Quest Journals
Journal of Software Engineering and Simulation
Volume 12 ~ Issue 8 (August 2026) pp: 01-19
ISSN(Online) :2321-3795 ISSN (Print):2321-3809
www.questjournals.org

Research Paper

Numerical Simulation of Gravity Responses of Infinite and Finite Solid Cylinders Using Analytical Matrix-Based MATLAB Implementation

La Hamimu[1]
[1]*(Geophysical Engineering Department, Halu Oleo University, Kendari Indonesia)*
Corresponding Author

ABSTRACT: *Forward modeling of gravity is a fundamental component of computational geophysics, which functions to map the quantitative relationship between subsurface mass distribution and gravity anomalies measured at the Earth's surface. In the case of idealized geological models, homogeneous solid cylinders of infinite and finite length are widely used to represent elongated subsurface structures such as dikes, intrusive bodies, magma conduits, and mineralized zones. This study presents a unified analytical matrix-based MATLAB framework for the numerical simulation of the gravity response generated by infinite and finite solid cylinders over a two-dimensional observation grid. The proposed framework combines closed-form analytical solutions derived from Newton's law of gravity with MATLAB matrix-oriented computation and meshgrid functions, enabling the simultaneous evaluation of gravitational acceleration at all observation points without explicit point-by-point iteration or numerical segmentation of the observation grid. For the infinite cylinder model, the gravitational acceleration is calculated as a function of the radial distance from the cylinder axis, while the finite cylinder formulation incorporates analytical boundary functions that explicitly account for the effects of finite length. The resulting gravity field is visualized through one-dimensional gravity profiles, three-dimensional surface maps, and two-dimensional contour maps to facilitate quantitative interpretation of the anomaly characteristics. Numerical simulations show that infinite cylinders produce a ridge-shaped gravity anomaly that is invariant to translation and governed only by radial distance with parallel contour lines. While finite cylinders produce a localized dome-shaped anomaly characterized by longitudinal attenuation and a closed elliptical contour pattern resulting from its limited axial extent. At various cylinder lengths of L = 15 km, L = 20 km, and L = 25 km, the numerical simulation results still show a closed elliptical contour and for L = 35 km shows a parallel contour pattern. The close agreement between theoretical predictions and numerical simulations indicates that the developed framework accurately reproduces the expected physical behavior of a cylindrical gravity field while maintaining numerical stability and computational efficiency. Thus, the proposed MATLAB implementation can be used as an effective platform for gravity forward modeling, algorithm verification, synthetic model generation, and future extensions towards gravity inversion modeling, sensitivity analysis, uncertainty quantification, and high-performance scientific computing.*

KEYWORDS: *Forward modeling of gravity, numerical simulation, matrix-based computing of MATLAB, infinite and finite solid cylinders, gravitational acceleration, computational geophysics.*

*Received 01 Aug., 2026; **R**evised 05 Aug, 2026; **A**ccepted 07 Aug., 2026*
P**ublished with open access at **www.questjournals.org

I. INTRODUCTION

Forward gravity modeling constitutes one of the fundamental computational techniques in applied geophysics because it establishes the physical relationship between subsurface density distributions and the gravity anomalies measured at the Earth's surface. The capability to accurately simulate gravity responses from subsurface geological bodies is essential for interpreting gravity survey data, validating inversion algorithms, designing synthetic models, and improving the understanding of potentialfield methods [1]. Consequently, efficient computational algorithms for gravity simulation have become increasingly important with the growing demand for highresolution numerical modeling and visualization in geophysical exploration.

The rapid development of scientific computing has significantly transformed the implementation of geophysical forward modeling. Modern computational environments such as MATLAB provide extensive numerical libraries, matrix-oriented computation, and powerful visualization capabilities that enable researchers to implement complex physical models with relatively concise program structures [2]. Matrix-based computation also offers substantial improvements in code readability, numerical stability, and computational efficiency compared with conventional procedural programming approaches. These advantages have made MATLAB one of the most widely adopted platforms for developing numerical algorithms in geophysics.

Over the last decade, our research has consistently focused on developing MATLAB-based computational frameworks for solving forward and inverse problems across several branches of applied geophysics. The initial work introduced MATLAB implementations for forward and inverse modeling of Vertical Electrical Sounding (VES), providing an integrated computational environment for Schlumberger-array simulations and genetic-algorithm inversion [3]. The developed program demonstrated that MATLAB could effectively integrate numerical computation, optimization, and visualization within a single platform for geophysical interpretation.Subsequent studies expanded this computational framework to surface-wave analysis through the development of ForLoveDCs, an open MATLAB software package for calculating multimode Love-wave dispersion curves [4]. The software implemented Haskell's matrix formulation for layered media and established a reusable forward-modeling engine intended for future inversion applications. This work emphasized software sustainability, modular programming, and extensibility of MATLAB implementations for computational geophysics.

Our recent investigations further extended MATLAB implementations toward seismic imaging and inversion methodologies. A computational framework based on the pseudo-bending ray-tracing algorithm was developed for anisotropic seismic refraction tomography, demonstrating the capability of MATLAB to solve complex forward modeling and tomographic imaging problems involving heterogeneous subsurface media [5]. In addition, a comprehensive numerical analysis of geophysical parameter sensitivity was performed for multimode Love-wave dispersion curves, providing an efficient strategy for identifying influential model parameters prior to inversion and reducing computational complexity [6]. More recently, a new inversion framework based on systematic search, guided permutation, and sensitivity analysis was proposed to improve the robustness and reliability of nonlinear geophysical inversion [7].

Although considerable progress has been achieved in developing MATLAB-based computational tools for electrical, seismic, and surface-wave methods, comparatively less attention has been devoted to developing an integrated MATLAB framework for gravity forward simulation of fundamental geological geometries. Among various subsurface models, cylindrical bodies are particularly important because numerous geological structuresincluding dykes, elongated intrusive bodies, mineralized zones, magma conduits, and fault-fill materialscan be reasonably approximated using infinite or finite solid cylinders. Accurate simulation of their gravity responses provides an essential foundation for gravity interpretation and for validating future inversion algorithms.

The gravity response of an infinite solid cylinder differs fundamentally from that of a finite cylinder. In the infinite-cylinder approximation, gravity depends solely on the radial distance from the cylinder axis because the end effects are neglected. Conversely, finite cylinders require three-dimensional numerical computation since gravity is influenced by both radial distance and the finite longitudinal extent of the body. Consequently, numerical implementations capable of efficiently handling both geometries within a unified computational framework remain valuable for both educational and research applications.

This paper presents an efficient MATLAB-based framework for forward gravity simulation of homogeneous infinite and finite solid cylinders over two-dimensional observation grids. The proposed implementation integrates analytical gravity formulations for infinite cylinders with numerical discretization techniques for finite cylinders within a common matrix-based computational architecture. Observation grids are generated using MATLAB's meshgrid function, enabling simultaneous computation of gravitational acceleration at all observation points. The computed responses are subsequently visualized as gravity profiles, three-dimensional surface distributions, and contour maps to facilitate quantitative interpretation of anomaly characteristics.

The primary contribution of this study lies in the development of a unified and computationally efficient MATLAB implementation capable of simulating gravity responses from both infinite and finite cylindrical bodies within the same programming framework. Unlike previous studies that primarily emphasized analytical formulations or isolated numerical examples, the proposed framework combines physical modeling, matrix-based computation, and comprehensive visualization into an integrated simulation environment. The resulting algorithm is expected to serve as a practical platform for geophysical education, synthetic model generation, benchmarking of gravity algorithms, and future development of gravity inversion methodologies.

Recent advances in computational geophysics have increasingly emphasized the development of efficient forward modeling algorithms capable of handling complex geological geometries while maintaining computational accuracy and scalability. With the rapid growth of high-performance scientific computing, modern forward modeling techniques have evolved from conventional analytical formulations toward hybrid numerical implementations involving adaptive discretization, matrix-based computation, finite-difference schemes, and parallel numerical algorithms. These developments are particularly important for gravity modeling because realistic geological structures rarely conform to idealized geometries and frequently require efficient numerical representations.

The growing interest in gravity forward modeling has also stimulated the development of open computational frameworks intended for both research and educational purposes. Several recent studies have proposed MATLAB-based implementations, graphical user interfaces, and open-source computational environments for gravity forward and inverse modeling, demonstrating that MATLAB remains one of the most effective scientific computing platforms for rapid algorithm development, visualization, and numerical experimentation. GUI-based gravity modeling systems have considerably simplified parameter manipulation and visualization while maintaining computational efficiency for large observation grids.

Recent methodological developments have further expanded gravity forward modeling beyond classical prism and simple analytical bodies. Adaptive mimetic finite-difference methods have been introduced for gravity computation on unstructured grids, providing improved numerical accuracy when modeling irregular geological interfaces. Likewise, modified Parker-based approaches have significantly accelerated forward gravity computation for general polyhedral bodies by combining Fourier-domain formulations with efficient numerical layering strategies. These studies clearly demonstrate that improving computational efficiency has become one of the principal research directions in contemporary gravity modeling [8].

In parallel with algorithmic developments, the availability of open-access scientific software has become increasingly important for ensuring research reproducibility and encouraging wider adoption of computational methods. Open computational platforms not only facilitate verification of numerical algorithms but also provide reusable frameworks for extending forward modeling toward inversion, sensitivity analysis, uncertainty quantification, and machine-learning-assisted interpretation. Consequently, developing reproducible MATLAB implementations with clear numerical workflows remains highly relevant within modern computational geophysics.

Within this context, gravity modeling of cylindrical bodies continues to receive considerable attention because elongated geological structuresincluding dykes, ore bodies, intrusive rocks, magma conduits, and fault zonesare commonly approximated by infinite or finite cylinders [9]. Although numerous studies have focused on sophisticated three-dimensional gravity formulations for arbitrary geometries, relatively few studies have presented a unified MATLAB framework that simultaneously models homogeneous infinite and finite solid cylinders over two-dimensional observation grids using computational procedures suitable for teaching, algorithm verification, and future inversion development. This gap motivates the present study.

II. MATHEMATICAL MODEL

2.1 Physical Model of an Infinite Solid Cylinder

Gravity forward modeling requires an idealized representation of subsurface geological bodies. In this study, elongated geological structures such as dykes, intrusive bodies, mineralized zones, and magma conduits are represented by homogeneous solid cylinders. Two different geometrical assumptions are considered, namely an infinite solid cylinder and a finite solid cylinder.For the infinite-cylinder model, the cylinder is assumed to extend indefinitely along the y-axis, so that its length is significantly larger than its radius. Under this assumption, the contribution from both cylinder ends can be neglected. Consequently, the gravitational field depends only on the radial distance between the observation point and the cylinder axis [1].

The observation system is defined on a horizontal two-dimensional grid located at the Earth's surface (z = 0). Each observation point is represented by Cartesian coordinates$P(x, y, 0)$ whereas the cylinder axis is positioned at $Q(x_q, z_q)$ with homogeneous density (ρ) and radius (a). Because the cylinder extends infinitely in the y-direction, gravitational acceleration is invariant along this axis and varies only in the x–z plane. This geometrical simplification considerably reduces the computational complexity while preserving the principal characteristics of gravity anomalies generated by elongated subsurface bodies [9].

2.2 Gravitational Acceleration of an Infinite Cylinder

According to Newton's law of gravitation, the gravitational field produced by an infinitely long homogeneous cylinder can be obtained by integrating the contribution of infinitesimal mass elements distributed continuously along the cylinder axis.The radial distance between an observation point and the cylinder axis is expressed as

$$r = \sqrt{(x - x_q)^2 + (z - z_q)^2} \tag{1}$$

where x and z denote the coordinates of the observation point, $x_q$and $z_q$represent the coordinates of the cylinder center, and r is the radial distance between the observation point and the cylinder axis. This radial distance is the primary geometric parameter governing the gravitational field of an infinite solid cylinder because the cylinder

is assumed to extend indefinitely along the y-axis. Consequently, the gravitational acceleration depends solely on the distance from the cylinder axis and is independent of the position along its longitudinal direction. This assumption substantially simplifies the mathematical formulation while preserving the essential characteristics of gravity anomalies generated by elongated geological structures.
The gravitational acceleration generated by an infinite cylinder is

$$g(r) = \frac{2G\lambda}{r} \tag{2}$$

where G is Newton's gravitational constant (G=6.673×10^{-11} m^{3}kg^{-1}s^{-2}) and λ represents the linear mass density

$$\lambda = \rho\pi a^2 \tag{3}$$

Substituting Equation (3) into Equation (2) gives

$$g(r) = \frac{2G\rho\pi a^2}{r} \tag{4}$$

Equation (4) indicates that gravity is directly proportional to density and the square of the cylinder radius while decreasing inversely with radial distance. This formulation differs from the spherical model, whose gravity decays proportionally to the inverse square of distance [1].
For numerical implementation, the gravity vector is decomposed into horizontal (g_x) and vertical components (g_z), that is

$$g_x = -\frac{2G\lambda\left(x - x_q\right)}{r^2} \tag{5}$$

$$g_x = -\frac{2G\lambda\left(z - z_q\right)}{r^2} \tag{6}$$

The total gravitational acceleration is subsequently calculated as

$$g_{tot} = \sqrt{g_x^2 + g_z^2} \tag{7}$$

2.3 Mathematical Model of a Finite Solid Cylinder

Unlike the infinite cylinder, whose gravitational field is independent of the longitudinal direction, the gravity response of a finite homogeneous solid cylinder is influenced by both its radius and its finite axial extent. The cylinder is assumed to be aligned with the y-axis, with radius a, length L, density contrast ρ, and center located at (x_q, y_q, z_q). The two end surfaces of the cylinder are positioned aty_1=y_q-L/2 and y_2=y_q+L/2. Instead of discretizing the cylinder into numerous elementary segments and applying numerical superposition, this study employs a closed-form analytical solution derived directly from Newton's law of gravitation. The finite length of the cylinder is incorporated analytically through two boundary functions corresponding to the lower and upper end surfaces. This formulation eliminates numerical approximation errors associated with discretization while substantially reducing computational cost.

The auxiliary boundary functions are expressed as

$$F_1 = \frac{u_1}{\sqrt{R^2 + u_1^2}} \tag{8}$$

$$F_2 = \frac{u_2}{\sqrt{R^2 + u_2^2}} \tag{9}$$

where $u_1 = y–y_1$ and $u_2= y–y_2$, and R^2=(x–x_q)^2+(z–z_q)^2
Using these auxiliary functions, the gravity components are obtained analytically as

$$g_x = -G\lambda\frac{x - x_q}{R^2}(F_1 - F_2) \tag{10}$$

$$g_y = G\lambda\left(\frac{1}{\sqrt{R^2 + u_1^2}} - \frac{1}{\sqrt{R^2 + u_2^2}}\right) \tag{11}$$

$$g_z = -G\lambda\frac{z - z_q}{R^2}(F_1 - F_2) \tag{12}$$

where λ=ρπ$a$denotes the linear mass density of the cylinder [9].
Finally, the total gravity magnitude is evaluated using

$$g_{tot} = \sqrt{g_x^2 + g_y^2 + g_z^2} \tag{13}$$

This analytical formulation provides an exact solution for a finite homogeneous solid cylinder and serves as the computational foundation of the MATLAB implementation presented in this study.

2.4 Numerical Formulation on a Two-Dimensional Observation Grid

To perform efficient forward gravity simulation, the observation area is discretized into a regular two-dimensional grid covering the entire survey domain. During the preprocessing stage, one-dimensional coordinate vectors are transformed into two-dimensional observation matrices using MATLAB's meshgrid function [2],

$$\left[X, Y\right] = \text{meshgrid}(x, y) \tag{14}$$

which simultaneously generates the Cartesian coordinates of all observation points.Once the observation matrices have been constructed, the computational engine selects the analytical solution corresponding to the specified cylinder geometry. For the infinite solid cylinder, the radial distance matrix is evaluated before computing the analytical gravity components. For the finite solid cylinder, the boundary coordinates and analytical boundary functions are first determined and subsequently used to calculate the gravity components. In both cases, the gravity field is evaluated through matrix-based analytical operations, allowing all observation points to be processed simultaneously without explicit point-by-point iteration. This matrix-oriented implementation fully exploits MATLAB's optimized numerical libraries, thereby improving computational efficiency while preserving the analytical accuracy of the forward gravity solution.

The computed gravity components are subsequently combined to obtain the total gravity magnitude, and the resulting gravity response matrix is stored for post-processing. The software then extracts the gravity profile along the central observation line, determines the maximum gravity response, and computes the statistical quantities required for quantitative analysis [10].Finally, the gravity field is visualized through three complementary graphical representations: a one-dimensional gravity profile, a three-dimensional gravity surface, and a two-dimensional contour map. These visualization products constitute the post-processing stage of the proposed software framework and provide both qualitative and quantitative insight into the gravity anomalies generated by infinite and finite solid cylinders.

III. MATLAB IMPLEMENTATION

3.1 Software Architecture

The proposed forward gravity simulation was entirely developed using MATLAB owing to its comprehensive numerical computing environment, extensive matrix manipulation capabilities, and advanced scientific visualization tools. The software integrates analytical formulations for the infinite solid cylinder with numerical discretization techniques for the finite solid cylinder into a unified computational framework. This unified implementation enables users to simulate gravity responses for both geometrical models using a consistent computational workflow while minimizing code redundancy.

The overall software architecture consists of four principal modules: (1) parameter initialization, (2) observation grid generation, (3) gravity computation, and (4) visualization. Figure 1 illustrates the computational workflow adopted in this study.First, the user specifies the physical parameters of the cylindrical body, including density contrast, radius, cylinder length (for the finite model), cylinder position, and observation grid dimensions. These parameters constitute the initial conditions for the forward simulation.

Subsequently, a two-dimensional observation grid is generated using MATLAB's meshgrid function, which transforms one-dimensional coordinate vectors into two coordinate matrices representing every observation location simultaneously. This matrix representation eliminates the need for explicit coordinate pairing and provides the basis for efficient matrix-based gravity computation. The gravity calculation module evaluates the analytical solution for the infinite cylinder or performs numerical superposition for the finite cylinder according to the selected simulation mode. Finally, the computed gravity values are visualized as one-dimensional gravity profiles, three-dimensional surface maps, and two-dimensional contour plots to facilitate geological interpretation.

3.2 Observation Grid Generation

Efficient gravity computation requires the systematic generation of observation coordinates. In this study, observation points are distributed over a regular two-dimensional grid covering the survey area. Rather than computing each observation point individually, the proposed algorithm employs MATLAB's built-in meshgrid function to generate the entire observation grid simultaneously.The observation coordinates are defined as

$$x = x_{\min} : \Delta x : x_{\max} \tag{15}$$

$$z = z_{\min} : \Delta z : z_{\max} \tag{16}$$

The coordinate matrices are subsequently constructed using MATLAB: [X,Z] = meshgrid(x,z), where X contains the horizontal coordinates of all observation points and Z contains the corresponding vertical coordinates. This matrix representation allows the gravitational response to be evaluated simultaneously over the entire observation domain through matrix operations instead of sequential point-by-point calculations.The use of meshgridconsiderably simplifies the implementation while improving computational efficiency, particularly for high-resolution observation grids consisting of thousands of observation points.

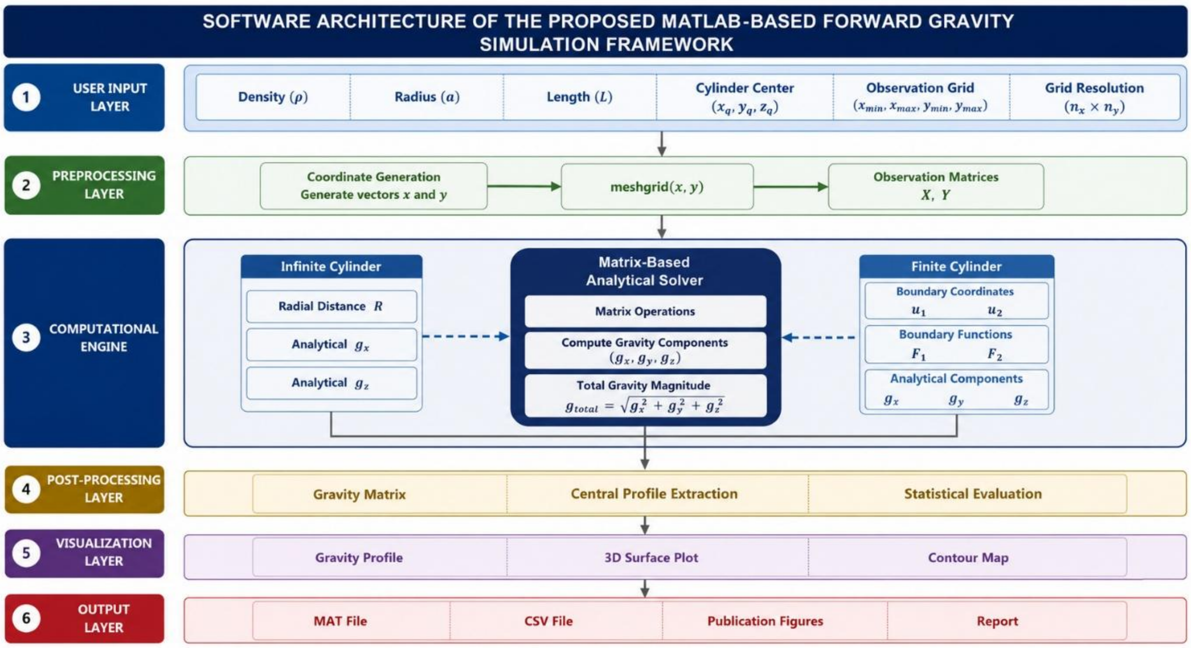

Figure 1: Software architecture of the proposed MATLAB-based forward gravity simulation system

3.3 Matrix-Based Gravity Computation

The computational implementation follows a fully matrix-based strategy in MATLAB to evaluate the analytical gravity equations over the entire observation grid simultaneously. Two-dimensional coordinate matrices are first generated using the meshgrid function, after which the relative coordinate differences between each observation point and the cylinder center are computed through element-wise matrix operations [11].Unlike conventional numerical implementations that discretize the finite cylinder into numerous elementary segments and accumulate their gravitational contributions using iterative loops, the present algorithm directly evaluates the analytical solution derived in Section 2.3. The finite length effect is incorporated through the analytical boundary functions F_1 and F_2, which represent the gravitational contributions of the two end surfaces of the cylinder.

Once the boundary functions have been computed, the gravity components g_x, g_y, and $g_z$are obtained by direct element-wise matrix operations without any longitudinal discretization. The total gravity magnitude is subsequently calculated using the Euclidean norm of the three gravity components. Since every arithmetic operation is performed on complete matrices rather than individual observation points, MATLAB executes the computations efficiently through optimized internal numerical libraries.

This implementation offers several advantages over discretization-based approaches. First, the analytical formulation eliminates approximation errors arising from numerical segmentation. Second, the absence of iterative superposition significantly reduces computational complexity. Finally, the algorithm preserves the simplicity of the mathematical formulation while maintaining excellent computational efficiency and numerical stability over large observation grids.

3.4 Data Visualization

Visualization constitutes the final stage of the computational workflow. Three complementary graphical representations are produced automatically after the gravity computation is completed.The first visualization is a one-dimensional gravity profile extracted along the central observation line. This profile is particularly useful for examining anomaly symmetry, peak amplitude, and lateral extent.The second visualization is a three-dimensional gravity surface generated using MATLAB's surf function. The resulting surface illustrates the spatial variation of gravity anomalies over the entire observation grid and provides an intuitive representation of anomaly geometry.

The third visualization is a contour map generated using the contourf function. Contour representations facilitate the identification of anomaly boundaries and are particularly useful for comparing the spatial responses of infinite and finite cylindrical models.These visualization modules enable both qualitative and quantitative evaluation of gravity responses while providing graphical outputs suitable for scientific publication.

3.5 Computational Workflow

Figure 1 presents the overall software architecture of the proposed MATLAB-based forward gravity simulation framework, illustrating the interactions among the major computational modules, including user input, preprocessing, computational engine, post-processing, visualization, and output generation. The modular organization enables each computational stage to be executed independently while maintaining a coherent data flow throughout the simulation process [8].The detailed computational workflow is illustrated in Figure 2. The simulation begins with the initialization of the physical parameters, including the density contrast, cylinder radius, cylinder length, cylinder center coordinates, observation domain, and grid resolution. These parameters are subsequently used to generate one-dimensional coordinate vectors, which are transformed into two-dimensional observation matrices using MATLAB's meshgrid function.The computational engine then selects the analytical solution according to the specified cylinder geometry. For the infinite cylinder, the radial distance matrix is evaluated to compute the analytical gravity components directly. For the finite cylinder, the boundary coordinates are first determined, followed by the evaluation of the analytical boundary functions. These functions are subsequently used to calculate the three gravity components through matrix-based analytical computation.

The resulting gravity components are combined to obtain the total gravity magnitude, after which the gravity response matrix is stored for subsequent post-processing. The software then extracts the central gravity profile, determines the maximum gravity response, and computes several statistical quantities required for quantitative interpretation.Finally, the computed gravity field is visualized as a gravity profile, a three-dimensional surface map, and a contour map. The numerical results together with publication-quality figures are subsequently exported for further analysis and documentation.The computational workflow presented in Figure 2 is formally summarized in Algorithm 1. While the workflow diagram illustrates the logical sequence of the computational stages, Algorithm 1 describes the corresponding matrix-based computational procedures performed during each stage. The combination of these two representations improves the clarity, reproducibility, and software-oriented presentation of the proposed MATLAB framework.

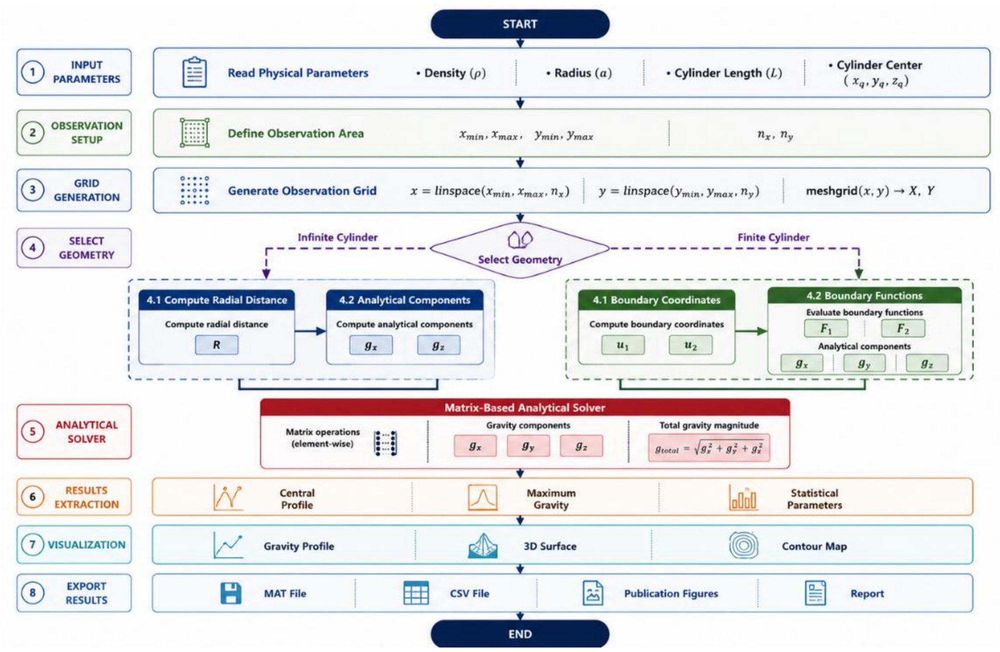

Figure 2: Flowchart of the MATLAB algorithm for forward gravity simulation on infinite and finite solid cylinders

Following the generation of the observation grid, the computational engine selects the appropriate analytical model according to the user-defined cylinder geometry. For the infinite solid cylinder, the radial distance matrix is first evaluated, after which the analytical gravity components are computed simultaneously at all observation points through matrix-based operations. For the finite solid cylinder, the algorithm computes the boundary coordinates associated with the upper and lower cylinder surfaces, evaluates the corresponding analytical boundary functions, and subsequently determines the three gravity components using a closed-form matrix-based analytical formulation. In both simulation modes, the total gravity magnitude is obtained directly from the computed gravity components without the need for numerical discretization or element-wise superposition.

The resulting gravity response matrix is subsequently stored for post-processing. The software extracts the gravity profile along the central observation line, determines the maximum gravity response, and computes the statistical parameters required for quantitative evaluation. The computed gravity field is then visualized as one-dimensional gravity profiles, three-dimensional surface maps, and two-dimensional contour maps before both the numerical data and publication-quality graphical outputs are exported for further interpretation.The complete computational procedure corresponding to the workflow illustrated in Figure 2 is formally summarized in Algorithm 1. The algorithm follows a matrix-oriented implementation in which the observation grid is processed simultaneously using MATLAB's optimized matrix operations rather than sequential point-by-point calculations. This strategy improves computational efficiency while preserving the analytical accuracy of the forward gravity solution. Furthermore, the modular organization of the computational workflow facilitates future extensions of the software, including inverse modeling, parameter optimization, sensitivity analysis, uncertainty quantification, parallel computing, and GPU-based acceleration without requiring substantial modifications to the existing software architecture [13].

Algorithm 1 provides a programming-language-independent description of the principal computational stages implemented in MATLAB. Together, Figure 2 and Algorithm 1 offer complementary representations of the proposed framework: the workflow diagram illustrates the logical sequence of the simulation process, whereas the pseudocode emphasizes the algorithmic operations performed at each computational stage. This dual representation enhances the clarity, reproducibility, and software-oriented presentation of the proposed analytical gravity simulation framework [14].

Algorithm 1 Matrix-based MATLAB forward gravity simulation

```
Require:
   Physical parameters (ρ, a, L, xc, yc, zc)
   Observation domain
   Grid resolution (nx, ny)
   Simulation geometry
Ensure:
   Gravity response matrix G(x,y)
1:   Read all physical and simulation parameters
2:   Define the observation domain
        (xmin, xmax, ymin, ymax)
3:   Generate observation vectors
        x ← linspace(xmin, xmax, nx)
        y ← linspace(ymin, ymax, ny)
4:   Construct observation matrices
        (X, Y) ← meshgrid(x, y)
5:   Select cylinder geometry
6:   if Geometry = Infinite Cylinder then
7:        Compute radial distance matrix R
8:        Evaluate analytical gravity components
            gx
            gz
9:        Compute total gravity magnitude
            gtotal
10:  else
11:       Compute boundary coordinates
            u1
            u2
12:       Evaluate analytical boundary functions
            F1
            F2
13:       Compute analytical gravity components
            gx
            gy
```

```
       gz
     Compute total gravity magnitude
       gtotal
end if
Store gravity response matrix
Extract gravity profile along the central observation line
Determine maximum gravity response
Compute statistical parameters
Generate gravity profile
Generate three-dimensional surface map
Generate contour map
Return G(x,y)
```

IV. RESULTS AND DISCUSSION

4.1 Verification of the Infinite Cylinder Model

The proposed MATLAB implementation was first verified using the analytical solution for a homogeneous infinite solid cylinder. The primary objective of this verification was to evaluate whether the matrix-based computational framework accurately reproduces the theoretical gravity response prior to its application to the more computationally demanding finite-cylinder model. The verification was performed by examining the computed gravity field through three complementary visualizations, namely a one-dimensional gravity profile, a three-dimensional gravity surface, and a two-dimensional contour map. These graphical representations collectively provide both quantitative and qualitative evidence for the correctness of the proposed numerical implementation [12].

Figure 3 presents the gravity profile extracted along the central observation line intersecting the cylinder axis. The computed anomaly exhibits a single positive and perfectly symmetric peak, with the maximum gravitational acceleration occurring directly above the cylinder center where the radial distance reaches its minimum value. Away from the cylinder axis, the gravity magnitude decreases smoothly and monotonically toward both sides of the profile. This behavior is fully consistent with the analytical solution presented in Section II, where the gravitational acceleration depends exclusively on the radial distance from the cylinder axis. The absence of secondary peaks or numerical oscillations further demonstrates that the matrix-based implementation accurately evaluates the analytical formulation over the entire observation profile.

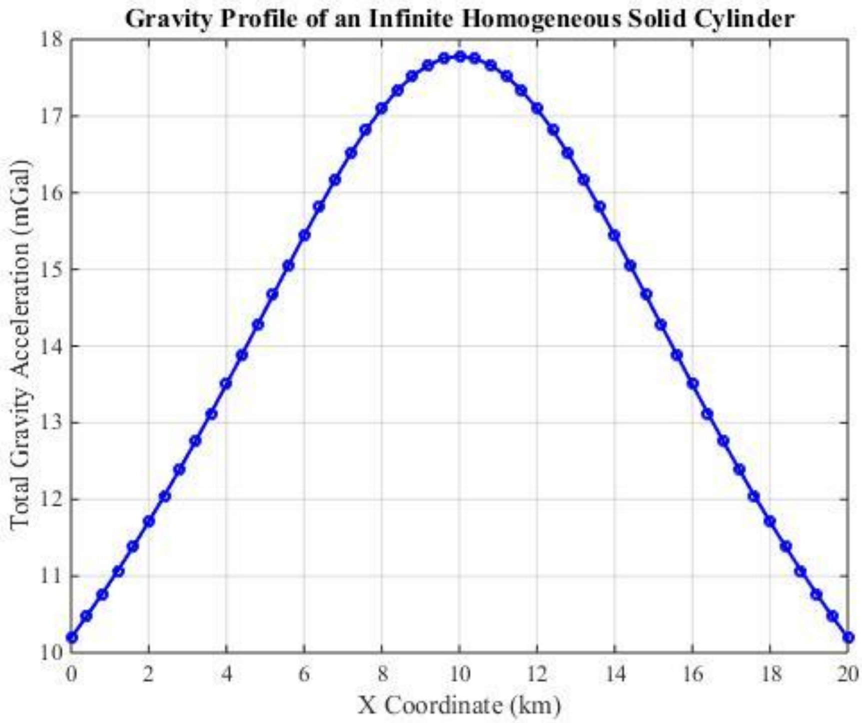

Figure 3: Gravity profile of the infinite cylinder

Although the gravity profile clearly illustrates the variation of gravitational acceleration along a single observation line, it does not fully describe the spatial distribution of the gravity field across the observation domain. Therefore, the simulated results are further visualized as a three-dimensional gravity surface, as shown in Figure 4. The surface plot reveals a continuous ridge-shaped anomaly extending uniformly along the longitudinal direction of the cylinder [15]. Unlike compact anomalies generated by finite geological bodies, the anomaly exhibits nearly constant amplitude along the y-axis because the infinite-cylinder approximation assumes translational invariance in this direction. Consequently, gravitational acceleration varies only with the radial distance in the x–z plane, while remaining essentially unchanged along the cylinder axis. The smooth geometry of the computed surface also confirms that the observation grid generated using MATLAB's meshgrid function and the subsequent matrix-based computations produce stable and continuous numerical solutions without noticeable discretization artifacts.

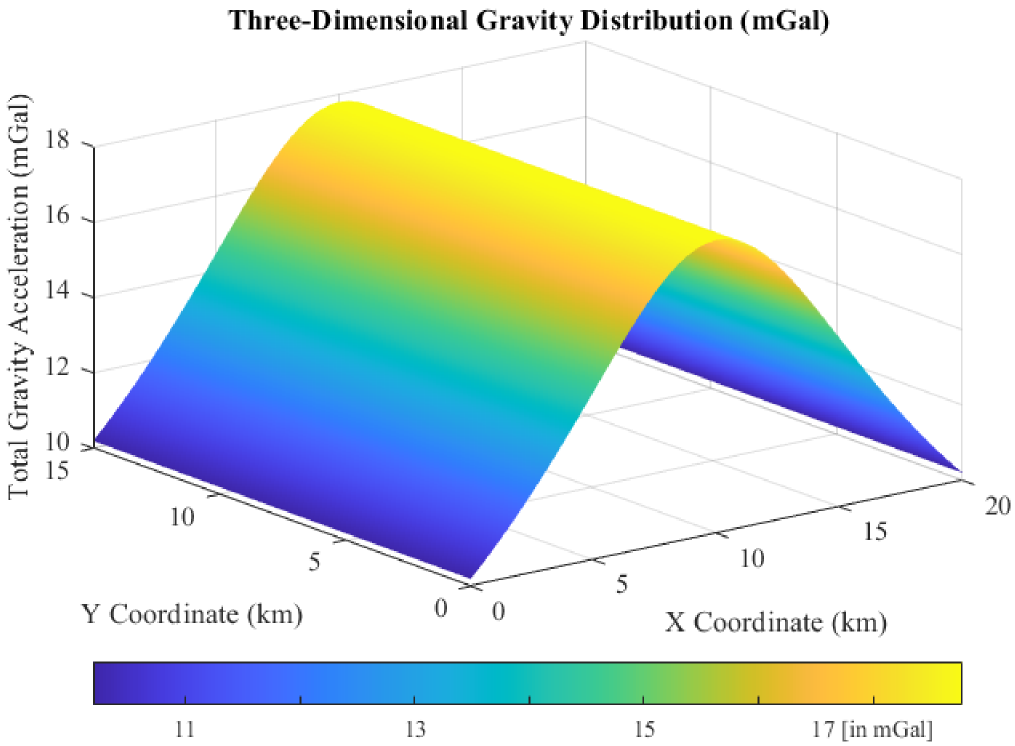

Figure 4:3D Gravity distribution of the infinite cylinder

Additional insight into the lateral distribution of the gravity field is provided by the contour representation shown in Figure 5. The contour map displays a series of nearly parallel contour bands that extend continuously along the longitudinal direction, indicating that the gravity field is invariant with respect to the y-coordinate. The highest gravity values are concentrated directly above the cylinder axis, whereas progressively lower gravity values occur symmetrically toward both sides of the observation area. The absence of closed contour patterns or localized anomaly concentrations further confirms that the computed gravity field corresponds to an infinitely long cylindrical body rather than a finite object. This spatial pattern agrees with the theoretical expectation that an infinite cylinder produces a gravity field controlled solely by radial distance, without longitudinal edge effects [16].

Taken together, the gravity profile, three-dimensional surface, and contour map provide mutually consistent evidence that the proposed MATLAB implementation successfully reproduces the analytical characteristics of the gravitational field generated by an infinite homogeneous solid cylinder. The agreement between the numerical simulation and the theoretical model demonstrates the correctness of the implemented analytical equations, the observation-grid formulation, and the matrix-based computational strategy. This verification establishes a reliable computational foundation for the subsequent simulation of finite solid cylinders, where gravitational responses are influenced not only by radial distance but also by finite-length effects that require numerical superposition.

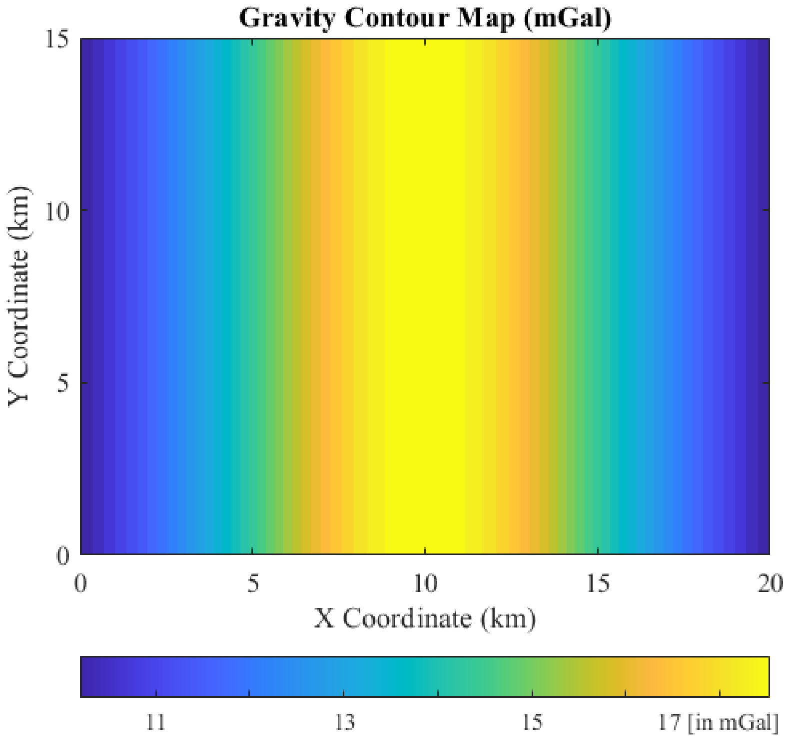

Figure 5: Gravity contour map of the infinite cylinder

4.2 Verification and Physical Interpretation of the Finite Cylinder Model

Following the successful verification of the infinite-cylinder formulation, the proposed MATLAB framework was subsequently applied to simulate the gravitational field generated by a homogeneous finite solid cylinder. In contrast to the infinite-cylinder approximation, whose gravity depends solely on the radial distance from the cylinder axis, the finite-cylinder formulation incorporates the influence of the cylinder's two end boundaries through the analytical boundary functions $F1$ and $F2$introduced in Equations (8)–(13). Consequently, the gravitational field becomes dependent not only on the radial distance but also on the longitudinal position relative to the finite extent of the cylinder. This additional geometric dependence fundamentally modifies both the amplitude and the spatial distribution of the resulting gravity anomaly.

Unlike conventional numerical approaches that subdivide the cylinder into numerous elementary segments and evaluate their contributions through numerical superposition, the present implementation directly evaluates the analytical expressions derived in Section 2.3 over the entire observation grid using matrix operations. The observation coordinates are first generated using MATLAB's meshgrid function, after which the auxiliary boundary functions are simultaneously evaluated for every observation point. The gravity components g_x, g_y, and $g_z$are subsequently obtained through element-wise analytical computations without any longitudinal discretization or iterative accumulation. This implementation preserves the exact analytical formulation while significantly reducing computational complexity and ensuring smooth numerical solutions over the entire observation domain [17].

Figure 6 presents gravity profiles extracted from two observation lines corresponding to the central axis of the cylinder (y=yq) and the upper end boundary (y=yq+L/2). The profile measured directly above the cylinder centre exhibits a single positive anomaly whose maximum occurs immediately above the projection of the cylinder axis. This behaviour is expected because the observation point experiences the largest combined contribution from the entire cylindrical mass when positioned at the longitudinal centre of the body. As the observation point moves away from the centre along the x-direction, the radial distance increases continuously, producing a smooth and symmetric reduction in gravitational acceleration.

A second profile extracted near the cylinder end displays noticeably smaller amplitudes throughout the observation line. This reduction directly reflects the finite-length effect incorporated analytically through the boundary functions F1 and F2. Near the cylinder ends, part of the surrounding mass is absent beyond the terminal boundary, reducing the total gravitational attraction compared with that at the cylinder centre. The resulting difference between the two profiles clearly demonstrates that, unlike the infinite-cylinder model, gravitational acceleration is no longer translationally invariant along the longitudinal direction. Instead, the anomaly amplitude varies systematically with longitudinal position because the finite cylinder possesses physical boundaries that modify the spatial distribution of mass contributing to the observed gravitational field. The absence of numerical oscillations, discontinuities, or artificial fluctuations along either profile further confirms that the analytical matrix-based implementation evaluates the governing equations consistently across the observation grid while maintaining excellent numerical stability.

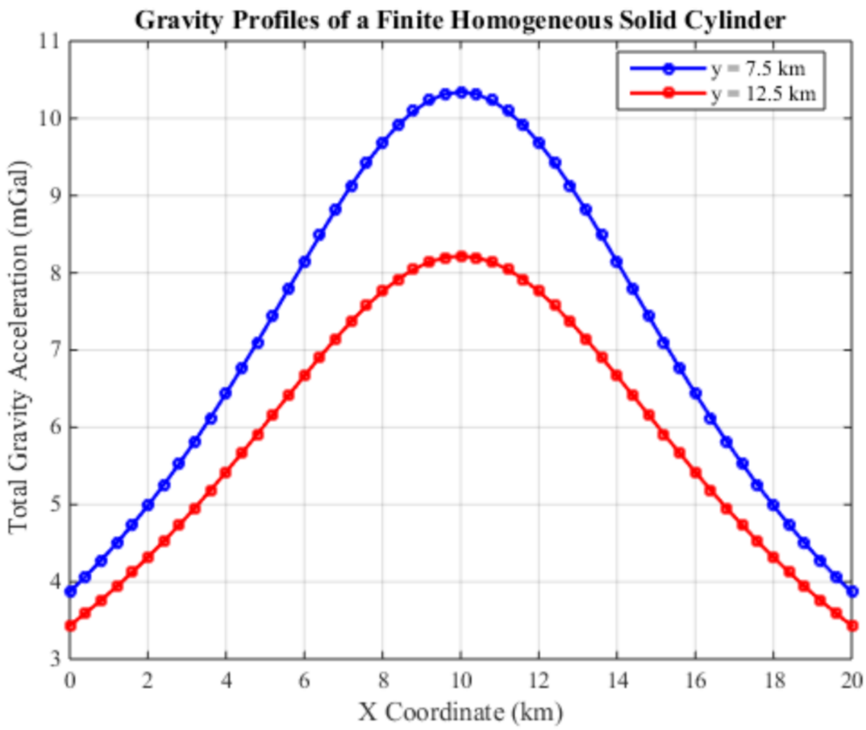

Figure 6: Gravity profile of the finite cylinder

The spatial characteristics of the finite-cylinder anomaly become more apparent in the three-dimensional surface representation shown in Figure 7. Unlike the elongated ridge observed for the infinite-cylinder model, the gravity surface generated by the finite cylinder forms a localized dome-shaped anomaly centred directly above the cylindrical body. The anomaly reaches its maximum amplitude above the geometric centre of the cylinder and decreases continuously in all horizontal directions.The reduction of gravity toward both cylinder ends is particularly significant because it reflects the finite axial extent explicitly represented in the analytical formulation. While the radial decay remains governed by the increasing distance from the cylinder axis, an additional longitudinal attenuation occurs as the observation position approaches either end of the cylinder. Consequently, the gravity surface exhibits curvature in both the *x*- and *y*-directions, producing a fully three-dimensional anomaly rather than the ridge-like geometry characteristic of an infinite cylinder.

The smoothness of the computed surface also demonstrates the effectiveness of the matrix-oriented implementation adopted in the MATLAB program. Since all observation points are processed simultaneously through vectorized analytical operations, the resulting gravity field remains continuous over the entire observation grid without introducing interpolation artefacts or discretization errors.

Additional insight into the spatial behaviour of the finite-cylinder anomaly is obtained from the contour representation shown in Figure 8. In contrast to the nearly parallel contour bands observed for the infinite-cylinder model, the finite-cylinder response is characterized by a series of concentric closed contours centred above the cylinder midpoint. These closed contour patterns indicate that the gravitational field decreases simultaneously in both the radial and longitudinal directions.

The highest contour values occupy a relatively compact region surrounding the cylinder centre, whereas progressively lower contour levels extend outward toward the edges of the observation domain. The gradual contraction of contour spacing near the anomaly peak indicates relatively steep gravity gradients immediately above the cylinder, while the wider spacing observed farther from the centre reflects the progressive weakening of the gravitational field. Most importantly, the closed contour geometry provides direct visual evidence of the finite-length effect incorporated into the analytical model. Because gravitational attraction diminishes near both terminal boundaries, the anomaly becomes spatially confined rather than infinitely elongated. This contour configuration therefore represents one of the most distinctive differences between finite and infinite cylindrical bodies and provides a useful qualitative indicator for distinguishing elongated geological structures of finite extent in practical gravity interpretation [1].

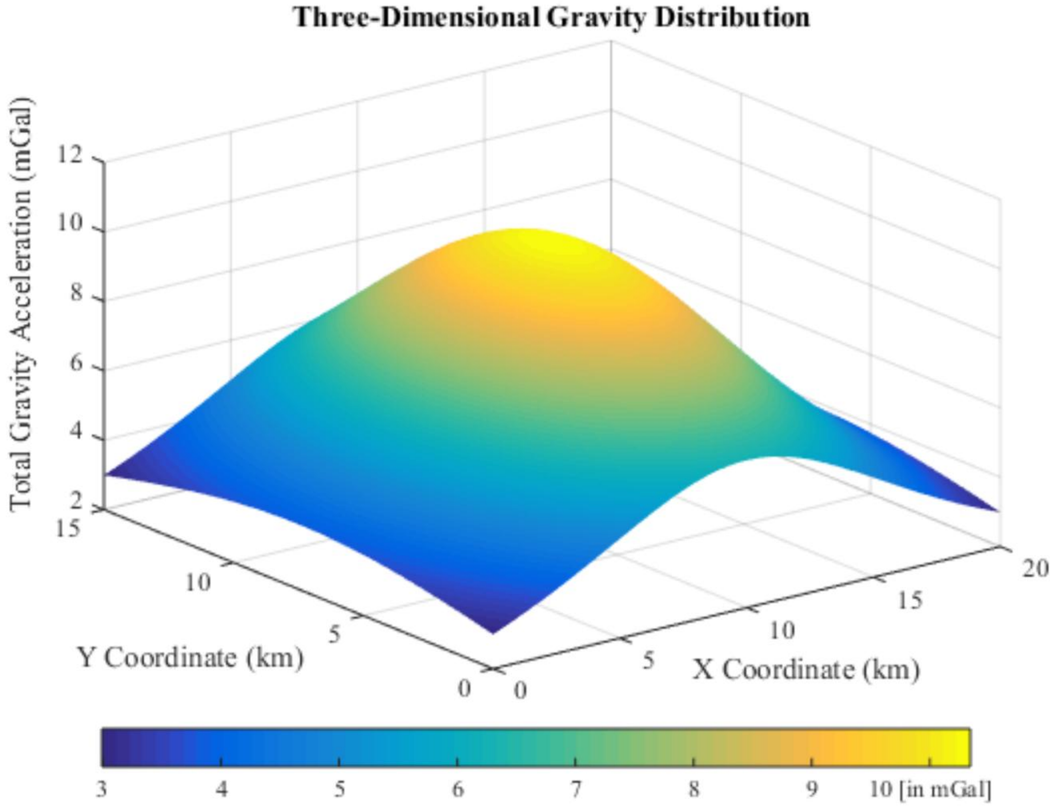

Figure 7: 3D Gravity distribution of the finite cylinder

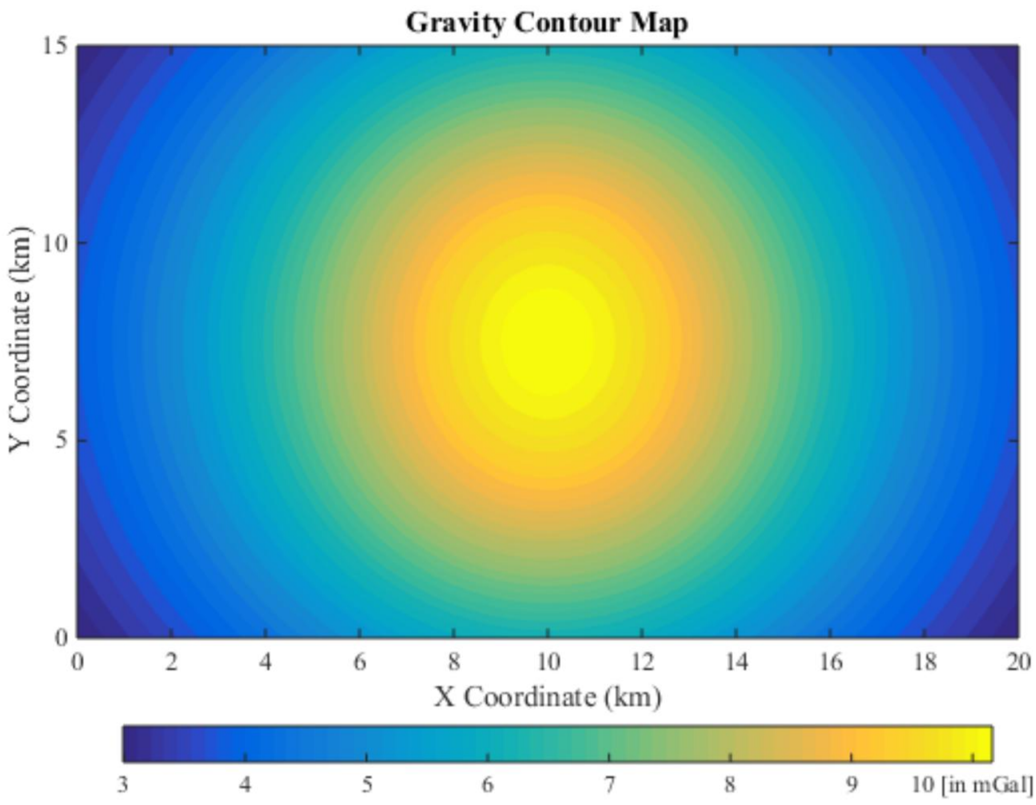

Figure 8: Gravity contour map of the infinite cylinder

4.3 Convergence of the Finite-Cylinder Gravity Field toward the Infinite-Cylinder Solution

Although Sections 4.1 and 4.2 demonstrated that the proposed MATLAB implementation accurately reproduces the gravitational fields of both infinite and finite solid cylinders, an important theoretical question remains regarding the underlying physical relationship between these two geometric models. In classical potential-field theory, the infinite cylinder model does not represent an isolated physical entity; rather, it constitutes the theoretical limiting case of a finite cylinder whose axial length (L) becomes significantly larger than both its radius (a) and the dimensions of the observation grid. Consequently, a physically consistent forward-modeling algorithm must exhibit a continuous transition from a localized gravity anomaly associated with a finite cylinder toward the translationally invariant gravity field produced by an infinite cylinder [18]. Demonstrating this convergence process serves as a crucial validation of both the analytical formulation developed in Section 2.3 and the matrix-based computational implementation detailed in previously section.

To investigate this behavior, a parametric experiment was conducted using the developed MATLAB program by varying solely the cylinder length (L), while keeping all other physical parameters constant (i.e., density contrast ρ, cylinder radius a, depth to center z_q, and observation grid dimensions). Simulations were carried out sequentially for cylinder lengths of 15 km, 20 km, 25 km, and 35 km using the same matrix-based computational procedure as in the previous simulations. Because the underlying analytical equations remained unchanged and only the cylinder length parameter was modified, any observed variation in the gravity field pattern is strictly attributable to the finite-length effect, which is analytically represented by the boundary functions $F_1$and F_2. This computational strategy provides a direct numerical experiment for examining the convergence behavior of the finite-cylinder solution without introducing additional sources of uncertainty or computational approximation.

Figure 9 presents the results of this parametric experiment as four contour maps corresponding to progressively increasing cylinder lengths. At L = 15 km, the gravity anomaly exhibits a highly localized pattern with closed elliptical contours centered around the anomaly peak. These contours indicate that the gravitational acceleration attenuates in both the radial and longitudinal directions due to the strong influence of the ends of cylinder. Thus, the gravity field distribution fully reflects the characteristics of a bounded geological body with distinct physical endpoints.

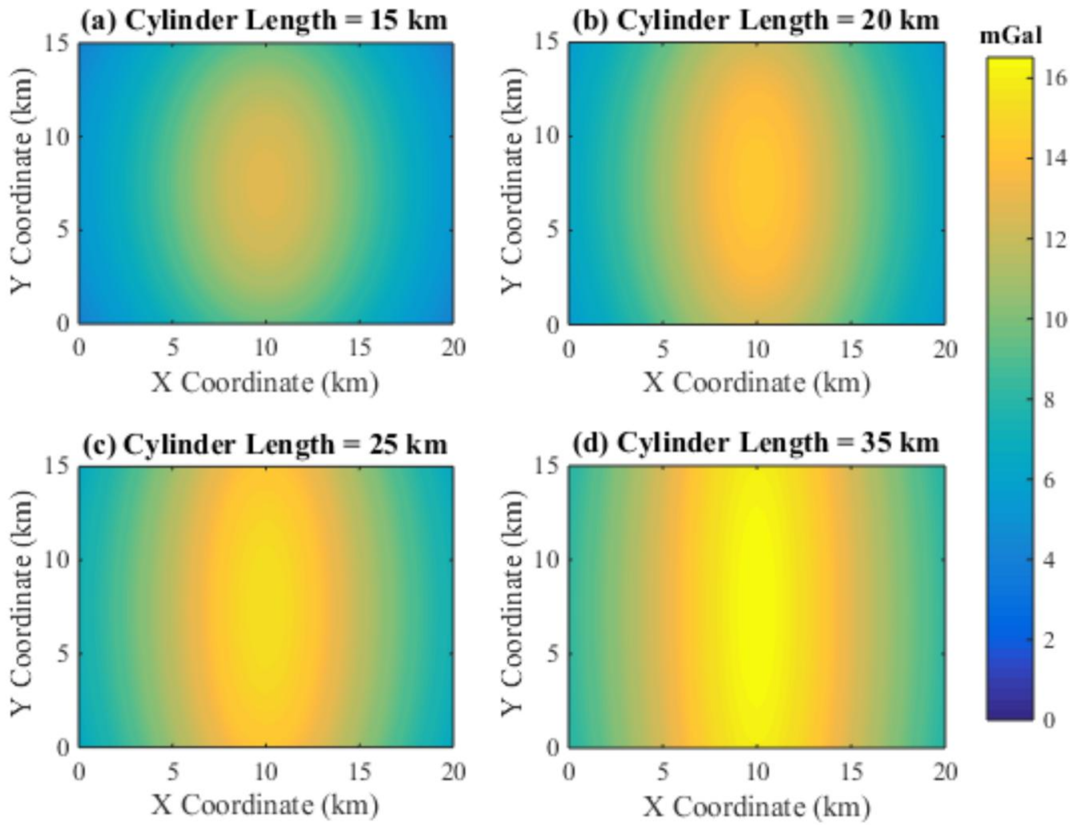

Figure 9: Progressive convergence of finite-cylinder gravity anomalies toward the infinite-cylinder solution with increasing cylinder length (L = 15, 20, 25, and 35 km).

As the cylinder length increases to 2o km and 25 km, a gradual change in the gravity field distribution becomes apparent. The gravity anomaly elongates along the longitudinal axis, while the boundary effects from both ends progressively diminish. The region exhibiting relatively constant gravity values expands along the cylinder axis, indicating that the central portion of the cylinder begins to behave as though it were unaffected by the presence of the terminal endfaces.

This transformation is also vividly captured in the contour geometry. At shorter cylinder lengths, the contours remain as closed elliptical shapes (closed elliptical contours) due to the bounded mass distribution in the longitudinal direction. However, as the cylinder length increases, these ellipses undergo a continuous deformation, stretching longitudinally such that their major axes expand along the cylinder strike. With further increases in length, the contours cease to form compact closed curves and instead transition into a nearly parallel contour pattern (parallel contour pattern). In other words, a smooth spatial transition occurs from an elliptical contour morphology to a parallel contour configuration as a direct consequence of the diminishing influence of the terminal end faces on the gravity field distribution.

Physically, this phenomenon occurs because the distance from observation points near the center of the survey area to the cylinder ends increases significantly. As a result, the relative contribution of the analytical boundary functions $F_1$and F_2 to the total gravitational acceleration decreases relative to the mass contribution from the central portion of the cylinder. Consequently, the gravity field around the central observation area progressively approaches the behavior predicted by the infinite cylinder model, where the field distribution is governed solely by the radial distance from the cylinder axis.

At L = 35 km, the simulation results show that the gravity anomaly pattern closely resembles the infinite-cylinder simulation presented in Section 4.1. Although the cylinder geometrically retains a finite length, the contour distribution within the observation domain becomes nearly invariant with respect to the longitudinal direction. The contour lines extend nearly parallel to the cylinder axis, while the end effects are restricted to regions far outside the primary observation grid. Therefore, the spatial transition from elliptical to parallel contour patterns serves as a clear visual indicator that the finite-cylinder solution has converged toward the infinite-cylinder solution.

This morphological evolution also establishes a strong conceptual link between the findings in Sections 4.1 and 4.2. Section 4.1 showed that an infinite cylinder generates a ridge-shaped anomaly with parallel contour lines because the gravity field depends exclusively on radial distance. Conversely, Section 4.2 showed that the presence of cylinder endpoints causes localized anomalies with closed elliptical contours and longitudinal attenuation. The results in Figure 9 bridge these two regimes by demonstrating that both models are interconnected through a continuous physical convergence process. Hence, the finite-cylinder model is not a distinct formulation, but rather a more general framework that naturally approaches the infinite-cylinder solution as the influence of the terminal ends vanishes.

From a computational standpoint, this convergence process provides robust validation for the proposed analytical implementation. All simulations utilized an identical observation grid generated via MATLAB's meshgrid function, identical analytical boundary functions, and identical element-wise matrix operations. The sole modified variable was the cylinder length. Therefore, the smooth and continuous shift in the anomaly pattern demonstrates that the algorithm responds consistently to geometric variations without introducing numerical instabilities, artificial discontinuities, or discretization artifacts. Furthermore, the simultaneous evaluation of analytical expressions across all observation points via vectorized matrix operations ensures both high numerical accuracy and superior computational efficiency [19].

The results in Figure 9 also carry important practical implications for gravity data interpretation. Many elongated geological structures, such as dikes, intrusive bodies, ore bodies, and magma channels, possess axial lengths far greater than the dimensions of a typical survey area. Under such conditions, these findings confirm that the infinite-cylinder approximation provides a suitable representation, as the influence of the body's endpoints on the observation domain is negligible. Conversely, when the survey area is situated near the terminal boundary of a geological structure, the finite-cylinder model becomes necessary to explicitly account for end effects. Thus, the transition from elliptical to parallel contour patterns shown in Figure 9 provides a valuable visual guideline for selecting the most appropriate geometric model based on the ratio between the length of the geological body and the survey domain dimensions [20].

In summary, the convergence experiment presented in Figure 9 not only validates the numerical stability of the developed algorithm, but also illustrates a fundamental physical property of the cylindrical gravity field: the finite-cylinder solution converges continuously toward the infinite-cylinder solution as the cylinder length increases. The most explicit evidence of this convergence is the gradual transition of contour patterns from closed ellipses to parallel lines, reflecting the progressive loss of end-face influence on the gravity field distribution. The excellent agreement between theoretical principles, physical interpretation, and numerical simulation results underscores the accuracy, consistency, and reliability of the proposed MATLAB-based gravity simulation framework.

V. CONCLUSSION

This study has successfully developed a MATLAB-based numerical simulation framework to model the gravitational field response of homogenous infinite and finite solid cylinders across a two-dimensional observation grid. The numerical results demonstrate that the implemented analytical formulations accurately capture the primary physical characteristics of both cylindrical geometries. For the infinite solid cylinder, the gravitational field exhibits translational invariance along the cylinder axis, manifested as a symmetric gravity profile, a 3D ridge-shaped anomaly, and parallel contour patterns governed strictly by radial distance. Conversely, for the finite solid cylinder, the inclusion of analytical boundary functions explicitly accounts for terminal end effects, producing a localized, dome-shaped 3D anomaly with closed elliptical contours that reflect longitudinal attenuation.

The primary scientific contribution of this research is demonstrated through the systematic convergence analysis. By progressively increasing the cylinder length (L), the finite-cylinder gravitational field seamlessly converges toward the infinite-cylinder solution as terminal boundary effects diminish within the survey domain. The most explicit visual evidence of this continuous physical transition is the morphological evolution of the anomaly contours, which deform gradually from closed ellipses into parallel contour patterns. This confirms that the finite-cylinder formulation serves as a generalized framework that naturally approaches the 2D limiting case when the axial length becomes significantly larger than the survey extent.

From a computational standpoint, leveraging MATLAB's vectorized matrix operations in conjunction with the meshgrid grid generation structure allows simultaneous evaluation across all observation points. This approach eliminates the need for iterative loops or numerical superposition, preserving pure analytical precision while delivering smooth, artifact-free spatial distributions with high computational efficiency and numerical stability.Overall, the excellent agreement between analytical theory, numerical simulations, and physical convergence confirms that the proposed MATLAB framework provides an accurate, efficient, and reliable tool for forward modeling, algorithm verification, and advanced geophysical research.

REFERENCES

[1]. R. J. Blakely, Potential Theory in Gravity and Magnetic Applications. Cambridge: Cambridge University Press, 1996.
[2]. MATLAB, MATLAB R2024b Documentation, MathWorks Inc., 2024.
[3]. L. Hamimu, J. Safani, and L. O. Ngkoimani, "Developed Forward and Inverse Modelling of Vertical Electrical Sounding (VES) Using MATLAB Implementation," International Journal of Science and Research, vol. 4, no. 11, pp. 1592–1597, 2015.
[4]. L. Hamimu, "ForLoveDCs: A Set of MATLAB Function for Calculating the Forward Modeling of Multimode Love Wave Dispersion Curves," Journal of Software Engineering and Simulation, vol. 9, no. 4, pp. 129–140, 2023.
[5]. L. Hamimu, S. R. Haraty, A. Rubaiyn, L. O. I. Juarzan, and Indrawati, "Computation and Modelling of Seismic Refraction Tomography for Anisotropic Medium Based on Pseudo Bending Method," Journal of Physics: Conference Series, vol. 2734, no. 1, Art. no. 012049, 2024.
[6]. L. Hamimu, E. Cahyono, H. Budiman, S. R. Haraty, and N. Ransi, "Numerical Analysis of the Geophysical Parameter Sensitivity on the Multimode Love Wave Dispersion Curves," Journal of Physics Communications, vol. 8, Art. no. 015007, 2024.
[7]. L. Hamimu, L. Aba, Muliddin, E. Cahyono, L. O. I. Juarzan, and Indrawati, "Implementing a New Inversion Technique of Geophysical Data Incorporating Global Optimization Method Based on Systematic Search, Permutation, and Sensitivity," IOP Conference Series: Earth and Environmental Science, vol. 1521, Art. no. 012017, 2025.
[8]. Wu, L., Modified Parker's Method for Gravitational Forward and Inverse Modeling Using General Polyhedral Models,Journal of Geophysical Research: Solid Earth, vol. 126, e2021JB022553, 2021.
[9]. Telford, W. M.,Geldart, L. P.,and Sheriff, R. E.,Applied Geophysics, 2nd ed. Cambridge: Cambridge University Press, 1990.
[10]. Li, Y., and Oldenburg, D. W., 3-D inversion of gravity data, Geophysics, 63(1), 109–119, 1998.
[11]. Last, B. J., and Kubik, K., Compact gravity inversion, Geophysics, 58(9), 1319–1337, 1993.
[12]. A. G. Camacho, P. Vajda, C. A. Miller, and J. Fernández, "A Free-Geometry Geodynamic Modelling of Surface Gravity Changes Using Growth-dg Software," Scientific Reports, vol. 11, Art. no. 23442, 2021.
[13]. L. T. Pham, E. Oksum, and M. N. Dolmaz, GRV_D_inv: A Graphical User Interface for 3D Forward and Inverse Modeling of Gravity Data,Geophysical Journal, vol. 43, no. 1, pp. 181–193, 2021.
[14]. [D. Sampietro and M. Capponi, Seismic Constrained Gravity Inversion: A Reliable Tool to Improve Geophysical Models Away from Seismic Information, Geosciences, vol. 11, no. 11, Art. no. 467, 2021.
[15]. David Miles, Closed-form expressions for the non-axial component of the gravitational field of an arbitrary cylinder segment, Journal of Applied Geophysics, Volume 159, Pages 621-630, 2018.
[16]. M F Asgharzadeh , H Hashemi , R RB von Frese, Comprehensive gravitational modeling of the vertical cylindrical prism by Gauss–Legendre quadrature integration, Geophysical Journal International, Volume 212, Issue 1, Pages 591–611, 2018.
[17]. Xu, X., & Lockerbie, N. A., Forward modelling of geophysical survey data using cylindrical elemental volumes. Nuovo Cimento B, 111(4), 497–506, 1996.
[18]. Damiata, B. N., & Lee, T. C., Gravitational attraction of solids of revolution: Part 2: General expressions. Journal of Applied Geophysics, 50, 351–373, 2002.
[19]. R.Nagendra, P.V.S. Prasad, V.L.S. Bhimasankaram, Forward and inverse computer modeling of a gravity field resulting from a density interface using Parker-Oldenberg method, Computers & Geosciences, Volume 22, Issue 3, 1996.
[20]. Robert L. Parker, The Theory of Ideal Bodies for Gravity Interpretation, Geophysical Journal International, Volume 42, Issue 2, Pages 315–334, 1975.

Appendix

```
%% Numerical Simulation of Gravity Responses of
% Infinite and Finite Solid Cylinders Using Analytical
% Matrix-Based MATLAB Implementation
% Journal: Journal of Software Engineering and
% Simulation (JSES)
% Description:
% This program computes the forward gravity response generated by a
% homogeneous finite solid cylinder using the analytical solution derived
% from Newton's law of gravitation. The gravity field is evaluated over a
% two-dimensional observation grid.
% Outputs:
%   (1) Gravity profiles
%   (2) Three-dimensional gravity distribution
%   (3) Gravity contour map
%   (4) Transition from finite cylinder to infinite cylinder
%% Version : 1.0
% MATLAB Release Compatibility : R2014b or later
% Computational Method :
% Analytical Forward Gravity Modeling
% Geometry :
% Homogeneous Solid Cylinder
% Observation Grid :
% Two-Dimensional Cartesian Grid

clc; clear; close all;
%% PHYSICAL CONSTANTS
G = 6.67430e-11;        % Universal gravitational constant (m^3 kg^-1 s^-2)
si2mg = 1.0e5;          % SI to mGal conversion factor
km2m = 1.0e3;           % Kilometer to meter conversion factor
%% CYLINDER PARAMETERS
xq = 10.0;              % X-coordinate of cylinder center (km)
yq = 7.5;               % Y-coordinate of cylinder center (km)
zq = -7.0;              % Depth to cylinder center (km)
a = 1.0;                % Cylinder radius (km)
L = 10.0;               % Cylinder length (km)
rho = 2970;             % Density contrast (kg/m^3)
%% DERIVED PARAMETERS
lambda = rho*pi*(a*km2m)^2;
y1 = yq - L/2;
y2 = yq + L/2;
%% TWO-DIMENSIONAL OBSERVATION GRID
xmin = 0; xmax = 20;
ymin = 0; ymax = 15;
nx = 51; ny = 31;
x = linspace(xmin,xmax,nx);
y = linspace(ymin,ymax,ny);
[xp,yp] = meshgrid(x,y);
zp = zeros(size(xp));
%% OBSERVATION GEOMETRY
rx = (xp-xq)*km2m;
rz = (zp-zq)*km2m;
R2 = rx.^2 + rz.^2;
R2(R2==0)=eps;
u1 = (yp-y1)*km2m;
u2 = (yp-y2)*km2m;
%% ANALYTICAL FUNCTIONS
F1 = u1 ./ sqrt(R2 + u1.^2);
F2 = u2 ./ sqrt(R2 + u2.^2);
```

```
%% GRAVITY COMPONENTS
gx = -G*lambda .* (rx./R2) .* (F1-F2);
gy =  G*lambda .* (1./sqrt(R2+u1.^2)- 1./sqrt(R2+u2.^2));
gz = -G*lambda .* (rz./R2) .* (F1-F2);
%% UNIT CONVERSION
gx = gx*si2mg;
gy = gy*si2mg;
gz = gz*si2mg;
%% TOTAL GRAVITY MAGNITUDE
gtot = sqrt(gx.^2 + gy.^2 + gz.^2);
%% OBSERVATION PROFILES
[~,iy1] = min(abs(y-yq));
yobs2 = yq + L/2;
[~,iy2] = min(abs(y-yobs2));
%% FIGURE 1: Gravity Profiles
figure(1)
plot(xp(iy1,:),gtot(iy1,:),'b-o','LineWidth',1.5,...
'MarkerSize',4)
hold on
plot(xp(iy2,:),gtot(iy2,:),'r-s','LineWidth',1.5,...
'MarkerSize',4)
xlabel('X Coordinate (km)')
ylabel('Total Gravity Acceleration (mGal)')
title('Gravity Profiles of a Finite Homogeneous Solid Cylinder')
legend(...
['y = ',num2str(y(iy1)),' km'],...
['y = ',num2str(y(iy2)),' km'],...
'Location','Best')
grid on; box on
%% FIGURE 2: Three-Dimensional Gravity Distribution
figure(2)
surf(xp,yp,gtot)
shading interp
xlabel('X Coordinate (km)')
ylabel('Y Coordinate (km)')
zlabel('Total Gravity Acceleration (mGal)')
title('Three-Dimensional Gravity Distribution')
% view(45,35)
cb = colorbar('horiz');
tick = get(cb,'XTick');
label = get(cb,'XTickLabel');
label{end} = sprintf('%g [in mGal]',tick(end));
set(cb,'XTickLabel',label)
grid on
%% FIGURE 3: Gravity Contour Map
figure(3)
[~,h] = contourf(xp,yp,gtot,40);
set(h,'LineColor','none')
xlabel('X Coordinate (km)')
ylabel('Y Coordinate (km)')
title('Gravity Contour Map')
cb = colorbar('horiz');
tick = get(cb,'XTick');
label = get(cb,'XTickLabel');
label{end} = sprintf('%g [in mGal]',tick(end));
set(cb,'XTickLabel',label)
% grid on
%% TRANSITION FROM FINITE CYLINDER TO INFINITE CYLINDER
Lset = [15 20 25 35];
```

```
gmax_global = 0;
for k = 1:length(Lset)
   Ltest = Lset(k);
   y1t = yq - Ltest/2;
   y2t = yq + Ltest/2;
   u1t = (yp-y1t)*km2m;
   u2t = (yp-y2t)*km2m;
   F1t = u1t ./ sqrt(R2 + u1t.^2);
   F2t = u2t ./ sqrt(R2 + u2t.^2);
   gx_t = -G*lambda .* (rx./R2) .* (F1t-F2t);
   gy_t =  G*lambda .* (1./sqrt(R2+u1t.^2) ...
       - 1./sqrt(R2+u2t.^2));
   gz_t = -G*lambda .* (rz./R2) .* (F1t-F2t);
   gx_t = gx_t*si2mg;
   gy_t = gy_t*si2mg;
   gz_t = gz_t*si2mg;
   gtot_t = sqrt(gx_t.^2 + gy_t.^2 + gz_t.^2);
   gmax_global = max(gmax_global,max(gtot_t(:)));
end
%% FIGURE 4
% Transition from Finite Cylinder to Infinite Cylinder
figure(4)
for k = 1:length(Lset)
   Ltest = Lset(k);
   y1t = yq - Ltest/2;
   y2t = yq + Ltest/2;
   u1t = (yp-y1t)*km2m;
   u2t = (yp-y2t)*km2m;
   F1t = u1t ./ sqrt(R2 + u1t.^2);
   F2t = u2t ./ sqrt(R2 + u2t.^2);
   gx_t = -G*lambda .* (rx./R2) .* (F1t-F2t);
   gy_t =  G*lambda .* (1./sqrt(R2+u1t.^2) ...
       - 1./sqrt(R2+u2t.^2));
   gz_t = -G*lambda .* (rz./R2) .* (F1t-F2t);
   gx_t = gx_t*si2mg;
   gy_t = gy_t*si2mg;
   gz_t = gz_t*si2mg;
   gtot_t = sqrt(gx_t.^2 + gy_t.^2 + gz_t.^2);
   subplot(2,2,k)
   [~,h] = contourf(xp,yp,gtot_t,40);
   set(h,'LineColor','none')
   caxis([0 gmax_global])
   hold on
   hold off
   title(sprintf('(%c) Cylinder Length = %.0f km',...
      char('a'+k-1),Ltest))
   xlabel('X Coordinate (km)')
   ylabel('Y Coordinate (km)')
   axis equal tight
end
cb = colorbar('Position',[0.92 0.11 0.02 0.815]);
title(cb,'mGal')
```