PPr701 Project Report on

Numerical Solution of the Poisson Equation in Cylindrical Geometry Using a Modal Decomposition Approach

Submitted by:
Madhurima Sarkar
Roll No: P0221617
Department of Physical Sciences
UM-DAE Centre for Excellence in Basic Sciences

Supervisor:
Prof. Bhooshan Paradkar
School of Physical Sciences
UM-DAE Centre for Excellence in Basic Sciences

Nurturing talent...promoting excellence

मौलिक विज्ञान प्रकर्ष केन्द्र

Certificate

This is to certify that Madhurima Sarkar, P0221617, student of UM-DAE CEBS, has undertaken project work from September 2025 to 30th November 2025 under the guidance of Prof. Bhooshan Paradkar, UM-DAE Centre for Excellence in Basic Sciences.

This submitted report titled **Numerical Solution of the Poisson Equation in Cylindrical Geometry using Azimuthal Modal Decomposition** is towards the academic requirements of the Integrated M.Sc. course at UM-DAE CEBS.

SMadhurima

Signature of Student

Signature of Guide

Madhurima Sarkar
UM-DAE CEBS

Prof. Bhooshan Paradkar
UM-DAE Centre for Excellence in Basic Sciences

Department of Physical Sciences
Centre for Excellence in Basic Sciences
Mumbai – 400098

Abstract

This report develops a Poisson solver tailored to the cylindrical, annular geometry of Hall thrusters using the azimuthal Fourier modal decomposition introduced by Lifschitz *et al.* [1]. Exploiting the natural periodicity in θ, the method reduces the full 3D (r, θ, z) problem to a small set of uncoupled 2D (r, z) equations, lowering computational cost from $O(N_r N_\theta N_z)$ to $O(N_r N_z M)$ with only a few modes required. Each modal equation is solved using second-order finite differences and a Successive Over-Relaxation (SOR) scheme. Benchmarking against the analytic $m = 0$ Laplace solution verifies correctness and near–second-order convergence. Higher modes reproduce the expected dipole and quadrupole structures, while the centrifugal term $-m^2/r^2$ improves diagonal dominance and accelerates convergence. Reconstructing the full potential shows that Hall-thruster electrostatics are dominated by the axisymmetric mode, with higher-mode corrections remaining small. These results demonstrate that modal decomposition provides an accurate and computationally efficient framework for Poisson solvers in Hall-thruster geometries.

Acknowledgements

I wholeheartedly thank Prof. Bhooshan Paradkar for his priceless inputs and guidance which led to the completion of this project. His constant guidance and inputs have been vital for the completion of this work. I thank him for his patience and reassurance which were showered on me in the project's entirety.

Lastly, I would like to thank my mother for her constant support and belief in all my academic endeavours.

Contents

Chapter 1

Introduction

The solution of Poisson's equation plays a central role in many areas of physics, including electrostatics, plasma dynamics, charged particle beams, and numerical simulation frameworks such as particle–in–cell (PIC) methods. In such systems, the electrostatic potential generated by a spatially varying charge density must often be computed repeatedly and with high accuracy. Because the Poisson equation is elliptic, its numerical solution can easily become a computational bottleneck, especially in three-dimensional geometries where the cost of standard solvers scales poorly with grid resolution.

For problems where the physics exhibits approximate or exact azimuthal symmetry, a more efficient formulation is obtained by decomposing the fields into Fourier harmonics in the azimuthal direction. This reduces the original three-dimensional Poisson equation to a set of uncoupled two-dimensional equations on an (r,z) grid. Each mode can be solved independently, and the resulting modal potentials can be recombined to reconstruct the full three-dimensional solution.

A powerful and widely used approach is the azimuthal modal decomposition introduced by Lifschitz *et al.* (J. Comput. Phys. **228**, 1804 (2009)). By expanding all fields in Fourier modes in the azimuthal direction, the original three-dimensional Poisson equation separates into a set of independent two-dimensional equations, one for each azimuthal harmonic. This formulation preserves the full three-dimensional structure of the electric potential while dramatically reducing the dimensionality and computational cost of the numerical solver. In particular, each mode can be treated independently, enabling efficient parallelization and simplified boundary treatments.

The original problem is three dimensional in(r,θ,z) and solving the Poisson equation directly in this geometry requires a three dimensional grid with computational cost scaling as $O(N_r N_\theta N_z) \sim O(N^3)$ for a uniform discretization. By exploiting the periodicity in the azimuthal direction and expanding all quantities in a Fourier series, the problem separates into independent equations for each azimuthal mode m. This effectively reduces the original 3D problem to a sequence of 2D problems posed on the (r, z) plane. The computational cost then becomes $O(N_r N_z M) \sim O(N^2 M)$, where M is the number of retained Fourier harmonics. In practice, only a few low-order modes are needed to capture the essential physics, which makes the modal approach substantially more efficient while still preserving the three-dimensional structure of the fields. For instance, if we use a 100 x 100 x 100 grid in (r,θ,z) , the computational cost will be of the order of $O(10^6)$, but instead if we use a 100 x 100 grid in (r,z) and compute for 10 modes to capture the physical picture , our computational cost will be $O(10^5)$, which is ten times lower than the full 3D computation.

1.1 Mathematical Method of Modal Decomposition of Electromagnetic Fields

The numerical model of Lifschitz *et al.* [1] is built around a cylindrical representation of space. In the paper, the laser pulse is assumed to be propagating along a distinguished axis, denoted by x, while the two directions perpendicular to the beam are described using the radial coordinate $r = \sqrt{y^2 + z^2}$ and the azimuthal angle θ.

The electric and magnetic fields are expressed in the cylindrical basis $(\hat{e}_r, \hat{e}_\theta, \hat{e}_x)$, which is related to the Cartesian vectors by

$$\hat{e}_r = \cos\theta\,\hat{y} + \sin\theta\,\hat{z}, \qquad \hat{e}_\theta = -\sin\theta\,\hat{y} + \cos\theta\,\hat{z}.$$

For a linearly polarized laser beam (polarization along y), the transverse fields take the form (as in Eq. (1) of the paper):

$$\mathbf{E}(r,\theta,x,t) = E_y(r,x,t)\,\hat{y} = E_y(r,x,t)\big(\cos\theta\,\hat{e}_r - \sin\theta\,\hat{e}_\theta\big), \tag{1.1}$$

$$\mathbf{B}(r,\theta,x,t) = B_z(r,x,t)\,\hat{z} = B_z(r,x,t)\big(\sin\theta\,\hat{e}_r + \cos\theta\,\hat{e}_\theta\big). \tag{1.2}$$

From these expressions, it is clear that a linearly polarized beam corresponds to a single azimuthal harmonic: the $\cos\theta$ and $\sin\theta$ factors indicate that the laser occupies the $m = 1$ Fourier mode.

Because fields and sources are periodic in θ, each quantity $F(r,\theta,x)$ is expanded as a Fourier series,

$$F(r,\theta,x) = \sum_{m=0}^{\infty} \tilde{F}_m(r,x)\,e^{-im\theta}, \tag{1.3}$$

following Eq. (2) of [1]. Each Fourier coefficient $\tilde{F}_m(r,x)$ represents a two–dimensional field defined on the (r,x) plane.

$$E_{1r} = a\cos\theta = \mathrm{Re}(a(\cos\theta - i\sin\theta)) = \mathrm{Re}(ae^{-i\theta}) \tag{1.4}$$

$$E_{1\theta} = -a\sin\theta = \mathrm{Re}(-ia(\cos\theta - i\sin\theta)) = \mathrm{Re}(-ia(e^{-i\theta})) \tag{1.5}$$

$$B_{1r} = a\sin\theta = \mathrm{Re}(ia(\cos\theta - i\sin\theta)) = \mathrm{Re}(ia(e^{-i\theta})) \tag{1.6}$$

$$B_{1\theta} = a\cos\theta = \mathrm{Re}(a(\cos\theta - i\sin\theta)) = \mathrm{Re}(a(e^{-i\theta})) \tag{1.7}$$

$$\tilde{E_{1r}} = a; \tilde{E_{1\theta}} = -ia; \tilde{B_{1r}} = ia; \tilde{B_{1\theta}} = a \tag{1.8}$$

Using the following Maxwell's Equation,

$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t}, \tag{1.9}$$

$$\nabla \times \mathbf{B} = \mathbf{J} + \frac{\partial \mathbf{E}}{\partial t}. \tag{1.10}$$

$$\nabla \times \mathbf{E} = \frac{1}{r}\begin{vmatrix} \hat{\mathbf{e}}_r & r\hat{\mathbf{e}}_\theta & \hat{\mathbf{e}}_x \\ \dfrac{\partial}{\partial r} & \dfrac{\partial}{\partial \theta} & \dfrac{\partial}{\partial x} \\ E_r & rE_\theta & E_x \end{vmatrix} =$$

$\left(\frac{1}{r}\left(\frac{\partial E_x}{\partial\theta}-\frac{\partial(rE_\theta)}{\partial x}\right)\right)\hat{\mathbf{e}}_r+\left(\frac{\partial E_r}{\partial x}-\frac{\partial E_x}{\partial r}\right)\hat{\mathbf{e}}_\theta+\left(\frac{1}{r}\left(\frac{\partial(rE_\theta)}{\partial r}-\frac{\partial E_r}{\partial\theta}\right)\right)\hat{\mathbf{e}}_x$. Equating component wise, we get the following

$$\left(\frac{1}{r}\left(\frac{\partial E_x}{\partial\theta}-\frac{\partial(rE_\theta)}{\partial x}\right)\right)=-\frac{\partial B_r}{\partial t}$$

$$\left(\frac{\partial E_r}{\partial x}-\frac{\partial E_x}{\partial r}\right)=-\frac{\partial B_\theta}{\partial t}$$

$$\left(\frac{1}{r}\left(\frac{\partial(rE_\theta)}{\partial r}-\frac{\partial E_r}{\partial\theta}\right)\right)=-\frac{\partial B_x}{\partial t}.$$

Using

$$\frac{\partial\tilde{F}_m e^{-im\theta}}{\partial\theta}=(-im)\tilde{F}_m e^{-im\theta}$$

and substituting it into the above equations, we get :

$$\boxed{\frac{im\,\widetilde{E}_x^m}{r}+\frac{\partial\widetilde{E}_\theta^m}{\partial x}=\frac{\partial\widetilde{B}_r^m}{\partial t}}$$

$$\boxed{-\frac{1}{r}\frac{\partial(r\widetilde{E}_\theta^m)}{\partial r}-\frac{im\,\widetilde{E}_r^m}{r}=\frac{\partial\widetilde{B}_x^m}{\partial t}}$$

$$\boxed{-\frac{\partial\widetilde{E}_r^m}{\partial x}+\frac{\partial\widetilde{E}_x^m}{\partial r}=\frac{\partial\widetilde{B}_\theta^m}{\partial t}}$$

Similarly for the other equation

$$\nabla\times\mathbf{B}-\mathbf{J}=\frac{\partial\mathbf{E}}{\partial t}$$

, we do the following treatment:

$$\nabla\times\mathbf{B}=\frac{1}{r}\begin{vmatrix}\hat{\mathbf{e}}_r & r\hat{\mathbf{e}}_\theta & \hat{\mathbf{e}}_x\\ \frac{\partial}{\partial r} & \frac{\partial}{\partial\theta} & \frac{\partial}{\partial x}\\ B_r & rB_\theta & B_x\end{vmatrix}.$$

Equating component wise, we get

$$\frac{1}{r}\left(\frac{\partial B_x}{\partial\theta}-\frac{\partial(rB_\theta)}{\partial x}\right)-J_r=\frac{\partial E_r}{\partial t},$$
$$\frac{\partial B_r}{\partial x}-\frac{\partial B_x}{\partial r}-J_\theta=\frac{\partial E_\theta}{\partial t},$$
$$\frac{1}{r}\left(\frac{\partial(rB_\theta)}{\partial r}-\frac{\partial B_r}{\partial\theta}\right)-J_x=\frac{\partial E_x}{\partial t}.$$

Using

$$\frac{\partial\tilde{F}_m e^{-im\theta}}{\partial\theta}=(-im)\tilde{F}_m e^{-im\theta}$$

and substituting it into the above equations, we get :

$$\boxed{\frac{-im\,\widetilde{B}_x^m}{r} - \frac{1}{r}\frac{\partial(r\widetilde{B}_\theta^m)}{\partial x} - \widetilde{J}_r^m = \frac{\partial \widetilde{E}_r^m}{\partial t}}$$

$$\boxed{\frac{\partial \widetilde{B}_r^m}{\partial x} - \frac{\partial \widetilde{B}_x^m}{\partial r} - \widetilde{J}_\theta^m = \frac{\partial \widetilde{E}_\theta^m}{\partial t}}$$

$$\boxed{\frac{1}{r}\frac{\partial(r\widetilde{B}_\theta^m)}{\partial r} + \frac{im\,\widetilde{B}_r^m}{r} - \widetilde{J}_x^m = \frac{\partial \widetilde{E}_x^m}{\partial t}}$$

Chapter 2

Generalization to the Electrostatic Case

2.1 Geometry of a Hall Thruster and Motivation for the Cylindrical Formulation

The cylindrical formulation follows directly from the geometry of a Hall thruster, whose discharge channel is annular: the radial direction spans the channel width, the axial direction aligns with ion acceleration, and the azimuthal direction hosts the characteristic E×B electron drift. Because the walls correspond to surfaces of constant radius, and because plasma quantities typically vary in r and z with only structured or weak dependence on θ, cylindrical coordinates are the most natural description.

This approximate azimuthal periodicity makes a Fourier decomposition in θ especially efficient. It allows the 3D problem to be separated into independent azimuthal modes, retaining non-axisymmetric structure without the cost of a full 3D simulation.

Electrostatics plays a central role in Hall thruster physics: ions are accelerated almost entirely by the axial electric field, while the magnetic field primarily shapes electron transport. As a result, obtaining the self-consistent electrostatic potential $\phi(r, \theta, z)$ through Poisson's equation is fundamental to predicting ion acceleration, electron dynamics, and overall discharge behaviour.

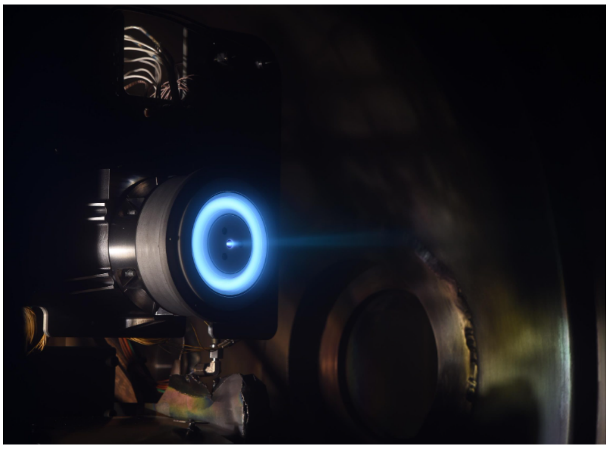


Figure 2.1: Image of a Hall Thruster. Credit: Northrop Grumman. Ref: [4]

2.2 Non-dimensional form of Poisson's Equation

In electrostatics, the potential generated by a charge density is governed by Poisson's equation. In Gaussian units, the dimensional form reads

$$\nabla^2\Phi(\mathbf{x}) = -4\pi\,\rho(\mathbf{x}), \tag{2.1}$$

where Φ is the electrostatic potential and ρ is the charge density. For numerical computations, it is often convenient to eliminate the physical constant and work with nondimensional quantities. This simplifies the equation and aligns with the normalized formulation employed by Lifschitz *et al.* (J. Comput. Phys. **228**, 1804 (2009)).

To nondimensionalize the equation,

$$e\left[\nabla^2\phi = -4\pi\rho = -4\pi e\,(n_i - n_e)\right] \tag{2.2}$$

$$\frac{\nabla^2(e\phi)}{m_e v_{th}^2} = -\frac{4\pi e^2 n_o}{m_e v_{th}^2}\left(\frac{n_i}{n_o} - \frac{n_e}{n_o}\right) \tag{2.3}$$

where $\dfrac{n_i}{n_0} = \tilde{n}_i$ and $\dfrac{n_e}{n_0} = \tilde{n}_e$

$$\nabla^2\left(\frac{e\phi}{m_e v_{th}^2}\right) = -\left(\frac{4\pi n_o e^2}{m_e}\right)\frac{1}{v_{th}^2}\left(\tilde{n}_i - \tilde{n}_e\right) \tag{2.4}$$

where $\tilde{\phi} = \dfrac{e\phi}{m_e v_{th}^2}$.

The plasma frequency $\omega_{pe}^2 = \dfrac{4\pi n_o e^2}{m_e}$ and the Debye length λ_{De} are related by:

$$\frac{\omega_{pe}^2}{v_{th}^2} = \frac{1}{\lambda_{De}^2} \tag{2.5}$$

So we can write $\tilde{\nabla}^2 = \lambda_{De}^2\nabla^2$. Using this , the equation can be written as:

$$\nabla^2(\tilde{\phi}) = -\frac{1}{\lambda_{De}^2}\left(\tilde{n}_i - \tilde{n}_e\right) \tag{2.6}$$

$$\left(\lambda_{De}\nabla\right)^2\tilde{\phi} = -\left(\tilde{n}_i - \tilde{n}_e\right) \tag{2.7}$$

$$\tilde{\nabla}^2\tilde{\phi} = -\left(\tilde{n}_i - \tilde{n}_e\right) = -\tilde{\rho} \tag{2.8}$$

We can drop the ˜ sign for convenience.

$$\boxed{\nabla^2\phi = -\,\rho}. \tag{2.9}$$

This nondimensional form is algebraically simpler and will be used throughout this report.

2.3 Generalization of the Method for Solving the Poisson Equation

To solve our non-dimensional Poisson's equation,

$$\nabla^2\phi = -\,\rho, \tag{2.10}$$

we use the Fourier decomposition in azimuthal angle method described in the section 1.1.

We use

$$\phi = \sum_m \tilde{\phi}_m(r,z)e^{-im\theta}$$

and

$$\rho = \sum_m \tilde{\rho}_m(r,z)e^{-im\theta}$$

$$\frac{\partial\tilde{\phi}_m e^{-im\theta}}{\partial\theta} = (-im)\tilde{\phi}_m e^{-im\theta} \tag{2.11}$$

Writing out the Laplacian in cylindrical coordinates, we get

$$\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial\phi}{\partial r}\right) + \frac{1}{r^2}\frac{\partial^2\phi}{\partial\theta^2} + \frac{\partial^2\phi}{\partial z^2} = -\rho \tag{2.12}$$

$$\frac{1}{r}\frac{\partial\phi}{\partial r} + \frac{\partial^2\phi}{\partial r^2} + \frac{1}{r^2}\frac{\partial^2\phi}{\partial\theta^2} + \frac{\partial^2\phi}{\partial z^2} = -\rho \tag{2.13}$$

Substituting into the equation above

$$\phi = \sum_m \tilde{\phi}_m(r,z)e^{-im\theta}$$

and

$$\rho = \sum_m \tilde{\rho}_m(r,z)e^{-im\theta}$$

, we get the following sets of equations:

$$\frac{1}{r}\frac{\partial\tilde{\phi}_m}{\partial r} + \frac{\partial^2\tilde{\phi}_m}{\partial r^2} + \frac{1}{r^2}\frac{\partial^2\tilde{\phi}_m}{\partial\theta^2} + \frac{\partial^2\phi}{\partial z^2} = -\rho \tag{2.14}$$

$$\frac{1}{r}\frac{\partial\tilde{\phi}_m}{\partial r} + \frac{\partial^2\tilde{\phi}_m}{\partial r^2} + \frac{(-im)^2}{r^2}\tilde{\phi}_m + \frac{\partial^2\phi}{\partial z^2} = -\rho \tag{2.15}$$

Finally we get the following equation:

$$\boxed{\frac{1}{r}\frac{\partial\tilde{\phi}_m}{\partial r} + \frac{\partial^2\tilde{\phi}_m}{\partial r^2} - \frac{m^2}{r^2}\tilde{\phi}_m + \frac{\partial^2\tilde{\phi}_m}{\partial z^2} = -\,\tilde{\rho}_m}$$

Chapter 3

Numerical Implementation on the (r,z) grid

Using the finite difference method, we discretize the Poisson's equation after azimuthal Fourier decomposition in a 2D grid in(r,z).

$$r_{\,i} = i\,\Delta r, \qquad i = 0, 1, \ldots, N_r \tag{3.1}$$

$$z_{\,j} = j\,\Delta z, \qquad j = 0, 1, \ldots, N_z \tag{3.2}$$

$$\tilde{\phi}_m(r_i, z_j) = \phi_{i,j}$$

and

$$\tilde{\rho}_m(r_i, z_j) = \rho_{i,j}$$

We consider the grid spacing in r to be Δr. Writing $\phi(r + \Delta r)$ and $\phi(r - \Delta r)$ and expanding about r, we have :

$$\phi(r + \Delta r) = \phi(r) + \phi'(r)\Delta r + \phi''(r)\frac{(\Delta r)^2}{2!} + \ldots \tag{3.3}$$

$$\phi(r - \Delta r) = \phi(r) - \phi'(r)\Delta r + \phi''(r)\frac{(\Delta r)^2}{2!} + \ldots \tag{3.4}$$

Adding Eq. 3.3 and Eq. 3.4, we get

$$\phi(r + \Delta r) + \phi(r - \Delta r) = 2\phi(r) + \phi''(r)(\Delta r)^2 \tag{3.5}$$

Rearranging the terms, we get

$$\frac{\phi(r + \Delta r) - 2\phi(r) + \phi(r - \Delta r)}{(\Delta r)^2} = \phi''(r) \tag{3.6}$$

So, we can write

$$\frac{\partial^2 \phi}{\partial r^2} = \frac{\phi_{i+1,j} - 2\phi_{i,j} + \phi_{i-1,j}}{(\Delta r)^2} \tag{3.7}$$

Similarly ,

$$\frac{\partial^2 \phi}{\partial z^2} = \frac{\phi_{i+1,j} - 2\phi_{i,j} + \phi_{i-1,j}}{(\Delta z)^2} \tag{3.8}$$

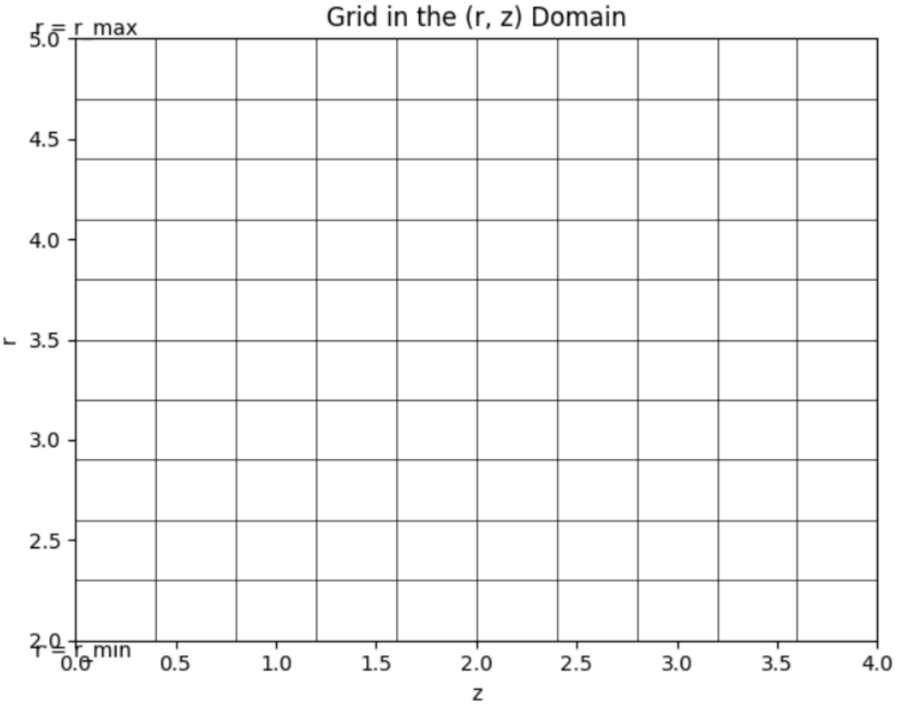


Figure 3.1: a sample 10 x 10 grid with r$_{min} = 2.0\ and\ r_{max} = 5.0\ and\ z_{min} = 0\ and\ z_{max} = 4$.

Using the standard central difference formula, we get

$$\frac{\partial \phi}{\partial r} = \frac{\phi_{i+1,j} - \phi_{i-1,j}}{2\Delta r} \tag{3.9}$$

So substituting the above into

$$\left(\frac{\partial^2}{\partial r^2} + \frac{1}{r}\frac{\partial}{\partial r} + \frac{\partial^2}{\partial z^2} - \frac{m^2}{r^2} \right) \phi = -\rho \tag{3.10}$$

We get

$$\frac{\phi_{i+1,j} - 2\phi_{i,j} + \phi_{i-1,j}}{(\Delta r)^2} \quad + \quad \frac{1}{r_i}\frac{\phi_{i+1,j} - \phi_{i-1,j}}{2\Delta r} \quad + \quad \frac{\phi_{i+1,j} - 2\phi_{i,j} + \phi_{i-1,j}}{(\Delta z)^2} \quad - \quad \frac{m^2}{r_i^2} = -\rho_{i,j}$$

Rearranging and collecting similar terms together , we can write

$$\begin{aligned} &\phi_{i+1,j}\left(\frac{1}{(\Delta r)^2} + \frac{1}{2r_i\,\Delta r}\right) + \phi_{i-1,j}\left(\frac{1}{(\Delta r)^2} - \frac{1}{2r_i\,\Delta r}\right) \\ &\quad + \phi_{i,j}\left(-\frac{2}{(\Delta r)^2} - \frac{2}{(\Delta z)^2} - \frac{m^2}{r_i^2}\right) + \frac{1}{(\Delta z)^2}\big(\phi_{i,j+1} + \phi_{i,j-1}\big) = -\rho_{i,j}. \end{aligned} \tag{3.11}$$

$$\begin{aligned} A_i^{(+r)} &= \boxed{\frac{1}{(\Delta r)^2} \; + \; \frac{1}{2\,\Delta r\, r_i}}, \\ A_i^{(0)} &= \boxed{-\frac{2}{(\Delta r)^2} \; - \; \frac{2}{(\Delta z)^2} \; - \; \frac{m^2}{r_i^2}}, \\ A_i^{(-r)} &= \boxed{\frac{1}{(\Delta r)^2} \; - \; \frac{1}{2\,\Delta r\, r_i}}. \end{aligned} \tag{3.12}$$

So we can the full discrete equation as :

$$\boxed{A_i^{(-r)}\phi_{i-1,j} + A_i^{(0)}\phi_{i,j} + A_i^{(+r)}\phi_{i+1,j} + \frac{1}{(\Delta z)^2}\left(\phi_{i,j-1} + \phi_{i,j+1}\right) = -\rho_{i,j}} \tag{3.13}$$

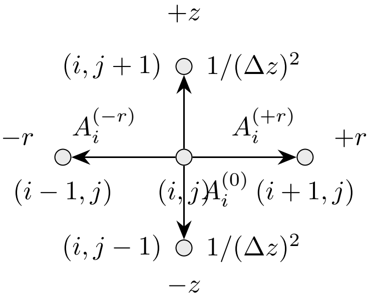


Figure 3.2: Five-point finite-difference stencil for the (r, z) discretization of the $m = 0$ Poisson equation. Thicker arrows highlight the coupling between neighboring grid points.

Chapter 4

Numerical Method

In this section, we present the numerical technique used to obtain the potential for the $m = 0$ mode of the discretized Poisson equation in cylindrical coordinates after azimuthal Fourier decomposition.

Since the system is periodic along the z–axis, the potential satisfies $\phi(r, z + L_z) = \phi(r, z)$, while in the radial direction the boundary conditions are $\phi(r = 0) = \phi_0$ and $\phi(r = R) = 0$, corresponding to a fixed inner potential and a grounded outer wall.

4.1 Discrete form of the $m = 0$ equation

The generalised modal Poisson equation derived in Chapter 2 , for any mode m is as follows:

$$\frac{1}{r}\frac{\partial\tilde{\phi}_m}{\partial r} + \frac{\partial^2\tilde{\phi}_m}{\partial r^2} - \frac{m^2}{r^2}\tilde{\phi}_m + \frac{\partial^2\tilde{\phi}_m}{\partial z^2} = -\,\tilde{\rho}_m \tag{4.1}$$

For the $m = 0$ azimuthal harmonic, the modal Poisson equation reduces to

$$\frac{1}{r}\frac{\partial\tilde{\phi}}{\partial r} + \frac{\partial^2\tilde{\phi}}{\partial r^2} + \frac{\partial^2\tilde{\phi}}{\partial z^2} = -\,\tilde{\rho}. \tag{4.2}$$

Discretizing on the (r, z) grid introduced in Chapter 3, the equation at grid point (i, j) takes the form

$$A_i^{(-r)}\phi_{i-1,j} + A_i^{(0)}\phi_{i,j} + A_i^{(+r)}\phi_{i+1,j} + \frac{1}{(\Delta z)^2}\left(\phi_{i,j-1} + \phi_{i,j+1}\right) = -\rho_{i,j}, \tag{4.3}$$

where the finite–difference coefficients are defined as

$$\begin{aligned} A_i^{(+r)} &= \frac{1}{(\Delta r)^2} + \frac{1}{2r_i\Delta r}, \\ A_i^{(-r)} &= \frac{1}{(\Delta r)^2} - \frac{1}{2r_i\Delta r}, \\ A_i^{(0)} &= -\frac{2}{(\Delta r)^2} - \frac{2}{(\Delta z)^2}. \end{aligned} \tag{4.4}$$

This system can be expressed compactly as a sparse linear equation

$$\mathbf{A}\boldsymbol{\phi} = \mathbf{b}, \tag{4.5}$$

where $\mathbf{A}$ is a symmetric, diagonally dominant matrix corresponding to the discrete Laplacian operator. Direct inversion of $\mathbf{A}$ is computationally expensive; hence we employ the successive over-relaxation technique.

4.2 Implementation of the SOR Method

The discretized Poisson equation for the $m = 0$ mode in cylindrical coordinates takes the five-point finite-difference form

$$A_i^{(-r)}\phi_{i-1,j} + A_i^{(0)}\phi_{i,j} + A_i^{(+r)}\phi_{i+1,j} + \frac{1}{(\Delta z)^2}\left(\phi_{i,j-1} + \phi_{i,j+1}\right) = -\rho_{i,j}, \tag{4.6}$$

where $A_i^{(\pm r)}$ and $A_i^{(0)}$ are the radial coefficients determined from the finite-difference discretization of the Laplacian operator in cylindrical geometry. The indices (i, j) correspond to the grid points along the radial (r) and axial (z) directions, respectively.

The Successive Over-Relaxation (SOR) algorithm provides an efficient iterative scheme for solving Eq. (4.6) without explicitly inverting the sparse matrix associated with the finite-difference operator. Starting from an initial guess $\phi_{i,j}^{(0)}$, the potential is updated iteratively according to

$$\phi_{i,j}^{(k+1)} = \phi_{i,j}^{(k)} + \omega \frac{-\rho_{i,j} - \left[A_i^{(-r)}\phi_{i-1,j}^{(k+1)} + A_i^{(+r)}\phi_{i+1,j}^{(k)} + \frac{1}{(\Delta z)^2}\left(\phi_{i,j-1}^{(k+1)} + \phi_{i,j+1}^{(k)}\right) + A_i^{(0)}\phi_{i,j}^{(k)}\right]}{A_i^{(0)}}, \tag{4.7}$$

where ω is the over-relaxation parameter $(1 < \omega < 2)$ and the terms with superscript $(k+1)$ correspond to points that have already been updated during the current iteration following the Gauss–Seidel ordering.

Boundary Conditions

The potential at the boundaries is specified as:

$$\phi(0, z) = \phi_0, \qquad \phi(R, z) = 0,$$

and periodicity is imposed along z as

$$\phi(r, 0) = \phi(r, L_z), \quad \phi(r, -\Delta z) = \phi(r, L_z - \Delta z).$$

These conditions are enforced after each iteration sweep, ensuring that the updated potential remains consistent with the physical constraints of the problem.

Convergence Criterion

At the end of each iteration k, the local residual

$$r_{i,j}^{(k)} = -\rho_{i,j} - \left[A_i^{(-r)} \phi_{i-1,j}^{(k)} + A_i^{(0)} \phi_{i,j}^{(k)} + A_i^{(+r)} \phi_{i+1,j}^{(k)} + \frac{1}{(\Delta z)^2} \left(\phi_{i,j-1}^{(k)} + \phi_{i,j+1}^{(k)} \right) \right] \tag{4.8}$$

is computed. In the implementation used here, convergence is monitored by tracking how much the solution changes between successive SOR iterations. At each iteration k, the updated potential field $\phi^{(k)}$ is compared with the previous iterate $\phi^{(k-1)}$, and the residual is defined as the relative update norm

$$\text{residual} = \frac{\|\phi^{(k)} - \phi^{(k-1)}\|_2}{\|\phi^{(k-1)}\|_2}.$$

The solver is considered to have converged once this quantity falls below the chosen tolerance which is chosen to be 1e-07 for our case.

Algorithmic Procedure

In practice, the implementation proceeds as follows:

1. Initialize the grid arrays (r_i, z_j), and specify $\phi_{i,j}^{(0)}$.

2. At each iteration k, perform a nested loop over i and j applying Eq. (4.7) to update $\phi_{i,j}$.

3. Enforce the boundary conditions at $r = r_{min}$, $r = r_{max}$, and across $z = 0$ and $z = L_z$.

4. After completing one full sweep, compute the residual norm and check for convergence.

5. If convergence is not achieved, increment k and repeat.

A schematic representation of the SOR workflow is shown in Fig. 4.1. The choice of the relaxation parameter ω significantly influences convergence speed; empirically, values in the range $1.5 \leq \omega \leq 1.9$ yield rapid convergence for uniform grids of moderate aspect ratio.

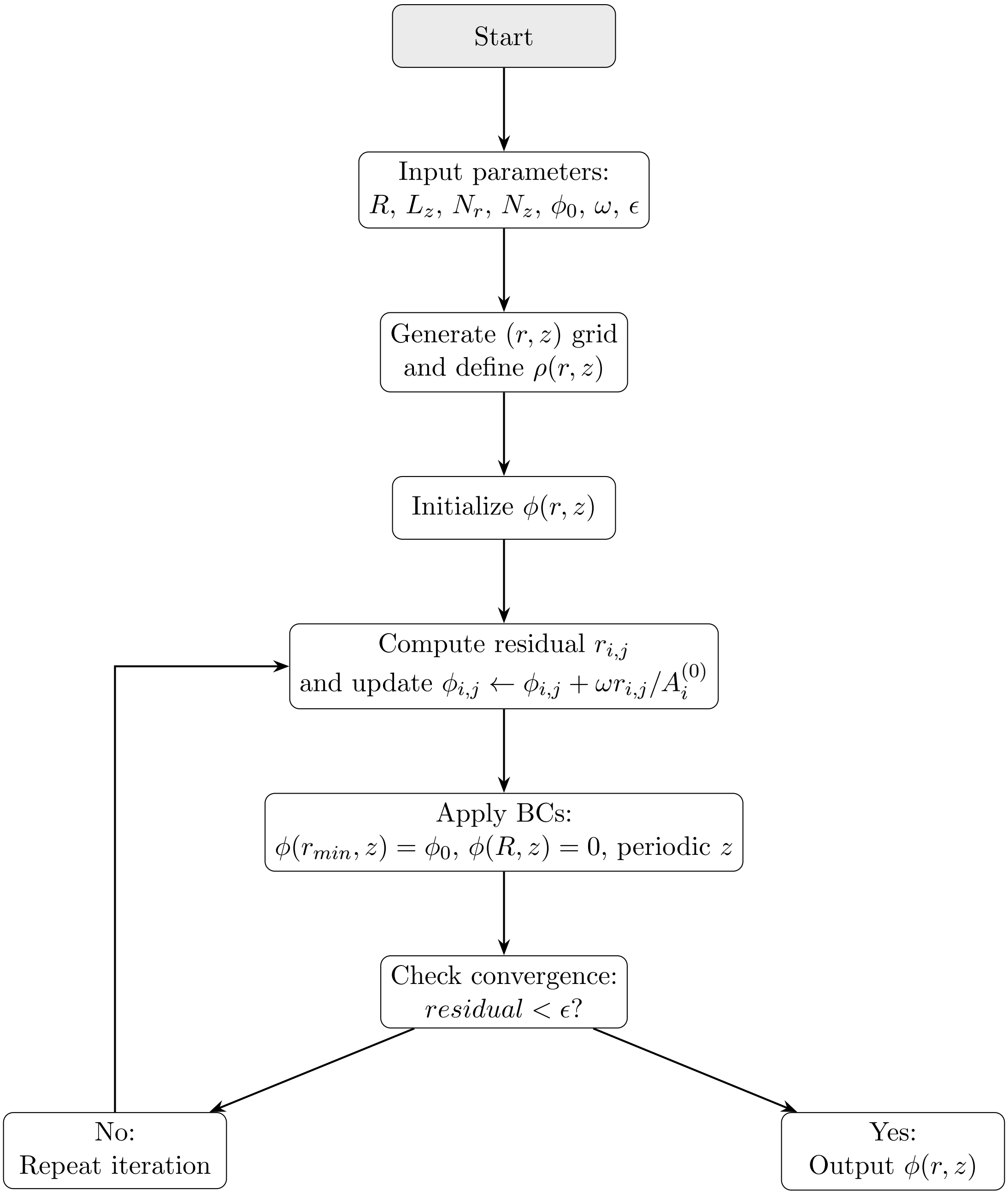

Start
Input parameters:
R, L_z, N_r, N_z, ϕ_0, ω, ϵ
Generate (r, z) grid
and define $\rho(r, z)$
Initialize $\phi(r, z)$
Compute residual $r_{i,j}$
and update $\phi_{i,j} \leftarrow \phi_{i,j} + \omega r_{i,j}/A_i^{(0)}$
Apply BCs:
$\phi(r_{min}, z) = \phi_0$, $\phi(R, z) = 0$, periodic z
Check convergence:
$residual < \epsilon$?
No:
Repeat iteration
Yes:
Output $\phi(r, z)$

Figure 4.1: Flow diagram of the SOR-based solver for the $m = 0$ modal Poisson equation.

Chapter 5

Results

In this section we present the numerical results obtained using the Successive Over-Relaxation (SOR) scheme applied to the discretised $m = 0$ modal Poisson equation in cylindrical coordinates. All simulations were performed on a uniform (r, z) grid with periodic boundary conditions along the axial direction and Dirichlet conditions at the radial boundaries. The results are organised as follows: first, we verify the solver using a Laplace problem with a known analytic solution; next, we analyse the convergence of the SOR method; finally, we examine the full two-dimensional potential distribution and the effect of grid refinement.

5.0.1 Verification Using the Laplace Problem ($\rho = 0$)

To check that our Poisson solver is implemented correctly, we first test it on the simplest case: the $m = 0$ mode with zero charge density ($\rho = 0$). For this benchmark, we choose $r_{\text{min}} = 2$ as the inner radius and $r_{\text{max}} = 5$ as the outer radius. The system is taken to be periodic along the z-direction, and the outer boundary at $r = r_{\text{max}}$ is set to ground. We have chosen the SOR relaxation parameter ω to be 1.8 . The grid size is taken to be 100 × 100.

For the axisymmetric $m = 0$ case with no charge ($\rho = 0$), and with boundary conditions $\phi(r_{\text{min}}) = 1$ and $\phi(r_{\text{max}}) = 0$ (outer wall grounded), the Laplace equation in cylindrical coordinates reduces to

$$\frac{1}{r}\frac{d}{dr}\left(r\frac{d\phi}{dr}\right) + \frac{\partial^2 \phi}{\partial z^2} = 0.$$

If the boundary conditions and the source are independent of z (and z is periodic), the physical solution has no z-dependence. Therefore $\partial^2\phi/\partial z^2 = 0$ and the equation becomes the ordinary differential equation

$$\frac{1}{r}\frac{d}{dr}\left(r\frac{d\phi}{dr}\right) = 0.$$

Integrate once with respect to r:

$$r\frac{d\phi}{dr} = C_1 \quad \Rightarrow \quad \frac{d\phi}{dr} = \frac{C_1}{r}.$$

Integrate a second time:

$$\phi(r) = C_1 \ln r + C_2,$$

where C_1 and C_2 are integration constants determined from the Dirichlet boundary conditions. Applying $\phi(r_{\min}) = 1$ and $\phi(r_{\max}) = 0$ gives the linear system

$$\begin{cases} C_1 \ln r_{\min} + C_2 = 1, \\ C_1 \ln r_{\max} + C_2 = 0. \end{cases}$$

Subtracting the two equations yields

$$C_1(\ln r_{\max} - \ln r_{\min}) = -1 \quad \Rightarrow \quad C_1 = -\frac{1}{\ln(r_{\max}/r_{\min})}.$$

Then

$$C_2 = 1 - C_1 \ln r_{\min} = 1 + \frac{\ln r_{\min}}{\ln(r_{\max}/r_{\min})}.$$

A more compact and often convenient form for the analytic solution is

$$\boxed{\phi_{\text{ana}}(r) \;=\; \frac{\ln\bigl(r_{\max}/r\bigr)}{\ln\bigl(r_{\max}/r_{\min}\bigr)}},$$

which manifestly satisfies $\phi(r_{\min}) = 1$ and $\phi(r_{\max}) = 0$.

Numeric example. For $r_{\min} = 2$ and $r_{\max} = 5$,

$$C_1 = -\frac{1}{\ln(5/2)} \approx -\frac{1}{0.9162907} \approx -1.0911,$$

$$C_2 = 1 - C_1 \ln 2 \approx 1 + \frac{\ln 2}{\ln(5/2)} \approx 1.7561.$$

So numerically

$$\phi_{\text{ana}}(r) \approx -1.0911 \ln r + 1.7561,$$

which is algebraically identical to the boxed compact form above.

Figure 5.1 shows a comparison between the analytic solution and the numerical mid-z slice of $\phi(r, z)$. Excellent agreement is observed across the entire radial domain. The convergence behaviour for SOR relaxation parameter ω=1.8 is shown in Figure 5.2 .For a 100 × 100 grid, it takes 1825 iterations to converge.

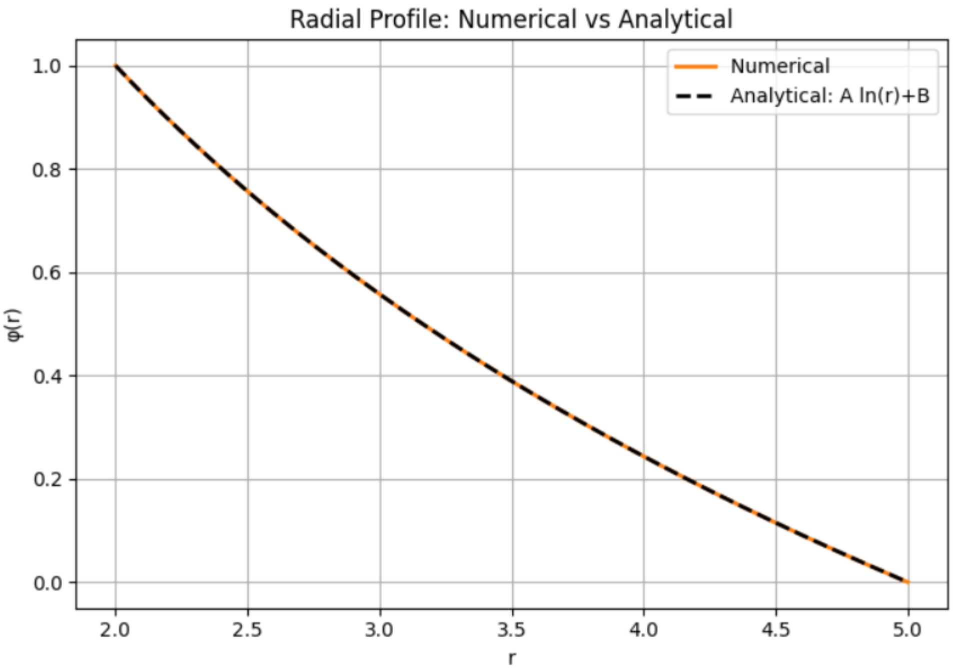


Figure 5.1: Comparison of the numerical potential profile $\phi_{\text{num}}(r)$ with the analytic solution $\phi_{\text{ana}}(r)$ for the Laplace case.

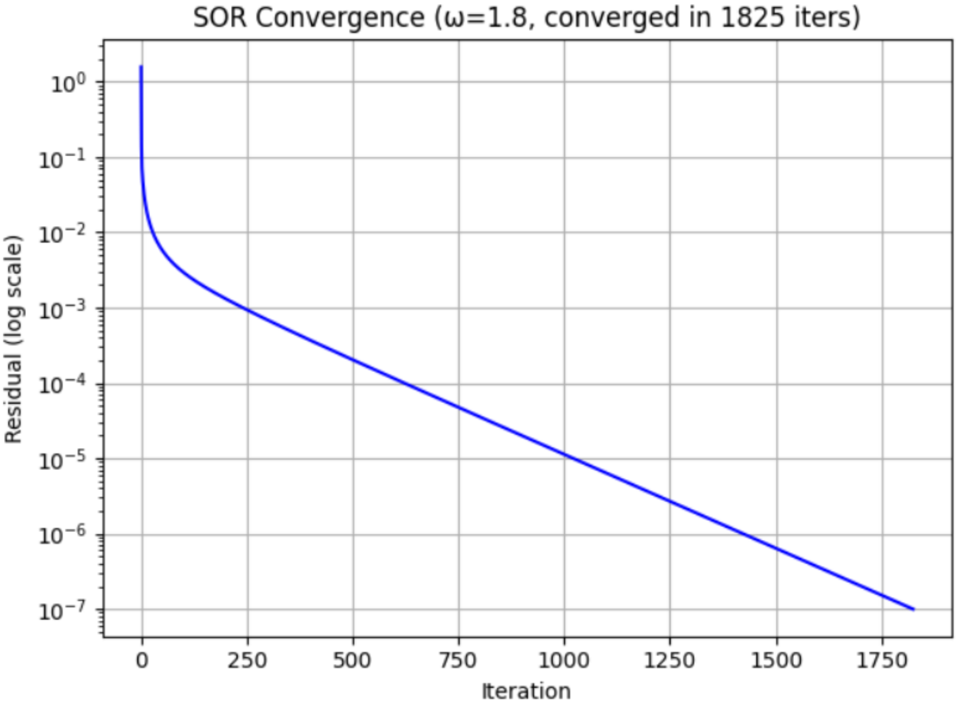


Figure 5.2: Convergence behaviour for SOR relaxation parameter ω =1.8 for a 100 × 100 grid.

5.0.2 Convergence Behaviour of the SOR Method

The convergence of the SOR scheme was examined by monitoring the decay of the the residual which in our case is the relative update norm

$$\text{residual} = \frac{\|\phi^{(k)} - \phi^{(k-1)}\|_2}{\|\phi^{(k-1)}\|_2}.$$

We vary the SOR relaxation parameter ω from 1.0 to 1.9 in units of 0.1 and see how many iterations it takes to converge. Our tolerance limit is set at 1e-07 and convergence takes places when residual < tolerance. The residual norm $\|r\|_2$ decreases monotonically for stable relaxation parameters. Figure 5.3 shows the decay of the residual as a function of iteration number for several choices of the relaxation parameter ω.

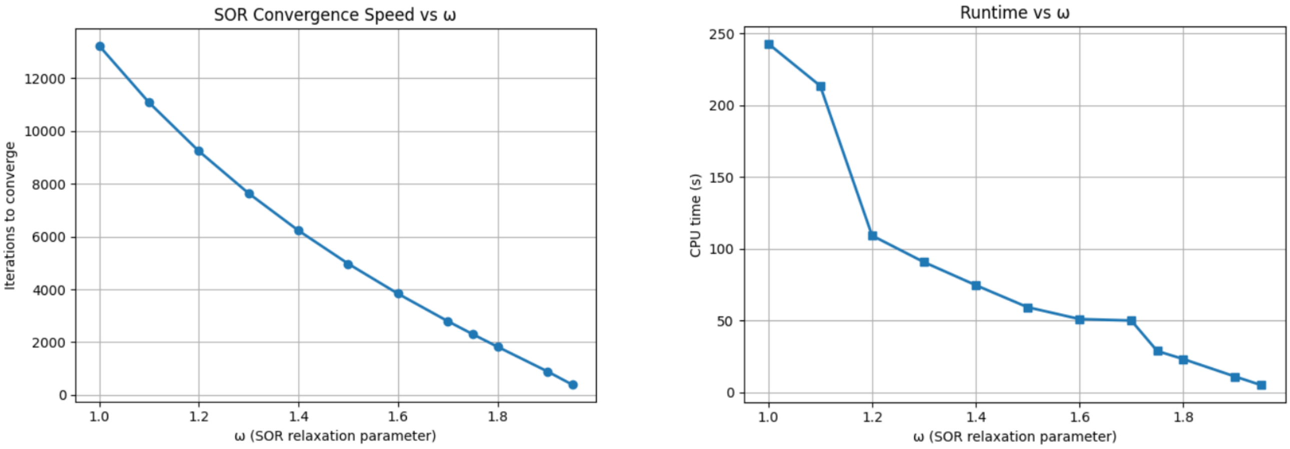


Figure 5.3: (Left)Iterations taken to converge versus different values of the relaxation parameter ω. (Right) The graph shows the CPU time in seconds vs different values of ω.

The total number of iterations required for convergence is summarised in Table 5.1. An optimal relaxation parameter in the range $1.6 \leq \omega \leq 1.9$ is found, consistent with expectations for elliptic problems.

Table 5.1: SOR Convergence Study: Number of Iterations and CPU Time for Different Relaxation Parameters ω

ω	Iterations	CPU time (s)
1.00	13238	242.898
1.10	11087	213.572
1.20	9245	109.286
1.30	7644	90.808
1.40	6233	74.744
1.50	4974	59.384
1.60	3837	51.137
1.70	2795	50.083
1.75	2303	28.863
1.80	1825	23.302
1.90	892	11.115
1.95	389	5.192

5.0.3 Two-Dimensional Potential Distribution

We now examine the full (r, z) structure of the potential obtained from the SOR solver. Figure 5.4 shows a contour plot of the numerical solution. Because the Laplace problem has no explicit z-dependence and the domain is periodic in z, the potential is expected to be independent of z. The numerical solution shows horizontal contour bands and a flat axial slice at $r = (r_{\min} + r_{\max})/2$, confirming that $\partial\phi/\partial z \approx 0$. Physically, this indicates that the electric field has no axial component and is purely radial, consistent with the symmetry of the problem.

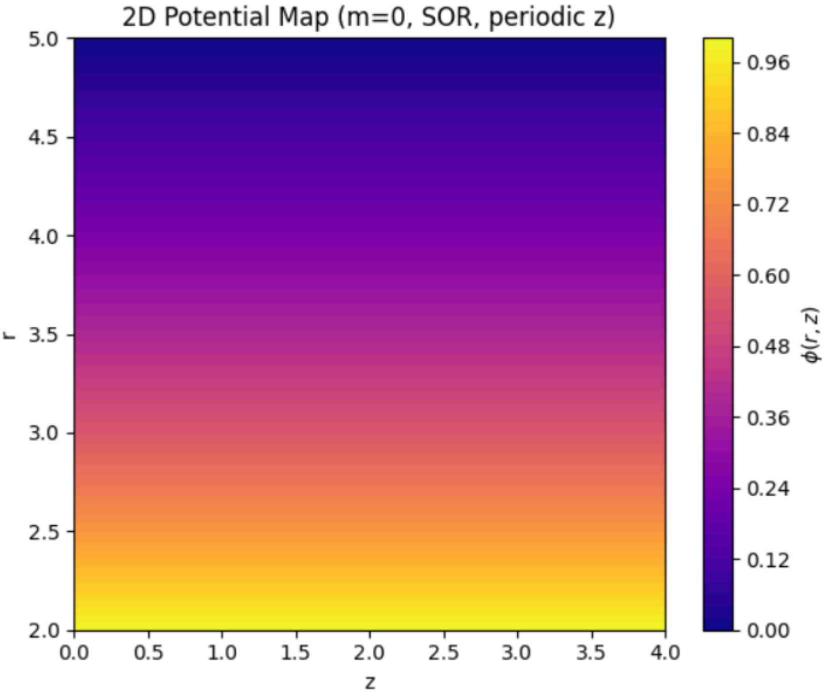


Figure 5.4: Contour plot of the computed potential $\phi(r, z)$ on the (r, z) domain.

Figure 5.5 shows representative slices: a radial slice at mid-z and an axial slice at mid-r. Both are smooth and free from numerical artefacts.

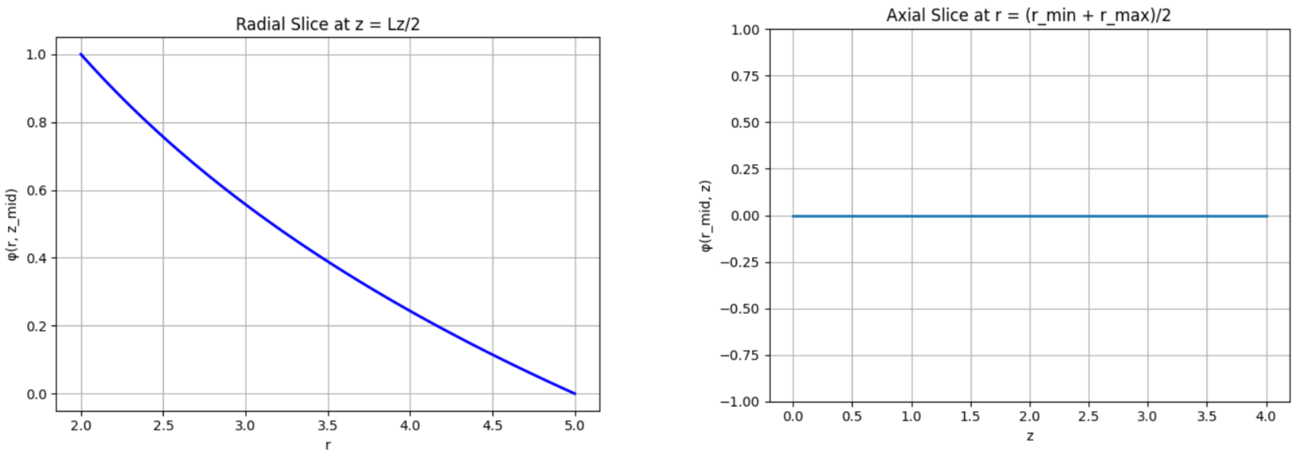


Figure 5.5: Radial(left) and Axial(right) slices of the potential.

5.0.4 Grid Refinement Study

To assess the accuracy of the finite-difference discretisation, we perform a grid-refinement study by repeating the Laplace test case on progressively finer grids. The Laplace equation ($m = 0$) in cylindrical coordinates was solved on uniform grids with $N_r = N_z = 50, 100, 150, 200, 250$.

The numerical solution $\phi_{\text{num}}(r, z)$ was averaged along the axial direction z to obtain a radial profile

$$\phi_{\text{avg}}(r_i) = \frac{1}{N_z} \sum_{j=1}^{N_z} \phi_{\text{num}}(r_i, z_j),$$

which was then compared to the analytical solution

$$\phi_{\text{exact}}(r) = \phi_{\text{rmin}} + (\phi_{\text{rmax}} - \phi_{\text{rmin}}) \frac{\ln(r/r_{\text{min}})}{\ln(r_{\text{max}}/r_{\text{min}})},$$

with $\phi_{\text{rmin}} = 1$ and $\phi_{\text{rmax}} = 0$.

The relative L2 error was computed as

$$\text{Error}_{L_2} = \frac{\|\phi_{\text{avg}} - \phi_{\text{exact}}\|_2}{\|\phi_{\text{exact}}\|_2} = \frac{\sqrt{\sum_{i=1}^{N_r} (\phi_{\text{avg}}(r_i) - \phi_{\text{exact}}(r_i))^2}}{\sqrt{\sum_{i=1}^{N_r} (\phi_{\text{exact}}(r_i))^2}}.$$

The effective radial grid spacing was defined as

$$h = \frac{r_{\text{max}} - r_{\text{min}}}{N_r - 1}.$$

For sufficiently fine grids, the L_2 error is expected to scale with the grid spacing according to

$$\text{Error}_{L_2} \sim h^p, \tag{5.1}$$

where p is the order of convergence.

Table 5.2: Grid Convergence Study: Relative L_2 Error vs Grid Spacing

$N_r = N_z$	h	Relative L_2 Error	Iterations to Converge
50	0.061224	6.6184×10^{-6}	490
100	0.030303	1.4776×10^{-5}	1825
150	0.020134	3.8510×10^{-5}	3843
200	0.015075	7.0220×10^{-5}	6470
250	0.012048	1.1063×10^{-4}	9653

A log-log plot of the relative L_2 error versus grid spacing h yields a slope of approximately -1.767, indicating that the numerical solution converges to the analytical solution at a rate close to second-order.

These results confirm that the SOR solver is consistent and the numerical solution approaches the exact solution as the grid is refined.

Figure 5.6 shows the error versus grid spacing on a log–log scale. The slope of approximately -1.77 confirms second-order convergence, as expected from the central-difference discretisation used for the radial and axial derivatives. For both radial and axial derivatives, the SOR solver employs central finite differences, which are formally second-order accurate. Therefore, it is anticipated that the relative L_2 error will decrease in proportion to h^2, resulting in a convergence slope of 2. Due to boundary effects at $r_{\min}$ and $r_{\max}$ and the moderate grid sizes chosen, which might not completely reach the asymptotic second-order regime, the measured slope of roughly 1.77 is significantly lower.

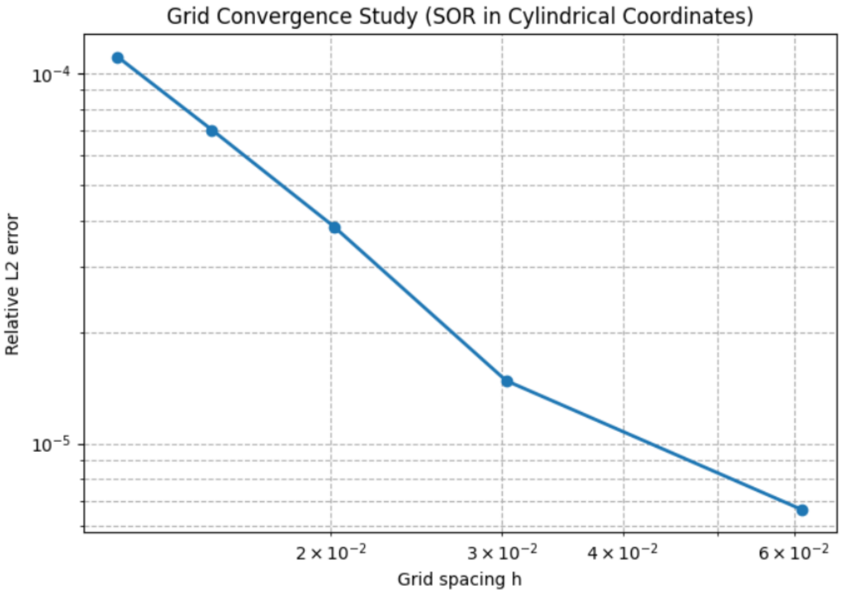


Figure 5.6: Grid refinement study showing order of convergence of the numerical method.

5.1 Extension to higher mode solutions

We now extend our Poisson solver to handle higher azimuthal modes. Previously, the Laplace case was solved mainly for benchmarking purposes, but in this section we introduce a finite charge distribution and examine how the solver performs for $m > 0$. The charge distribution used is

$$\rho_m(r, z) = r^m \sin(\pi r) \cos(2\pi z),$$

which provides a smooth and mode-selective source term. The results for each mode are presented separately in the following subsections, together with radial and axial slices, electric field visualisations, and convergence histories.

5.1.1 General Behaviour of the Modal Solutions

The modal Poisson operator contains a centrifugal contribution,

$$\nabla_m^2 \phi_m = \frac{\partial^2 \phi_m}{\partial r^2} + \frac{1}{r}\frac{\partial \phi_m}{\partial r} + \frac{\partial^2 \phi_m}{\partial z^2} - \frac{m^2}{r^2}\phi_m,$$

which strongly influences the behaviour of higher modes near the axis. The results therefore show distinct patterns for each value of m. We analyse these in detail below.

5.1.2 Mode $m = 0$: Axisymmetric Case

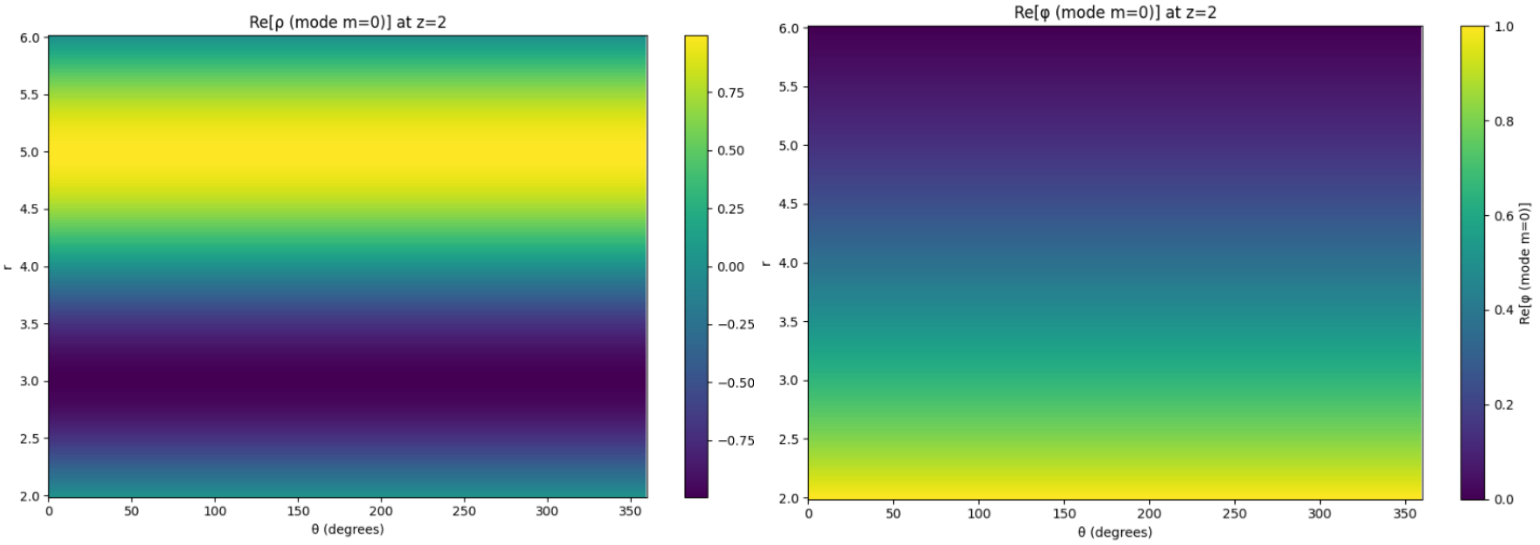


Figure 5.7: (left)Charge density $\rho_0(r, \theta)$ for the axisymmetric case. (right)Potential $\phi_0(r, \theta)$ for the $m = 0$ mode.

For the $m = 0$ mode, the charge distribution is purely radial and exhibits no dependence on the azimuthal angle θ, consistent with perfect cylindrical symmetry. The charge density oscillates smoothly in the radial direction, featuring a positive band at larger radii and a corresponding negative band at smaller radii. This produces horizontal, θ-independent stripes in the r–θ plot, reflecting a radially varying but angularly uniform distribution.

The potential is purely radial with no angular dependence. It remains constant in θ for all radii, consistent with axial symmetry. The magnitude is highest at the inner radius and decreases smoothly toward the outer boundary, where it reaches zero. The resulting profile is a monotonic radial decay, forming a uniform band when plotted over θ.

5.1.3 Mode $m = 1$: Dipole-Like Mode

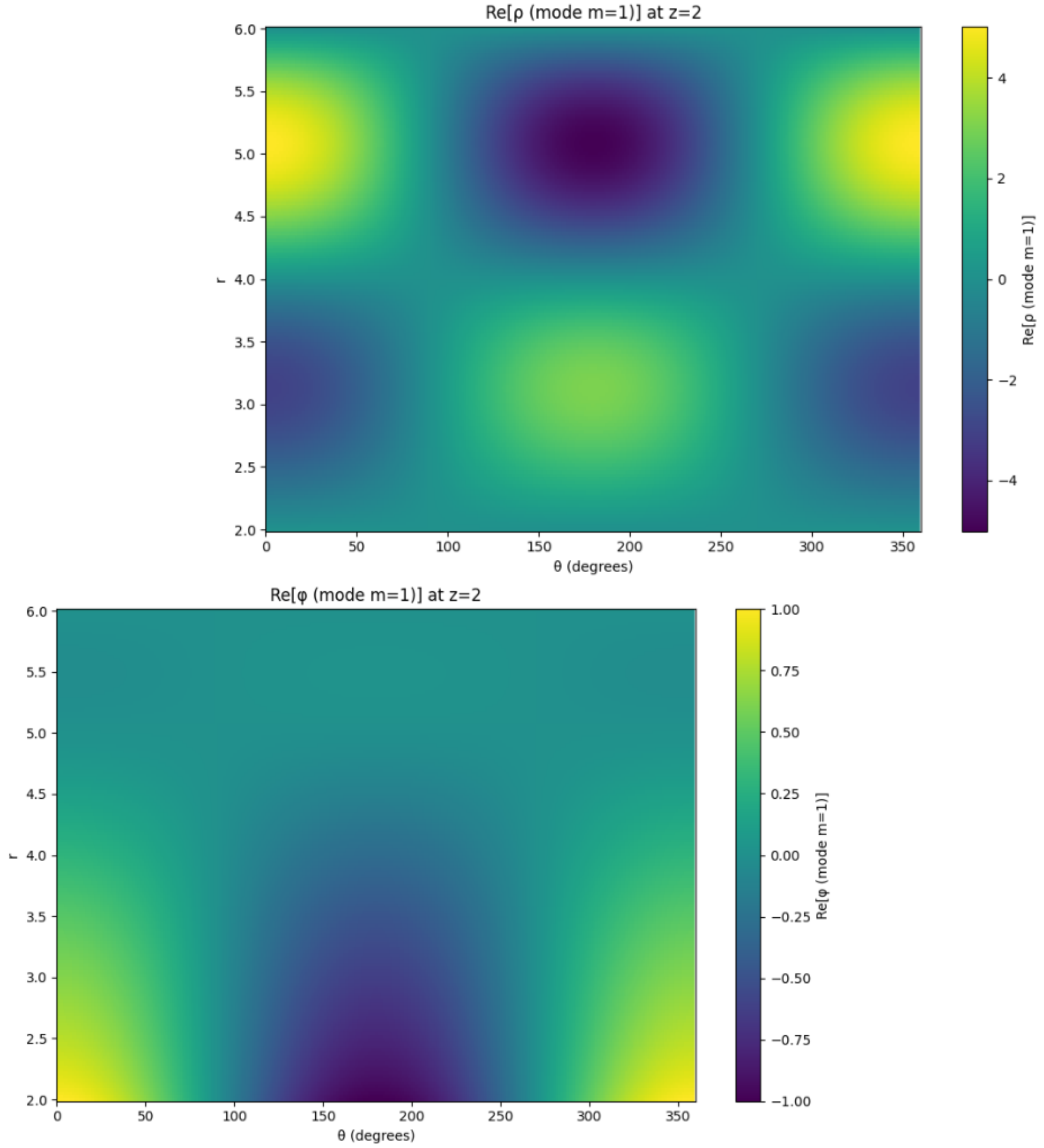
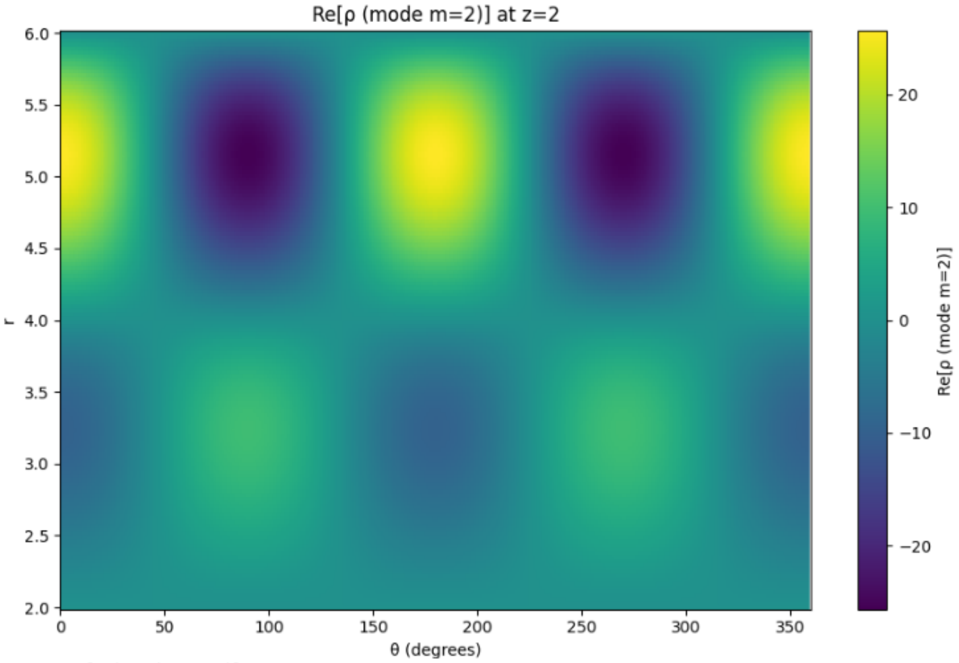


Figure 5.8: (left)Charge density $\rho_1(r, \theta)$ for the dipole-like mode.(right)Potential $\phi_0(r, \theta)$ for the $m = 1$ mode

For the $m = 1$ mode, the charge varies both in r and in θ. There is one bright region and one dark region as we go around the angle, giving a pattern that looks like a single wave around the circle. The charge also changes with radius, forming broad bands as r increases.

The potential exhibits a dipolar angular pattern with a positive lobe near $\theta = 0°$ and a negative lobe near $\theta = 180°$. Its magnitude is largest at the inner radius and decays smoothly outward, approaching zero near the outer boundary.

5.1.4 Mode $m = 2$: Quadrupole Mode



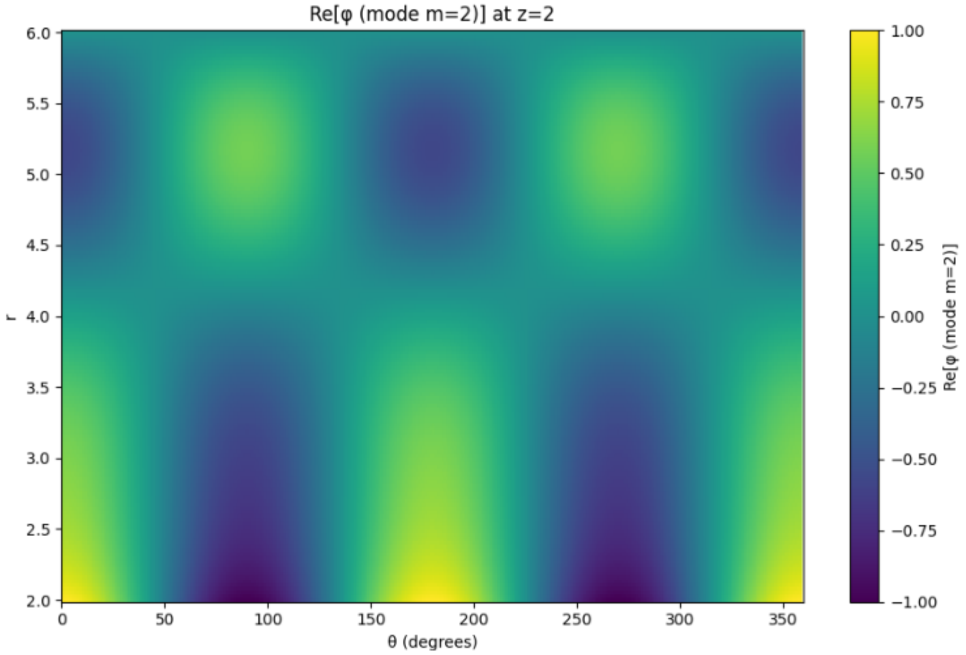


Figure 5.9: (left)Charge density $\rho_1(r, z)$ for the quadrupole-like mode.(right)Potential $\phi_0(r, z)$ for the $m = 2$ mode

For the $m = 2$ mode, the charge repeats twice as we go around the angle. This creates two bright areas and two dark areas in one full turn. The radial direction also shows smooth changes, so the plot has stacked bands in r along with the repeating pattern in θ.

The potential displays a quadrupolar angular structure, featuring two positive and two negative lobes over the full 0°–360° range. The pattern repeats twice around the azimuthal direction, consistent with its $m = 2$ symmetry. The amplitude is largest near the inner radius and decays monotonically with increasing r, approaching zero toward the outer boundary.

Table 5.3: Convergence statistics for modes $m = 0, 1, 2$ using SOR with $\omega = 1.8$.

Mode m	**Iterations to Converge**	**CPU Time (s)**
0	3383	179.240
1	3133	142.391
2	2548	67.231

5.1.5 Convergence for modes m=0,1,2 using SOR with $\omega = 1.8$

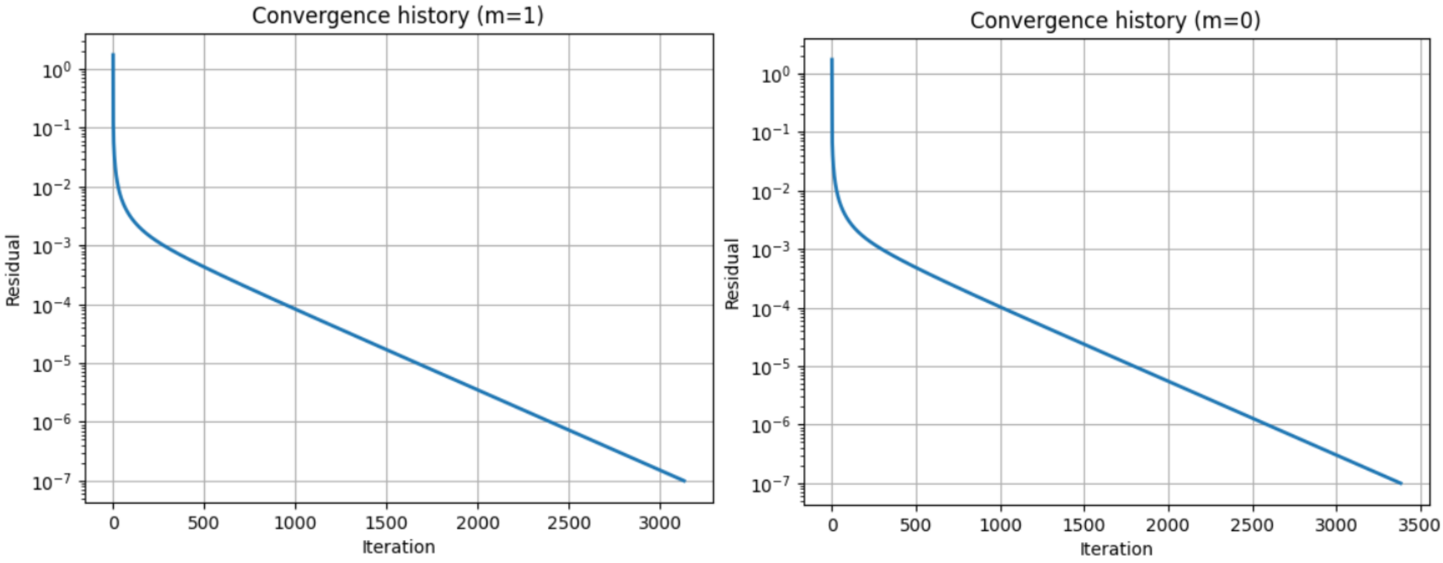


Figure 5.10: Residual v/s No of iterations for m=0 mode and m=1 mode respectively .

Table 5.3 highlights a clear trend: the SOR solver converges faster for higher azimuthal modes. While the axisymmetric mode $m = 0$ requires 3383 iterations, the dipole mode $m = 1$ converges slightly faster, and the quadrupole mode $m = 2$ converges the fastest at only 2548 iterations. This behaviour is expected, because the centrifugal term $-m^2\phi/r^2$ makes the operator increasingly diagonally dominant for larger m, thereby improving the efficiency of the SOR relaxation. The CPU time also decreases correspondingly, falling from 179 s for $m = 0$ to just 67 s for $m = 2$, showing that higher modes not only converge in fewer iterations but also in substantially less time.

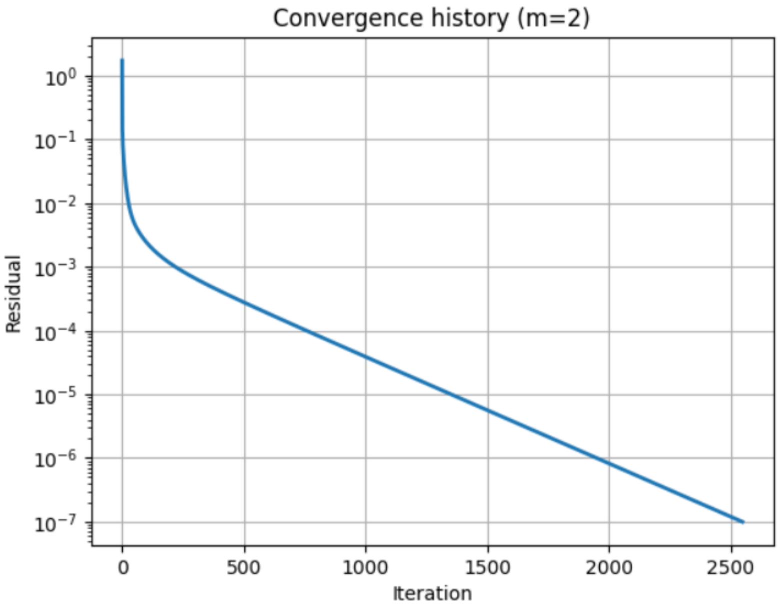


Figure 5.11: Residual v/s No of iterations for m=2 mode .

5.2 Reconstruction of the full 3D potential

Before presenting the fully reconstructed potential, it is useful to clarify what the modal solutions represent. Each $\phi_m(r,z)$ is not the physical potential on its own, but the amplitude of the m-th azimuthal Fourier mode. Higher modes change sign because they describe angular oscillations such as $\cos(m\theta)$, while the axisymmetric $m = 0$ mode remains positive and carries the main structure enforced by the boundary conditions. The physical potential is obtained by recombining all modes through

$$\Phi(r,\theta,z) = \mathrm{Re}\left[\sum_m \phi_m(r,z)\, e^{-im\theta}\right],$$

so the dominant component is $m = 0$ with the higher modes contributing only small angular variations. The following figure shows this fully reconstructed potential $\Phi(r,\theta,z)$ on an (r,θ) slice at fixed z.

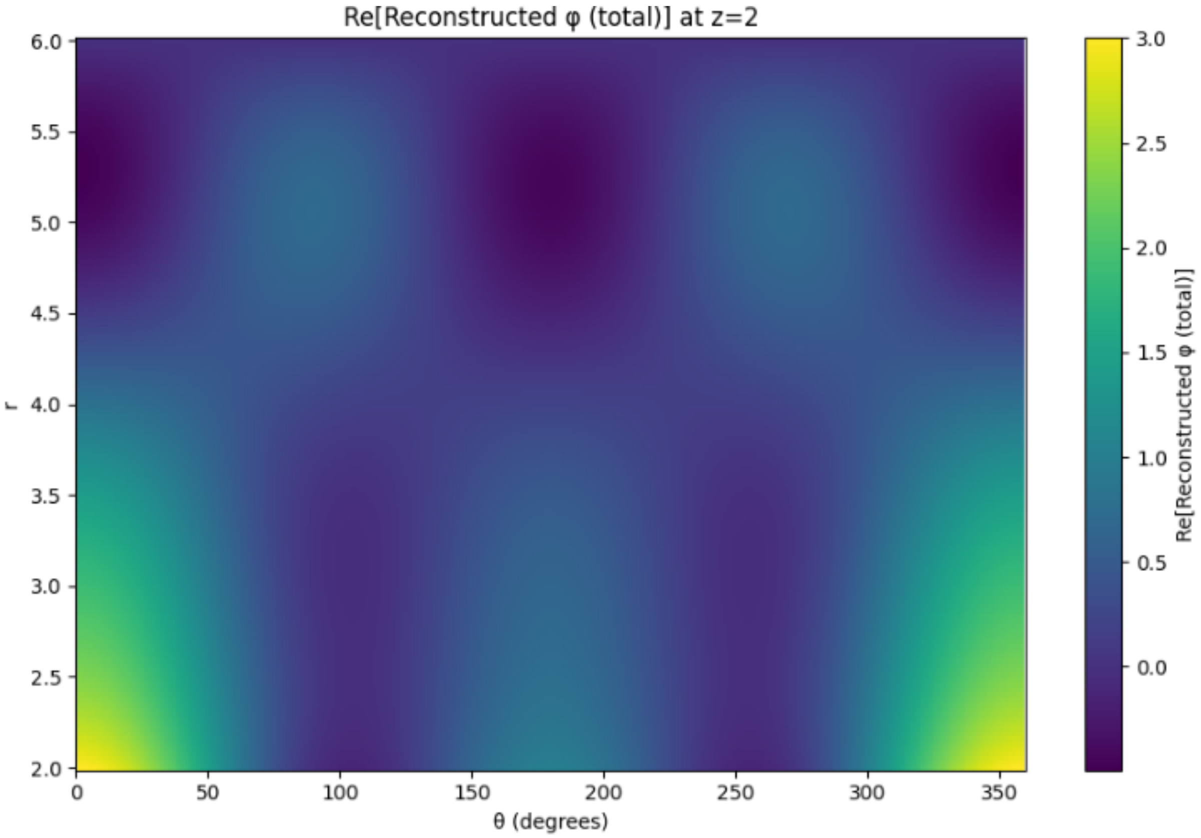


Figure 5.12: Fully reconstructed potential using modes m=0,1,2

In addition to the potential, we also show the reconstructed charge density $\rho(r,\theta,z)$ obtained from the same modal coefficients. Since the modal source term scales as

$$\rho_m(r,z) = r^m \sin(\pi r)\cos(2\pi z), \tag{5.2}$$

higher-order modes produce charge distributions whose angular oscillations increase rapidly with radius. This behaviour is clearly visible in the reconstructed $\rho(r,\theta,z)$, where the amplitude at larger r is significantly larger due to the r^m dependence of the source.

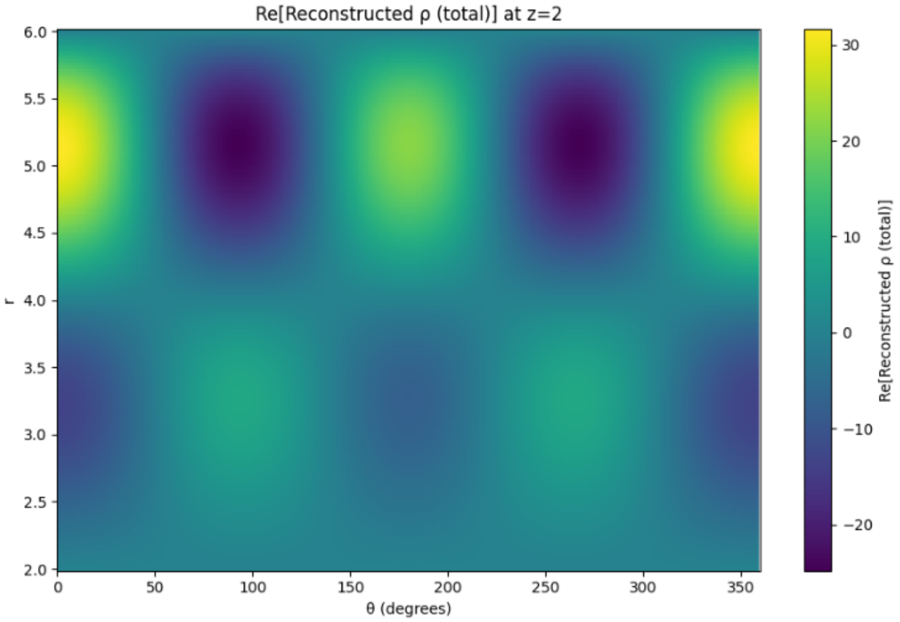


Figure 5.13: Reconstructed charge density $\rho(r, \theta, z)$ on an (r, θ) slice at fixed $z = 2$.

To quantify the behaviour , Fig. 5.14 presents lineouts of the reconstructed potential $\Phi(r, \theta, z)$ as a function of θ at $z = 2$ for several radii. The potential varies only mildly with angle, and its amplitude decreases with increasing radius. This confirms that higher-order angular oscillations are strongly attenuated in the solution, and that the physical potential remains dominated by the axisymmetric $m = 0$ mode.

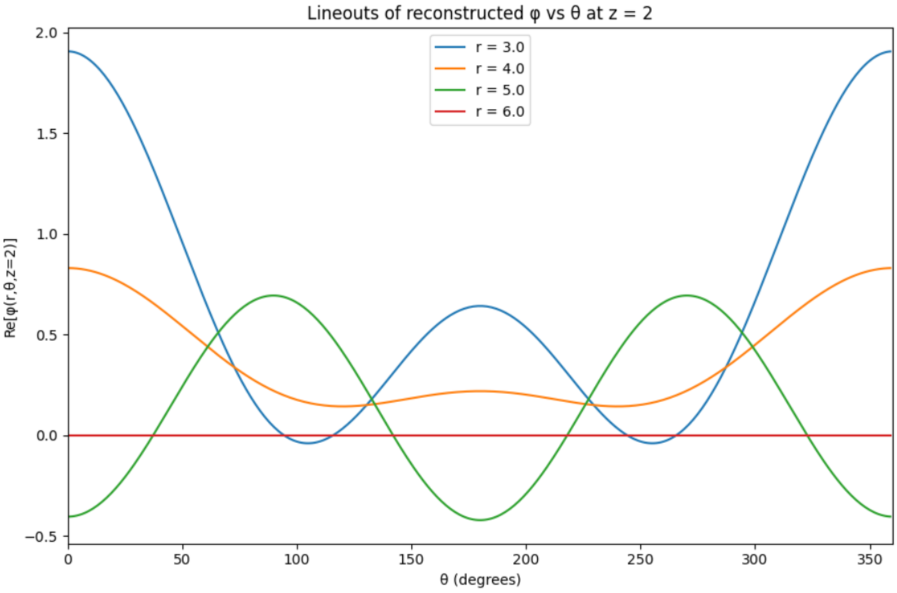


Figure 5.14: Lineouts of the reconstructed potential $\Phi(r, \theta, z)$ vs. θ at $z = 2$ for several radii.

Finally, Fig. 5.15 shows the corresponding lineouts of the reconstructed charge density. The charge displays large and rapidly growing angular oscillations as r increases, reflecting the r^m scaling of the source. The stark contrast between these oscillations and the smoother potential lineouts highlights how high-m contributions in the charge are strongly suppressed in the potential.

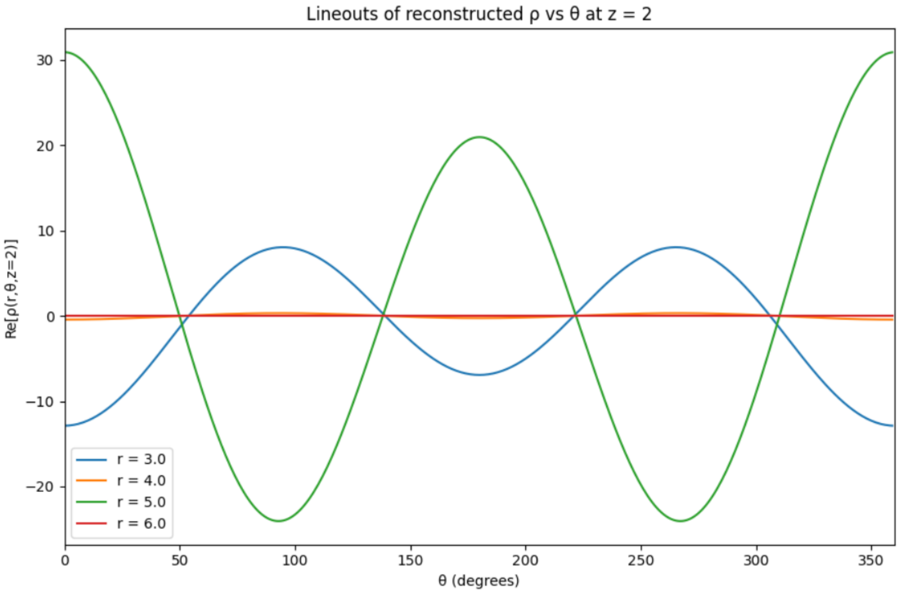


Figure 5.15: Lineouts of the reconstructed charge density $\rho(r, \theta, z)$ vs. θ at $z = 2$ for the same radii as in Fig. 5.14.

Chapter 6

Discussion

The results presented in this report demonstrate that the azimuthal Fourier modal decomposition provides a robust and efficient framework for solving the Poisson equation in cylindrical geometry. The benchmarking against the analytic Laplace solution confirms that the numerical discretisation, together with the SOR iterative method, is implemented correctly. The close agreement with the analytical profile, as well as the expected second-order convergence obtained from the grid refinement study, validate both the finite-difference stencil and the treatment of boundary conditions.

The numerical results obtained across different azimuthal modes highlight the central role of the centrifugal term

$$-\frac{m^2}{r^2}\,\phi_m,$$

in shaping both the behaviour of the modal solutions and the performance of the iterative solver. This term modifies the effective operator differently for each Fourier mode, and its influence becomes increasingly significant as the mode number m increases.

From a physical perspective, the centrifugal term acts as a "repulsive" contribution that suppresses the potential close to the axis. For $m > 0$, the contribution ϕ_m/r^2 diverges as $r \to 0$; therefore, the physically acceptable modal solution must vanish near the axis in order to satisfy regularity conditions. This naturally forces the potential profiles of the $m = 1$ and $m = 2$ modes to flatten toward zero in the central region. The numerical solutions clearly reflect this: the dipole and quadrupole potentials noticeably pull away from the axis, while the $m = 0$ mode remains free from such constraints.

This behaviour has an important numerical consequence. The centrifugal term effectively increases the diagonal dominance of the discrete Laplacian operator. In the finite-difference stencil, the term $-m^2/r^2$ adds a large negative contribution to the diagonal entries of the matrix A, especially near small r. A more diagonally dominant system is generally easier for iterative solvers such as SOR to handle, because the updated values at each iteration depend less sensitively on neighbouring grid points and the relaxation scheme becomes more stable. This explains why the convergence accelerates markedly as the mode number increases: the operator for $m = 2$ is substantially stiffer in the radial direction than for $m = 0$, and the SOR updates are therefore more effective.

The effect is clearly visible in the convergence metrics. While the axisymmetric mode takes the longest to converge due to the absence of stabilising terms, the dipole and quadrupole modes converge progressively faster. The quadrupole mode, which experiences the strongest centrifugal contribution, converges in almost one-third the number of iterations required for the $m = 0$ case. This behaviour is fully consistent with theoretical expectations for elliptic equations with enhanced diagonal dominance.

In terms of solution structure, the centrifugal term also dictates the spatial localisation of higher modes. The $m = 1$ and $m = 2$ potentials are pushed outward, creating modal shapes that increasingly favour larger radii. This is evident in the radial slices, where the amplitude of the potential scales approximately as r^m and remains suppressed near the axis. The electric field maps reflect this redistribution: rather than the purely radial field seen in the $m = 0$ case, higher modes exhibit twisted, multi-lobed field patterns determined by the interplay between the radial dependence and the axial periodicity.

The results reinforce the effectiveness of modal decomposition as an alternative to full three-dimensional Poisson solvers, particularly in systems where cylindrical symmetry plays a central role.

Chapter 7

Conclusion and Outlook

In this project, I implemented and examined a Fourier–modal Poisson solver in cylindrical geometry, following the formulation introduced by Lifschitz *et al.* The modal decomposition naturally separates the full three-dimensional Poisson equation into a set of independent two-dimensional problems, and this structure made it straightforward to study each azimuthal harmonic in isolation. Establishing the $m = 0$ benchmark was crucial for validating the finite-difference discretisation and the SOR iteration scheme. Once this baseline was confirmed, the extension to higher modes required no additional numerical machinery, which highlights one of the practical advantages of the modal framework.

The comparison with the analytical Laplace solution demonstrated that both the stencil and boundary conditions were implemented correctly. The grid-refinement study showed that the method approaches second-order accuracy, even though the cylindrical geometry introduces additional subtleties near the inner radius. The behaviour of the higher modes offered further insight: the centrifugal term m^2/r^2 modifies not only the structure of the modal potentials but also the conditioning of the discrete operator. The noticeably faster convergence for $m = 1$ and $m = 2$ compared to the axisymmetric mode is consistent with the increased diagonal dominance introduced by this term, and this trend became apparent through direct numerical experimentation.

Beyond these validations, the project provided a more practical appreciation for how cylindrical discretisations behave in computation. There are several clear directions for improving the solver. The slow convergence of the $m = 0$ mode suggests that multigrid acceleration or Krylov-based methods could reduce iteration counts substantially. A non-uniform radial mesh would likely improve behaviour near $r = 0$, where the cylindrical operator varies most rapidly. Finally, because each mode solves independently, the approach is naturally amenable to parallelisation, making it well suited for integration into a particle-in-cell workflow. These extensions would enhance the solver's applicability to more demanding plasma and beam-physics simulations.

Bibliography

[1] A. F. Lifschitz, X. Davoine, E. Lefebvre, J. Faure, C. Rechatin and V. Malka, “Particle-in-Cell modelling of laser–plasma interaction using cylindrical coordinates,” *Journal of Computational Physics*, vol. 228, no. 5, pp. 1803–1814, 2009.

[2] D. M. Young, “Iterative Methods for Solving Partial Difference Equations of Elliptic Type,” *Transactions of the American Mathematical Society*, vol. 76, no. 1, pp. 92–111, 1954.

[3] R. J. LeVeque, *Finite Difference Methods for Ordinary and Partial Differential Equations*, SIAM, 2007.

[4] NASA. (2024, April 29). *NASA unveils game-changing electric propulsion technology for future space missions.* SciTechDaily. [online]

[5] Code for the higher m modes. code in my github repo.

.1 Benchmarking code : for comparison with Laplace's equation

Listing 1: Python implementation for benchmarking

```python
import numpy as np
import matplotlib.pyplot as plt
from time import perf_counter

# ============================================================
# SOR solver function
# ============================================================
def solve_sor(Nr=100, Nz=100, r_min=2.0, r_max=5.0, Lz=4.0,
              omega=1.8, tol=1e-7, max_iter=20000, verbose=False):
    """Solve Laplace equation using SOR for given omega."""
    r = np.linspace(r_min, r_max, Nr)
    z = np.linspace(0, Lz, Nz)
    dr = r[1] - r[0]
    dz = z[1] - z[0]
    dr2, dz2 = dr**2, dz**2

    phi = np.zeros((Nr, Nz))
    rho = np.zeros_like(phi)

    # Boundary conditions
    phi_rmin, phi_rmax = 1.0, 0.0
    phi[0, :] = phi_rmin
    phi[-1, :] = phi_rmax

    # Precompute coefficients
    def coeffs(i):
        ri = r[i]
        A_plus = 1/dr2 + 1/(2*ri*dr)
        A_minus = 1/dr2 - 1/(2*ri*dr)
        A0 = -2/dr2 - 2/dz2
        return A_plus, A_minus, A0

    # SOR iteration
    res_history = []
    start = perf_counter()

    for it in range(max_iter):
        phi_old = phi.copy()

        for i in range(1, Nr - 1):
            A_plus, A_minus, A0 = coeffs(i)
            for j in range(Nz):
                j_plus = (j + 1) % Nz
                j_minus = (j - 1) % Nz
                rhs = -rho[i, j]
                lap = (A_minus * phi[i - 1, j] +
                       A_plus * phi[i + 1, j] +
                       (phi[i, j_plus] + phi[i, j_minus]) / dz2)
                phi_new = (lap + rhs) / (-A0)
                phi[i, j] = (1 - omega)*phi[i, j] + omega*phi_new

        phi[0, :] = phi_rmin
        phi[-1, :] = phi_rmax

        res = np.linalg.norm(phi - phi_old) / (np.linalg.norm(phi_old) + 1e-15)
        res_history.append(res)

        if verbose and it % 50 == 0:
            print(f"  ={omega:.2f} | Iter {it:5d} | Residual = {res:.3e}")

        if res < tol:
            end = perf_counter()
            if verbose:
                print(f"Converged (  ={omega:.2f}) in {it} iterations, res={res:.2e}, time={end-start:.2f}s")
            return phi, r, z, res_history, it, end-start

    end = perf_counter()
    print(f"Did not converge (  ={omega:.2f}), final residual={res:.2e}")
    return phi, r, z, res_history, max_iter, end-start


# ============================================================
# Single-run visualization for    = 1.8
# ============================================================
phi, r, z, res_history, n_iter, runtime = solve_sor(omega=1.8, verbose=True)

# Analytical solution
A = (phi_rmax - phi_rmin) / np.log(r_max/ r_min)
B = phi_rmin - A * np.log(r_min)
phi_analytic = A * np.log(r) + B
phi_avg = np.mean(phi, axis=1)

# ---- Plot residual history ----
plt.figure()
plt.semilogy(res_history, 'b-')
plt.xlabel("Iteration")
plt.ylabel("Residual (log scale)")
plt.title(f"SOR Convergence (  =1.8, converged in {n_iter} iters)")
plt.grid(True)
plt.tight_layout()

# ---- Two Subplots: Numerical and Analytical   (r) ----
fig, axes = plt.subplots(1, 2, figsize=(10, 4), sharey=True)

# (1) Left subplot     Numerical solution
axes[0].plot(r, phi_avg, 'r-', lw=2)
axes[0].set_title("Numerical Solution (SOR)")
axes[0].set_xlabel("r")
axes[0].set_ylabel("  (r)")
axes[0].grid(True)

# (2) Right subplot     Analytical solution
axes[1].plot(r, phi_analytic, 'k--', lw=2)
axes[1].set_title("Analytical Solution (A ln r + B)")
axes[1].set_xlabel("r")
```

```
axes[1].grid(True)

plt.suptitle("Radial Profiles: Numerical vs Analytical", fontsize=13)
plt.tight_layout(rect=[0, 0, 1, 0.95])  # leave space for the suptitle
plt.show()




# ---- Plot radial and axial slices ----
mid_z = Nz // 2
mid_r = Nr // 2

plt.figure()
plt.plot(r, phi[:, mid_z], 'b-', lw=2)
plt.xlabel("r")
plt.ylabel("  (r, z_mid)")
plt.title("Radial Slice at z = Lz/2")
plt.grid(True)
plt.tight_layout()




plt.figure()
plt.plot(z, phi_line, lw=2)
plt.ylim(-1,1)  # keep stable range
plt.xlabel("z")
plt.ylabel("  (r_mid, z)")
plt.title("Axial Slice at r = (r_min + r_max)/2")
plt.grid(True)
plt.tight_layout()

# ---- 2D potential map ----
plt.figure(figsize=(6,5))
plt.contourf(z, r, phi, levels=50, cmap='viridis')
plt.colorbar(label=r'$\phi(r,z)$')
plt.xlabel("z")
plt.ylabel("r")
plt.title("2D Potential Map (m=0, SOR, periodic z)")
plt.tight_layout()

#Study: convergence for different $\omega$ values
# ============================================================
omegas = [1.0,1.1,1.2, 1.3,1.4, 1.5,1.6, 1.7,1.75, 1.8, 1.9,1.95]
iters_needed = []
times_needed = []

for w in omegas:
    _, _, _, res_hist, n_iter, runtime = solve_sor(omega=w)
    iters_needed.append(n_iter)
    times_needed.append(runtime)

plt.figure()
plt.plot(omegas, iters_needed, 'o-', lw=2)
plt.xlabel("   (SOR relaxation parameter)")
plt.ylabel("Iterations to converge")
plt.title("SOR Convergence Speed vs   ")
plt.grid(True)
plt.tight_layout()

plt.figure()
plt.plot(omegas, times_needed, 's-', lw=2)
plt.xlabel("   (SOR relaxation parameter)")
plt.ylabel("CPU time (s)")
plt.title("Runtime vs   ")
plt.grid(True)
plt.tight_layout()

plt.show()
```