_{3,1} & \cdots & u_{N_r-1,N_\theta} \end{bmatrix}^T$$

Accordingly, the global index corresponding to the node (i, j) is written as:

$$k = (i-2)N_\theta + j, \qquad i = 2, 3, \ldots, N_r - 1, \quad j = 1, 2, \ldots, N_\theta \tag{2.13}$$

With this indexing, each interior finite difference equation becomes one row of the matrix system. The diagonal entry of that row corresponds to the coefficient of $u_{i,j}$, while the off-diagonal entries correspond to the neighboring points $u_{i-1,j}$, $u_{i+1,j}$, $u_{i,j-1}$, and $u_{i,j+1}$. Therefore, each row of $\mathbf{A}$ contains at most five nonzero entries. This reflects the five-point stencil of the second-order finite difference discretization in polar coordinates.

This idea can be shown schematically by writing one representative row inside the matrix system as:

$$\underbrace{\begin{bmatrix} \ddots & & & & & & & & \\ & \cdots & a_i & \cdots & d_i & b_i & d_i & \cdots & c_i \\ & & & & & & & & \ddots \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} \vdots \\ u_{i-1,j} \\ \vdots \\ u_{i,j-1} \\ u_{i,j} \\ u_{i,j+1} \\ \vdots \\ u_{i+1,j} \\ \vdots \end{bmatrix}}_{\overrightarrow{x}} = \underbrace{\begin{bmatrix} \vdots \\ \left(\overrightarrow{b}\right)_k \\ \vdots \end{bmatrix}}_{\overrightarrow{b}}$$

In this representative matrix form, the coefficient b_i multiplies the central unknown $u_{i,j}$. The coefficients a_i and c_i multiply the radial neighboring values, while the two d_i coefficients multiply the angular neighboring values. This row is only schematic. The actual column positions of these entries depend on the global indexing defined in equation 2.13.

The matrix $\mathbf{A}$ may be interpreted as follows: each row represents the discrete equation written for one interior node, and each nonzero entry represents either that node or one of its neighboring nodes. The vector $\overrightarrow{x}$ contains the nodal values to be determined, whereas the vector $\overrightarrow{b}$ contains the known information coming from the boundary conditions.

At radial layers adjacent to the boundaries, one of the radial neighbors is known. Therefore, its coefficient is not placed in $\mathbf{A}$. Instead, the known boundary term is moved to the right hand side vector, $\overrightarrow{b}$.

For example, since $u_{1,j} = 0$, the inner boundary term is zero. The outer boundary term is nonzero since $u_{N_r,j} = 1.5 \sin\theta_j$. Thus, the known boundary values enter the system through $\overrightarrow{b}$, while the unknown interior values remain in $\mathbf{A}\overrightarrow{x}$.

Although $\mathbf{A}$ represents the coefficient matrix of the complete algebraic system, it is highly sparse because each finite difference equation involves only a small number of neighboring nodes. In other words, most entries of the full matrix are zero. Therefore, the system is not stored as a dense matrix. Instead, only the nonzero coefficients are stored together with their row and column positions.

In the present discretization, the nonzero coefficients of $\mathbf{A}$ come directly from the finite difference stencil. For each interior node, the coefficient of the central value $u_{i,j}$ and the coefficients of the neighboring values $u_{i-1,j}$, $u_{i+1,j}$, $u_{i,j-1}$, and $u_{i,j+1}$ are placed into the corresponding row of $\mathbf{A}$. All other entries in that row remain zero. Therefore, the sparse matrix is not a different numerical approximation; it is the same coefficient matrix $\mathbf{A}$, but stored only through its nonzero entries.

In MATLAB sparse matrix notation, the sparse structure can be described by three arrays: i, j, and v. The array i stores the row locations of the nonzero entries, j stores the column locations, and v stores the corresponding numerical values [1]. Therefore, each stored entry is represented by (i_p, j_p, v_p), where p is only the storage order of the nonzero entry. After these arrays are filled, they define the row index, column index, and value of every nonzero coefficient in the sparse matrix. Thus, the complete coefficient matrix is formed from these arrays without storing the zero entries.

This storage strategy is advantageous because each row contains at most five nonzero entries, regardless of the total matrix size. As a result, the number of stored coefficients grows approximately in proportion to the number of unknowns, whereas a dense matrix would require storage proportional to N_{unk}^2. Therefore, sparse storage allows the same algebraic system to be stored and solved much more efficiently.

Once the sparse matrix $\mathbf{A}$ and the right hand side vector $\overrightarrow{b}$ are assembled, the numerical solution is obtained by solving equation 2.12.

A direct solution strategy can be explained conceptually through LU factorization. The purpose of this part is to describe the mathematical idea behind a direct matrix solution. In its basic form, the coefficient matrix is factorized into a lower triangular matrix and an upper triangular matrix:

$$\mathbf{A} = \mathbf{LU}$$

where $\mathbf{L}$ is a lower triangular matrix and $\mathbf{U}$ is an upper triangular matrix. The factorization can be represented schematically as:

$$\mathbf{A} = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1n} \\ A_{21} & A_{22} & \cdots & A_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ A_{n1} & A_{n2} & \cdots & A_{nn} \end{bmatrix} = \underbrace{\begin{bmatrix} L_{11} & 0 & \cdots & 0 \\ L_{21} & L_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ L_{n1} & L_{n2} & \cdots & L_{nn} \end{bmatrix}}_{\mathbf{L}} \underbrace{\begin{bmatrix} U_{11} & U_{12} & \cdots & U_{1n} \\ 0 & U_{22} & \cdots & U_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & U_{nn} \end{bmatrix}}_{\mathbf{U}}$$

Substituting this factorization into the algebraic system gives:

$$\mathbf{LU}\overrightarrow{x} = \overrightarrow{b}$$

Introducing an intermediate vector $\overrightarrow{y}$, the solution is obtained in two steps:

$$\mathbf{L}\overrightarrow{y} = \overrightarrow{b}\ , \qquad \mathbf{U}\overrightarrow{x} = \overrightarrow{y}$$

The first triangular system is solved by forward substitution, and the second triangular system is solved by backward substitution. In this way, all interior nodal values are obtained simultaneously from the assembled matrix system. For a sparse matrix, an explicit LU-based form can also be written in MATLAB by using `[L,U,P,Q]=lu(A)`. In this case, permutation matrices may appear, and the factorization is written as $\mathbf{PAQ} = \mathbf{LU}$[2]. The solution can then be recovered through the sequence `y=L\(P*b)`, `z=U\y`, and `x=Q*z`. The same assembled sparse system can also be solved directly using the MATLAB command `x=A\b`.

After the vector $\overrightarrow{x}$ is obtained, its entries are mapped back onto the two dimensional grid. The interior nodal values are placed into their corresponding (i, j) locations, while the boundary values are inserted directly from the prescribed boundary conditions. In this way, the complete numerical solution field, $U = [u_{i,j}]$, is reconstructed over the whole computational domain. This reconstructed field is then compared with the analytical solution in order to evaluate the numerical error and the convergence behavior discussed in the following section.

2.5. 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:

$$Error = \frac{\|u_{i,j} - u_{\text{analytic}}\|_2}{\sqrt{i_{\max} j_{\max}}} \tag{2.14}$$

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:

$$p_{12_{\Delta r}} = \frac{\log\left(E_{\text{old}}/E_{\text{new}}\right)}{\log\left(\Delta r_{\text{old}}/\Delta r_{\text{new}}\right)} \tag{2.15}$$

For the angular direction, convergence rate is given by:

$$p_{12_{\Delta\theta}} = \frac{\log\left(E_{\text{old}}/E_{\text{new}}\right)}{\log\left(\Delta\theta_{\text{old}}/\Delta\theta_{\text{new}}\right)} \tag{2.16}$$

Where the p values represents convergence rate for consecutive meshes.

3. RESULTS AND DISCUSSION

3.1. Error Evaluation

The numerical error decreases consistently as the mesh is refined. In particular, the error is reduced from 4.027×10^{-5} on the 81×41 mesh to 1.036×10^{-5} on the 161×81 mesh, and then to 2.629×10^{-6} on the 321×161 mesh. This reduction indicates that the numerical approximation improves systematically as both Δr and $\Delta\theta$ decrease.

Figure 3.1 shows the variation of the error with respect to radial grid spacing in logarithmic scale, while Fig. 3.2 shows the same behavior with respect to angular grid spacing. Taken together, the two figures confirm that the numerical solution becomes progressively more accurate under mesh refinement in both coordinate directions.

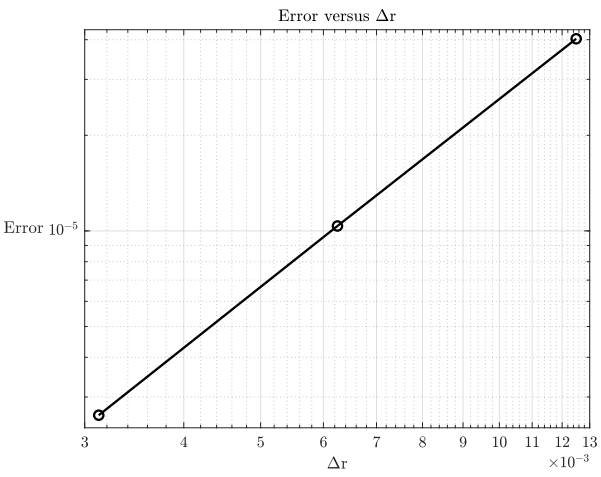


Figure 3.1: Variation of the numerical error with radial grid spacing, shown in logarithmic scale.

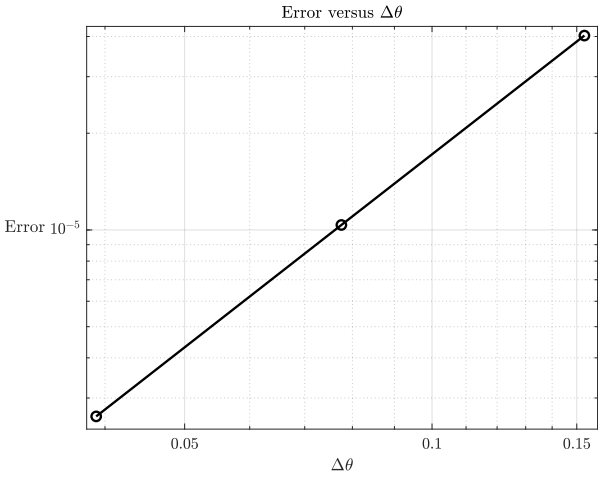


Figure 3.2: Variation of the numerical error with angular grid spacing, shown in logarithmic scale.

Since the error is evaluated by equation 2.14, the reported values represent the difference between the analytical and numerical solutions all over the computational domain. From a numerical perspective, the observed trend is significant: when the grid spacing is reduced by approximately half, the error is reduced by almost a factor of four. This behaviour is consistent with second order spatial accuracy.

3.2. Convergence

The observed convergence rates are very close to 2 in both coordinate directions. For the radial spacing, the computed rates are 1.958 between the first and second meshes and 1.979 between the second and third meshes. For the angular spacing, the corresponding values are 1.993 and 1.997. These results show that the method yields barely ideal second order accuracy over the tested mesh range.

This conclusion is also supported by Fig. 3.3. In the convergence graph, the slope of the error curve has the same meaning as the observed convergence rate given in equations 2.15 and 2.16. Since the plotted data follow an almost straight line with a slope very close to 2, the figure confirms that the numerical solution converges with approximately second order accuracy in both directions.

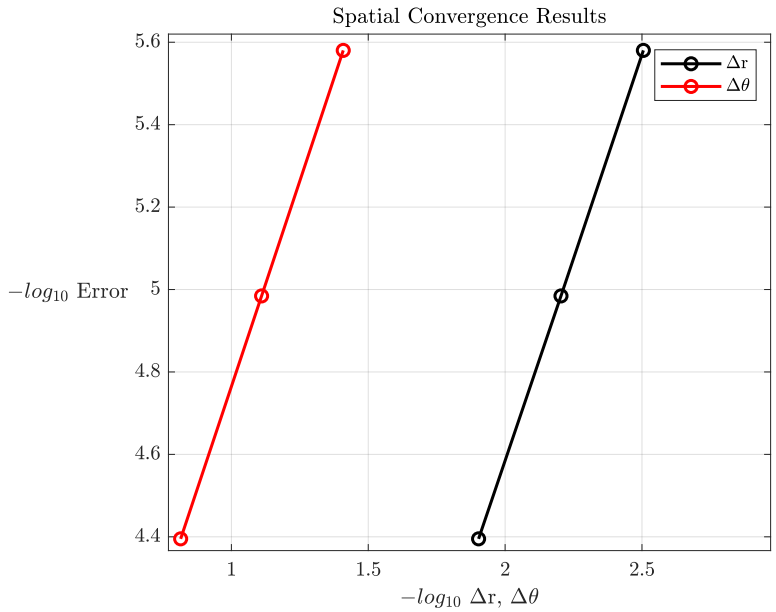


Figure 3.3: Spatial convergence results.

The small deviation of the measured values from the exact value 2 is not unexpected in computations. For a finite set of meshes, the observed rate may differ slightly from the expected value. Nevertheless, the present results are sufficiently close to the ideal value to validate the expected convergence behavior of the scheme.

3.3. Visualization

Figure 3.4 presents a visual representation of the results in polar coordinates. In this way, the analytical solution, numerical solution, and absolute error distribution can be examined together for different mesh levels.

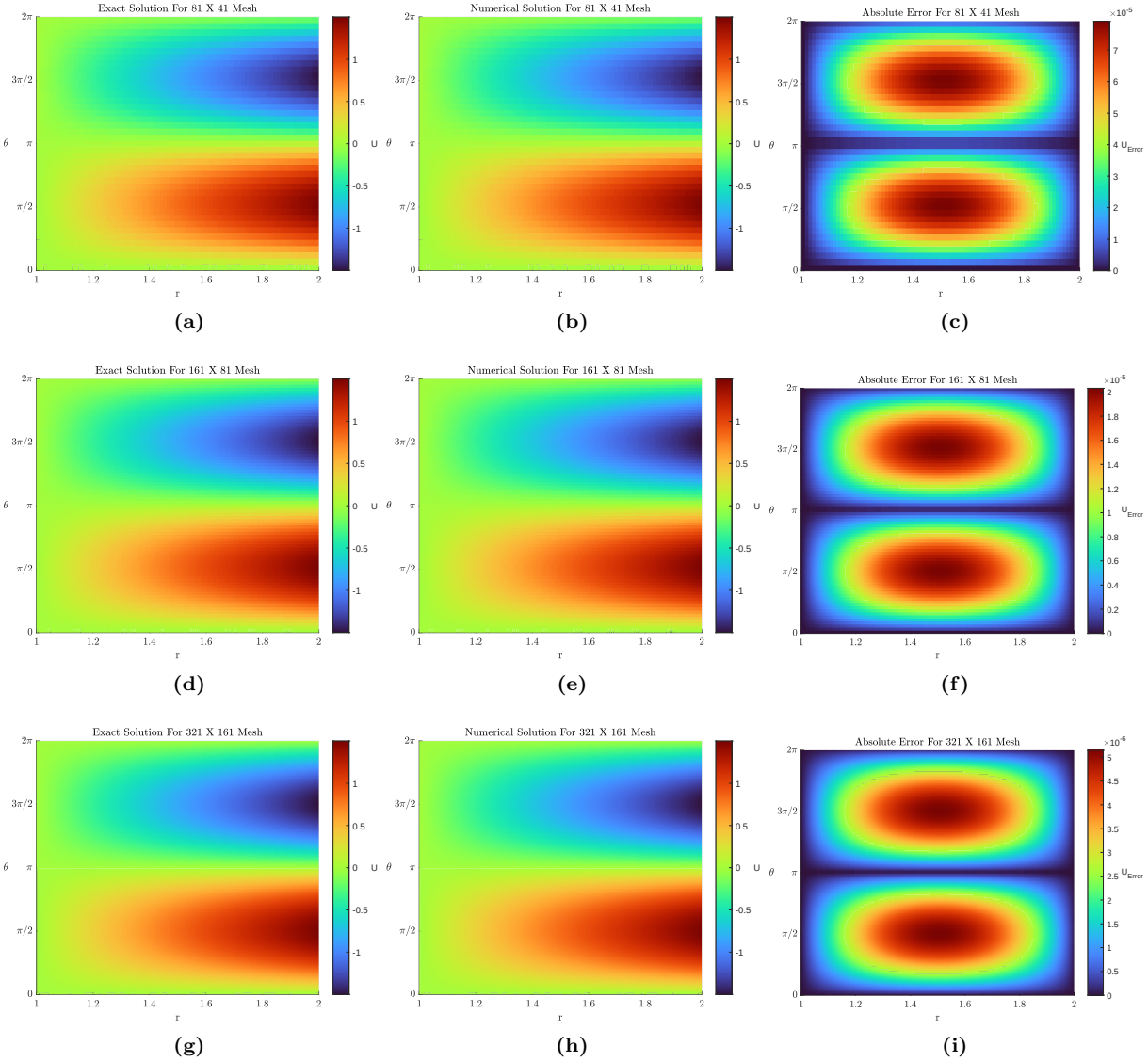


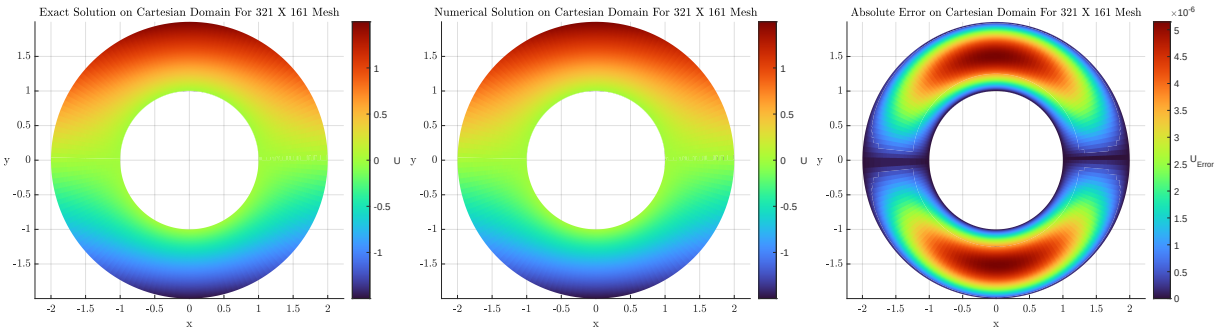
Figure 3.4: Visualization of the exact solution, numerical solution, and absolute error on the polar coordinate domain for different mesh resolutions. Subfigures (a), (d), and (g) present the exact solution; (b), (e), and (h) present the numerical solution; and (c), (f), and (i) present the absolute error. The first, second, and third rows correspond to the coarse, medium, and fine meshes, respectively.

The absolute error plots are especially useful because they show how the absolute error is distributed over the domain. Even when the numerical and analytical solution surfaces appear visually almost identical, the error plots still indicate the regions where the approximation is relatively less accurate. In this way, the figure provides not only the overall error level but also a visual representation of its spatial distribution.

The analytical solution, numerical solution, and absolute error distribution for the fine mesh are also displayed in the cartesian plane in Fig. 3.5. Although the governing equation is solved in polar coordinates, the computed field can be mapped onto the physical (x, y) plane through:

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

This representation places the results directly on the physical circular geometry of the problem. As a result, the spatial variation of the analytical solution, numerical solution, and absolute error around the cylinder can be interpreted more clearly than in the (r, θ) computational domain. The cartesian plots therefore provide a physical-domain visualization of the fine-mesh results.



(a) Analytical solution **(b)** Numerical solution **(c)** Absolute error

Figure 3.5: Analytical solution, numerical solution, and absolute error distribution represented on the cartesian domain for fine mesh.

CONCLUSION

In this study, the two dimensional Laplace equation in polar coordinates was solved using a second order finite difference method. The governing equation was discretized on a uniform mesh, and the resulting algebraic system was constructed by applying the finite difference equations at the interior nodes while the boundary conditions included into the right hand side vector. The system was solved using a sparse matrix formulation.

The results show that the numerical solution becomes more accurate as the mesh is refined. The error decreases consistently for all mesh levels, and the observed convergence rates are close to two in both radial and angular directions. This is consistent with the expected second order accuracy of the method.

The comparison between the numerical and analytical solutions also shows a good agreement. In addition, the visualised error plots indicate that the remaining error is small and smoothly distributed over the domain.

Overall, the implemented method gives a reliable and accurate solution for the problem. The results confirm that the numerical approach is working as expected.

BIBLIOGRAPHY

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

[2] The MathWorks, Inc., lu: LU Matrix Factorization, https://www.mathworks.com/help/matlab/ref/lu.html, accessed: 7 April 2026.