Supplementary Information for

An alternative approach to accurate calculation of gravitational and magnetic fields of a spherical prism

Pan Zhang[1,4], Yinglin Li[4], Jinsong Du[1,2,3*], Liang Yin[4], Jiangsong Gui[1], Changqing Yuan[1], Shuang Liu[1,2], Chao Chen[1*], Keke Zhang[4]

1. Hubei Subsurface Multi-scale Imaging Key Laboratory, School of Geophysics and Geomatics, China University of Geosciences, Wuhan 430074, China

2. Key Laboratory of Geological Survey and Evaluation of Ministry of Education, China University of Geosciences, Wuhan 430074, China

3. State Key Laboratory of Geological Processes and Mineral Resources, China University of Geosciences, Wuhan 430074, China

4. Macau Institute of Space Technology and Application, Macau University of Science and Technology, Macau 999078, China

* Corresponding authors: J. Du (jinsongdu@cug.edu.cn), C. Chen (chenchao@cug.edu.cn)

Contents of this file

1. Captions for auxiliary and computational codes.

Additional Supporting Information (Files uploaded separately)

1. Captions for MATLAB codes demonstrating the numerical stability of integral kernels;

2. Figures illustrating the morphology of integral kernels and captions for corresponding MATLAB codes;

3. Captions for C++ codes used to calculate the gravitational and magnetic fields of a spherical prism.

Introduction

This supporting information provides captions for two MATLAB code sets: one demonstrating the numerical stability of integral kernels, and another illustrating their morphological characteristics. In addition, it includes captions for a C++ code set used for the forward computation of the gravitational and magnetic fields of a spherical prism.

1. Numerical stability of the integral kernels

In this section, we present a synthetic example designed to illustrate the numerical stability of the integral kernels listed in Appendix A of the main text. MATLAB codes demonstrating this numerical behavior are provided in the sub-folder: "SupplementaryMaterial_1_KernelCancellation", which contains 20 MATLAB scripts covering the gravitational potential, acceleration vector, gradient tensor, and curvature components, respectively.

To highlight potential numerical issues, we simulate computationally extreme conditions, specifically, when the computational point is located very close to the mass surface, and the angular separation ψ approaches zero. The codes are executed in both a double-precision environment (which is computationally efficient but may suffer from numerical instability) and a symbolic computation environment (which is more accurate but computationally slower).

By adjusting the parameters in the provided scripts, users can directly observe the numerical instability behavior of the integral kernels. Consistent with the findings presented in the main text, our results confirm that the integral kernels for the V_{xx}, V_{xy} and V_{yy} components exhibit relatively minor numerical cancellation issues when the angle ψ is less than approximately 0.0001°. In contrast, the V_{xxx}, V_{xxy}, V_{yyx} and V_{yyy} components display significantly greater numerical instability when ψ is less than approximately 0.01°.

2. Morphology of the integral kernels

In this section, we examine how the morphology of the integral kernels varies with the position of the computational point. To this end, we consider a single spherical prism defined by the following coordinate parameters: upper and lower boundaries $r_2 = R_E$, $r_1 = R_E - 30$ km, latitudinal boundaries $\varphi_2 = 50°$, $\varphi_1 = 30°$, and longitudinal boundaries $\lambda_2 = 130°$ and $\lambda_1 = 110°$.

Three test cases are designed to illustrate how the morphology of the integral kernels changes with respect to the computational point location:

(1) **Test No. S1**: The computational point is located at $r = R_E + 300$ km, $\varphi = 40°$ and $\lambda = 120°$, i.e., 300 km directly above the spherical prism. The resulting kernel morphologies are shown in **Figure S1**;

(2) **Test No. S2**: The computational point is located at $r = R_E + 50$ km, $\varphi = 40°$ and $\lambda = 120°$, i.e., 50 km directly above the spherical prism. The resulting kernel morphologies are shown in **Figure S2**;

(3) **Test No. S3**: The computational point is located at $r = R_E + 50$ km, $\varphi = 90°$ and $\lambda = 0°$, i.e., far from the spherical prism. The resulting kernel morphologies are shown in **Figure S3**.

Based on the morphological variations observed in these tests, the following conclusions can be drawn: (1) A sharp peak appears in the integral kernels when the line OP (connecting the origin to the computational point) intersects the mass surface (i.e. $\varphi_1 \le \varphi \le \varphi_2$ and $\lambda_1 \le \lambda \le \lambda_2$); (2) The shape of the peak resembles that of the gravitational field generated by a point mass; and (3) The closer the computational point P is to the mass surface, the sharper the observed peak, which is consistent with the theoretical conclusions presented in the main text.

MATLAB codes used to visualize the morphology of the integral kernels are provided in the sub-folder: "SupplementaryMaterial_2_KernelMorphology". This folder contains 20 MATLAB functions and 4 driver scripts, covering the gravitational potential, acceleration vector, gradient tensor, and curvature components, respectively.

3. C++ codes for the computation

In this section, we provide caption for C++ codes used to calculate the gravitational and magnetic fields of a spherical prism in the main text.

The main calculation functions are all included in the sub-folder "include/internal", where "_integral2.h" contains the main integration program, "_tesseroid_integralkernel.h" contains the integral kernels calculation program, "_tesseroid_estimate.h" contains the calculation program for the gravitational and magnetic fields of spherical prisms, and the outer "tesseroid.h" header file can be referenced to include all the header files. In addition, we provide six script programs for calculating gravitational and magnetic fields, which can perform serial or parallel calculations based on "OpenMP" or "MPI", demonstrating how to use the calculation interface.

Software and Compilation Instructions

To compile and run the provided C++ codes for gravitational and magnetic field calculations, we recommend using the **Intel oneAPI Base Toolkit** and **Intel oneAPI HPC Toolkit**, which include optimized compilers and the MPI library. These toolkits can be downloaded from the following links:

- Intel oneAPI Base Toolkit:

  https://www.intel.cn/content/www/cn/zh/developer/tools/oneapi/base-toolkit-download.html;

- Intel oneAPI HPC Toolkit:

  https://www.intel.cn/content/www/cn/zh/developer/tools/oneapi/hpc-toolkit-download.html.

In addition, the Eigen linear algebra library must be installed. It can be obtained from:

- Eigen Library:

  https://gitlab.com/libeigen/eigen.

Platform-Specific Recommendations

- **Windows/macOS users**: It is recommended to install Visual Studio 2022 for C++ development and integration with Intel compilers.
- **Linux users**: Compilation should be accompanied by an appropriate Makefile.

However, alternative compilers such as **gcc** or **CLang** are also supported and can be used to perform the compilation.

Code Location

The C++ source codes are provided in the sub-folder:

- "SupplementaryMaterial_3_ForwardCodes"

Additionally, we provide four MATLAB scripts designed to assist users in:

1. Setting up the forward model parameter files, and

2. Reading the output result files in binary format.

These auxiliary scripts serve as a convenient interface between the numerical model setup and result analysis.

Of course, if you encounter any issues during usage—whether related to running the program, understanding the methodology, or if you have comments or suggestions—we warmly welcome you to contact us for further discussion and collaboration. Please feel free to reach out through the contact information provided below.

Email: panzhang@cug.edu.cn, 1393791486@qq.com

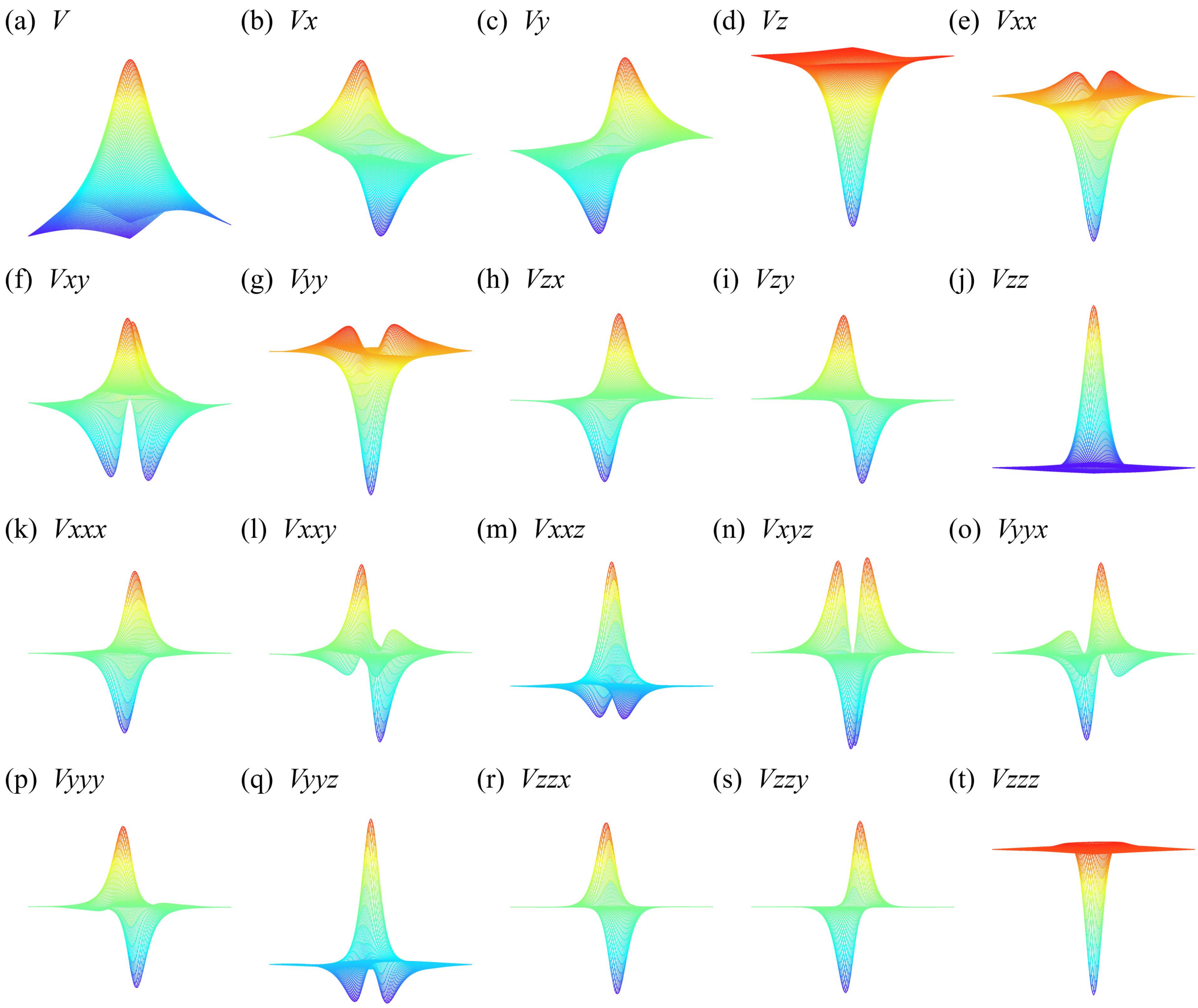

Fig. S1 Test No. S1: morphology of the integral kernel

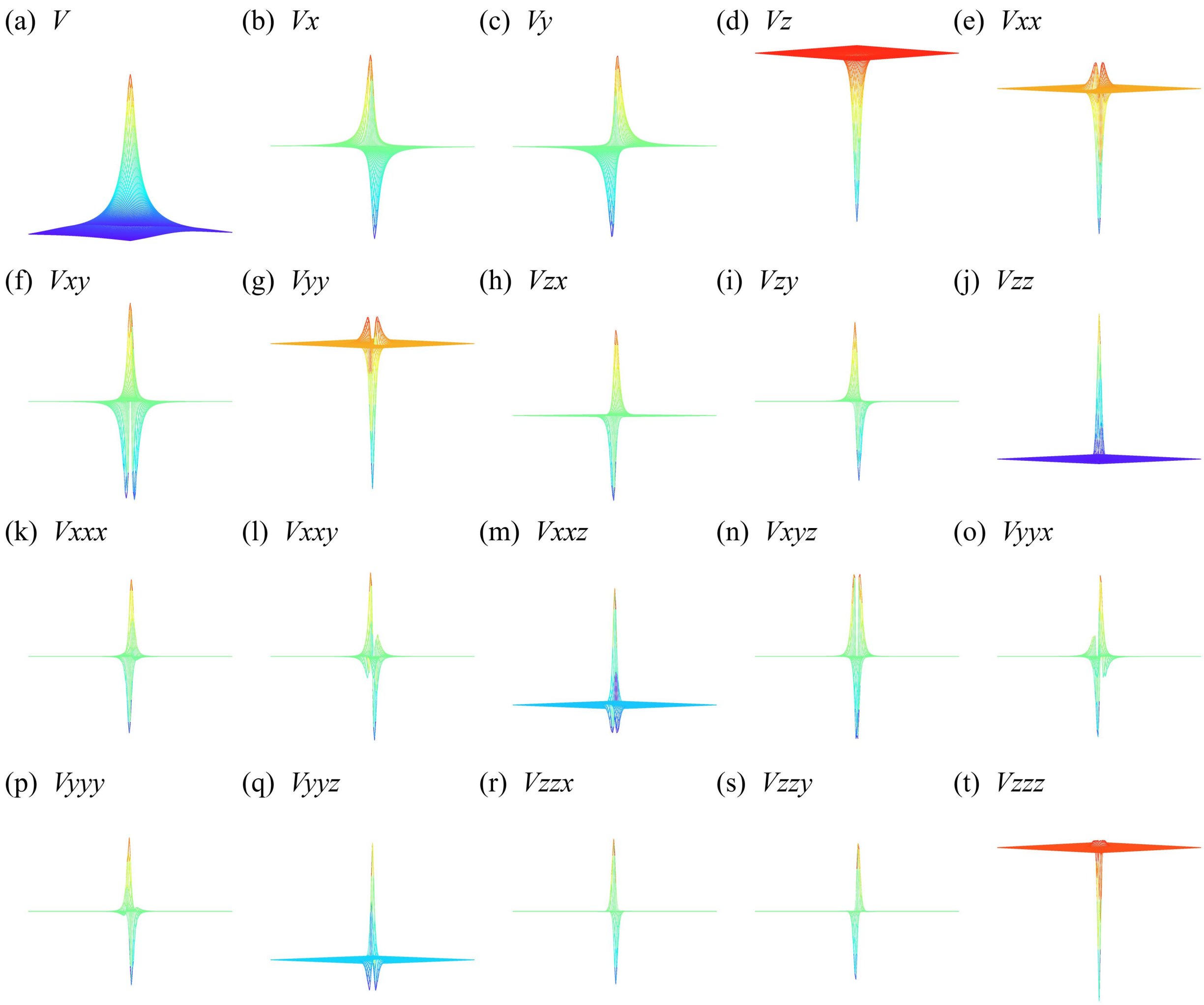

Fig. S2 Test No. S2: morphology of the integral kernel

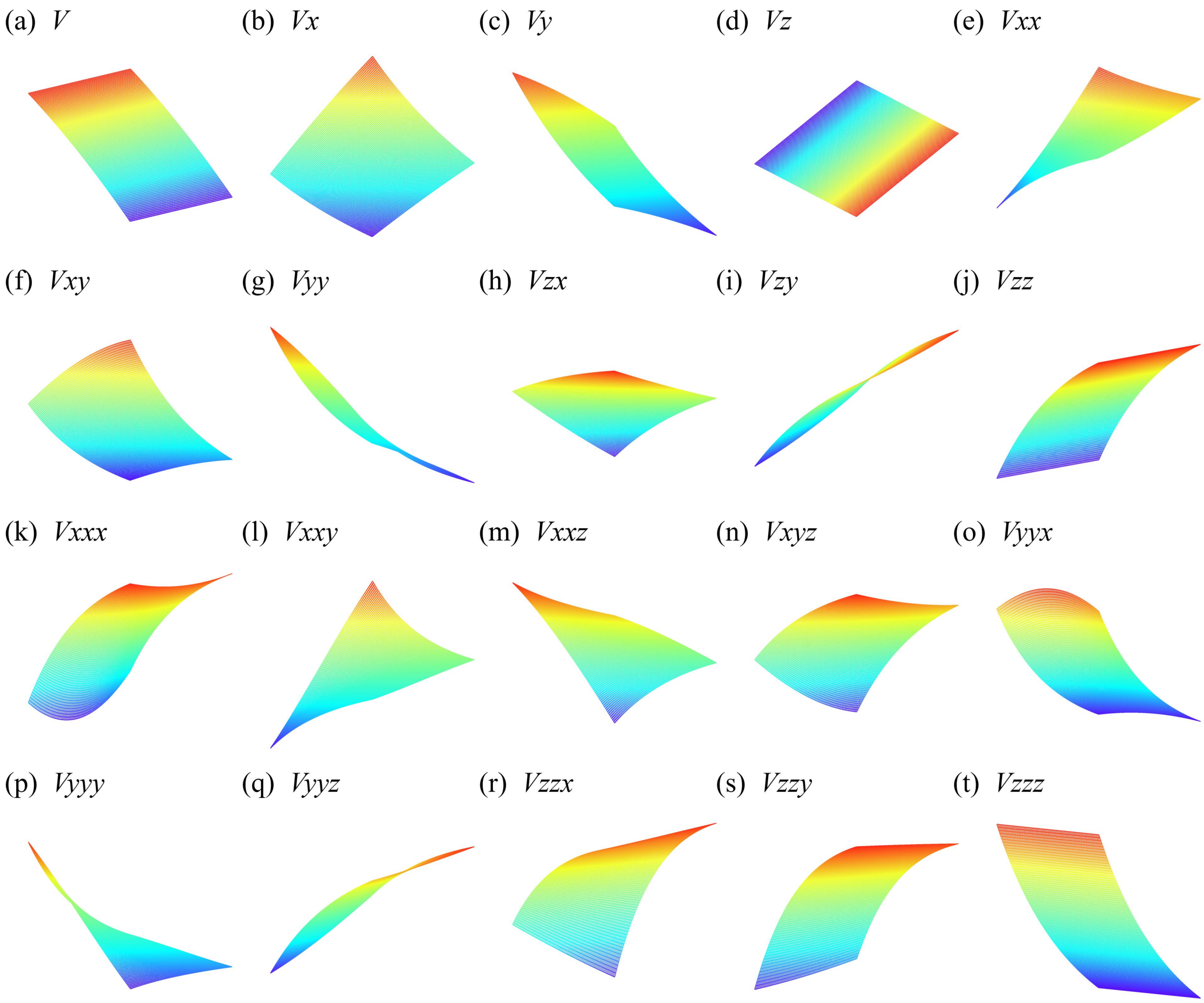

Fig. S3 Test No. S3: morphology of the integral kernel