Physics International

Research Article

A Generalized Algorithm for the Gravitational Potential of a Homogeneous Cuboid

Thunendran Periyandy[1,2] and Michael Bevis[1]

[1]*Division of Geodetic Science, School of Earth Sciences, The Ohio State University, Columbus, OH 43210, USA*
[2]*Faculty of Geomatics, Sabaragamuwa University of Sri Lanka, Sri Lanka*

Article history
Received: 08-05-2025
Revised: 18-07-2025
Accepted: 01-08-2025

Corresponding Author:
Thunendran Periyandy
Division of Geodetic Science, School of Earth Sciences, The Ohio State University, Columbus, OH 43210, USA;
Faculty of Geomatics, Sabaragamuwa University of Sri Lanka, Sri Lanka
Email: periyandy.1@osu.edu

Abstract: This paper presents a mathematical formulation for computing the gravitational potential of a cuboid (or right rectangular prism) with uniform density. The resulting analytical expressions are valid inside, on and outside the cuboid. The methodology systematically organizes logarithmic and arctangent terms using a hexagonal graph approach, promoting numerical stability and correct sign allocation. We provide a structured pseudocode algorithm to facilitate computational implementation, and actual codes written in Python, Matlab and Julia. We also introduce a test, that we call a Laplacian test, used to validate our algorithm and codes.

Keywords: Gravitational Potential, Cuboid, Hexagonal Graph, Finite Differences, Laplacian Test

Introduction

There is a large scientific literature, developed over more than one century, addressing the gravitational field produced by three-dimensional bodies with simple or idealized shapes such as cubes, plates, prisms and polyhedra (Macmillan, 1930; Telford et al., 1976; Werner, 1994; Chappell et al., 2012; Tsoulis and Gavriilidou, 2021). In most cases, the solutions for the gravitational potential (and/or gravitational acceleration) are valid outside of the body, but not in its interior.

In situations in which the density and shape of a mass distribution is varying in two dimensions but not in a third, we can model the gravitational field as that due to a polygon or a set of polygons. Hubbert (1948) showed that the gravitational attraction due to a polygon can be expressed in terms of a line integral around its periphery. This idea was further developed by Talwani et al. (1959) and Grant and West, (1965), and led to computer codes used by geophysicists to model gravitational measurements. Won and Bevis (1987) revisited this problem. They began by formulating expressions already widely used to compute the gravitational attraction exterior to the polygon, and showed that by introducing some special case handling related to terms involving logarithms and arctangents of quotients, the modified computations were valid inside, on and outside the polygon.

Given that many published algorithms for the gravitational field due to 3-D bodies of uniform density, such as a cube or a polyhedron, are valid only when evaluating the field exterior to the body, we wondered if similar special case handling could increase the utility of such algorithms by allowing the field to be computed anywhere inside, on or outside of the body. We decided to focus initially on the cuboid, also called a right rectangular prism, following Chappell et al. (2012), who addressed its exterior gravitational potential, V. So, the goal of this work is to modify the earlier solution so as to allow us to compute V anywhere inside, on or outside of the cuboid.

In what follows, we first describe our mathematical methods and the development of our algorithm. We then validate our algorithm and codes. This includes a discussion of a novel numerical test which we call the Laplacian test. This requires us to describe how we can control numerical round-off problems which otherwise would limit the precision of our tests, and which might constitute something of a mine field for some of the less experienced users of our codes. We end with a discussion and conclusions section, which includes consideration of potential applications and possible future extensions of this work.

Methods

We present a complete computational framework for evaluating the gravitational potential of a homogeneous

DOI: 10.3844/pisp.2025.7.17

cuboid. This includes a novel mathematical formulation, singularity handling techniques, and a structured algorithm with pseudocode for implementation.

Mathematical Formulation

The gravitational potential at a point $P(x, y, z)$ due to a continuous mass distribution in some finite volume Ω is given by:

$$V(x, y, z) = -G \iiint_{\Omega} \frac{\rho \, d\Omega}{r} \tag{1}$$

where G is the gravitational constant, ρ is the mass density, r is the Euclidean distance from the mass element to the evaluation point, and $d\Omega$ is the volumetric element (Heiskanen and Moritz, 1967).

We consider the case of a cuboid of constant density ρ, and dimensions $2L$, $2B$, and $2D$ corresponding to length, breadth, and depth, respectively (Fig. 1). Chappell et al. (2012) showed that the gravitational potential due to this cuboid is given by Eq. 2.

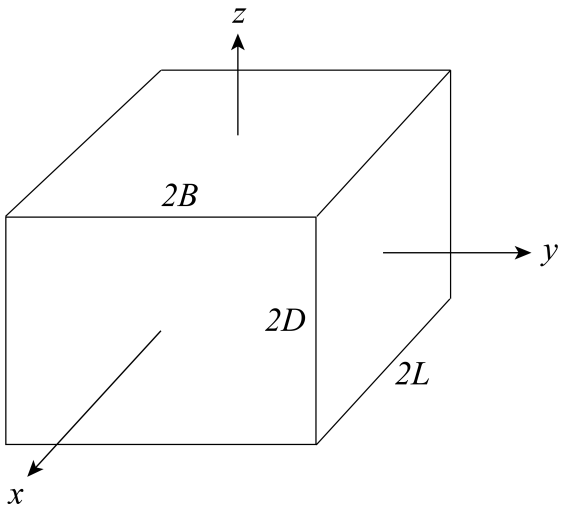


Fig. 1: A cuboid whose center coincides with the origin of a cartesian axis system. The dimensions of the cuboid are 2L, 2B and 2D in the directions of the x, y and z axes, respectively. In the special case that $L = B = D$ the cuboid becomes a cube

$$V(x,y,z) = G\rho \Bigg[\Bigg[\Bigg[YZ\log(X+r) - \frac{X^2}{2}\tan^{-1}\left(\frac{YZ}{Xr}\right) + XZ\log(Y+r) - \frac{Y^2}{2}\tan^{-1}\left(\frac{XZ}{Yr}\right) + XY\log(Z+r) - \frac{Z^2}{2}\tan^{-1}\left(\frac{XY}{Zr}\right) \Bigg]_{Z=X_6}^{X_5} \Bigg]_{Y=X_4}^{X_3} \Bigg]_{X=X_2}^{X_1} \tag{2}$$

where the limits X_i, defined thus

$$\begin{aligned} X_1 &= L - x, \quad X_2 = L + x, \quad X_3 = B - y, \\ X_4 &= B + y, \quad X_5 = D - z, \quad X_6 = D + z \end{aligned} \tag{3}$$

represent projected distances from the evaluation point (x, y, z) to the boundaries of the cuboid.

Hexagonal Graph Representation

To systematically organize the integral terms in the gravitational potential algorithm, we arranged the six coordinates $X_1, X_2, X_3, X_4, X_5, X_6$ as the nodes of a hexagonal graph.

Each node represents one of the six limits, and the edges indicate valid interactions, explicitly excluding connections to the opposite coordinate. This hexagonal graph systematically defines the structure of the integration components of Eq. 4 (Fig. 2).

$$V = G\rho \sum_{i=1}^{6} \sum_{\text{triples } (\mathcal{T}_i)} \Bigg[S_i X_j X_k \log(S_i X_i + R) - \frac{X_i^2}{2} \arctan\left(\frac{X_j X_k}{X_i R}\right) \Bigg] \tag{4}$$

where $R = \sqrt{X_i^2 + X_j^2 + X_k^2}$ and S_i is a sign function.

Mathematical Definition of Valid Triples $\mathcal{T}_i$

For each coordinate X_i, a valid triple (X_i, X_j, X_k) satisfies the following conditions:

1. X_i is always included in the triple.
2. The second and third elements (X_j, X_k) are chosen from the four nodes connected to X_i, ensuring
$$X_j, X_k \in \{X_a, X_b, X_c, X_d\}, \quad X_j \neq X_k \tag{5}$$
3. The opposite coordinate X_{i+1} is excluded, such that
$$X_j, X_k \neq X_{i+1} \tag{6}$$

Thus, the selection rule ensures a complete yet non-redundant interaction structure.

General Form of the Gravitational Potential

The gravitational potential of the cuboid is decomposed as

$$V = G\rho \textstyle\sum_{i=1}^{6} \left(V_L^i + V_T^i\right) \tag{7}$$

where V_L^i and V_T^i represent the logarithmic and arctangent-based components, respectively.

1) Logarithmic Terms

Each logarithmic component is expressed as

$$V_L^i = S_i \textstyle\sum_{(X_i, X_j, X_k) \in \mathcal{T}_i} X_j X_k \log\left(S_i X_i + \sqrt{X_i^2 + X_j^2 + X_k^2}\right) \tag{8}$$

where the summation over triples $\mathcal{T}_i$, determined by the local geometric structure of the cuboid.

The sign function $S_i \in \{-1, +1\}$ encodes the orientation of the coordinate axis and is defined as (Fig. 2a):

$$S_i = \begin{cases} +1, & \text{if } X_i \text{ points in the positive direction,} \\ -1, & \text{if } X_i \text{ points in the negative direction.} \end{cases} \tag{9}$$

DOI: 10.3844/pisp.2025.7.17

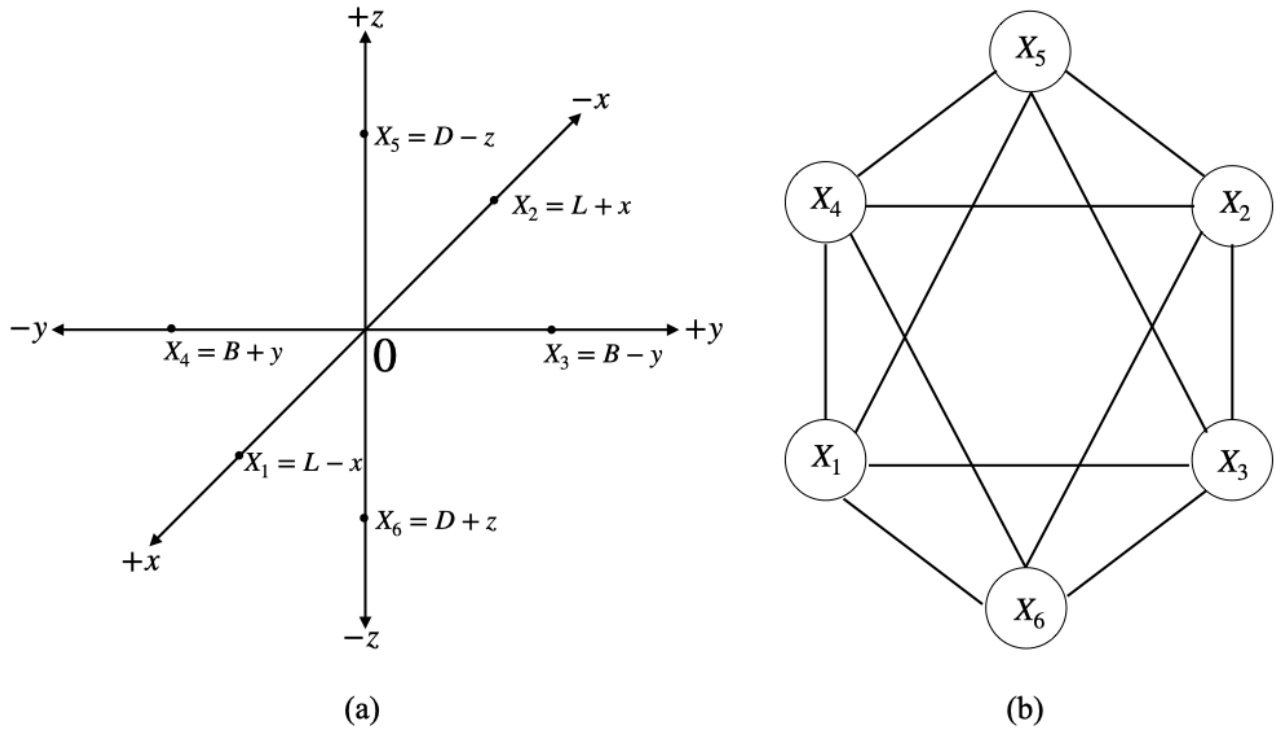


Fig. 2: (a) Directional representation $(+x, -x, +y, -y, +z, -z)$ and corresponding limits for the cuboid. (b) Hexagonal graph depicting the six directional components. Each node corresponds to one of the six limits $X_1, X_2, X_3, X_4, X_5, X_6$, with each connected to four others, excluding its opposite coordinate

The structure of the $\mathcal{T}_i$ ensures that each triple (X_i, X_j, X_k) includes the base coordinate X_i, and selects X_j and X_k according to the admissibility conditions outlined in Eqs. 5 and 6.

This formulation inherently reflects the directional asymmetry of logarithmic singularities across the surface, ensuring alignment with the cuboid's topological and geometrical properties. Notably, at the critical interface where $X_i = 0$, the logarithmic expression simplifies to a finite form: $\log\left(\sqrt{X_j^2 + X_k^2}\right)$, provided that $X_j^2 + X_k^2 > 0$, ensuring continuity and numerical stability of the integrand.

It is important to note that in Eq. 8 the factor S_i appears outside the logarithm. If S_i were set to zero when $X_i = 0$, the entire term V_L^i would incorrectly vanish. However, V_L^i remains finite at $X_i = 0$ due to the contributions from X_j and X_k within the logarithmic function. Therefore, S_i must be assigned based on the orientation of the coordinate axis relative to the cube's center, and not based on the instantaneous value of X_i. For example, $S_i = +1$ if the corresponding axis points in the positive direction, and $S_i = -1$ if it points in the negative direction, even when $X_i = 0$.

2) Arctangent Terms

The arctangent component is given by

$$V_T^i = -\frac{X_i^2}{2} \sum_{(X_i, X_j, X_k) \in \mathcal{T}_i} \tan^{-1}\left(\frac{X_j X_k}{X_i \sqrt{X_i^2 + X_j^2 + X_k^2}}\right) \quad (10)$$

Singularity Handling

We now explain the special case handling that we introduce for the two functions that are capable of producing numerical singularities.

1) Logarithmic Function Singularities

The logarithmic term becomes singular when

$$S_i X_i + \sqrt{X_i^2 + X_j^2 + X_k^2} \leq 0 \quad (11)$$

To ensure the function remains well-defined, we require

$$\sqrt{X_i^2 + X_j^2 + X_k^2} > -S_i X_i \quad (12)$$

Since square roots are non-negative, this condition is always satisfied.

2) Arctangent Function Singularities

The arctangent expression is undefined when

$$X_i \sqrt{X_i^2 + X_j^2 + X_k^2} = 0 \quad (13)$$

To prevent division by zero, we define a modified function

$$\text{special\_arctan}(X_jX_k,\, X_iR) = \begin{cases} \tan^{-1}\left(\frac{X_jX_k}{X_iR}\right), & \text{if } X_i \neq 0, \\ 0, & \text{if } X_i = 0, \end{cases} \tag{14}$$

where $R = \sqrt{X_i^2 + X_j^2 + X_k^2}$.

Compact Mathematical Representation

The full computation is expressed as

$$V = G\rho \sum_{i=1}^{6} \sum_{\text{triples}} \left[S_i X_j X_k \cdot \text{special\_log}\left(S_i X_i + R\right) - \frac{X_i^2}{2} \cdot \text{special\_arctan}\left(X_jX_k, X_iR\right) \right] \tag{15}$$

where $R = \sqrt{X_i^2 + X_j^2 + X_k^2}$.

Algorithm 1: Computation of Gravitational Potential of a Cuboid

Input: $x, y, z, L, B, D, \rho, G$

Output: V (Gravitational potential at (x, y, z))

Compute transformed coordinates:

$X_1 \leftarrow L - x, X_2 \leftarrow L + x$

$X_3 \leftarrow B - y, X_4 \leftarrow B + y$

$X_5 \leftarrow D - z, X_6 \leftarrow D + z$

Initialize $V \leftarrow 0$

for each direction dir in {Front, Back, Right, Left, Top, Bottom} **do**

Compute sign allocation S_i

for each valid (X_i, X_j, X_k) **do**

Check singularity condition:

if $S_iX_i + \sqrt{X_i^2 + X_j^2 + X_k^2} \leq 0$ **then**

Set $V_L^{\text{dir}} = 0$ (handle singularity)

else

Compute V_L^{dir} using logarithm

end if

if $X_i = 0$ **then**

Set Set $V_T^{\text{dir}} = 0$ (handle singularity)

else

Compute V_T^{dir} using arctangent

end if

end for

Update potential: $V \leftarrow V + V_L^{\text{dir}} + V_T^{\text{dir}}$

end for

Return: $V \times G \times \rho$

This algorithm outlines the computational procedure for evaluating the gravitational potential of a cuboid. The algorithm comprises three main components:

1. Singularity handling, which ensures numerical stability in logarithmic and arctangent evaluations.
2. Directional contributions, computed from the six directional components.
3. Summation of all contributions to obtain the total gravitational potential.

Validation

Computational tests demonstrate the accuracy and robustness of the formulation. These include validating the gravitational potential and acceleration, comparing results with a point-mass model, and verifying consistency through the Laplacian test.

Testing the Computation of Gravitational Potential

The gravitational potential V should be a continuous function inside, on and outside the cuboid, and we tested this by evaluating V at special locations on the surface and outside of the cuboid at pairs of points separated by distances that are tiny relative to the size of the cuboid. This enabled us to check that our special-case handling of the arctangent and logarithm terms was working as intended. In particular we examined exterior points in planes containing a face, and on lines containing an edge. On the surface of the cuboid we evaluated V on faces, on edges and on vertices. Examples results are tabulated in Appendix 1. Table 1 may prove useful to those adopting our codes who wish to test them on their computers.

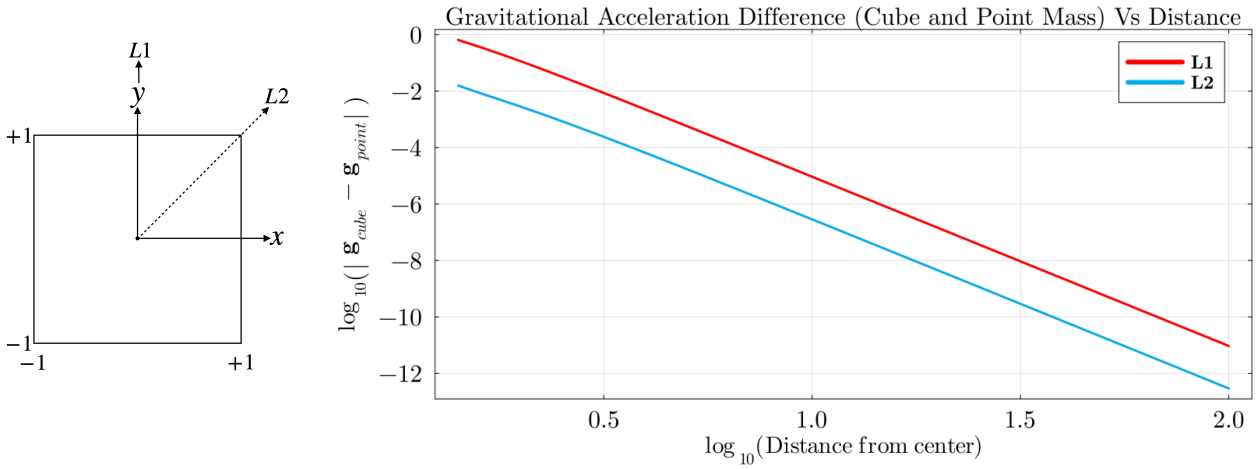
Computation of Gravitational Acceleration

We computed gravitational acceleration $\mathbf{g}$ using first-order finite differences. The vector $\mathbf{g}$ at a point (x, y, z) is given by $\mathbf{g} = -\nabla V(x, y, z)$. The components of $\mathbf{g}$ are approximated thus:

$$\begin{aligned} g_x &= -\tfrac{\partial V}{\partial x} \approx -\tfrac{V(x+h,y,z) - V(x-h,y,z)}{2h}, \\ g_y &= -\tfrac{\partial V}{\partial y} \approx -\tfrac{V(x,y+h,z) - V(x,y-h,z)}{2h}, \\ g_z &= -\tfrac{\partial V}{\partial z} \approx -\tfrac{V(x,y,z+h) - V(x,y,z-h)}{2h} \end{aligned} \tag{16}$$

where h is the finite difference step size. (We must switch to one-sided finite differences when we wish to compute $\mathbf{g}$ at the origin). The selection of h is crucial to avoid errors introduced by the linear approximation when h is too large, or by floating-point cancellation when h is too small. The optimal value of h varies with location. We can most easily avoid these problems by computing the finite difference approximation using many significant digits, using extended-precision arithmetic (`BigFloat` in Julia). We discuss the problem of selecting the value of h in more detail in Appendices 2 and 3.

DOI: 10.3844/pisp.2025.7.17



(a) (b)

Fig. 3: (a) Layout of the x-y plane at $z = 0$, showing the two evaluation paths: L1 along the y-axis and L2 along the diagonal direction $x = y$. (b) Gravitational acceleration difference between a homogeneous cube and an equivalent point mass along paths L1 and L2. The cube of dimensions $2 \times 2 \times 2$, with uniform density $\rho = 1$ and gravitational constant $G = 1$. The plotted quantity is $\log_{10}\left(|\mathbf{g}_{\text{cube}} - \mathbf{g}_{\text{point}}|\right)$, as a function of radial distance from the cube's center

We computed $\mathbf{g}$ at discrete points with 'special' locations, such as exterior to the cube but in a plane containing a face, exterior to the cube but on a line containing an edge, on a face, on an edge, and on a vertex (Appendix 1). No numerical difficulties occurred in any of these locations. As with V, we found that $\mathbf{g}$ varied continuously in ϵ-regions surrounding these points (to the extent that the number of significant digits allowed us to resolve).

Comparison with Point Mass Approximation

A classical result in potential theory is that the gravitational field produced by a finite mass approaches that due to a point mass as the distance (r) from the center of the mass becomes very large in comparison to the size of the mass. The point-mass approximation is given by:

$$\mathbf{g}_{\text{point}} = -\frac{GM}{r^2}\hat{\mathbf{r}}, \quad M = \rho \cdot (2L)(2B)(2D),$$
$$r = \sqrt{x^2 + y^2 + z^2}, \quad \hat{\mathbf{r}} = \frac{\mathbf{r}}{|\mathbf{r}|} = \frac{1}{r}\begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{17}$$

The gravitational acceleration due to the cuboid is computed using finite differences. Fig. 3 shows the logarithmic measure $\log_{10}|\mathbf{g}_{\text{cube}} - \mathbf{g}_{\text{point}}|$ versus $\log_{10}(r)$, along the y axis and along the diagonal direction $(x = y)$. We observe that the gravitational acceleration due to the cube converges (as r increases) with that due to an equivalent point mass.

As noted above, the accuracy of gravitational acceleration computed via numerical differentiation is highly sensitive to the choice of finite difference step size (Appendix 2, Fig. 6). In this test, central difference gradients were computed using high-precision arithmetic in Julia's `BigFloat` environment (Appendix 2, Fig. 7), and step sizes were carefully selected to balance discretization error and numerical precision. As shown in Appendix 2, this approach ensures reliable evaluation of acceleration discrepancies across a wide spatial domain, from near-field to far-field regions.

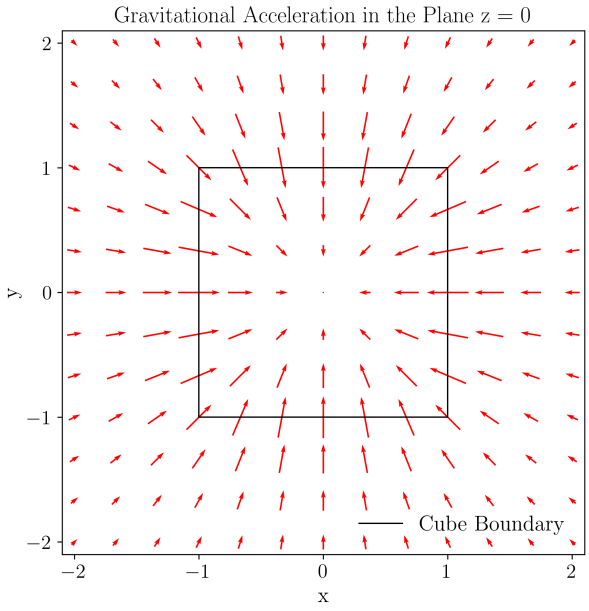


Fig. 4: Gravitational acceleration vectors inside and outside a cube, evaluated on a grid in the $z = 0$ plane. Vectors $\mathbf{g}$ were computed via central differences of the potential $V = 0$. Because $z = 0$, symmetry requires $g_z = 0$ everywhere. The boundary of the cube is shown in black

Another means to test our algorithm and codes is to examine the directionality of gravitational acceleration. Fig. 4 illustrates the gravitational acceleration due to a

DOI: 10.3844/pisp.2025.7.17

cube centered on the origin, with side lengths 2, computed on a grid established in the $z = 0$ plane. The vector field demonstrates the expected directional symmetry imposed by the cube's geometry. The magnitude of the exterior vector field decays with increasing distance from the origin, reflecting Newtonian inverse-square behavior, and the magnitude of the internal vector field also decays towards zero as the evaluation point approaches the center of the cube.

The Laplacian Test

In classical potential theory, the gravitational potential V satisfies Poisson's equation or Laplace's equation (Macmillan, 1930), depending on the location:

$$\nabla^2 V = \begin{cases} -4\pi G\rho & \text{inside the mass,} \\ 0 & \text{outside the mass (in free space)} \end{cases} \quad (18)$$

where G is the gravitational constant, ρ is the mass density, and ∇^2 is the Laplacian operator. To validate the correctness of the computed gravitational potential, we perform a series of Laplacian tests implemented using finite difference approximations. The Laplace operator is discretized via the second-order central difference scheme:

$$\begin{aligned} \nabla^2 V(x,y,z) \approx &+ \frac{V(x+h,y,z) - 2V(x,y,z) + V(x-h,y,z)}{h^2} \quad (19) \\ &+ \frac{V(x,y+h,z) - 2V(x,y,z) + V(x,y-h,z)}{h^2} \\ &+ \frac{V(x,y,z+h) - 2V(x,y,z) + V(x,y,z-h)}{h^2} \end{aligned}$$

where h is the step size. (We must switch to one-sided finite differences if we wish to compute $\nabla^2 V$ at the origin). This discretization enables the evaluation of the Laplacian at various locations, allowing for the distinction between interior and exterior regions of the homogeneous cube of density ρ.

We set $G = 1$ and $\rho = 1$, and evaluated $\nabla^2 V$ on dense 2D grids established on various sections through the cube, for example, on the x-y plane (Fig. 5). We computed the value of $\nabla^2 V$ at every grid point and color coded the result. We did this in Matlab, Python, and later using the `BigFloat` environment in Julia, which allowed us to compute $\nabla^2 V$ to any desired degree of accuracy.

- Interior ($\nabla^2 V = -4\pi G\rho$): colored in blue.
- Exterior ($\nabla^2 V = 0$): colored in red.
- Boundary: shown in white.

For every section we examined (not just the x-y plane) we found that $\nabla^2 V$ obeyed Poisson's equation inside the cube, and Laplace's equation outside the cube. Since $\nabla^2 V$ is discontinuous on the boundary, it is not defined there. The solutions to Laplace's equation are unique, so this test validates our algorithm and codes.

Discussion and Conclusions

We have presented a numerically stable formulation for computing the gravitational potential of a cuboid with uniform density. The derived expressions are valid inside, on and outside the cuboid. We achieved this using a hexagonal graph to properly structure the logarithmic and arctangent terms, and keep track of the sign of the result.

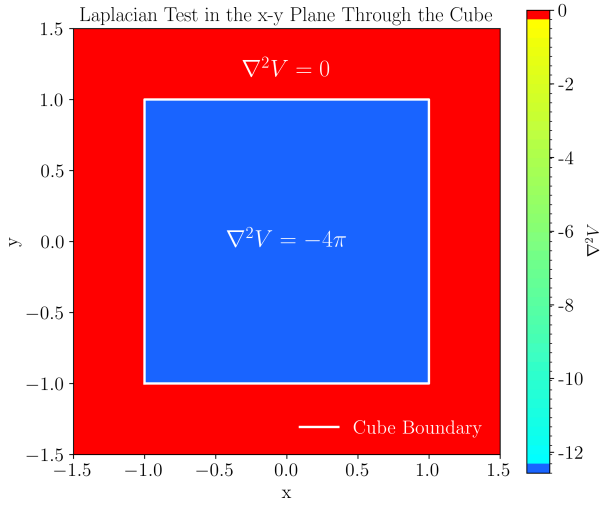


Fig. 5: The value of $\nabla^2 V$ computed on a 300 by 300 grid on the x-y plane through the center of a cube with uniform density $\rho = 1$. The computations assume that $G = 1$. The resulting grid values are colored coded. Interior regions satisfy $\nabla^2 V = -4\pi$, and exterior regions satisfy $\nabla^2 V = 0$, to within the numerical precision obtained by the computations. The cube's boundary is indicated in white. Note that the Laplacian is not defined on the boundary

To verify the correctness of the formulation and its implementations, we introduced the *Laplacian test*, which evaluates whether the computed potential satisfies Poisson's equation within the cuboid and Laplace's equation outside of it. The results of this test, visualized over representative planar slices, confirm the theoretical expectations and thus the fidelity of our algorithm and codes.

We suggest that the Laplacian test can be used to validate similar codes for other simply-shaped, constant-density bodies.

Importantly, our experience with the Laplacian test highlights the sensitivity of numerical differentiation, particularly with respect to the choice of finite difference step size (Appendix 3, Fig. 8). Inappropriate step sizes can lead to misleading results, even when the underlying potential expression is analytically correct. Therefore, careful calibration of discretization parameters is essential for accurate numerical validation. Robust numerical validation requires the use of arbitrary precision arithmetic, such as that provided by the `BigFloat` environment in Julia.

We provide pseudocode and Matlab, Julia and Python implementations to assist our readers with their

gravitational modeling applications. As far as we know, ours is the only algorithm published to date that can compute the gravitational potential anywhere ---inside, on and outside the cuboid, with no spatial restrictions whatsoever. Werner (1994) addressed the polyhedron, which is a more general shape than the cuboid, but his solution is not valid inside the polyhedron, just as the Chappel et al. (2012) solution is not valid inside the cuboid. Our special case handling for arctangents and logarithms of quotients is essentially the same scheme employed by Won and Bevis (1987), but they were addressing constant density mass elements in two dimensions, and our treatment of the cuboid is fully three dimensional. Thus our work complements that of these previous authors.

We point out that sets of abutting but not overlapping cubes (or cuboids) can be used to model planetary bodies, such as asteroids, with complex shapes. This can be achieved through linear superposition. This is most easily done if all individual cubes have edges that are parallel to the axes of the global cartesian system in which their geometry is described. Then, the transformation from global coordinates to local coordinates (in which the solution is formulated) is simply a translation that maps the global coordinates of the center of each cube onto the origin of the local system.

A future extension of this work might involve computing not only the potential V, but also its partial derivatives, using a similarly analytical approach. This would allow us to compute gravitational acceleration, $\mathbf{g}$, without resorting to numerical derivatives.

Another future research direction is to seek a spatially-general solution for other simple constant-density mass elements, such as a tetrahedron. Since the tetrahedron can be viewed as a building block for any polyhedron, if this effort was successful, then it should also be possible to compute the gravitational potential inside, on or outside any constant-density polyhedron.

Acknowledgement

We thank Chappell et al. (2012), whose work led us to focus on cuboids rather than the elemental shapes most widely invoked in physical geodesy.

Funding Information

This research did not receive external funding.

Author's Contributions

Thunendran Periyandy: Conceptualization, Methodology, Software, Validation, Writing - Original Draft.

Michael Bevis: Conceptualization, Validation - Laplacian Test, Writing - Review & Editing.

Ethics

There are no ethical conflicts associated with this manuscript.

Data Availability

The supplementary code and materials referenced in this article are available in a public GitHub repository at https://github.com/Thunendran/GravitationalPotentialCuboid.

References

Chappell, J. M., Chappell, M. J., Iqbal A., & Abbott, D. (2012). The Gravity Field of a Cube. *Physics International*, *3*(2), 50-57. https://doi.org/10.3844/pisp.2012.50.57

Grant, F. S., & West, G. F. (1965). *Interpretation theory in applied geophysics*. McGraw-Hill Book Co., New York

Heiskanen, W., & Moritz, H. (1967). *Physical Geodesy*. W. H. Freeman and Company, San Francisco.

Hubbert, M. K. (1948). A Line-integral Method of Computing the Gravimetric Effects of Two-dimensional Masses. *Geophysics*, *13*(2), 215-225. https://doi.org/10.1190/1.1437395

Macmillan, W. D. (1930). *Theory of the Potential*. McGraw-Hill Book Co., New York.

Talwani, M., Worzel, J. L., & Landisman, M. (1959). Rapid gravity computations for two-dimensional bodies with application to the Mendocino submarine fracture zone. *Journal of Geophysical Research*, *64*(1), 49-59. https://doi.org/10.1029/jz064i001p00049

Telford, W. M., Geldart, L., Sheriff, R., & Keys, D. (1976). *Applied Geophysics*. Cambridge University Press, London.

Tsoulis, D., & Gavriilidou, G. (2021). A computational review of the line integral analytical formulation of the polyhedral gravity signal. *Geophysical Prospecting*, *69*(8-9), 1745-1760. https://doi.org/10.1111/1365-2478.13134

Werner, R. A. (1994). The gravitational potential of a homogeneous polyhedron or don't cut corners. *Celestial Mechanics & Dynamical Astronomy*, *59*(3), 253-278. https://doi.org/10.1007/bf00692875

Won, I. J., & Bevis, M. (1987). Computing the gravitational and magnetic anomalies due to a polygon: Algorithms and Fortran subroutines. *Geophysics*, *52*(2), 232-238. https://doi.org/10.1190/1.1442298

DOI: 10.3844/pisp.2025.7.17

Appendix 1: Example Computations for Gravitational Potential and Acceleration

Table 1 was computed using our Julia code, which utilizes Julia's `BigFloat` environment. All digits in the table are significant. Similar results, but not exact ones, can be computed using Matlab, since its carries only 15 significant digits in its floating point operations, and is susceptible to numerical precision problems.

Table 1: Gravitational potential and gravity vector at selected test points due to a cube of dimensions $2 \times 2 \times 2$ (that is, L = B = D = 1), with gravitational constant $G = 1.0$ and density $\rho = 1.0$. The locations of the test points are described in the right hand column. By `on extended edge' we mean that the test point is on a line that incorporates an edge, but not on the edge itself. By `on extended face' we mean that the point lies in a plane that incorporates a face, but not in the face itself

ID	Test Point (x,y,z)	Potential	Gravitational acceleration	Location
1	$\left(\frac{1}{2}, \frac{1}{2}, \frac{1}{2}\right)$	$8.043586363964623e+00$	$g_x = -1.845778162369231e+00$ $g_y = -1.845778162369231e+00$ $g_z = -1.845778162369231e+00$	Interior
2	$\left(\frac{1}{2}, \frac{1}{2}, 1\right)$	$6.504625741605996e+00$	$g_x = -1.296479052953900e+00$ $g_y = -1.296479052953900e+00$ $g_z = -4.515510073496715e+00$	On face
3	$\left(\frac{1}{2}, \frac{1}{2}, 1+\varepsilon\right)$	$6.504625741151316e+00$	$g_x = -1.296479052819571e+00$ $g_y = -1.296479052819571e+00$ $g_z = -4.515510073568636e+00$	Exterior (by ε)
4	$\left(\frac{1}{2}, \frac{1}{2}, 1-\varepsilon\right)$	$6.504625742060676e+00$	$g_x = -1.296479053088229e+00$ $g_y = -1.296479053088229e+00$ $g_z = -4.515510073424793e+00$	Interior (by ε)
5	$(1, 1, 1)$	$4.760154727959107e+00$	$g_x = -1.930929097605087e+00$ $g_y = -1.930929097605087e+00$ $g_z = -1.930929097605087e+00$	On vertex
6	$(1, 1+\varepsilon, 1)$	$4.760154727765229e+00$	$g_x = -1.930929097016367e+00$ $g_y = -1.930929097657447e+00$ $g_z = -1.930929097016367e+00$	On extended edge
7	$(1+\varepsilon, 1, 1)$	$4.760154727765229e+00$	$g_x = -1.930929097657447e+00$ $g_y = -1.930929097016367e+00$ $g_z = -1.930929097016367e+00$	On extended edge
8	$(0, 2, 1)$	$3.569191738087612e+00$	$g_x = -8.380208987159455e-248$ $g_y = -1.426422695721153e+00$ $g_z = -6.791372154057112e-01$	On extended face
9	$(4, 4, 4)$	$1.154780286871141e+00$	$g_x = -9.625844144561081e-02$ $g_y = -9.625844144561081e-02$ $g_z = -9.625844144561081e-02$	Far exterior point
10	$(4+\varepsilon, 4+\varepsilon, 4+\varepsilon)$	$1.154780286842264e+00$	$g_x = -9.625844144079449e-02$ $g_y = -9.625844144079449e-02$ $g_z = -9.625844144079449e-02$	Far exterior + ε

Note: The perturbation used in evaluations is $\varepsilon = 10^{-10}$

DOI: 10.3844/pisp.2025.7.17

Appendix 2: First-Order Finite Difference Step Size Selection

This analysis examines the sensitivity of gravitational acceleration computations for a homogeneous cube with dimensions $2 \times 2 \times 2$ (that is, L = B = D = 1) and density $\rho = 1$, to the choice of the step size of the finite difference. Again, we choose the normalization $G = 1$. We emphasize the impact of step size on numerical accuracy. The computed gravitational acceleration is compared against the analytical solution of an equivalent point mass located at the cubeâ€™s center. Analytically, we expect the magnitude of the vector difference $\mathbf{g}_{\text{cube}} - \mathbf{g}_{\text{point}}$ to decrease rapidly, and continuously, as r increases. In practice, finite-precision computations can cause departures from this expectation.

The MATLAB implementation (Fig. 6) uses machine-precision arithmetic (approximately 15 significant digits), which makes the results highly sensitive to the selected step size. Inappropriate step sizes introduce significant numerical noise, as reflected in the fluctuating difference curves.

The analysis is conducted along the diagonal direction ($x = y$, $z = 0$), covering radial distances $r \in [\sqrt{2}, 500]$. Gravitational acceleration is computed using central finite differences of the gravitational potential (Eq. 16), and the absolute difference relative to the point-mass model is evaluated for various step sizes $h \in \{10^{-4}, 10^{-8}\}$. Results are shown in log-log scale to illustrate the sensitivity to step size across different distances.

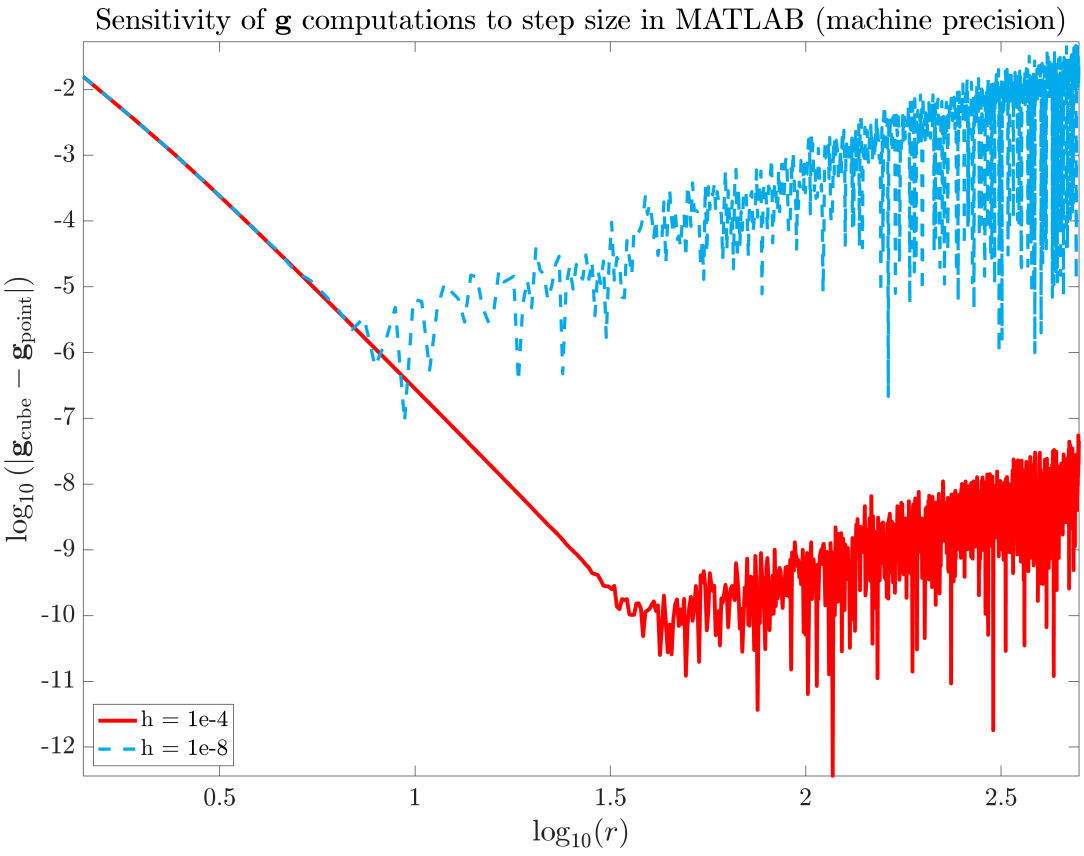


Fig. 6: Sensitivity of gravitational acceleration computations for a homogeneous cube to finite difference step size along the diagonal direction ($x = y, z = 0$), across radial distances $r \in [\sqrt{2}, 500]$. MATLAB implementation using machine precision (15 significant digits) shows significant variability in error magnitude depending on the h, highlighting the strong sensitivity to numerical discretization.

There is a way to mitigate this numerical error: make the step size a judiciously chosen function of r. However, there is an easier alternative, which we now describe.

Using Julia's `BigFloat` environment with 250-digit precision, gravitational acceleration is also computed via central finite differences of the potential. The absolute error relative to the point-mass model is evaluated for step sizes $h \in$

DOI: 10.3844/pisp.2025.7.17

$\{10^{-10}, 10^{-20}, \ldots, 10^{-60}\}$. Results are likewise presented in log-log scale to reveal the sensitivity to step size over distance (Fig. 7).

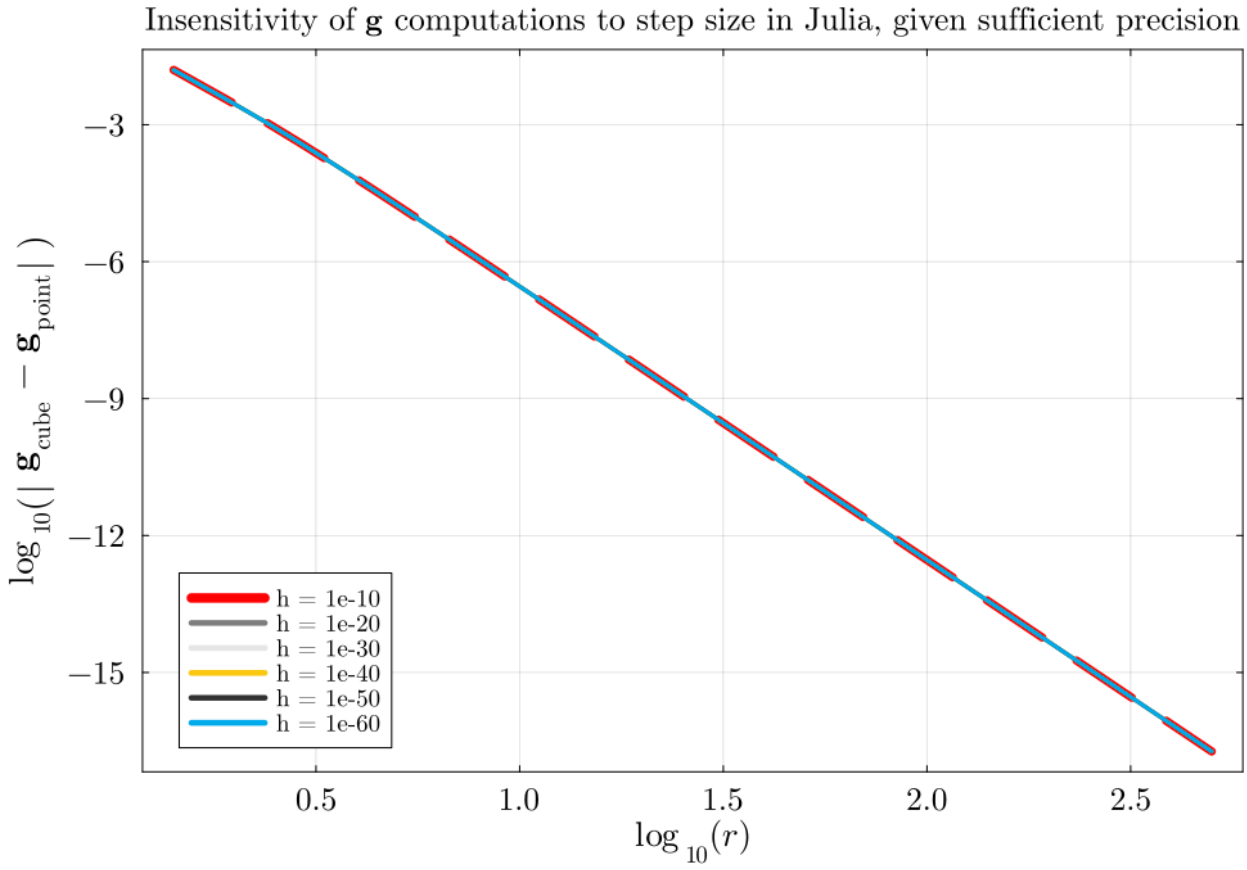


Fig. 7: Logarithmic plot of the gravitational acceleration difference between a homogeneous cuboid and an equivalent point mass, evaluated along the diagonal direction ($x = y, z = 0$) for distances ranging from $r = \sqrt{2}$ to $r = 500$. The gravitational acceleration of the cube is computed numerically using central differences with multiple step sizes h, in a high-precision (`BigFloat`) environment. The quantity plotted is $\log_{10}(|\mathbf{g}_{\text{cuboid}} - \mathbf{g}_{\text{point}}|)$, with curves labeled by their respective step sizes. Smaller step sizes yield improved accuracy up to a limit, beyond which round-off errors dominate. The optimal step size minimizes the error, depending on the local curvature of the potential field and numerical precision.

Clearly, if we perform our finite difference computations using sufficient numerical precision, we can eliminate numerical round-off problems even for relatively large values of r.

DOI: 10.3844/pisp.2025.7.17

Appendix 3: Laplacian Test RMS Error vs Step Size at BigFloat Precision

The results shown in Fig. 8 computed using our Julia code, executed within Julia's `BigFloat` environment at 250-digit and 500-digit precision levels. In this Laplacian test, the true values for the interior and exterior regions are known analytically as $-4\pi G\rho$ and 0, respectively. The figure presents the root mean square (RMS) of the error across a range of step sizes h, plotted as $\log_{10}(h)$ versus $\log_{10}(\text{RMS error})$. The horizontal axis indicates the number of significant digits in the step size, while the vertical axis reflects the relative error magnitude. The optimal step size minimizing the RMS error is identified by the minima of the plotted curves.

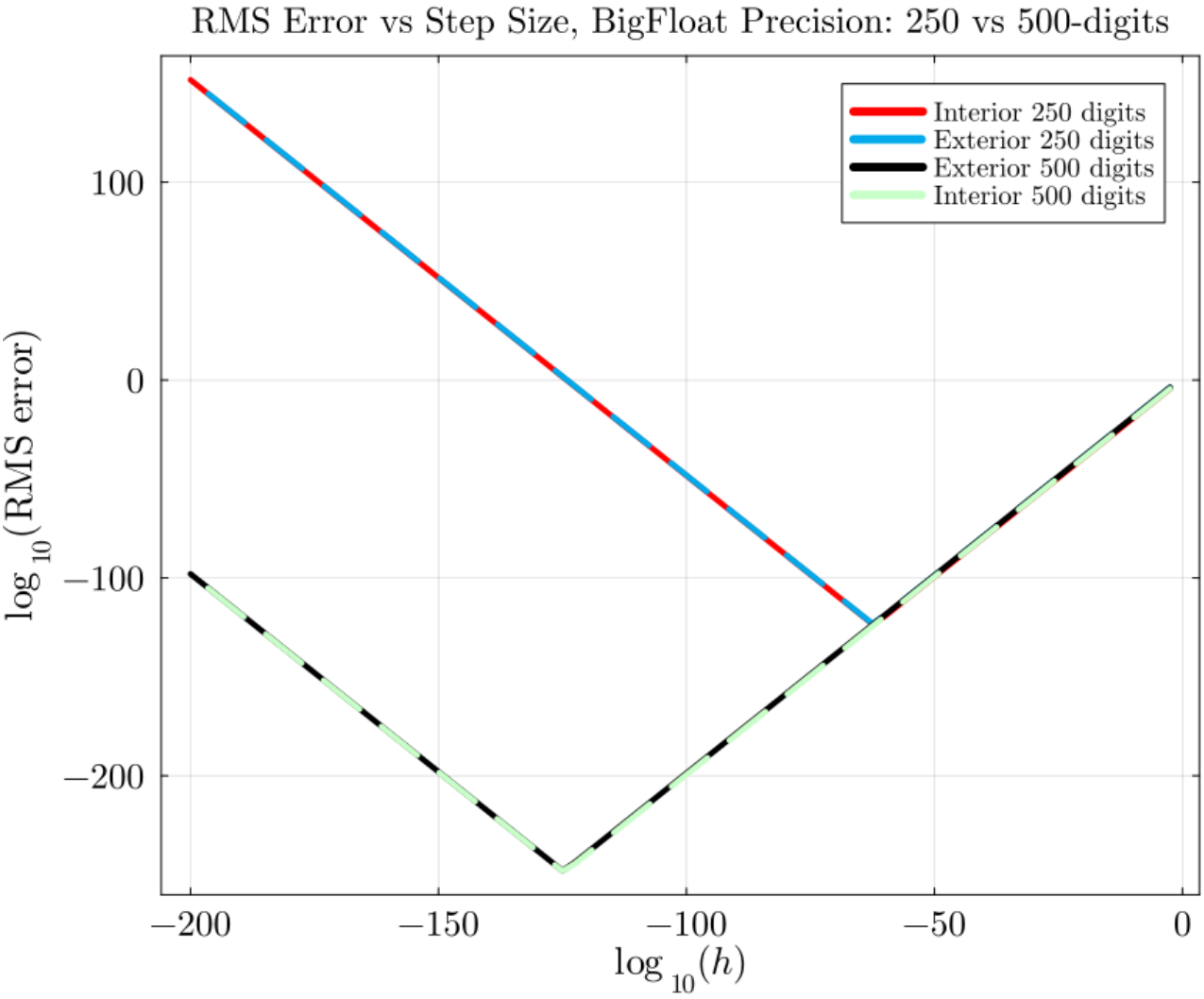


Fig. 8: RMS error comparison for Laplacian test using different step sizes h, evaluated for interior and exterior regions of a cuboid. Results are shown for `BigFloat` precision set to 250 digits and 500 digits. The optimal step size minimizing RMS error can be identified by the minima in each curve