

# 基于Python的直流电阻率法三维有限元数值模拟

石 靖<sup>1</sup>, 庞满莹<sup>2</sup>, 梁 卓<sup>3</sup>, 崔杨洋<sup>3</sup>, 陈汉波<sup>1</sup>, 喻佳敏<sup>1</sup>, 李耀昕<sup>1</sup>

<sup>1</sup>桂林理工大学地球科学学院, 广西 桂林

<sup>2</sup>江西省地质局物化探大队, 江西 南昌

<sup>3</sup>中国冶金地质总局第三地质勘查院, 山西 太原

收稿日期: 2024年11月13日; 录用日期: 2024年12月24日; 发布日期: 2024年12月31日

## 摘要

利用Python平台强大的科学计算库, 拟实现电导率分块均匀的直流电阻率法三维有限元数值模拟。基于Galerkin法给出稳定电流场三维双电源总场电位边值问题的控制方程, 采用六面体单元剖分和三线性插值进行有限单元分析, 将控制方程化为线性方程组, 分别采用压缩稀疏列矩阵(CSC)与压缩稀疏行矩阵(CSR)存储总体系数矩阵, 对比科学计算库SciPy及PyPardiso的求解速度与准确度, 试算层状介质模型及单一、组合异常体模型, 结果表明: H型层状介质的视电阻率测深解析解与有限元数值解最大相对误差低于3.00%, 平均相对误差1.54%, 本文算法有效; 在Python平台求解地球物理正演中大型稀疏线性方程组时, PyPardiso与SciPy精度一致, 当方程未知个数达37万时, PyPardiso比SciPy求解速度快约66倍; 以CSC与CSR形式存储的大型稀疏线性系统, 在同一求解器中求解效率相当。

## 关键词

直流电阻率法, 有限单元法, 数值模拟, 压缩存储, Python

# Three-Dimensional Finite Element Numerical Simulation of Direct Current Resistivity Method Based on Python

Jing Shi<sup>1</sup>, Manying Pang<sup>2</sup>, Zhuo Liang<sup>3</sup>, Yangyang Cui<sup>3</sup>, Hanbo Chen<sup>1</sup>, Jiamin Yu<sup>1</sup>, Yaxin Li<sup>1</sup>

<sup>1</sup>College of Earth Sciences, Guilin University of Technology, Guilin Guangxi

<sup>2</sup>Geophysical & Geochemical Exploration Brigade of Jiangxi, Nanchang Jiangxi

<sup>3</sup>The Third Geological Exploration Institute of China Metallurgical Geology Bureau, Taiyuan Shanxi

Received: Nov. 13<sup>th</sup>, 2024; accepted: Dec. 24<sup>th</sup>, 2024; published: Dec. 31<sup>st</sup>, 2024

文章引用: 石靖, 庞满莹, 梁卓, 崔杨洋, 陈汉波, 喻佳敏, 李耀昕. 基于 Python 的直流电阻率法三维有限元数值模拟[J]. 地球科学前沿, 2024, 14(12): 1604-1615. DOI: [10.12677/ag.2024.1412150](https://doi.org/10.12677/ag.2024.1412150)

## Abstract

Utilizing the powerful scientific computing libraries on the Python platform, this study aims to implement a three-dimensional finite element numerical simulation for direct current (DC) resistivity with uniform conductivity blocks. By applying the Galerkin method, we derived control equations for the boundary value problem of a steady-state electric field with dual-source potential. Hexahedral meshing and trilinear interpolation were used for finite element analysis, converting the control equations into linear algebraic systems. The global coefficient matrix was stored in Compressed Sparse Column (CSC) and Compressed Sparse Row (CSR) formats to evaluate the speed and accuracy of solvers in SciPy and PyPardiso. Test cases included layered media models and single and combined anomaly models. Results show that, for H-type layered media, the maximum relative error between the apparent resistivity depth profiling analytical solution and the finite element numerical solution was below 3.00%, with an average relative error of 1.54%, validating the algorithm. When solving large-scale sparse linear systems in geophysical forward modeling on Python, PyPardiso and SciPy demonstrated consistent accuracy; for systems with up to 370,000 unknowns, PyPardiso achieved approximately 66 times faster performance than SciPy. For large sparse systems stored in either CSC or CSR formats, both achieved comparable efficiency with the same solver.

## Keywords

**DC Resistivity Method, FEM, Numerical Simulation, Compressed Storage, Python**

Copyright © 2024 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

## 1. 引言

目前，直流电阻率法三维有限元数值模拟已经取得了显著的进展。阮百尧等[1][2]给出了电导率分块均匀及电导率分块连续变化的三维地电断面电阻率测深的有限元数值模拟方法。在处理边值问题方面，黄俊革[3]简化异常电位算法的边界条件以节约计算时间，汤井田[4]提出有限元 - 无限元耦合数值模拟方法以解决传统有限元截断边界所引起的问题。在优化网格剖分方面，任政勇等[5]实现了对复杂模型的完全非结构化四面体全自动剖分，给出了三维直流电阻率模拟中四面体网格的质量评价标准和最优指标，杨志龙[6]基于梯度恢复的局部网格自适应加密技术实现了网格合理剖分，显著节约计算内存并减少正演时间，赵宁等[7]应用 h 型自适应加密算法结合高阶形函数提高有限元解的精度并节约计算资源。在提高求解效率方面，刘斌等[8]采用预条件共轭梯度法求解大型稀疏线性方程组以提高正演速度，张钱江等[9]提出了一种近似边界条件方法将与场源位置相关的边界系数矩阵从刚度矩阵中分离出来以实现线性方程组的快速回代求解，欧东新等[10]利用改进的对称超松弛预条件共轭梯度法进行线性方程组求解，朱姣[11]利用直接求解器 MUMPS 对大型稀疏矩阵进行快速分解及线性方程组求解。

Python 作为一种强大的科学计算平台，其具备丰富的科学计算和数据处理库，如 NumPy, SciPy, Matplotlib, Pandas 和 PyPardiso 等，能极大简化数据操作、数值计算与可视化工作。NumPy 提供了高效的数组操作，SciPy 则扩展了这些能力至科学计算领域，而 PyPardiso 专门优化稀疏矩阵的求解过程，显著提高处理大型稀疏线性代数方程组的计算效率。Matplotlib 则是一个在数据分析和科学计算中功能强大的可视化工具。值得一提的是，在地球物理学领域，Python 平台同样拥有众多出色的数据处理库。如用

于地球物理正反演的开源库 SimPEG [12]，其支持电阻率法、激发极化法和电磁法等多种物探方法；专注于地震学数据处理和分析的 Pyrocko 与 ObsPy [13]；以及支持地球物理建模与反演的 pyGIMLi [14]等。

基于 Python 丰富成熟的科学生态系统，本文将尝试利用 Python 进行直流电阻率法三维有限元数值模拟。为保证数值模拟的精度与效率，将测试对比 SciPy 与 PyPardiso 求解器的性能及准确度，同时分析在两种求解器中采取 CSC 与 CSR 两种不同的方式存储大型稀疏系数矩阵对其性能与精准度的影响，以挑选出更适合求解地球物理有限元正演中大型稀疏线性系统的科学计算工具。将水平层状介质模型的滤波算法计算结果与本文算法数值模拟结果对比，平均相对误差 1.54%，再对单一低阻异常体与组合异常体模型进行数值模拟，验证了本文方法的有效性。

## 2. 边值问题

三维地电断面双电源总电位的边值问题可归纳为[2]：

$$\nabla \cdot (\sigma \nabla U) = -2I\delta(A) + 2I\delta(B), \quad \in \Omega \quad (1)$$

$$\frac{\partial U}{\partial n} = 0, \quad \in \Gamma_s \quad (2)$$

$$\frac{\partial U}{\partial n} + \frac{1}{r_B - r_A} \left[ \frac{r_B}{r_A} \cos(r_A, n) - \frac{r_A}{r_B} \cos(r_B, n) \right] U = 0, \quad \in \Gamma_\infty \quad (3)$$

式中， $\Omega$  是地面边界  $\Gamma_s$  及无穷边界  $\Gamma_\infty$  所围的区域， $U$  是总场电位， $\sigma$  是介质的电导率， $\delta(A)$ ， $\delta(B)$  分别表示以电源点  $A, B$  为中心的  $\delta$  函数。 $r_A, r_B$  分别是测点至电源点  $A, B$  的距离， $\cos(r_A, n), \cos(r_B, n)$  分别是矢径  $\mathbf{r}_A, \mathbf{r}_B$  与外法向  $\mathbf{n}$  之间夹角的余弦。

## 3. Galerkin 法建立控制方程

式(1)的余项可写为

$$R = \nabla \cdot (\sigma \nabla U) + 2I\delta(A) - 2I\delta(B), \quad (4)$$

这里， $R$  是标量余项。由此，Galerkin 的加权积分可表示成

$$\iiint_{\Omega} W_i R dV = 0,$$

将式(4)代入上式，同时借助 Green 第一定理

$$\iiint_V \left( \phi \nabla^2 \psi + \nabla \phi \cdot \nabla \psi \right) dV = \oint_S \phi \frac{\partial \psi}{\partial n} dS,$$

可以得到

$$\begin{aligned} & \iiint_{\Omega} W_i R dV \\ &= \iiint_{\Omega} W_i [\nabla \cdot (\sigma \nabla U) + 2I\delta(A) - 2I\delta(B)] dV \\ &= - \iiint_{\Omega} \sigma \nabla U \cdot \nabla W_i dV + \oint_{\Gamma} W_i \sigma \frac{\partial U}{\partial n} dS + 2 \iiint_{\Omega} W_i I \delta(A) dV - 2 \iiint_{\Omega} W_i I \delta(B) dV \\ &= 0, \end{aligned}$$

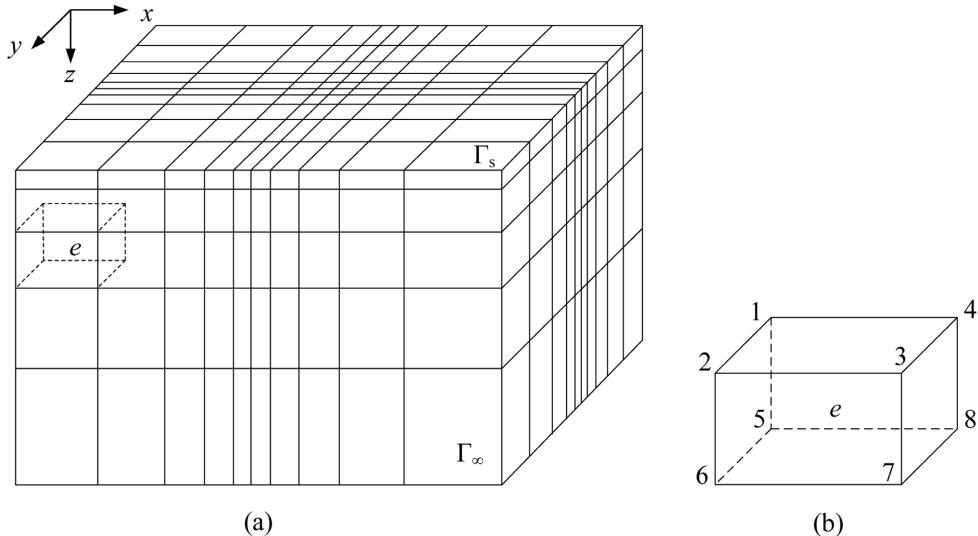
再将式(2)和(3)代入上式，得到控制方程

$$\begin{aligned} & \iiint_{\Omega} \sigma \nabla U \cdot \nabla W_i dV + \oint_{\Gamma_\infty} W_i \sigma \frac{1}{r_B - r_A} \left[ \frac{r_B}{r_A} \cos(r_A, n) - \frac{r_A}{r_B} \cos(r_B, n) \right] U dS \\ &= 2 \iiint_{\Omega} W_i I \delta(A) dV - 2 \iiint_{\Omega} W_i I \delta(B) dV. \end{aligned} \quad (5)$$

## 4. 有限单元法

### 4.1. 求解区域离散化

将求解区剖分为一系列互不重叠的六面体单元(图 1), 即可将式(5)中对区域和无穷边界积分分解为对各小单元的体积分与面积分, 每个单元的顶点称为节点。



**Figure 1.** Schematic diagram of hexahedral mesh generation (a) and hexahedral element (b)

**图 1.** 六面体网格剖分(a)与单元(b)示意图

在完成网格剖分的同时, 于求解区形成若干节点。待定函数(如函数 \$U\$)及其导数在这些节点上的数值, 称为节点函数值。这样, 便用求解区内有限个离散的待定节点函数值来近似表征待定函数在连续空间中的分布。这些待定的节点函数值 \$U\_1\$、\$U\_2\$、……、\$U\_n\$, 组成一组独立变量。所谓待定函数的数值解, 就是确定这些节点函数值[15]。

### 4.2. 线性插值

通常地, 利用各节点函数值在各单元内作线性内插来逐个单元计算参数 \$U\$。采用六面体单元和三线性插值, 图 2(a)是母单元(边长为 2 的正方体), 图 2(b)是子单元, 两者的坐标关系为

$$x = x_0 + \frac{a}{2}\xi, \quad y = y_0 + \frac{b}{2}\eta, \quad z = z_0 + \frac{c}{2}\zeta,$$

其中, \$x\_0\$、\$y\_0\$、\$z\_0\$ 是子单元的中点, \$a\$、\$b\$、\$c\$ 是子单元的三个边长, 两单元间的微分关系为

$$dx = \frac{a}{2}d\xi, \quad dy = \frac{b}{2}d\eta, \quad dz = \frac{c}{2}d\zeta,$$

$$dxdydz = \frac{abc}{8}d\xi d\eta d\zeta.$$

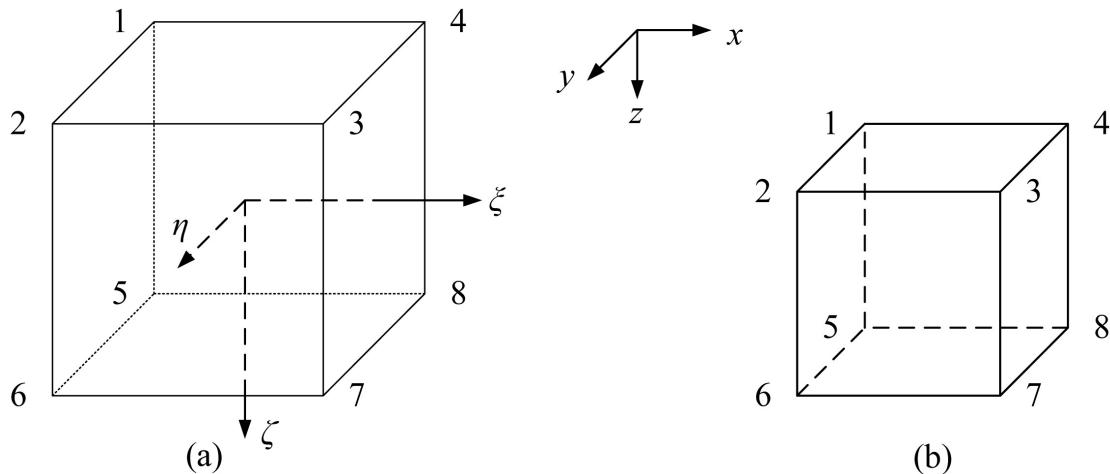
三线性插值的形函数为

$$N_i = \frac{1}{8}(1+\xi_i\xi)(1+\eta_i\eta)(1+\zeta_i\zeta), \quad (6)$$

这里 \$\xi\_i\$、\$\eta\_i\$、\$\zeta\_i\$ 是点 \$i(i=1,2,\dots,8)\$ 的坐标。在单元中, \$U\$ 的插值函数是

$$U = \sum_{i=1}^8 N_i U_i,$$

式中  $U_i$  是六面体单元八个顶点处的待定  $U$  值。



**Figure 2.** Parent element (a) and subelement (b)  
**图 2.** 母单元(a)与子单元(b)

### 4.3. 单元分析

用形函数  $N_i$  作权函数  $W_i$ , 将式(5)中的积分分解为各单元  $e$  上的积分。

#### 4.3.1. 式(5)左端的体积分

$$\iiint_e \sigma \nabla U \cdot \nabla N_i dV = (k_{ij}) U_e, \quad (7)$$

其中

$$U_e = (U_1, U_2, \dots, U_8)^T,$$

$$K_{1e} = (k_{ij}),$$

$$k_{ij} = \iiint_e \sigma \left( \frac{\partial N_i}{\partial \xi} \frac{\partial \xi}{\partial x} \frac{\partial N_j}{\partial \xi} \frac{\partial \xi}{\partial x} + \frac{\partial N_i}{\partial \eta} \frac{\partial \eta}{\partial y} \frac{\partial N_j}{\partial \eta} \frac{\partial \eta}{\partial y} + \frac{\partial N_i}{\partial \zeta} \frac{\partial \zeta}{\partial z} \frac{\partial N_j}{\partial \zeta} \frac{\partial \zeta}{\partial z} \right) \frac{abc}{8} d\xi d\eta d\zeta,$$

将式(6)对  $\xi$ 、 $\eta$ 、 $\zeta$  微商后, 代入上式积分, 可得  $k_{ij}$ , 且  $k_{ij} = k_{ji}$ 。 $k_{ij}$  的具体计算公式如下

$$\begin{aligned} \begin{pmatrix} k_{11} \\ k_{21} \\ k_{31} \\ k_{41} \\ k_{51} \\ k_{61} \\ k_{71} \\ k_{81} \\ k_{22} \end{pmatrix} &= \begin{pmatrix} 4 & 4 & 4 \\ 2 & -4 & 2 \\ -2 & -2 & 1 \\ -4 & 2 & 2 \\ 2 & 2 & -4 \\ 1 & -2 & -2 \\ -1 & -1 & -1 \\ -2 & 1 & -2 \\ 4 & 4 & 4 \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \\ \gamma \end{pmatrix}, \\ \begin{pmatrix} k_{32} \\ k_{42} \\ k_{52} \\ k_{62} \\ k_{72} \\ k_{82} \\ k_{33} \\ k_{43} \\ k_{53} \end{pmatrix} &= \begin{pmatrix} -4 & 2 & 2 \\ -2 & -2 & 1 \\ 1 & -2 & -2 \\ 2 & 2 & -4 \\ -2 & 1 & -2 \\ -1 & -1 & -1 \\ 4 & 4 & 4 \\ 2 & -4 & 2 \\ -1 & -1 & -1 \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \\ \gamma \end{pmatrix}, \end{aligned}$$

$$\begin{pmatrix} k_{63} \\ k_{73} \\ k_{83} \\ k_{44} \\ k_{54} \\ k_{64} \\ k_{74} \\ k_{84} \\ k_{55} \end{pmatrix} = \begin{pmatrix} -2 & 1 & -2 \\ 2 & 2 & -4 \\ 1 & -2 & -2 \\ 4 & 4 & 4 \\ -2 & 1 & -2 \\ -1 & -1 & -1 \\ 1 & -2 & -2 \\ 2 & 2 & -4 \\ 4 & 4 & 4 \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \\ \gamma \end{pmatrix}, \quad \begin{pmatrix} k_{65} \\ k_{75} \\ k_{85} \\ k_{66} \\ k_{76} \\ k_{86} \\ k_{77} \\ k_{87} \\ k_{88} \end{pmatrix} = \begin{pmatrix} 2 & -4 & 2 \\ -2 & -2 & 1 \\ -4 & 2 & 2 \\ 4 & 4 & 4 \\ -4 & 2 & 2 \\ -2 & -2 & 1 \\ 4 & 4 & 4 \\ 2 & -4 & 2 \\ 4 & 4 & 4 \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \\ \gamma \end{pmatrix},$$

这里

$$\alpha = \frac{\sigma}{36} \frac{bc}{a}, \quad \beta = \frac{\sigma}{36} \frac{ca}{b}, \quad \gamma = \frac{\sigma}{36} \frac{ab}{c}.$$

#### 4.3.2. 式(5)左端的面积分

若单元  $e$  的一个面  $S_{3478}$  落在无穷远边界上, 则边界积分

$$\iint_e W_i N_i \sigma \frac{1}{r_B - r_A} \left[ \frac{r_B}{r_A} \cos(r_A, n) - \frac{r_A}{r_B} \cos(r_B, n) \right] U dS = \mathbf{K}_{2e} \mathbf{U}_e, \quad (8)$$

其中

$$\mathbf{K}_{2e} = \frac{bc}{36} \cdot \frac{\sigma}{r_B - r_A} \left[ \frac{r_B}{r_A} \cos(r_A, n) - \frac{r_A}{r_B} \cos(r_B, n) \right] \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 4 & 2 & 0 & 0 & 2 & 1 \\ 0 & 0 & 2 & 4 & 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 1 & 0 & 0 & 4 & 2 \\ 0 & 0 & 1 & 2 & 0 & 0 & 2 & 4 \end{pmatrix}.$$

同理, 可得出其他落在无穷边界上的面积分。

#### 4.3.3. 式(5)右端的积分

根据  $\delta$  函数的积分性质, 即

$$\iiint_{\Omega} \delta(A) dV = \begin{cases} 0, & A \notin \Omega \\ \frac{1}{2}, & A \in \Gamma \end{cases},$$

容易写出式(5)右端的体积分

$$2 \iiint_{\Omega} N_i I \delta(A) dV - 2 \iiint_{\Omega} N_i I \delta(B) dV = I_A - I_B, \quad (9)$$

仅与两电源的  $I_A$ ,  $I_B$  有关。

#### 4.4. 总体合成

在单元内, 将式(7)~(9)相加, 然后扩展成全体节点组成的矩阵或列阵, 再把全部单元相加, 得到

$$\begin{aligned} F(\mathbf{U}) &= \sum F_e(\mathbf{U}) = \sum (\mathbf{K}_{1e} + \mathbf{K}_{2e}) \mathbf{U}_e - I_A + I_B \\ &= \sum \mathbf{K}_e \mathbf{U}_e - I_A + I_B = \sum \bar{\mathbf{K}}_e \mathbf{U} - \mathbf{P} \\ &= \mathbf{K}\mathbf{U} - \mathbf{P}, \end{aligned} \quad (10)$$

其中  $\mathbf{K}_e = \mathbf{K}_{1e} + \mathbf{K}_{2e}$ ,  $\bar{\mathbf{K}}_e$  是  $\mathbf{K}_e$  的扩展阵,  $\mathbf{K} = \sum \bar{\mathbf{K}}_e$ ,  $\mathbf{P} = (0, \dots, I_A, \dots, I_B, \dots, 0)^T$ 。

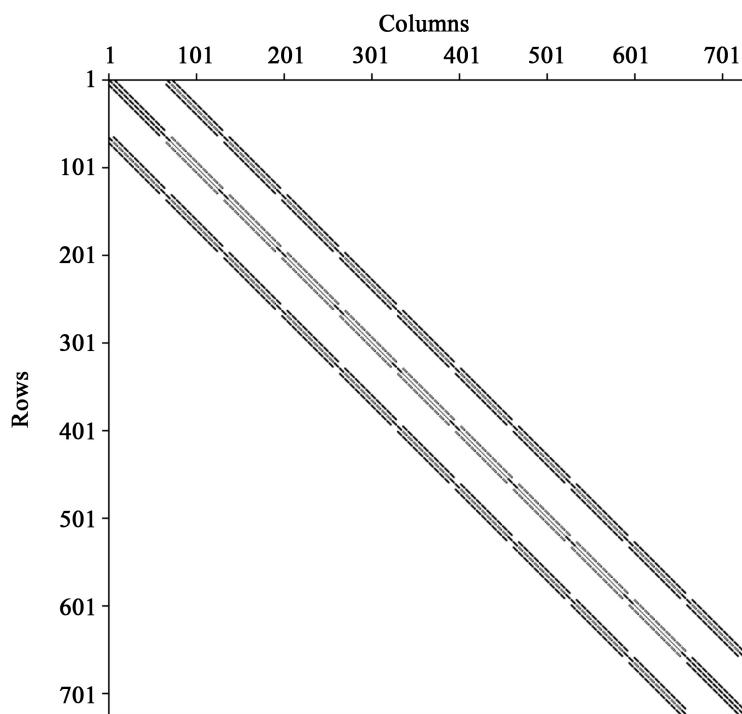
令式(10)为零, 得到线性代数方程组

$$\mathbf{KU} = \mathbf{P}, \quad (11)$$

解方程组, 得到各节点的电位函数值  $U$ 。

## 5. 求解器与存储方式

基于结构化网格剖分的有限单元法中, 由于单元的连通性是固定且有规律的, 每个节点只与邻近的节点有联系, 即只有邻近的节点会出现在相同的单元内, 因此, 在系数矩阵中对应的非零元素大多集中在矩阵的对角线上或对角线附近, 形成了条带状的非零元素区域。



**Figure 3.** Distribution map of non-zero elements in a structured grid's coefficient matrix

**图 3.** 结构化网格系数矩阵中非零元素分布图

以六面体网格剖分为例, 如图 1(a), 网格剖分的节点数量为  $11 \times 11 \times 6 = 726$ , 其全体节点组成的系数矩阵( $726 \times 726$ )非零元素分布如图 3, 其非零元素占比 2.42%。随着网格节点数量的增加, 系数矩阵的稀疏程度将越来越大, 可通过定带宽[16]或者变带宽[17]-[20]的存储方式以减少存储量。而定带宽或变带宽的存储方式依然存储了很多零元素, 采用稀疏存储方式可进一步压缩存储量[21]。

基于 Python 平台, 分别采用压缩稀疏列矩阵(CSC)与压缩稀疏行矩阵(CSR)存储总体系数矩阵, 对比分析 SciPy 及 PyPardiso 两种科学计算库的求解器速度。其中, SciPy 采用专门处理稀疏线性系统的 `scipy.sparse.linalg.spsolve` [22]方法进行求解; PyPardiso 则采用 `pypardiso.spsolve` [23]方法。本文的有限元数值模拟程序均在同一计算机上运行及测试, 操作系统 Windows10 64 位, CPU 型号 Intel (R) Core (TM) i5-8500, 内存 48 GB (DDR4 2666 MHz)。如表 1, 在求解地球物理正演中的大型稀疏线性方程组时, 以 CSC 或 CSR 形式存储的大型稀疏线性系统, 在同一求解器中两者求解速度相差微乎其微; 随着网格单元

节点总数的增加, SciPy 与 PyPardiso 两种求解器用时均有所增长; 当节点总数超过 5000 后, PyPardiso 求解效率明显高于 SciPy; 当节点总数达 37 万时, PyPardiso 求解时间约 17 s, 而 SciPy 耗时急剧增长, 约 1130 s, 两者相差约 66 倍, 这是由于 SciPy 求解线性系统的未知向量非稀疏时会提高计算成本[22], 而直流电阻率三维有限元正演中待求解的全体节点电位为稠密向量, 与基于 SuperLU 分解默认单线程运行的 SciPy 而言, PyPardiso 是专门针对稀疏矩阵优化的求解器, 且支持多进程并行计算, 具有高性能、稳健、节约内存等优点。

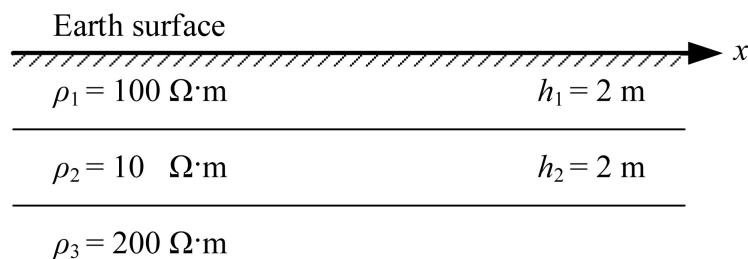
**Table 1.** Solving speed of SciPy and PyPardiso with different storage methods**表 1.** SciPy 与 PyPardiso 中不同存储方式的求解速度

网格剖分节点总数	求解用时(s)			
	SciPy	PyPardiso	CSC	CSR
$021 \times 21 \times 11 = 004851$	0.14	0.14	0.14	0.14
$073 \times 23 \times 17 = 028543$	5.13	5.11	0.38	0.38
$135 \times 35 \times 23 = 108675$	61.55	61.96	1.83	1.87
$141 \times 45 \times 28 = 177660$	191.64	191.74	3.37	3.32
$141 \times 59 \times 35 = 291165$	816.95	812.41	11.74	11.82
$181 \times 59 \times 35 = 373765$	1129.30	1132.46	16.95	16.97

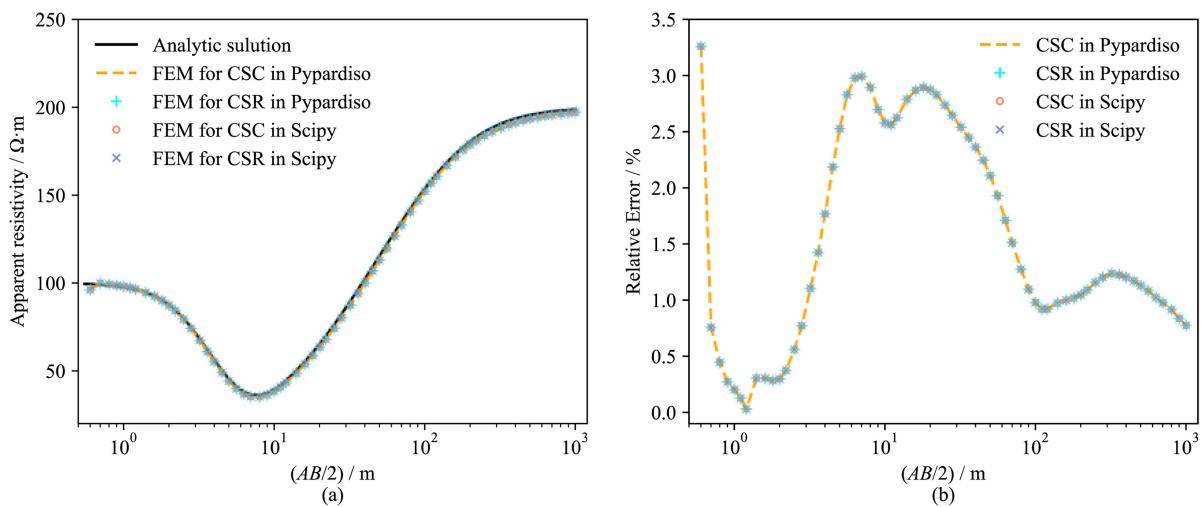
## 6. 数值算例

### 6.1. 模型 1: 水平层状介质模型

首先利用水平层状介质模型的对称四极测深数据检验本文算法的有效性。模型 1 为 H 型水平层状介质(图 4), 第 1 层厚度为 2 m, 电阻率为  $100 \Omega \cdot \text{m}$ , 第 2 层厚度为 2 m, 电阻率为  $10 \Omega \cdot \text{m}$ , 第 3 层电阻率为  $200 \Omega \cdot \text{m}$ 。

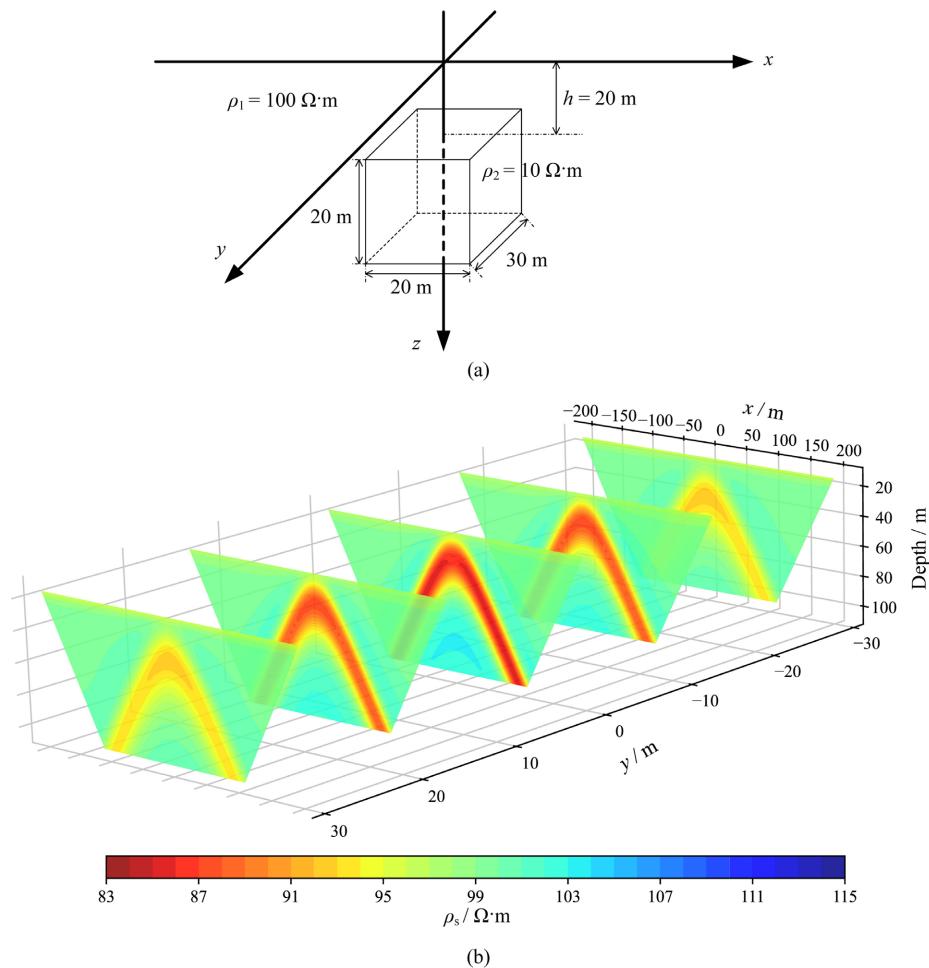
**Figure 4.** Model 1: H-layered media**图 4.** 模型 1: H 型层状介质

采用 140 点一阶 Hankel 滤波系数的滤波算法[24]得到的正演结果, 作为模型 1 的解析解, 同时对比分析本文算法中 SciPy 与 PyPardiso 在 CSC 与 CSR 存储方式下的精度。模型 1 的视电阻率测深曲线如图 5(a), 可见 SciPy 与 PyPardiso 求解器在不同存储方式下的正演结果与解析解拟合较好。两种求解器在不同存储方式下数值模拟结果的相对误差曲线如图 5(b), 容易看出它们精度一致, 由于总场算法电源点附近电位的奇异性, 距场源最近的浅部测深点相对误差为 3.26%, 其他位置最高相对误差均低于 3.00%, 平均相对误差为 1.54%, 表明本文算法行之有效。



**Figure 5.** Model 1 sounding curve of apparent resistivity (a) and relative error curves for two solvers with different storage methods (b)

**图 5.** 模型 1 视电阻率测深曲线(a)及两种求解器在不同存储方式下的相对误差曲线(b)

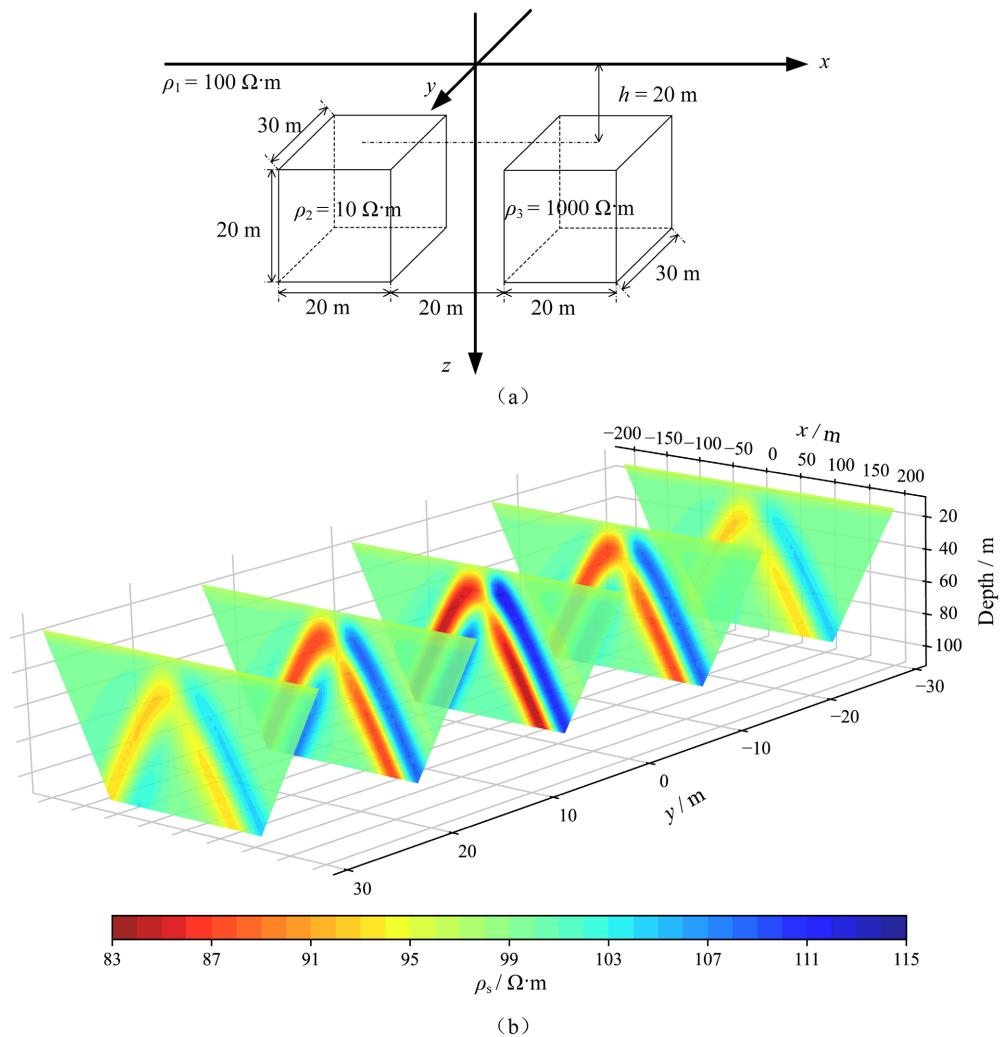


**Figure 6.** Schematic diagram of Model 2 (a) and contour section of apparent resistivity for dipole device (b)

**图 6.** 模型 2 示意图(a)及偶极装置视电阻率等值线断面图(b)

## 6.2. 模型 2：单一异常体模型

模型 2 为均匀大地中单一低阻块状体，如图 6(a)，其沿  $x$ ,  $y$ ,  $z$  方向的规模为  $20 \text{ m} \times 30 \text{ m} \times 20 \text{ m}$ ，电阻率为  $10 \Omega \cdot \text{m}$ ，顶面埋深  $20 \text{ m}$ ，围岩电阻率为  $100 \Omega \cdot \text{m}$ ，以低阻体顶面中心在地面上的投影作为坐标原点。采用偶极装置，偶极长度  $AB = MN = 10 \text{ m}$ ，分别在  $y = 30 \text{ m}$ ,  $15 \text{ m}$ ,  $0 \text{ m}$ ,  $-15 \text{ m}$ ,  $-30 \text{ m}$  测线上进行偶极测深的数值模拟结果如图 6(b)。在主测线上( $y = 0 \text{ m}$ )，低阻体异常最明显，从低阻体埋深处开始，视电阻率呈现极小值，随着深度的增加，视电阻率极小值圈向两侧扩展形成“八”字型双谷。当测线远离主测线，视电阻率极小值逐渐趋于背景场，低阻体产生的异常随之减弱。各断面上的视电阻率等值线图能较好表示出低阻体产生的视电阻率异常。



**Figure 7.** Schematic diagram of Model 3 (a) and contour section of apparent resistivity for dipole device (b)

**图 7.** 模型 3 示意图(a)及偶极装置视电阻率等值线断面图(b)

## 6.3. 模型 3：组合异常体模型

模型 3 为均匀大地中平行排列的高阻与低阻组合异常体，如图 7(a)，两者规模、走向及顶面埋深与模型 2 中单一低阻块状体一致，横向间距  $20 \text{ m}$ 。其中，低阻块状体电阻率为  $10 \Omega \cdot \text{m}$ ，高阻块状体电阻率

为  $1000 \Omega\cdot m$ , 围岩电阻率为  $100 \Omega\cdot m$ 。采用与模型 2 一致的排列方式进行数值模拟, 得到多条测线上视电阻率等值线断面图, 见图 7(b)。在主测线上( $y=0 m$ ), 高阻体与低阻体产生的异常相互叠加, 视电阻率极小值圈顶、极大值圈顶大致对应低阻体、高阻体埋深位置。当测线远离主测线, 高、低阻体产生的异常随之减弱, 视电阻率极小值逐渐增大趋于背景场, 视电阻率极大值逐渐减小趋于背景场。各断面上的视电阻率等值线断面图能较好表示出组合异常体产生的视电阻率异常。

## 7. 结论

- 1) 本文基于 Python 平台, 实现了电导率分块均匀的直流电阻率法三维有限元数值模拟, 通过对比水平层状介质模型对称四极视电阻率测深曲线的解析解与有限元数值解, 除近场源处相对误差达 3.26%, 其他位置相对误差均低于 3.00%, 平均相对误差 1.54%, 表明本文算法准确、有效。
- 2) 通过对比分析 SciPy 与 PyPardiso 求解器性能、精度的试验发现: 两者计算精度相当, 而 PyPardiso 效率更高, 当方程组未知个数达 37 万时, PyPardiso 比 SciPy 求解速度快约 66 倍; 以 CSC 或 CSR 形式存储的大型稀疏线性系统, 在同一求解器中计算效率相当。因此, 基于 Python 平台求解直流电阻率法三维有限元数值模拟中的大型稀疏线性方程组时, 优选支持多进程并行计算的 PyPardiso 高性能求解器可有效保证计算效率。

## 基金项目

广西自然科学基金(2024GXNSFBA010257); 广西人才基地专项基金(桂科 AD23026231); 江西省地质局青年科学技术带头人培养计划项目(2024JXDZKJRC10)资助。

## 参考文献

- [1] 阮百尧, 熊彬. 电导率连续变化的三维电阻率测深有限元模拟[J]. 地球物理学报, 2002, 45(1): 131-138.
- [2] 阮百尧, 熊彬, 徐世浙. 三维地电断面电阻率测深有限元数值模拟[J]. 地球科学, 2001, 26(1): 73-77.
- [3] 黄俊革. 三维电阻率/极化率有限元正演模拟与反演成像[D]: [博士学位论文]. 长沙: 中南大学, 2003.
- [4] 汤井田, 公劲皓. 三维直流电阻率有限元-无限元耦合数值模拟[J]. 地球物理学报, 2010, 53(3): 717-728.
- [5] 任政勇, 汤井田. 基于局部加密非结构化网格的三维电阻率法有限元数值模拟[J]. 地球物理学报, 2009, 52(10): 2627-2634.
- [6] 杨志龙. 基于非结构自适应有限元的三维直流电阻率各向异性响应正演研究[D]: [硕士学位论文]. 长春: 吉林大学, 2018.
- [7] 赵宁, 黄明卫, 申亚行, 等. 高阶自适应有限元三维直流电阻率正演方法及其在沁水盆地煤气层压裂监测中的应用[J]. 石油地球物理勘探, 2021, 56(1): 209-216.
- [8] 刘斌, 李术才, 李树忱, 等. 基于预条件共轭梯度法的直流电阻率三维有限元正演研究[J]. 岩土工程学报, 2010, 32(12): 1846-1853.
- [9] 张钱江, 戴世坤, 陈龙伟, 等. 多源条件下直流电阻率法有限元三维数值模拟中一种近似边界条件[J]. 地球物理学报, 2016, 59(9): 3448-3458.
- [10] 欧东新, 王超, 闫凯暄, 等. 电导率线性变化直流电测深三维有限单元法正演模拟[J]. 桂林理工大学学报, 2019, 39(3): 606-614.
- [11] 朱姣. 基于四面体网格谱元法三维电磁正演模拟研究[D]: [博士学位论文]. 长春: 吉林大学, 2022.
- [12] Cockett, R., Kang, S., Heagy, L.J., Pidlisecky, A. and Oldenburg, D.W. (2015) Simpeg: An Open Source Framework for Simulation and Gradient Based Parameter Estimation in Geophysical Applications. *Computers & Geosciences*, **85**, 142-154. <https://doi.org/10.1016/j.cageo.2015.09.015>
- [13] Krischer, L., Megies, T., Barsch, R., Beyreuther, M., Lecocq, T., Caudron, C., et al. (2015) Obspy: A Bridge for Seismology into the Scientific Python Ecosystem. *Computational Science & Discovery*, **8**, Article ID: 014003. <https://doi.org/10.1088/1749-4699/8/1/014003>

- 
- [14] Rücker, C., Günther, T. and Wagner, F.M. (2017) Pygimli: An Open-Source Library for Modelling and Inversion in Geophysics. *Computers & Geosciences*, **109**, 106-123. <https://doi.org/10.1016/j.cageo.2017.07.011>
  - [15] 余洋. 垂直地电剖面法数值模拟及其分析[D]: [硕士学位论文]. 成都: 西南交通大学, 2006.
  - [16] 冯凯, 秦策. 大地电磁(MT)自适应有限元各向异性正演[J]. 吉林大学学报(地球科学版), 2020, 50(6): 1887-1896.
  - [17] 欧东新, 阮百尧. 电性轴对称分布电阻率测井的有限元法模拟[J]. 桂林工学院学报, 2002, 22(2): 123-128.
  - [18] 韩思旭, 李勇, 陈卫营, 等. 基于 CUDA 并行计算的大地电磁二维有限元数值模拟研究[J]. 地球物理学进展, 2016, 31(3): 1095-1102.
  - [19] 崔杨洋. 基于 OpenMP 大地电磁二维有限元正演模拟[D]: [硕士学位论文]. 桂林: 桂林理工大学, 2017.
  - [20] 强建科, 罗延钟. 三维地形直流电阻率有限元法模拟[J]. 地球物理学报, 2007, 50(5): 1606-1613.
  - [21] 王亮, 王鹤, 席振铢, 等. 基于预处理广义极小残量法的大地电磁正演计算[J]. 地球物理学进展, 2013, 28(1): 165-170.
  - [22] Virtanen, P., Gommers, R., Oliphant, T.E., Haberland, M., Reddy, T., Cournapeau, D., et al. (2020) Scipy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, **17**, 261-272. <https://doi.org/10.1038/s41592-019-0686-2>
  - [23] Haas, A. (2024) Python Interface to the Intel MKL Pardiso Library to Solve Large Sparse Linear Systems of Equations. <https://github.com/haasad/PyPardiso>
  - [24] 熊彬, 徐志锋, 蔡红柱. MATLAB 地球物理科学计算实战[M]. 武汉: 中国地质大学出版社, 2020: 279-283.