libEMM：一款连接顺序实现与 GPU 并行实现的虚拟波域三维受控源电磁学建模库

Pengliang Yang[1] 哈尔滨工业大学数学学院，中国哈尔滨，邮编 150001。电子邮箱：ypl.2100@gmail.com

2023年4月4日

arXiv:2304.00233v1 [physics.geo-ph] 1 Apr 2023

摘要

本文介绍了一款用于虚拟波域三维可控源电磁法（CSEM）建模的软件——libEMM。该软件基于一种新近开发的非均匀网格高阶时域有限差分（FDTD）方法。数值模拟可通过基于 MPI 的高性能计算架构，在多个处理器上并行执行。采用 C 编写的 FDTD 内核已通过 OpenMP 并行化，并利用本地共享内存实现加速。此外，该软件还提供基于 CUDA 编程语言、采用相同算法的 GPU 实现，可用于交叉验证和效率比较。展望未来，libEMM 将应用于陆地和海洋环境中的三维 CSEM 反演。

1 引言

受控源电磁学（CSEM）是一项成熟的技术，可用于探测含烃地层，开展地球物理勘探（Chave and Cox, 1982; Constable et al., 1986）。CSEM 已应用于空中（Chang-Chun et al., 2015）和井间（Alumbaugh and Newman, 1997）几何构型，以及陆地（Grayver et al., 2014; Ward and Hohmann, 1988; Zhdanov and Keller, 1994）和／或海洋环境（Constable, 2010; Eidesmo et al., 2002; Ellingsrud et al., 2002; MacGregor and Tomlinson, 2014）。这些应用基于低频电磁信号对地下介质中高电阻率对比的高度敏感性。钻探前需要确定井位部署时，这使 CSEM 成为勘探目标降险的理想工具（MacGregor et al., 2007）（Meju, 2019）。

可控源电磁法（CSEM）数据反演是一种强大的成像方法，可用于识别显著的高阻异常，进而帮助揭示地球内部潜在的烃类分布。CSEM反演依赖于反复调用基于扩散型麦克斯韦方程的三维数值建模引擎。为此，人们开发了多种方法，包括频域有限差分法（Mulder, 2006；Newman 和 Alumbaugh, 1995；Smith, 1996；Streich, 2009）、频域有限元法（da Silva 等人, 2012；Key, 2016；Li 和 Key, 2007；Rochlitz 等人, 2019）以及时域有限差分法（Oristaglio 和 Hohmann, 1984；Taflove 和 Hagness, 2005；Wang 和 Hohmann, 1993）。

已有多种开源代码在公共领域发布，用于CSEM建模，例如 SimPEG（Cockett 等人, 2015）、MARE2DEM（Key, 2016）、PETGEM（Castillo-Reyes 等人, 2018）、custEM（Rochlitz 等人, 2019）和 emg3d（Werthm¨uller 等人, 2019）。其中，Fortran 编写的 MARE2DEM 专用于二维建模和反演，其余四种则使用 Python 构建，用于三维建模。SimPEG 和 emg3d 均支持正演模拟以及反演模式下的梯度估计。SimPEG（Cockett 等人, 2015）和 emg3d（Werthm¨uller 等人, 2019）的正演模拟依赖有限体积网格，而 PETGEM（Castillo-Reyes 等人, 2018）和 custEM（Rochlitz 等人, 2019）则采用有限元，旨在对复杂模型进行精确建模。所有这些软件包都基于频域方法，通过离散化频域麦克斯韦方程并求解线性系统来实现。部分软件包支持输出时间序列形式的电磁场：先在经过审慎选择的频率上进行建模，再将频域场通过傅里叶变换转换到时域（Rochlitz 等人, 2021）。

本文的主要贡献是提供时域三维CSEM建模代码 libEMM。该代码实现了我们新开发的、可在均匀网格和非均匀交错网格上运行的高阶FDTD格式（Yang 和 Mittet，2023），并采用高效的虚拟波域方法进行三维CSEM建模（Mittet，2010年）。libEMM 与其他软件的主要区别在于，它通过直接离散虚拟时域中的转换后麦克斯韦方程来计算电磁场。显著

与常用的非均匀网格二阶格式相比，该方法在精度和计算效率方面均取得了提升（Yang 和 Mittet，2023）。

本文重点介绍软件框架中的计算机实现，而非从准确性的角度对方法进行数值论证。除了介绍如何通过消息传递接口（MPI）在多个分布式 CPU 核心上实现该方法外，本文还展示了若干实现多线程 GPU 并行化的关键技术。这也是首次将该方法的 GPU 实现从均匀网格（Yang，2021）扩展到非均匀网格，并以开源软件的形式发布，希望能得到广泛采用。因此，libEMM 使用 C 和 Nvidia 统一计算设备架构（CUDA）进行编程，并通过 MPI 实现并行化。随后，可通过 Shell 脚本在高性能计算集群上启动建模任务。

本文结构安排如下。首先，简要回顾虚拟波域中的 CSEM建模方法。随后，介绍实现高效计算的关键要素，包括边界条件（顶部的空气-水边界条件和其他侧面的吸收边界条件）、时间步进、介质均匀化和最优网格生成。为便于在支持 GPU 的硬件上进行 CSEM建模，我们还列举了若干使用 CUDA 编程语言优化代码的策略。接着，介绍 libEMM 的主要特性，以便用户轻松使用该软件。然后，通过性能分析展示应用示例。最后，简要探讨该方法在实际三维 CSEM 反演中的潜在应用。

2 虚波域中的 CSEM建模

2.1 扩散型麦克斯韦方程

CSEM物理受扩散型麦克斯韦方程（由法拉第定律和安培定律组成）支配，该方程在频域中的形式为

$$\begin{cases} \nabla \times \mathbf{E} - \mathrm{i}\omega\mu\mathbf{H} = -\mathbf{M} \\ -\nabla \times \mathbf{H} + \sigma\mathbf{E} = -\mathbf{J} \end{cases}, \tag{1}$$

其中 $\mathbf{E} = (E_x,\ E_y,\ E_z)^T$ 和 $\mathbf{H} = (H_x,\ H_y,\ H_z)^T$ 分别为电场和磁场；$\mathbf{J} = (J_x,\ J_y,\ J_z)^T$ 和 $\mathbf{M} = (M_x,\ M_y,\ M_z)^T$ 分别为电源和磁源。这里，在时域与频域之间切换时，我们稍微滥用了相同的记号来表示相同的量，并采用了傅里叶变换约定 $\partial_t \leftrightarrow -\mathrm{i}\omega$ 。磁导率为 μ。电导率是一个对称的 3×3 张量：

$$\sigma = \begin{bmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{xz} \\ \sigma_{yx} & \sigma_{yy} & \sigma_{yz} \\ \sigma_{zx} & \sigma_{zy} & \sigma_{zz} \end{bmatrix} \tag{2}$$

其中 $\sigma_{ij} = \sigma_{ji},\ i,\ j \in \{x, y, z\}$。各向同性介质是指电导率张量只有对角元素非零，且各方向上的对角元素相同：$\sigma_{xx} = \sigma_{yy} = \sigma_{zz}$；$\sigma_{ij} = 0,\ i \neq j$。垂直横向各向同性（VTI）介质也只有对角元素，即 $\sigma_h := \sigma_{xx} = \sigma_{yy}$，$\sigma_v = \sigma_{zz}$，其中 σ_h 和σ_v 分别表示水平电导率和垂直电导率。CSEM 系统中也经常使用电阻率，它是电导率的逆矩阵（$\rho_{ij} = 1/\sigma_{ij}$）。

2.2 虚拟波域建模

通过定义虚拟介电常数 ϵ 使得 $\sigma = 2\omega_0\varepsilon$，并将第二个方程乘以 $\sqrt{-\mathrm{i}\omega/2\omega_0}$，Mittet，2010年将该方程转换为

$$\begin{cases} \nabla \times \mathbf{E}' - \mathrm{i}\omega'\mu\mathbf{H}' = -\mathbf{M}' \\ -\nabla \times \mathbf{H}' - \mathrm{i}\omega'\varepsilon\mathbf{E}' = -\mathbf{J}' \end{cases} \tag{3}$$

其中，我们通过以下方式引入撇号来定义所得的波域场

$$\mathbf{E}' = \mathbf{E},\ \mathbf{H}' = \sqrt{\frac{-\mathrm{i}\omega}{2\omega_0}}\mathbf{H},\ \mathbf{M}' = \mathbf{M},\ \mathbf{J}' = \sqrt{\frac{-\mathrm{i}\omega}{2\omega_0}}\mathbf{J},\ \omega' = (1+\mathrm{i})\sqrt{\omega\omega_0}. \tag{4}$$

方程（3）可转换为时域形式：

$$\begin{cases} \nabla \times \mathbf{E}' + \mu\partial_t\mathbf{H}' = -\mathbf{M}' \\ -\nabla \times \mathbf{H}' + \varepsilon\partial_t\mathbf{E}' = -\mathbf{J}' \end{cases} \tag{5}$$

在波动域中，可在建模过程中通过虚拟波变换实时计算频域场。

$$E_i'(\mathbf{x},\omega') = \int_0^{T_{\max}} E_i'(\mathbf{x},t)e^{\mathrm{i}\omega' t}\mathrm{d}t, \tag{6a}$$

$$H_i'(\mathbf{x},\omega') = \int_0^{T_{\max}} H_i'(\mathbf{x},t)e^{\mathrm{i}\omega' t}\mathrm{d}t, \tag{6b}$$

其中，$i=\{x,y,z\}$ 为电磁场的分量索引；$T_{\max}$ 为电磁场达到稳态前的最终时间。

点源的格林函数通过将电磁场与源电流归一化得到。在不存在磁源（$\mathbf{M}=0$）的情况下，由电流J_j（$\mathbf{x}_s$, ω）产生的格林函数为

$$G_{ij}^{EE}(\mathbf{x},\omega;\mathbf{x}_s) = \frac{E_i(\mathbf{x},\omega;\mathbf{x}_s)}{J_j(\mathbf{x}_s,\omega)} = \sqrt{\frac{-\mathrm{i}\omega}{2\omega_0}}\frac{E_i'(\mathbf{x},\omega;\mathbf{x}_s)}{J_j'(\mathbf{x}_s,\omega)}, \tag{7a}$$

$$G_{ij}^{HE}(\mathbf{x},\omega;\mathbf{x}_s) = \frac{H_i(\mathbf{x},\omega;\mathbf{x}_s)}{J_j(\mathbf{x}_s,\omega)} = \frac{H_i'(\mathbf{x},\omega;\mathbf{x}_s)}{J_j'(\mathbf{x}_s,\omega)}, \tag{7b}$$

其中，G_{ij}^{EE}（$\mathbf{x}$ ω $\mathbf{x}_s$）和 G_{ij}^{HE}（$\mathbf{x}$ ω；$\mathbf{x}_s$）分别表示电和磁格林函数的第 i 个分量 's 函数，对应于角 频率 ω 在空间位置 $\mathbf{x}$处。

3 计算机实现

本节介绍基于我们新开发的非均匀网格高阶FDTD方法（Yang 和 Mittet，2023），高效实现虚拟波域可控源电磁法建模的关键要素。为便于代码实现，我们为电磁场定义了一种数据结构，并使用指针 emf 访问所有相关参数的信息。

3.1 非均匀网格上的高阶FDTD

与（Maaø，2007）中的衰减形式不同，方程（3）是一个纯波动域方程，可采用交错网格上的蛙跳式交错网格有限差分方法进行离散：

$$\begin{cases} H_i^{'n+1/2} = & H_i^{'n-1/2} - \Delta t\mu^{-1}(\xi_{ijk}\partial_j E_k^{'n} + M_i^{'n}) \\ E_i^{'n+1} = & E_i^{'n} + \Delta t(\varepsilon'^{-1})_{ij}(\xi_{jkl}\partial_k H_l^{'n+1/2} - J_j^{'n+1/2}) \end{cases} \tag{8}$$

其中，Δt 表示时间采样，ξ_{ijk} 表示列维-奇维塔符号。各场在非均匀交错网格上的位置与均匀网格上的交错 FDTD 类似。均匀网格与非均匀网格上的 FDTD 实现之间的主要区别在于这些空间导数（$\partial_i E_k'$和 $\partial_k H_l'$）的离散化

为了计算任意网格间距下的电磁场及其导数，我们必须使用若干节点 x_0，x_1，$\cdots$，x_n进行多项式插值。根据泰勒展开，我们有

$$f(x_i) = f(x) + f'(x)(x_i-x) + \frac{1}{2}f''(x)(x_i-x)^2 + \cdots + \frac{1}{n!}f^{(n)}(x)(x_i-x)^n + \cdots \tag{9}$$

其中 $i=0$，1，$\cdots$，n。定义 $a_i(x) := f^{(i)}(x)/i!$，并使用前 $n+1$ 个互异节点 x_0，x_1，$\cdots$，x_n 构建一个矩阵方程组

$$\underbrace{\begin{bmatrix} f(x_0) \\ f(x_1) \\ \vdots \\ f(x_n) \end{bmatrix}}_{\mathbf{f}} = \underbrace{\begin{bmatrix} 1 & x_0-x & (x_0-x)^2 & \cdots & (x_0-x)^n \\ 1 & x_1-x & (x_1-x)^2 & \cdots & (x_1-x)^n \\ \vdots & \ddots & & \vdots & \\ 1 & x_n-x & (x_n-x)^2 & \cdots & (x_n-x)^n \end{bmatrix}}_{\mathbf{V}^T(x_0-x,\cdots,x_n-x)} \underbrace{\begin{bmatrix} a_0(x) \\ a_1(x) \\ \vdots \\ a_n(x) \end{bmatrix}}_{\mathbf{a}} \tag{10}$$

其中，$\mathbf{V}^T(x_0-x,\cdots,x_n-x)$ 是由 $x_0-x,\cdots,x_n-x$ 确定的范德蒙德矩阵的转置。上述表达式表明，通过对范德蒙德矩阵 $\mathbf{a} = [\mathbf{V}^T]^{-1}\mathbf{f}$ 求逆，可以得到函数 $f(x)$ 及其在任意位置 x 处直至 n 阶的导数。当范德蒙德矩阵的阶数

增大时，范德蒙德矩阵求逆会变得高度病态。幸运的是，通过利用范德蒙德矩阵求逆与拉格朗日多项式插值之间的等价性，可以采用一种准确且高效的算法（Bj¨orck 和 Pereyra，1970）来完成这一具有挑战性的任务。以下代码片段按照（Golub，1996，算法 4.6.1）实现了该算法。

```
/* 求解范德蒙德系统 V^T(x0,x1,...,xn)a = f *，其中 n+1 个点由向量 x=[
x0,x1,...,xn]指定； * 解存储在向量 a 中。 */ void vandermonde(int n, float *x,
float *a, float*f) { int i,k; /* 计算插值多项式的牛顿表示 */ for(i=0; i<=n; ++i) a[i
] = f[i]; for(k=0; k<n; k++){ for(i=n; i>k; i--){ a[i] = (a[i]-a[i-1])/(x[i]-x[i-k-1]); } }
for(k=n-1; k>=0; k--){ for(i=k; i<n; i++){ a[i]-= a[i+1]*x[k]; } } }
```

设逆矩阵 $[\mathbf{V}^T]^{-1}$ 的第 i 行、第 j 列元素为 w_{ij}，即 $([\mathbf{V}^T]^{-1})_{ij} = w_{ij},\ \ i,\ \ j = 0,\ \cdots,\ n$。由此可得

$$\underbrace{\begin{bmatrix} a_0(x) \\ a_1(x) \\ \vdots \\ a_n(x) \end{bmatrix}}_{\mathbf{a}} := \underbrace{\begin{bmatrix} w_{00} & w_{01} & \cdots w_{0n} \\ w_{10} & w_{11} & \cdots w_{1n} \\ \vdots & & \\ w_{n0} & w_{n1} & \cdots w_{nn} \end{bmatrix}}_{[\mathbf{V}^T]^{-1}} \underbrace{\begin{bmatrix} f(x_0) \\ f(x_1) \\ \vdots \\ f(x_n) \end{bmatrix}}_{\mathbf{f}} \tag{11}$$

在每个时间步都需要反复使用插值权重的情况有很多：

1. 在同一网格位置进行数值建模时，将使用相同的有限差分系数；2. 在不同时间步于同一位置记录场时，可能使用相同的插值权重。3. 注入源时间函数时，需要应用相同的插值权重。

后两种情况是由于源和接收器可能位于任意位置，而不一定位于网格上。因此，有必要将这些权重制成表格，而不是根据反演算法重复计算。由于以算子形式应用算法时不需要显式的矩阵系数，因此如何获得插值权重 w_{ij} 并不明显。本文介绍一种有效的方法来实现这一点。其思路如下：将向量 f 输入算法，其中第 j 个元素为 1，其余元素为 0，Vandermonde 反演算法的输出将是矩阵 $[\mathbf{V}^T]^{-1}$ 的第 j 列：

$$\mathbf{f} = [0, \cdots, 1, \cdots, 0]^T, \quad \mathbf{a} = [w_{1j}, \cdots, w_{nj}]^T. \tag{12}$$

通过将索引 j 从 0 遍历到 n，我们可以得到 $[\mathbf{V}^T]^{-1}$ 的所有列。在直角非均匀网格上，我们使用三个一维插值算子的张量积构造多维有限差分模板。

考虑使用以$x = x_{i+1/2}$ 为中心、用于磁场分量的 $2r$ 个非等距节点和用于电场分量的 $x = x_i$ 个非等距节点，在 x 方向上的一阶导数的交错有限差分近似。

用于离散化电场一阶导数的前向算子 D_x^+ 为

$$\begin{aligned} D_x^+ u|_{x=x_{i+1/2}} =& \Big(c_1^+(x_{i+1/2})u(x_{i+1}) + c_0^+(x_{i+1/2})u(x_i)\Big) \\ &+ \Big(c_2^+(x_{i+1/2})u(x_{i+2}) + c_{-1}^+(x_{i+1/2})u(x_{i-1})\Big) \\ &+ \cdots \\ &+ \Big(c_r^+(x_{i+1/2})u(x_{i+r}) + c_{-r+1}^+(x_{i+1/2})u(x_{i-r+1})\Big) \end{aligned} \tag{13}$$

而后向算子 D_x^- 用于离散磁场的一阶导数，其形式为

$$\begin{aligned} D_x^- u|_{x=x_i} =& \Big(c_1^-(x_i)u(x_{i+1/2}) + c_0^-(x_i)u(x_{i-1/2})\Big) \\ &+ \Big(c_2^-(x_i)u(x_{i+3/2}) + c_{-1}^-(x_i)u(x_{i-3/2})\Big) \\ &+ \cdots \\ &+ \Big(c_r^-(x_i)u(x_{i+r-1/2}) + c_{-r+1}^-(x_i)u(x_{i-r+1/2})\Big), \end{aligned} \tag{14}$$

这里我们将差分项配对，以突出其与均匀网格交错有限差分的相似性。系数 c_j^+、 c_j^-、 $j = -r+1, \cdots, r$ 分别是式（11）中矩阵$\mathbf{V}^T(x_{i+r-1/2} - x_i, \ldots, x_{i-r+1/2} - x_i)$ 和$\mathbf{V}^T(x_{i+r} - x_i, \cdots, x_{i-r+1} - x_i)$ 的逆矩阵的第二行。使用 $2r$ 个节点，我们可在空间中实现最高达 $2r$ 阶的精度。

将半径 r 记为 emf->rd。通过范德蒙德矩阵求逆，我们计算 $2r$ 阶有限差分系数，并将其存储在 emf->v3s 中（其中 3 表示 z 方向，s 表示网格交错），以计算 $\partial_z H_x$ 和 $\partial_z H_y$，得到

```
//i1min,i1max=i1 的下边界和上边界 - x 方向 //i2min,i2max=i2 的下边界和上边界 - y 方向 //i3min,i3max=i3 的下边界和上边界 - z 方向 for(i3=i3min; i3<=i3max;  ++i3){
  for(i2=i2min; i2<=i2max;  ++i2){ for(i1=i1min; i1<=i1max;  ++i1){ ... if(电磁场->rd==1){ D3H2 = 电磁场->v3s[i3][0]*电磁场->H2[i3-1][i2][i1]+ 电磁场->v3s[i3][1]*电磁场->H2[i3][i2][i1]; D3H1 = 电磁场->v3s[i3][0]*电磁场->H1[i3-1][i2][i1]+ 电磁场->v3s[i3][1]*电磁场->H1[i3][i2][i1]; }else if(电磁场->rd==2){ D3H2 = 电磁场->v3s[i3][0]*电磁场->H2[i3-2][i2][i1]+ 电磁场->v3s[i3][1]*电磁场->H2[i3-1][i2][i1]+ 电磁场->v3s[i3][2]*电磁场->H2[i3][i2][i1]+ 电磁场->v3s[i3][3]*电磁场->H2[i3+1][i2][i1]; D3H1 = 电磁场->v3s[i3][0]*电磁场->H1[i3-2][i2][i1]+ 电磁场->v3s[i3][1]*电磁场->H1[i3-1][i2][i1]+ 电磁场->v3s[i3][2]*电磁场->H1[i3][i2][i1]+ 电磁场->v3s[i3][3]*电磁场->H1[i3+1][i2][i1]; }else if(电磁场->rd==3){ D3H2 = 电磁场->v3s[i3][0]*电磁场->H2[i3-3][i2][i1]+ 电磁场->v3s[i3][1]*电磁场->H2[i3-2][i2][i1]+ 电磁场->v3s[i3][2]*电磁场->H2[i3-1][i2][i1]+ 电磁场->v3s[i3][3]*电磁场->H2[i3][i2][i1]+ 电磁场->v3s[i3][4]*电磁场->H2[i3+1][i2][i1]+ 电磁场->v3s[i3][5]*电磁场->H2[i3+2][i2][i1]; D3H1 = 电磁场->v3s[i3][0]*电磁场->H1[i3-3][i2][i1]+ 电磁场->v3s[i3][1]*电磁场->H1[i3-2][i2][i1]+ 电磁场->v3s[i3][2]*电磁场->H1[i3-1][i2][i1]+ 电磁场->v3s[i3][3]*电磁场->H1[i3][i2][i1]+ 电磁场->v3s[i3][4]*电磁场->H1[i3+1][i2][i1]
```

```
            + 电磁场->v3s[i3][5]*电磁场->H1[i3+2][i2][i1];
        }
        ...
      }
    }
}
```

上述过程使我们能够利用高阶有限差分格式，准确计算电磁场及其导数，且通常可在直线网格中采用任意网格间距。这为在统一框架下，使用非均匀网格进行高阶时域有限差分可控源电磁法建模铺平了道路。

3.2 稳定性条件和色散误差

根据标准的冯·诺依曼分析，作者已证明（Yang 和 Mittet，2023），对于通用的直线型非均匀网格，要实现稳定的数值建模，必须满足以下条件

$$\eta\Delta t \leq 1 \quad \text{with} \quad \eta = \frac{1}{2} v_{\max} \sqrt{(D_x^{\max})^2 + (D_y^{\max})^2 + (D_z^{\max})^2}, \tag{15}$$

其中 $v_{\max} = \max\{v\}$，其中 $v := 1/\sqrt{\mu\epsilon}$ 是传播电磁场的速度；$D_x^{\max}$、$D_y^{\max}$ 和 $D_z^{\max}$ 分别是沿 x、y 和 z 方向离散化一阶导数的最大值。具体而言，我们有

$$D_x^{\max} = \max\Big(\sum_{i=-r+1}^{r} |c_i^+|, \sum_{i=-r+1}^{r} |c_i^-|\Big) \tag{16}$$

以及对 $D_y^{\max}$ 和 $D_z^{\max}$ 的类似估算。为使时间步数 N_t 最少，我们希望在不违反上述稳定性条件的情况下使用尽可能大的 Δt。因此，时间采样 Δt 会根据预先确定的网格间距自动确定，即 $\Delta t = 0.99/\eta$。

如附录所示，均匀网格上的色散误差很容易测量。然而，在非均匀网格上，式（13）和式（14）中的离散空间导数算子使色散分析变得困难（波数也随空间位置变化）。鉴于实际测量非均匀网格上的色散误差存在困难，我们未对波动域中的色散误差进行控制。不进行此类控制的另一个充分理由是，我们最终计算的电磁场位于扩散域，而非波动域。从物理角度而言，扩散型电磁场本身就具有色散性，因此，从波动域转换到扩散域后，物理色散将在计算得到的场中占主导地位。这意味着波动域中的数值色散误差最终将被物理色散淹没。需要注意的是，在我们的方法中，物理色散已根据方程（4）以解析方式实现，因此不含数值误差。

3.3 顶部边界条件

在每个时间步中，都必须采用高阶有限差分时域法妥善处理顶部边界（海洋可控源电磁法中的空气-海水界面或陆地可控源电磁法中的空气-地层界面）（Oristaglio和Hohmann，1984；Wang和Hohmann，1993）。在空气中，电导率为零。不存在电流源时，安`培定律便简化为

$$\nabla \times \mathbf{H} = 0, \tag{17}$$

从而得到波数域关系

$$H_x = \frac{k_x}{k_z} H_z, \quad H_z = \frac{k_y}{k_z} H_z, \tag{18}$$

其中 k_x、k_y 和 k_z 分别是 x、y 和 z 方向上的波数。由于 $\nabla\cdot\mathbf{H} = 0$，根据关系式 $\nabla\times\nabla\times\mathbf{H} = \nabla(\nabla\cdot\mathbf{H}) - \Delta\mathbf{H}$ 可得

$$\Delta\mathbf{H} = 0, \tag{19}$$

这意味着 $k_x^2 + k_y^2 + k_z^2 = 0$，因此 $k_z = \pm\mathrm{i}\sqrt{k_x^2 + k_y^2}$。此处选择正确的符号至关重要，这样场才会在无限远处消失（$\mathbf{H} \to 0$ 当 $z \to 0$ 时），从而得到 $k_z = -\mathrm{i}\sqrt{k_x^2 + k_y^2}$。由此可得以下实现

$$H_x = \frac{\mathrm{i}k_x}{\sqrt{k_x^2 + k_y^2}} H_z, \quad H_z = \frac{\mathrm{i}k_y}{\sqrt{k_x^2 + k_y^2}} H_z. \tag{20}$$

请注意，我们对（Mittet，2010年，方程 46）中的符号进行了修正。为了使用高阶FDTD格式，我们还需要通过强制（Oristaglio和Hohmann，1984）来外推电场，

$$\Delta \mathbf{E} = 0 \tag{21}$$

在空气界面处。

上述边界条件会导致通过傅里叶变换将场外推至虚点。由于快速傅里叶变换（FFT）要求网格间距相等，空气-水边界条件要求 x 和 y 方向均采用均匀网格。通过在非均匀网格与均匀网格之间进行插值，可以在 x 和 y 方向采用非均匀网格。我们的数值结果（Yang 和 Mittet，2023）表明，这会降低建模精度，因此 li bEMM 未实现此方法。

3.4 吸收边界条件

为模拟波向无穷远传播，我们在模型顶部以外的其他方向采用完美匹配层（PML）边界条件（Chew 和 Weedon，1994）。它有助于利用截断计算网格衰减潜在反射。应用 PML 相当于进行坐标拉伸 $\partial_{\tilde{x}} = s_x^{-1}\partial_x$ 并乘以复因子 s_x ($|s_x| > 1$)。卷积完美匹配层（CPML）（Roden 和 Gedney，2000）采用 $s_x = \kappa_x + \frac{d_x(x)}{\alpha_x + i\omega}$($\kappa_x \geq 1$)，使得

$$\partial_{\tilde{x}} = \frac{1}{\kappa_x}\partial_x + \psi_x, \tag{22}$$

wher记忆变量 ψ_x 随后可以在时间步进过程中以递归方式计算 g

$$\psi_x^{n+1} = b_x \psi_x^n + a_x \partial_x^{n+1/2} \tag{23}$$

其中

$$b_x = e^{-(d_x/\kappa_x + \alpha_x)\Delta t}, a_x = \frac{d_x}{\kappa_x(d_x + \kappa_x \alpha_x)}(b_x - 1). \tag{24}$$

阻尼剖面 $d_x(x)$ 和常数 κ_x 根据（Komatitsch和Martin，2007）进行选择。我们以与时域有限差分中相同的方式，为 x、y 和 z 方向实现CPML。假设电阻率/电导率模型的尺寸为 n1*n2*n3 的立方体。模型将在每一侧填充 nb 个CPML层和 ne 个额外缓冲区：

```
emf->nbe = emf->nb + emf->ne; emf->n1pad = emf->n1 + 2*emf->
nbe; // x方向的网格点总数 emf->n2pad = emf->n2 + 2*emf->nbe; // y方向
的网格点总数 emf->n3pad = emf->n3 + 2*emf->nbe; // z方向的网格点总数
```

以下给出了与 $\partial_z H_x$ 和 $\partial_z H_y$ 相关的记忆变量示例 :

```
... if(i3<电磁场->nb){ 电磁场->memD3H2[i3][i2][i1] = 电磁场->b3[i
3]*电磁场->memD3H2[i3][i2][i1]+ 电磁场->a3[i3]*D3H2; 电磁场->
memD3H1[i3][i2][i1] = 电磁场->b3[i3]*电磁场->memD3H1[i3][i2][i1]
+ 电磁场->a3[i3]*D3H1; D3H2 += 电磁场->memD3H2[i3][i2][i1]; D3H
1 += 电磁场->memD3H1[i3][i2][i1]; }else if(i3>电磁场->n3pad-1-电磁场-
>nb){ j3 = 电磁场->n3pad-1-i3; k3 = j3+电磁场->nb; 电磁场->memD3H2[
k3][i2][i1] = 电磁场->b3[j3]*电磁场->memD3H2[k3][i2][i1]+ 电磁场->
a3[j3]*D3H2; 电磁场->memD3H1[k3][i2][i1] = 电磁场->b3[j3]*电磁场->
memD3H1[k3][i2][i1]+ 电磁场->a3[j3]*D3H1; D3H2 += 电磁场->
memD3H2[k3][i2][i1]; D3H1 += 电磁场->memD3H1[k3][i2][i1]; } ...
```

经过这一步，t 随后即可求得场的旋度。对于磁场的旋度，公式 la

$$\nabla \times \mathbf{H} = \left[\partial_y H_z - \partial_z H_y, \partial_z H_x - \partial_x H_z, \partial_x H_y - \partial_y H_x\right]^T$$

将转换为

```
... 电磁场->curlH1[i3][i2][i1] =
D2H3-D3H2; 电磁场->curlH2[i3][i2][i1] =
D3H1-D1H3; 电磁场->curlH3[i3][i2][i1] =
D1H2-D2H1; ...
```

上述代码实 现了 CPML区域和内部区域，并采用了一种优雅且兼容的框架。

3.5 时间积分

方程 (6) 表明，计算频域电磁场可能需要很多时间步。得益于离散时间傅里叶变换（DTFT），应在正演建模过程中实时积分频域数据。

$$u(\mathbf{x}, \omega) = \sum_{n=0}^{N_t - 1} u(\mathbf{x}, t_n) \exp(\mathrm{i}\omega' t_n), \tag{25}$$

其中，时间已离散化为 $t_n = n\Delta t$； $u \in \{E_x, E_y, E_z, H_x, H_y, H_z\}$ 是电磁场的一个分量； N_t 是所需的时间步总数。以下代码片段展示了在第 it 个时间步对分量 E_x 进行 DTFT 的示例：

```
void dtft_emf(emf_t *emf, int it, float ***E1, float_复杂 ****fwd_E1)
/*<电磁场（emf）的离散时间傅里叶变换>*/ { int i1,i2,i3,ifreq; float_
复因子; int i1min=电磁场->nb;//索引 i1 的下界 int i2min=电磁场->
nb;//索引 i2 的下界 int i3min=电磁场->nb;//索引 i3 的下界 int i1max=
电磁场->n1pad-1-电磁场->nb;//索引 i1 的上界 int i2max=电磁场->
n2pad-1-电磁场->nb;//索引 i2 的上界 int i3max=电磁场->n3pad-1-电磁
场->nb;//索引 i3 的上界 for(ifreq=0;ifreq<电磁场->nfreq; ++ifreq){
factor = 电磁场->expfactor[it][ifreq]; for(i3=i3min; i3<i3max; ++
i3){ for(i2=i2min;i2<i2max; ++i2){ for(i1=i1min; i1<i1max; ++i1){
fwd_E1[ifreq][i3][i2][i1] += E1[i3][i2][i1]*factor; } } } } }
```

其中，E1 和 fwd_E1 分别表示时域场和频域场。请注意，我们通过指定索引边界，仅在不含吸收边界的感兴趣区域内进行计算。对所有其他场分量，我们也采用相同的处理方式。

文献中已有大量工作致力于估算总建模时间 $T_{\max}$（Mittet，2010年；Wang 和 Hohmann，1993）。估算的关键在于确保频域电磁场达到稳态。实践中，为了稳妥起见，最终时间步数通常会估得偏多。因此，在我们的实现中，我们会定期检查频域场的收敛情况，直到

增加更多时间步后不再有贡献。如果收敛检查过于频繁，计算成本可能会很高。若要输出每个关注频率下每个网格点的格林函数，这一点尤其明显。在这里，我们基于审慎的物理直觉采用了一种简便方法。从物理上看，根据频率与趋肤深度之间的关系 $\delta = \sqrt{2/(\omega\mu\sigma)}$，可控源电磁法中最低频率分量的穿透深度最大。因此，使用计算区域边界处的最低频率，应足以监测场的演变。一场一旦收敛，模拟便会自动终止，从而避免额外计算。

3.6 介质均匀化

当介质中存在界面时，为了实现精确建模，需要额外处理较大的介质对比度。可通过对介质进行平均来实现这一点（Davydycheva 等人，2003）。尽管该方法能够处理完全各向异性，libEMM 仍将其用于 VTI 介质。该方法的核心思想是通过对网格单元尺寸取平均，计算水平电导率和垂向电阻率的有效介质参数

$$\bar{\sigma}_{xx}(x_{i+0.5}) = \frac{1}{x_{i+1}-x_i}\int_{x_i}^{x_{i+1}} \sigma_{xx}(x)\mathrm{d}x, \tag{26a}$$

$$\bar{\sigma}_{yy}(y_{j+0.5}) = \frac{1}{y_{j+1}-y_j}\int_{y_j}^{y_{j+1}} \sigma_{yy}(y)\mathrm{d}y, \tag{26b}$$

$$\bar{\sigma}_{zz}(z_{k+0.5}) = \left(\frac{1}{z_{k+1}-x_k}\int_{z_k}^{z_{k+1}} \sigma_{zz}^{-1}(z)\mathrm{d}z\right)^{-1}. \tag{26c}$$

上述过程基于这样一个事实：切向场分量将模型视为并联电阻的组合，而法向场分量则将模型视为串联电阻的组合。按照苏联文献中的惯例，这种方法通常称为均匀化，因为无论网格点位于均匀区域内还是界面邻域内，所有网格点都适用同一公式。

3.7 最优非均匀网格生成

我们的有限差分建模在直角网格上进行，该网格是多个一维非等距网格的张量积。生成这些一维网格时，采用几何级数法（Mulder，2006，附录 C）。假设总网格长度为 L，划分为 n 个区间（$n+1$ 个节点），公比为 $q>1$。将最小区间记为 $\Delta x = x_1 - x_0$。因此，L 与 Δx 之间的关系为

$$L = \Delta x(1 + q + \cdots + q^{n-1}) = \Delta x\frac{q^n - 1}{q - 1}. \tag{27}$$

由于稳定性要求以及建模中由此产生的计算成本，我们受到限制，只能在最小区间 Δx 上选取给定数量的节点，对一定距离进行离散化。由于方程 (27) 无法给出拉伸因子 q的显式表达式，问题归结为在固定的 n、Δx 和 L条件下，求最优的q 。式 (27) 中的关系等价于

$$q = \underbrace{\left(\frac{L}{\Delta x}(q-1)+1\right)^{\frac{1}{n}}}_{g(q)} \tag{28}$$

这启发我们执行若干次不动点迭代：

$$q^{k+1} = g(q^k), \quad k = 0, 1, \cdots. \tag{29}$$

由于对于所有 $q>1$，$|g'(q)|<1$ 均成立，因此这里提出的方案保证收敛。这一思路已在以下代码片段中实现。

float create_nugrid(int n, float len, float dx, float *x) /*<通过等比数列生成幂
律非均匀网格 >*/ {

  int i;

```
float q, qq; if(fabs(n*dx-len)<1e-15) { for(i=0; i<=n; i++) x[i] =
i*dx; return 1; } q = 1.1;//解的初始猜测 while(1){ qq =
pow(len*(q-1.)/dx + 1., 1./n); if(fabs(qq-q)<1e-15) break; q = qq; }
for(x[0]=0,i=1; i<=n; i++) x[i] = (pow(q,i) - 1.)*dx/(q-1.); return q; }
```

3.8 一致的命名与内存安全编程

由于三维CSEM建模涉及使用指针进行复杂的内存分配和初始化，libEMM 的每个计算模块都遵循相同的命名约定，以确保代码的内存安全。在整个软件中，名称为 xxxx_init() 和 xxxx_clos() 的所有例程分别用作构造函数和析构函数。这使得 C 例程 xxxx.c 与 C++类相似，并与 Fortran模块相对应。以下概述了基于时域有限差分的虚拟波域 CSEM建模主要步骤，如算法 1 所示:

- 指向电磁场数据结构的指针 emf 将通过 emf_init(emf) 进行初始化。需要计算内存的相关参数（例如 emf->rho11、emf->rho22 和 emf->rho 33)
  也将在 emf_init(emf) 中分配，以便读取输入模型。另一个名为 emf_close(emf) 的例程用于销毁之前分配的变量，并将与其一同放在源文件 emf_init_close.c 中。
- 遵循相同的约定，指向测量采集数据结构的指针 acqui 将通过 acqui_init() 进行初始化。此初始化会分配变量，并由不同的 MPI处理器读取输入文件。建模完成后，名为 acqui_close() 的例程将释放这些变量。这些内容都包含在源文件 acqui_init_close.c 中。
- 插值算子将在建模前通过 interpolation_init() 初始化，并在建模任务完成后通过 interpolation_close() 销毁。
- 计算模型必须扩展 CPML层和一些额外的缓冲层。这在 extend_model.c 中完成，通过 extend_model_init() 进行初始化，并通过 extend_model_close() 释放变量。
- 频域电磁场通过 dtft_emf_init() 初始化，并通过 dtft_emf_close() 释放。每个时间步都会在同一源文件 dtft_emf.c 中通过 dtft_emf() 对电磁场执行离散时间傅里叶变换。
- 空气波边界条件在 airwave_bc.c 中实现，通过 airwave_bc_init() 初始化，并在 airwave_bc_close() 中销毁。在每个时间步，分别通过 airwave_bc_update_E() 和 airwave_bc_update_H() 对电场和磁场进行空气波处理。
- 用于时域有限差分建模的时域电磁场在 fdtd_init() 中初始化，并在 fdtd_close() 中销毁。每个时间步的蛙跳时间步进都会调用 fdtd_curlH 计算 $\nabla \times \mathbf{H}$，调用 fdtd_update_E() 更新电场 $\mathbf{E}$，调用 fdtd_curlE 计算 $\nabla \times \mathbf{E}$，并调用 fdtd_update_H() 更新磁场 $\mathbf{H}$。这些例程集中在 fdtd.c 中。
- 正演模拟中电磁源的注入由 inject_electric_src_fwd() 和 inject_magnetic_src_fwd() 完成，位于 inject_src_fwd.c 中。
- 每经过 100 个时间步，都会通过 check_convergence() 检查收敛情况。

- 时域建模后，通过 compute_green_function() 计算格林函数。 .
- 最后通过 extract_emf() 提取 CSEM数据，并由 write_data() 写出。

Alg算法 1 虚拟波域中的基于FDTD的CSEM建模

1：emf_init(emf)；2：acqui_init(acqui,emf)；3：sanity_check(emf)；4：interpolation_init()；5：extend_model_init(emf)；6：dtft_emf_init(emf)；7：airwave_bc_init(emf)；8：fdtd_init(emf)；9：对 it $= 0, \cdots, N_t - 1$ 执行 10：fdtd_curlH(emf, it)；11：inject_electric_src_fwd()；12：fdtd_update_E(emf, it)；13：airwave_bc_update_E(emf)；14：dtft_emf(emf, it)；15：fdtd_curlE(emf, it)；；16：inject_magnetic_src_fwd()；17：fdtd_update_H(emf, it)；18：airwave_bc_update_H(emf)；19：如果 it%100 == 0，则 20：check_convergence(emf)；21：如果已收敛，则跳出；22：结束 if 23：结束 for 24：compute_green_function(emf)；25：extract_emf()；26：write_data(acqui, emf)；27：fdtd_init(emf)；28：airwave_bc_close(emf)；29：dtft_emf_close(emf)；30：extend_model_close(emf)；31：interpolation_close()；32：emf_close(emf)；33：acqui_close(acqui)；

为 CSEM 建模中使用的变量分配内存时，我们通过使用连续的内存块，动态分配由指针数组构成的指针数组……以构建多维数组，使其能够像 C 中的静态数组一样进行引用。也就是说，第（i, j, k）个元素可以直接通过 array[k] [j] [i] 引用。最右侧的索引 i 是变化最快的索引，而最左侧的索引 k 是变化最慢的索引。大小为 n1*n2*n3 的数组从第一个元素 array[0] [0] [0]，开始，到最后一个元素 array[n3-1] [n2-1] [n1-1] 结束。

4 GPU 多线程并行化

4.1 GPU 建模策略

经过十多年的发展，GPU 加速技术已日趋成熟。除了在地震学界的成功应用，例如逆时偏移和全波形反演（Yang 等，2014，2015）之外，GPU 在三维 CSEM 中的应用鲜有报道。本文重点介绍实现基于 GPU 的三维 CSEM 建模时的一些关键策略。

1. 将计算密集型部分放在 GPU 上执行，避免频繁的 CPU-GPU 数据传输。请注意，中央处理器侧重于低延迟，而 GPU 侧重于高吞吐量。为了避免大量通信延迟，我们设计代码，在移植到基于 GPU 的时间步进之前，尽可能完成所有必要的预计算

。预计算包括对输入参数进行有效性检查、在扩展计算域上准备介质，以及将震源和接收器位置处的插值权重制成查找表。空气-水边界条件直接通过 CUFFT 库在设备上计算。

2. 使用共享内存和分块算法（Micikevicius，2009）。其核心思想是利用共享内存的快速 L1 缓存；共享内存容量通常较小，但访问速度比全局内存快两个数量级。分块促使多个线程在每个执行阶段协同处理输入数据的一个子集。如下图 1 所示，我们将全局内存映射到二维共享内存块，并沿第三个维度滑动。GPU 上的数据将按块读取和写入，而非逐个元素处理，因此效率更高。由于有限差分模板中的同一数值需要在不同位置反复使用，采用共享内存的分块技术可以增强数据访问局部性。

3. 采用线性递增索引并连续访问内存。每个 CUDA 线程束包含 32 个处理核心，并以单指令多线程方式执行。当元素位于不同线程束中时，CUDA 内核会将操作串行化，从而导致性能大幅下降。因此，采用线性递增索引并连续访问内存，对设计高效的 CUDA 内核至关重要。

4. 改进的场收敛检查方案。只有当频域电磁场在每个网格点都收敛时，才能终止建模。这自然意味着每次进行收敛检查时都需要执行归约操作。为在 GPU 上高效完成这一操作，作者在（Yang，2021）中实现了并行归约方案。本文采用了更为审慎的方法，仅检查不含吸收边界的内部计算立方体的 8 个角点，如图 2 所示。这样便无需进行并行归约，并大幅减少所需的浮点运算次数。该方案同样适用于串行实现。

5. 使用映射页锁定内存。由于需要进行收敛检查，时间步进建模涉及设备（GPU）与主机（中央处理器）之间的通信，这违反了前述第一项原则。为避免这一问题，我们采用映射页锁定内存，使中央处理器和 GPU 能够共享缓冲区，而无需显式分配和复制缓冲区。众所周知，过度使用映射页锁定内存可能会影响性能。在此，仅需使用零拷贝页锁定内存备份复值电磁场的 8 个角点，从而最大限度地减少潜在的性能下降。

4.2 CUDA 实现

上述技术意味着，GPU 并行实现需要将算法 1 中的每个子程序大幅改写为不同的 CUDA 内核函数，而计算工作流保持不变，但运行在 GPU 设备上，而不是 CPU 主机上。在 GPU 模式下，必须先初始化设备，然后通过这些 CUDA 内核函数执行专用操作，并使用不同数量的多线程块和网格大小。cuda_modelling.cu 已完成这些工作，用于替代由 do_modeling.c 执行的串行建模。一些简单计算可能较难实现，但只需在 CPU 主机上执行一次即可轻松完成。这些操作包括扩展模型以设置吸收边界、设置 CPML层和 DTFD 系数、计算用于注入震源和提取接收器位置数据的插值权重；这些操作已在 cuda_modeling.cu 之前完成：GPU 将直接把它们复制到设备内存中，以便直接使用。这凸显了探索 CPU 与 GPU 计算各自独特优势的重要性。

所有相关的 CUDA 内核均位于 cuda_fdtd.cuh 中。具体而言，时域有限差分中的空间导数计算为 $\nabla \times \mathbf{E}$ 和 $\nabla \times \mathbf{H}$，已分别通过 GPU内核 cuda_fdtd_curlE_nugrid 和 cuda_fdtd_curlH_nugrid 实现并行化，以替代中央处理器例程 fdtd_curlE 和 fdtd_curlH。根据分块算法，前两个维度使用共享内存实现并行化，而第三个维度则按顺序步进，由此得到以下代码

```
__全局__空函数 cuda_时域有限差分_curlH_网格(...) { const inti1 =
threadIdx.x + blockDim.x*blockIdx.x; const inti2 =
threadIdx.y + blockDim.y*blockIdx.y; const intt1 =
threadIdx.x + 2;//t1>=2 && t1<BlockSize1+2const intt2 =
threadIdx.y + 2;//t2>=2 && t2<BlockSize2+2
```

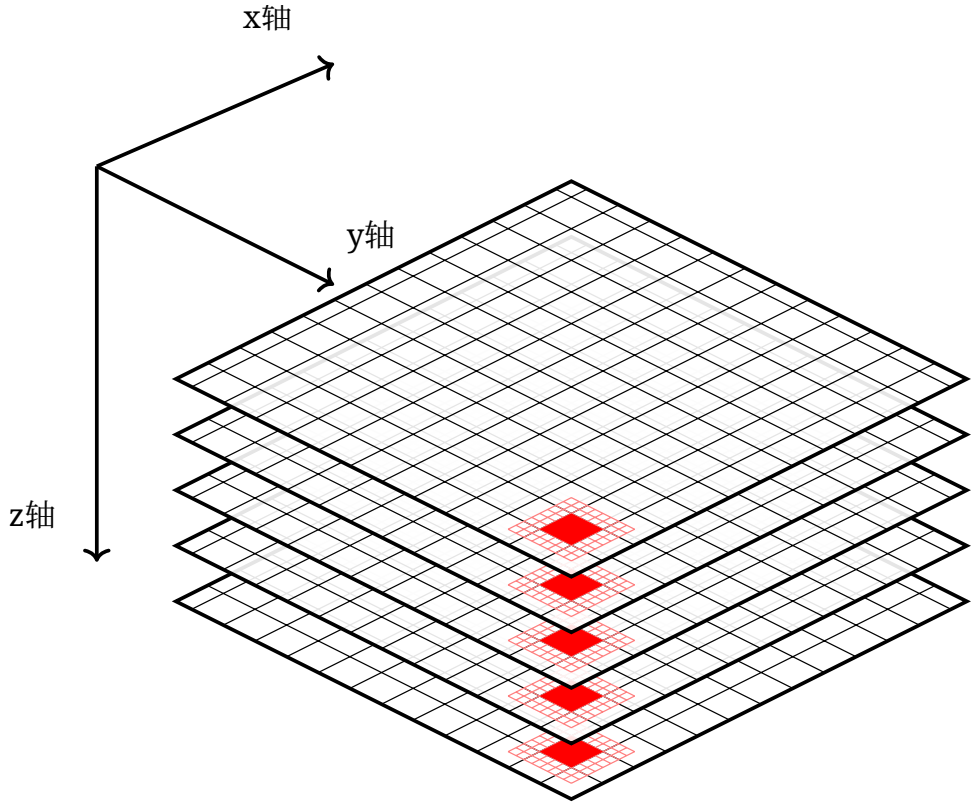

图1：沿z维度的x-y平面平铺块

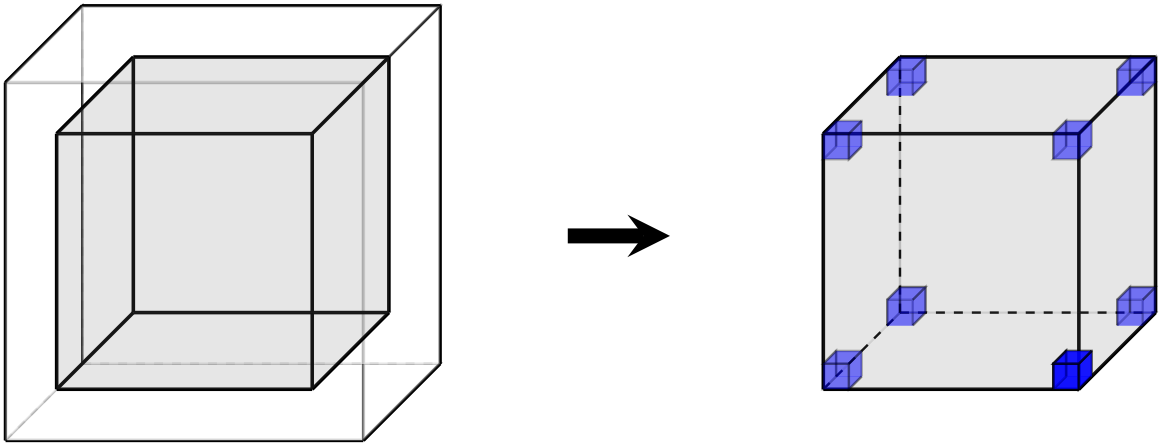

图2：不含吸收边界时，使用内部计算立方体八个角点进行收敛检查的示意图

__共享__浮点数 sH1[BlockSize2+3][BlockSize1+3];__共享__浮点数 sH2[BlockSize2+3][BlockSize1+3];__共享__浮点数 sH3[BlockSize2+3][BlockSize1+3];

int in_idx = i1+n1pad*i2; int out_idx = 0; int stride = n1pad*n2pad;

for(i3=i3min; i3<=i3max; i3++){//i3-2>=0 && i3+1<n3pad

....__syncthreads();

```
    if(validrw){ ... curlH1[out_idx] =
D2H3 - D3H2; curlH2[out_idx] =
D3H1 - D1H3; curlH3[out_idx] =
D1H2 - D2H1; } }//结束 for i3 }
```

请注意，在进入下一阶段之前同步 GPU线程非常重要。为提高效率，**E** 和**H** 场的时间步进也遵循分块映射，即：

```
__全局__无返回值 cuda_时域有限差分_更新_E(...) { const int
i1 = threadIdx.x + blockDim.x*blockIdx.x; const int i
2 = threadIdx.y + blockDim.y*blockIdx.y; int i3, id;
for(i3=0;i3<n3pad; i3++){ if(i1<n1pad&& i2<n2pad){ id =i1+
n1pad*(i2+ n2pad*i3); E1[id] +=dt*inveps11[id]*curlH1[
id]; E2[id] +=dt*inveps22[id]*curlH2[id]; E3[id] +=
dt*inveps33[id]*curlH3[id]; } } }
```

通过 cuda_check_convergence 在 GPU 上直接检查场的收敛情况。由于收敛检查的结果必须在主机上可见，因此我们使用 cudaHostAlloc 而非常用的 cudaMalloc 例程来分配内存，以记录三维立方体中已收敛角点的数量。相应地，必须使用 cudaFreeHost 而不是 cudaFree 来释放固定内存。具体来说，这些对应于以下代码行：

```
cudaHostAlloc(&h_ncorner, sizeof(int), cudaHostAllocMapped);
```

and

```
cudaFreeHost(h_ncorner);
```

5 软件特性

5.1 可执行文件构建依赖项

libEMM 是一款轻量但功能强大的三维CSEM建模工具包，可在单个 CPU处理器、多个 MPI 并行进程以及支持CUDA的GPU模式下运行。这意味着该软件具有以下依赖项：

- 用于快速傅里叶变换的 FFTW；
- 用于在分布式内存架构上实现多处理器并行化的 MPICH（或其他类型的 MPI）；
- 如果希望利用设备共享内存上的 GPU 并行化，则需要 Nvidia CUDA 软件包。

CUDA 软件包的安装是可选的，而前两个软件包则是必需的。libEMM 使用 C 编程语言实现高性能计算。要编译代码，可根据上述软件包的安装路径编辑 Makefile，然后运行 make 命令。编译生成的可执行文件名为 fdtd，并放置在 bin 文件夹中。

5.2 参数解析

libEMM 能够根据令牌中指定的参数自动解析参数列表。每个关键字的值在等号 = 后读取。同一关键字可以通过逗号分隔的参数指定多个值。不同关键字之间以空格分隔。

5.3 测量数据采集

要指定测量布局，必须提供三个 ASCII 格式的采集文件，用于规定震源和接收器位置及其连接表。这些文件将分别赋值给以下参数：

fsrc=sources.txt frec=receivers.txt fsrcrec=src_rec_table.txt

其中，sources.txt 文件给出震源位置；receivers.txt 文件给出接收器位置；src_rec_table.txt 文件建立震源与接收器之间的连接。使用 MPI 运行代码时，每个进程都会读取这些文件，然后根据其进程索引（该索引唯一确定一个震源）执行独立的建模任务。震源/接收器文件的格式如下：

```
      x      y       z        方位角       dip         iTx/iRx
-8000.0    0.0    275.0         0          0            1
-7975.0    0.0    275.0         0          0            2
-7950.0    0.0    275.0         0          0            3
-7925.0    0.0    275.0         0          0            4
    ...
```

以上 6 列分别对应 (x, y, z) 坐标（单位：米）、以弧度为单位的方位角和倾角，以及源/接收器的索引。借助通过求逆范德蒙德矩阵计算出的插值算子，无需将源和接收器放置在网格节点上，即可在任意位置准确地注入/提取信号。连接表如下：

```
 iTx iRx
 1     1
 1     2
 1     3
 ...

 2     1
 2     2
 2     3
 ...
```

该表将每个由 iTx 索引的源/发射器与若干个由 iRx 索引的接收器关联起来。

5.4 物理域、网格尺寸和坐标

libEMM 在一个三维立方体中进行可控源电磁法建模，其物理尺寸由边界指定。在包含 $X_1 = X_2 = [-10000, 10000]$ 米、 $X_3 = [0, 3000]$ 的区域 $X = X_1 \times X_2 \times X_3$ 上进行建模时，读取内容如下：

x1最小值=-10000 x1最大值=10000 x2最小值=-10000 x2最大值=10000x3最小值=0 x3最大值=3000

参数 n1、n2 和 n3 指定模型在 x、y 和 z 方向上的维度，而 d1、d2 和 d3 指定沿每个坐标的最小网格间距（单位为米）。要在由101^3 网格点组成的电阻率立方体中进行建模，在均匀网格上使用 $\Delta x = \Delta y = 200$ 米、 $\Delta z = 50$ 米，必须在参数列表中写入

n1=101n2=101 n3=101 d1=200 d2=200 d3=50

这些参数将由 libEMM 检查，以确保与区域边界相匹配。

由于 libEMM 支持在均匀网格和非均匀网格上进行 CSEM建模，因此若要沿 z 方向在非均匀网格上建模，必须为对应 z 方向网格划分的参数 fx3nu 提供二进制格式的网格坐标。请注意，该文件必须恰好存储 n3 个浮点数。由于会降低建模精度，x 方向和 y 方向的非均匀网格划分已被禁止（Yang 和 Mittet，2023）。在均匀网格上建模时，无需为 fx3nu 提供输入。

5.5 电阻率文件

数值建模所需的参数列表中必须包含三个二进制格式的电阻率文件，具体如下:

frho11=rho11 frho22=rho22 frho33=rho33

其中，文件 rho11、rho22 和 rho33 分别表示均匀化后的电阻率，数据尺寸为 n1*n2*n3，并分别对应 x、y 和 z 方向上的半个网格偏移。这些文件应使用建模内核之外的模型构建工具生成。

5.6 源通道、接收通道与频率

libEMM 通过关键字 - chsrc 和 chrec 指定活动源通道和接收通道。例如，记录电流源 J_x 的 E_x 和 E_y 分量的 CSEM数据时，写作

ch源=Ex ch接收=Ex,Ey

同样，可以将模拟频率指定为

freqs=0.25,0.75,1.25

其中 0.25 赫兹 、0.75 赫兹和 1.25 赫兹将是从时域可控源电磁法建模
ng 引擎。

5.7 算子长度、CPML层和缓冲区

有限差分算子的半长由参数 rd 决定。在 CPU 模式下，可以选择 rd=1、2 或 3，但 GPU 仅支持 rd=2 。这是因为从二阶格式切换到四阶格式时，建模精度会显著提升；但从四阶格式进一步提高阶数需要更多计算量，却不会带来显著的精度提升（Yang 和 Mittet，2023）。

我们在立方体的每一侧都填充相同数量的 CPML 层，该数量由参数 nb 指定。此外，我们添加 ne 层缓冲区，作为 CPML 与内部区域之间的平滑过渡区。选择 ne 的一般原则是确保缓冲层数大于有限差分模板的半长，例如，ne $=$ rd $+5$。

5.8 CSEM建模输出

运行三维CSEM建模后，结果将输出为 ASCII 文件，文件名为 emf_XXXX.txt，其中 XXXX 是源/发射器的索引。使用 4 个 MPI 进程运行模拟会产生

电磁场_0001.txt 电磁场_0002.txt 电磁场_0003.txt 电磁场_0004.txt

每个文件都将遵循相同的约定，输出发射器索引 iTx、接收器索引 iRx、接收器通道 Ex/Ey/Ez/Hx/Hy/Hz、频率索引 ifreq，以及频域复数电磁场的实部和虚部:

发射机编号 接收机编号 通道记录 频率 电磁场_实部 电磁场_虚
部 1 1 Ex 1 -1.086630e-14 2.026699e-16 1 2 Ex 1 -
1.148639e-14 3.976523e-16 1 3 Ex 1 -1.213745e-14
6.254443e-16 1 4 Ex 1 -1.282039e-14 8.894743e-16 1 5 Ex
1 -1.353619e-14 1.192894e-15 1 6 Ex 1 -1.428555e-14
1.539961e-15 1 7 Ex 1 -1.506933e-14 1.934350e-15 1 8 Ex
1 -1.588794e-14 2.380965e-15 1 9 Ex 1 -1.674208e-14
2.884079e-15 1 10 Ex 1 -1.763176e-14 3.449250e-15 1 11
Ex 1 -1.855739e-14 4.081327e-15

. . .

5.9 运行 indifferentmodes

要运行三维CSEM建模，可以将参数写入 Shell 脚本 run.sh 中，并使用 bash run.sh 启动代码。下面是一个示例 Shell 脚本：该脚本使用 25 个 MPI 进程运行，每个 MPI 进程通过 4 个 OpenMP线程进行并行化：

```
#!/usr/bin/bash

n1=101
n2=101
n3=101
d1=200
d2=200
d3=50

export OMP_NUM_THREADS=4
mpirun -n 25 ../bin/fdtd \ mode=0 \
fsrc=sources.txt \ frec=receivers.txt \
fsrcrec=src_rec_table.txt \ frho11=
rho11 \ frho22=rho22 \ frho33=r
ho33 \ chsrc=Ex \ chrec=Ex \ x1min=-
10000 x1max=10000 \ x2min=-10000
x2max=10000 \ x3min=0 x3max
=5000 \ n1=$n1 n2=$n2 n3=$n3 \ d1=$
d1 d2=$d2 d3=$d3 \ nb=12 ne=6\
freqs=0.25,0.75,1.25 \ rd=2
```

当采集中有 25 个震源时，也可以针对某些特定震源进行建模，例如，

```
mpirun-n2../bin/时域有限差分 shots=12,14 ...
```

这将只模拟发射机-12 和发射机-14 的 CSEM数据。输出的可控源电磁响应将命名为 emf_0012.txt 和 emf_0014.txt。如果通过 mpirun 启动的进程数超过 sources.txt 文件中给出的源总数，程序将终止。

使用 nvcc 编译器可通过命令 ‘make GPU=1’ 编译代码，以进行 GPU 建模。假设笔记本电脑上只有一张 GPU 卡。编译后生成可执行文件，即可通过执行带有适当参数的命令直接运行代码：

```
../bin/fdtd ...
```

在脚本中添加 nvprof，可以方便地分析不同 CUDA 内核所耗费的计算时间。

```
nvprof --log-file profiling.txt ../bin/fdtd ...
```

性能分析日志将记录在 profiling.txt 中。

6 应用示例

该软件支持在均匀网格（使用默认标志 emf->nugrid=0）和非均匀网格（emf->nugrid=1）上进行建模。我们首先使用具有一维结构的层状模型，此时采用均匀网格即可满足需求。针对需要在计算域的部分区域进行密集采样的实际应用，开发了基于非均匀网格的高阶有限差分时域法，用于

模拟三维CSEM响应。第二个示例包含变化的海底地形，因此可作为示例说明。本实验考察了三个在实际应用中具有代表性的频率，即 0.25 赫兹、0.75 赫兹和 1.25 赫兹。时间间隔和时间步数会自动选取（Mittet, 2010年），且不会违反稳定性条件。相对振幅误差通过 $|E_x^{FD}|/|E_x^{ref}|-1$ 测量。建模精确时，该误差应接近 0；如果 $|E_x^{FD}|>|E_x^{ref}|$，则为正；如果正值，则为负。相位误差通过 $\angle E_x^{FD}-\angle E_x^{ref}$ 以度为单位测量。

6.1 层状模型

第一个测试采用图 3 所示的层状电阻率模型：该模型包含 5 层，分别为空气层（ $10^{12}\,\Omega\cdot$m）、水层（ 0.3125 Ω ·m）和两个地层。在两个地层之间有一个电阻率为 100 Ω·m 的高阻层，用于模拟含烃地层。现在将三维 FDTD 代码与 Dipole1D 程序（Key，2009）计算的半解析解进行交叉验证。我们对尺寸为 10 km × 10 km × 5 km 的模型进行离散化，其中在 1250–1350 m 深度处包含一个高阻层，网格间距为 $\Delta x=\Delta y=100$ m、$\Delta z=50$m。由此得到一个规模为 101^3 的大型三维网格。点源位于海床上方 50 m 处。图 4a 和 4b 显示，FDTD 方法与半解析解在 E_x 分量的振幅和相位方面吻合良好。图 4c 和 4d 显示，在大多数偏移距处，振幅误差小于 1.5%，相位误差小于 1° 。近偏移距处的误差非常大。幸运的是，近偏移距数据与实际 CSEM 成像无关。此示例证实了 FDTD 方法具有良好的精度。

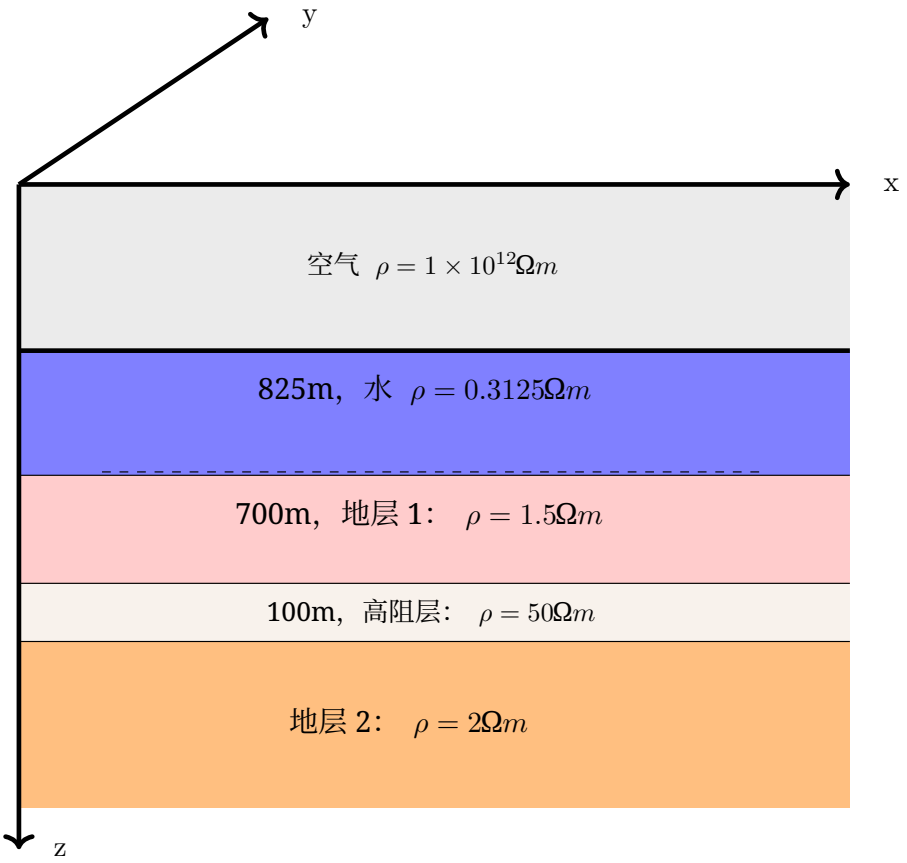

图 3：电阻率模型的剖面图：虚线表示源的深度。

6.2 含海底地形的模型

为模拟实际的海洋可控源电磁法环境，我们使用一个包含海底地形的二维结构三维电阻率模型进行可控源电磁法建模，如图5所示。为捕捉海底地形的影响，我们在海床附近对模型进行密集网格剖分，同时沿下方深度方向逐渐拉伸网格。网格的x-z剖面如图6所示。基于该模型，我们使用MARE2DEM计算参考解

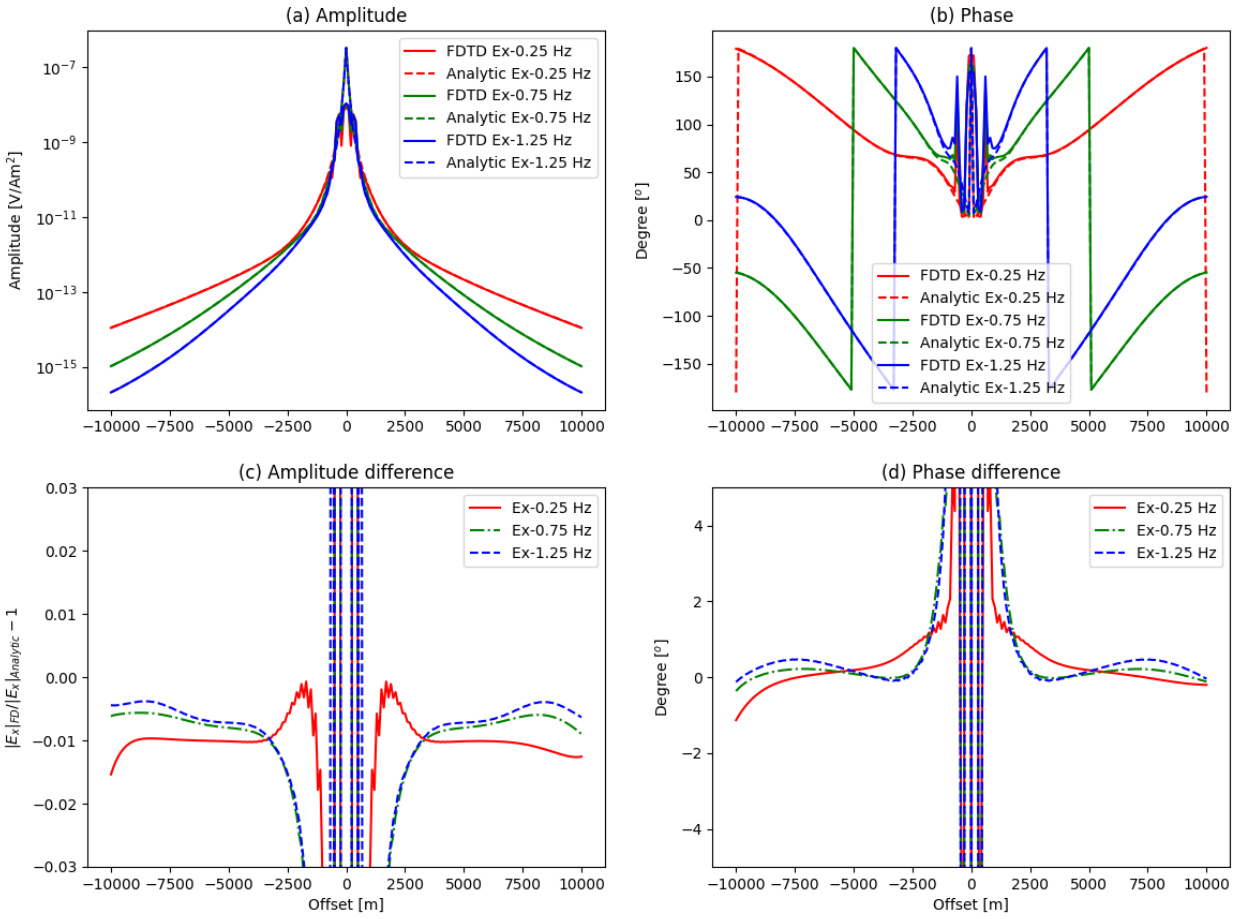

图4：浅水场景下三维CSEM模拟中时域有限差分与解析解的比较。(a) 振幅；(b) 相位；(c) 振幅误差；(d) 相位误差。

（Key，2016年）有限元法。模型向各个方向延伸数十千米，以模拟电磁场传播到很远距离的情况，同时避免可能的边缘/边界效应。感兴趣区域采用Delaunay三角剖分进行密集网格划分，而远离模型内部的区域则使用较大的单元。在我们的有限差分建模中，PML边界条件在数十个网格范围内衰减计算域中的人工反射，以实现相同的行为。正如图7所示，FDTD建模与MARE2DEM解之间的振幅和相位均非常吻合。图7c显示，在相关偏移距处的所有频率下，振幅误差均控制在3%以内；当远偏移距和较高频率接近噪声水平（10^{-15} V/m^2）时，振幅误差会变大。

使用多个MPI进程启动libEMM时，我们可以在所有接收器均处于活动状态的情况下，同时对与不同发射器相关的CSEM数据进行建模。这模拟了实际海洋可控源电磁法采集中的情况。对于图8所示的测量布局，libEMM可以帮助我们获取内联数据和横向数据，如图9所示。

7 性能分析：OpenMP与GPU的比较

使用前述规模为 101^3, 的层状电阻率模型，我们通过比较使用GPU和中央处理器进行相同模拟的运行时间来分析计算效率。我们的GPU卡为Nvidia Geforce GTX860M，计算能力为5.0，配备2048 MB GDDR5内存，运行频率为797 MHz。我们将其与Intel(R) Core(TM) i7-4710HQ CPU @ 2.50GHz进行比较。C代码经过高度优化，并使用不同数量的OpenMP线程进行并行化。

在 GPU 模式下，CPU时钟记录的性能分析信息会产生误导。此时，可以使用 nvprof 以较高的时间分辨率分析每个GPU内核的运行时间。CUDA事件

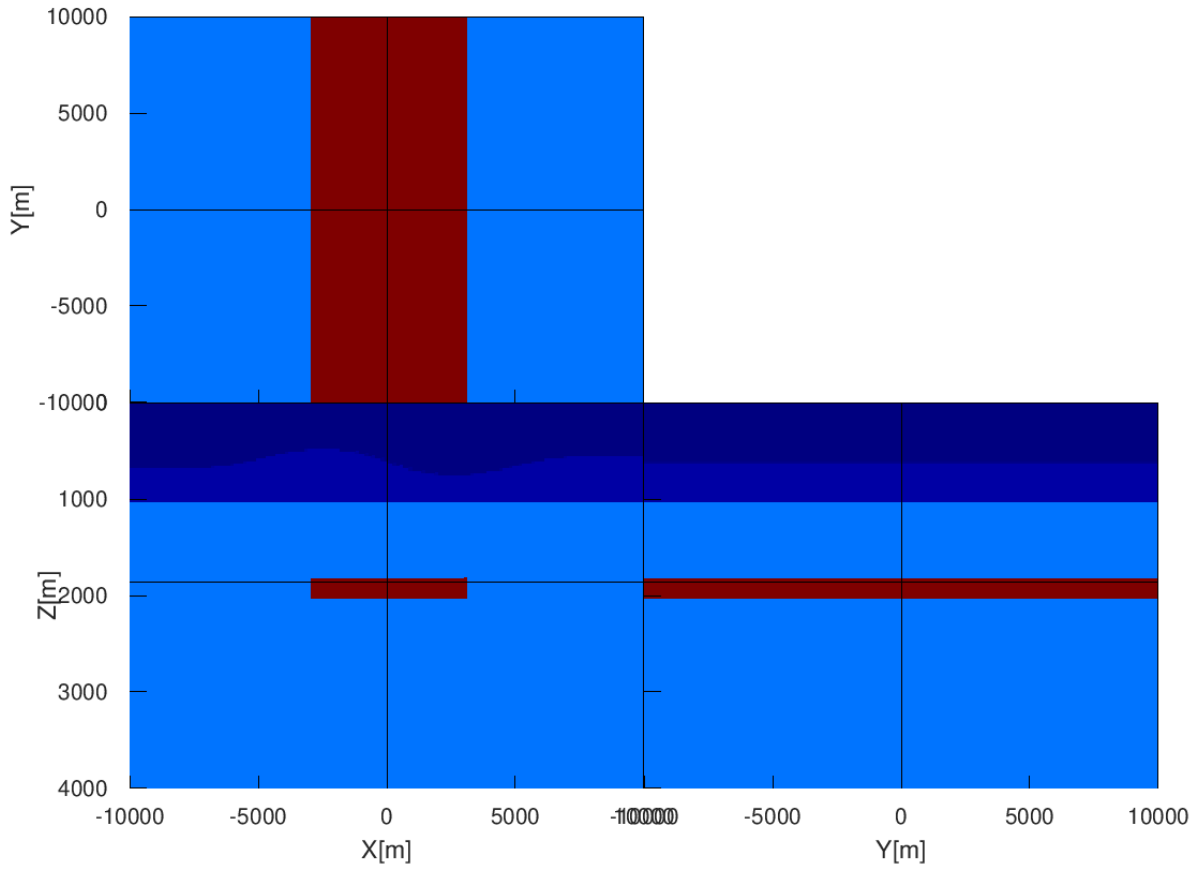

图5：具有海底地形的海洋可控源电磁法电导率模型

用作同步标记，以准确测量总耗时。

图 10 展示了不同 GPU 内核的性能分析日志截图。表 1 汇总了不同测试中各个阶段计算的 CPU运行时间。由于存在高电阻层，时域建模不得不采用非常小的时间采样。三维建模在完成 3000 多个时间步后结束，在 GPU 上仅需约 53 秒。为了更好地体现 GPU 相较于多线程CPU所带来的加速比，我们在图 11a 中展示了总运行时间。GPU 代码相较于单线程 CPU 的加速比为 13.7，相较于由 8 个 OpenMP线程并行化的 CPU 则为 8.3。可以看出，随着 OpenMP线程数量增加，代码在 CPU 上的性能并未呈线性扩展。图 11b 以饼图展示了全局 GPU建模过程各计算阶段所占的比例。计算量最大的操作包括计算 $\nabla \times \mathbf{E}$和 $\nabla \times \mathbf{H}$、更新场 $\mathbf{E}$ 和 $\mathbf{H}$、应用空气波边界条件以及离散时间傅里叶变换。震源注入和收敛检查所耗费的时间可以忽略不计。

8 讨论

我们强调，不同的介质均匀化方法可能会产生不同程度的准确性。本文采用了（Davydycheva 等人，2003）中的均匀化方法。界面也可以使用抗混叠带限阶跃函数（Mittet，2021）进行处理，这种方法同样可以达到很高的准确性（Yang 和 Mittet，2023）。libEMM 未采用阶跃函数方法，因为在三维非均匀介质中进行建模时，需要对二维界面进行连续描述，而这在实践中通常很难实现。

除了准确性，效率也是另一个重要考量。作者报告称，与单线程 GPU 相比，使用 GPU 可获得 40 倍的加速比（Yang，2021）。在 C代码中应用多项有效优化，尤其是取消对网格逐点进行收敛检查后，加速比有所降低，但仍然显著。需要注意的是，我们使用的是 2014 年购置的 GPU 卡，可能已经相当

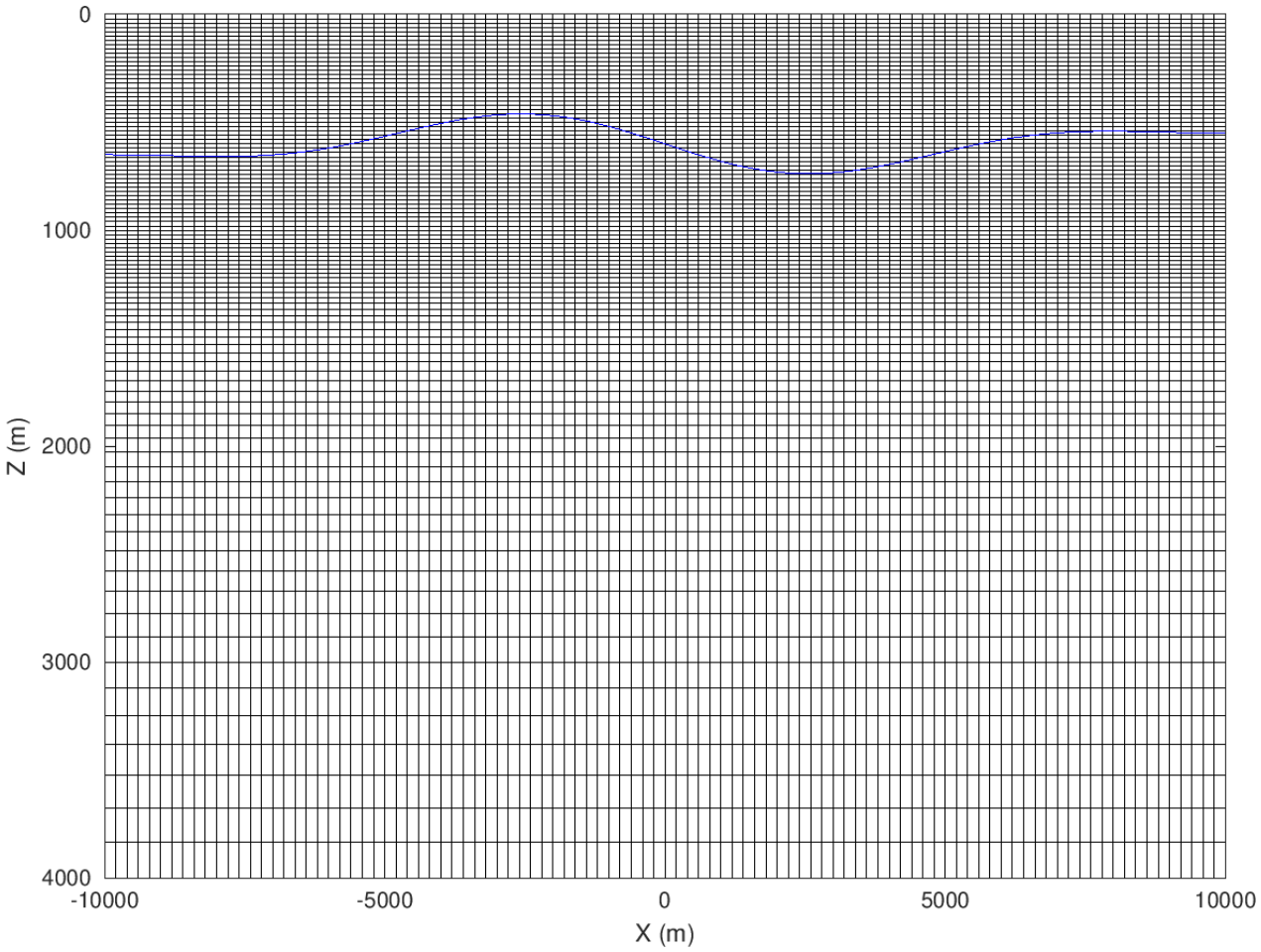

图6：用于网格化包含水深地形的电阻率的非均匀网格

过时。使用最新硬件时，预期可获得更令人瞩目的加速比。不过，发布该软件更有价值，因为这能让用户进一步提升性能，而不是只关注特定硬件上的某个具体数值。

当前版本的 libEMM 存在一些局限性。libEMM 还实现了有限长度扩展源的建模，即沿天线长度均匀分布源电流。然而，该实现的准确性仍然未知。FDTD方法能够处理完全各向异性介质的数值模拟。这些显然意味着其在实际问题中的重要应用，但目前尚未在 libEMM 中实现。用于构建一维和二维结构三维模型的简单工具已经提供，但已从建模功能中拆分出来，作为一个独立部分。用户可以使用其他先进且功能强大的工具构建包含地质结构的复杂三维模型，然后利用此前提供的工具在非均匀网格上进行介质均匀化，再开始建模任务。

9 结论

本文介绍了一款轻量而强大的软件 libEMM，用于在虚拟波域中进行三维CSEM建模。它同时包含 CPU实现和 GPU 并行化。因此，libEMM 不仅能让用户受益于多核并行集群，还能在 GPU 资源可用时借助图形卡提升效率。文中详细介绍了软件特性，便于用户使用，并通过应用示例展示了性能分析。未来将把这一实现部署到集群上，用于三维 CSEM 反演，从而高效处理大规模应用。

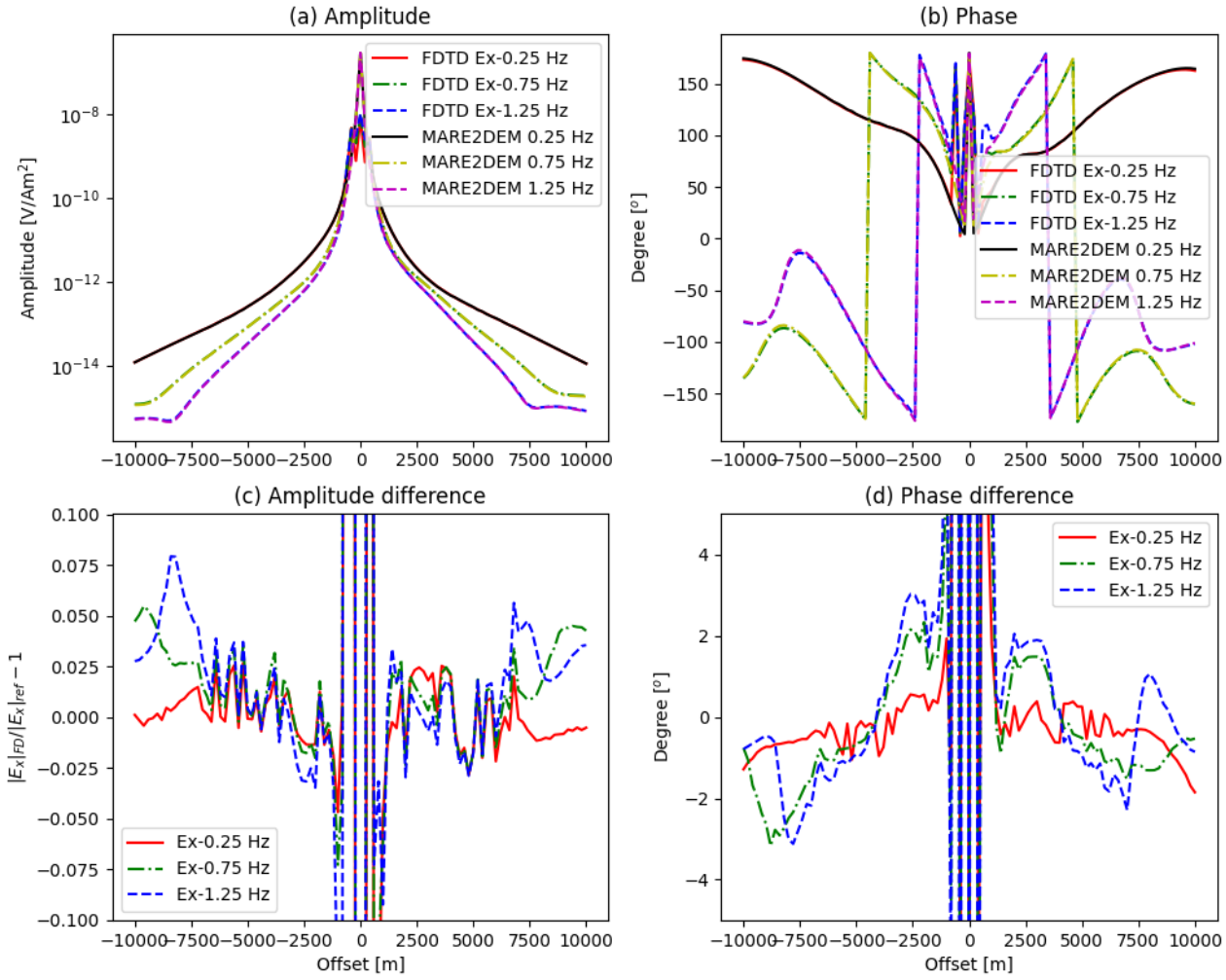

图7：时域有限差分与 MARE2DEM 模拟的可控源电磁响应对比

致谢

Pengliang Yang 获得了中央高校基本科研业务费（AUGA5710010121）和中国国家自然科学基金（42274156）的资助。作者感谢 Rune Mittet 和 Daniel Shantsev 进行了许多富有成效的讨论。作者也感谢一位匿名审稿人提出的关于色散误差分析的深刻问题。libEMM 可通过 GitHub代码仓库获取：https://github.com/yangpl/libEMM。libEMM 的许多易用特性和开发理念受到知名计算地球物理学开源软件 Seismic Unix 和 Madagascar 的启发。

A 色散分析

下面对不含震源项的波动域数值色散进行一些理论分析。对第一个子方程应用旋度算子，然后将其代入方程（3）中的第二个子方程，消去电场，得到

$$(\mu\epsilon\partial_{tt} + \nabla\times\nabla\times)\mathbf{H}' = (\mu\epsilon\partial_{tt} - \Delta)\mathbf{H}' = 0, \tag{30}$$

其中 $\Delta = \nabla\cdot\nabla$；第二个等号是由 $\nabla\cdot\mathbf{H} = 0$ 以及以下事实得出的

$$\nabla\times\nabla\times\mathbf{H}' = \nabla\nabla\cdot\mathbf{H}' - \nabla\cdot\nabla\mathbf{H}'. \tag{31}$$

考虑网格上电磁场的时间谐波平面波表示

$$\mathbf{H}' \propto e^{-\mathrm{i}(\omega t - k_x x - k_y y - k_z z)}, \tag{32}$$

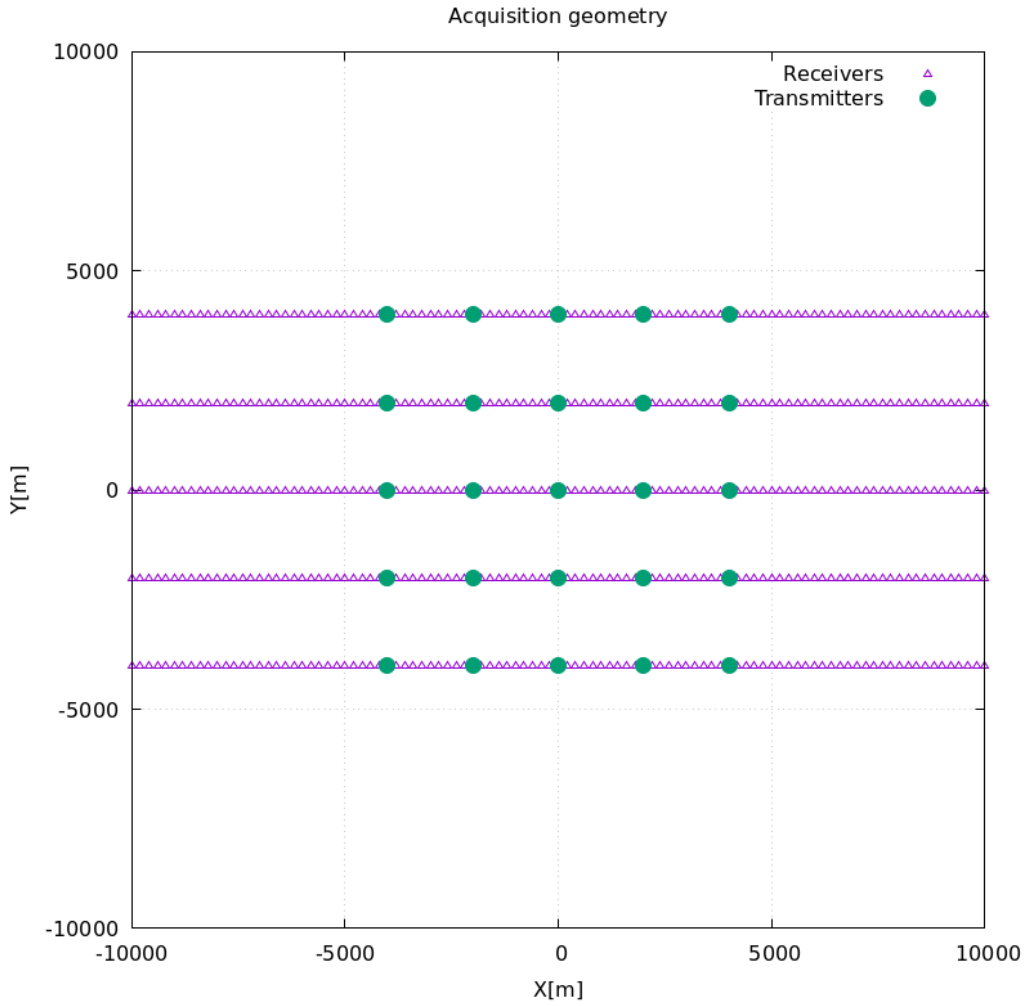

图8：测量布置图

其中， k_x、 k_y 和 k_z 分别是 x、y 和 z 方向的波数。在网格间距均为 Δx、Δy 和 Δz 的情况下，网格可直接由索引指定： $x_i = i\Delta x$、 $y_i = i\Delta y$ 和 $z_i = i\Delta z$。有限差分模板的 $2r$ 系数与网格位置无关，满足

$$c_i^+ = -c_{-i+1}^+ = c_i^- = -c_{-i+1}^- := \frac{c_i}{\Delta x}, \quad i = 1, \cdots, r \tag{33}$$

其中， c_i 是标准有限差分系数，可使用 (Fornberg, 1998) 中的方法轻松求得。离散空间导数变为

$$D_x^+ = D_x^- = \frac{1}{\Delta x}\sum_{i=1}^{r} c_i (e^{\mathrm{i}\frac{(2i-1)k_x\Delta x}{2}} - e^{-\mathrm{i}\frac{(2i-1)k_x\Delta x}{2}}) = \frac{2\mathrm{i}}{\Delta x}\sum_{i=1}^{r} c_i \sin((i-\frac{1}{2})k_x\Delta x), \tag{34}$$

从而得到离散拉普拉斯算子

$$\begin{aligned}\Delta \approx & D_x^+ D_x^- + D_y^+ D_y^- + D_z^+ D_z^- \\ = & -\frac{(2\sum_{i=1}^{r} c_i \sin((i-\frac{1}{2})k_x\Delta x))^2}{\Delta x^2} - \frac{(2\sum_{i=1}^{r} c_i \sin((i-\frac{1}{2})k_y\Delta y))^2}{\Delta y^2} - \frac{(2\sum_{i=1}^{r} c_i \sin((i-\frac{1}{2})k_z\Delta z))^2}{\Delta z^2}\end{aligned} \tag{35}$$

时间导数采用二阶蛙跳格式进行近似，因此

$$\partial_{tt} \approx D_t^+ D_t^- = -\left(\frac{2\sin(\frac{\omega\Delta t}{2})}{\Delta t}\right)^2. \tag{36}$$

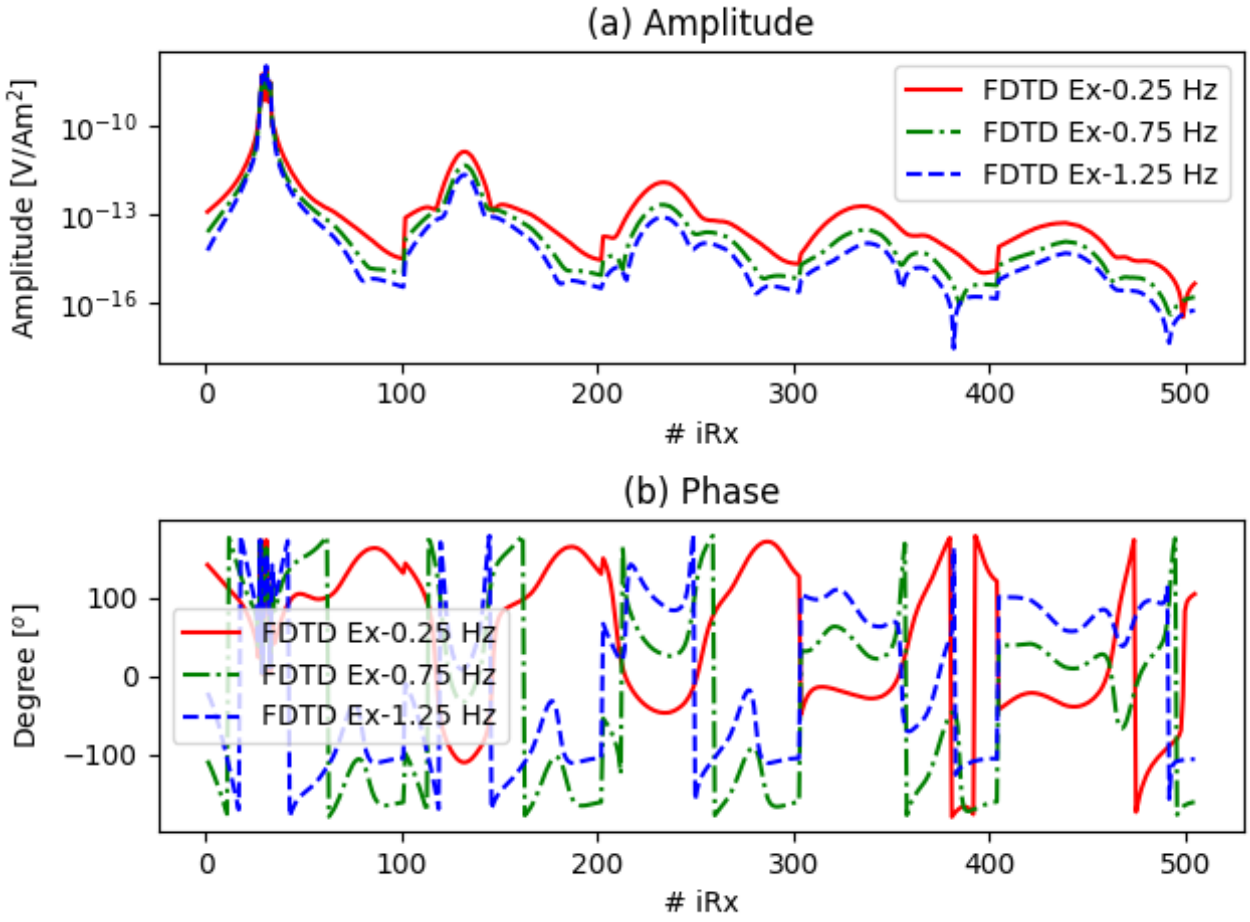

图9：使用部署在 (x,y,z)=(-4000 m, -4000 m, 550 m) 处的发射机建模的 CSEM数据（包括内联数据和横向数据）的振幅和相位。

Time(%)	Time	Calls	Avg	Min	Max	Name
21.47%	11.5934s	3301	3.5121ms	3.4560ms	4.8247ms	cuda_fdtd_curlE
21.00%	11.3392s	3301	3.4351ms	3.3582ms	5.4086ms	cuda_fdtd_curlH
14.95%	8.07114s	3301	2.4451ms	2.2912ms	3.7809ms	cuda_fdtd_update_E
11.48%	6.19826s	3301	1.8777ms	1.7759ms	3.1461ms	cuda_dtft_emf
11.01%	5.94660s	3301	1.8015ms	1.7170ms	2.7286ms	cuda_fdtd_update_H
7.16%	3.86615s	29709	130.13us	102.79us	985.57us	void composite_2way_fft
5.11%	2.75920s	29709	92.874us	80.576us	1.0691ms	void composite_2way_fft
2.89%	1.56005s	13204	118.15us	111.11us	1.1795ms	cuda_airwave_bc_scale_FH
1.88%	1.01344s	13204	76.752us	73.376us	485.99us	[CUDA memcpy DtoD]
1.20%	650.54ms	6602	98.536us	92.512us	684.90us	cuda_airwave_bc_scale_FE
0.69%	374.63ms	9903	37.830us	36.416us	903.78us	cuda_airwave_bc_copy
0.66%	358.98ms	19806	18.124us	4.6080us	492.39us	cuda_airwave_bc_back2emf
0.27%	146.75ms	6	24.458ms	21.449ms	27.399ms	[CUDA memcpy DtoH]
0.14%	77.052ms	3301	23.342us	22.944us	26.016us	cuda_inject_electric_source
0.03%	13.997ms	3301	4.2400us	3.7440us	8.2880us	cuda_inject_magnetic_source
0.02%	10.932ms	27	404.90us	1.1840us	2.8186ms	[CUDA memcpy HtoD]
0.01%	7.1433ms	31	230.43us	2.3680us	852.71us	[CUDA memset]
0.00%	153.67us	34	4.5190us	3.9040us	5.1520us	cuda_check_convergence

图10：GPU活动性能分析日志的屏幕截图

公式（30）得出

$$\frac{\sin^2(\frac{\omega\Delta t}{2})}{v^2\Delta t^2} = \frac{(\sum_{i=1}^{r} c_i \sin((i-\frac{1}{2})k_x\Delta x))^2}{\Delta x^2} + \frac{(\sum_{i=1}^{r} c_i \sin((i-\frac{1}{2})k_y\Delta y))^2}{\Delta y^2} + \frac{(\sum_{i=1}^{r} c_i \sin((i-\frac{1}{2})k_z\Delta z))^2}{\Delta z^2}. \quad (37)$$

表 1：使用中央处理器和不同数量 OpenMP线程时的运行时间（秒）

计算	omp=1	omp=2	omp=4	omp=8
$\nabla\times\mathbf{E}$	2.07e+02	1.39e+02	1.05e+02	8.56e+01
注入 $\mathbf{H}$	1.12e-03	1.73e-03	1.48e-03	2.10e-03
更新 $\mathbf{H}$	4.18e+01	3.36e+01	2.91e+01	2.66e+01
$\nabla\times\mathbf{H}$	1.88e+02	1.22e+02	9.21e+01	7.65e+01
注入 $\mathbf{E}$	3.94e-03	5.28e-03	4.72e-03	6.07e-03
更新 $\mathbf{E}$	5.49e+01	4.23e+01	3.66e+01	3.41e+01
空中波	2.06e+02	1.91e+02	2.03e+02	1.93e+02
DTFT	3.07e+01	2.71e+01	2.58e+01	2.62e+01
收敛	5.57e-04	8.01e-04	8.97e-04	2.35e-03
总计	7.30e+02	5.56e+02	4.93e+02	4.42e+02

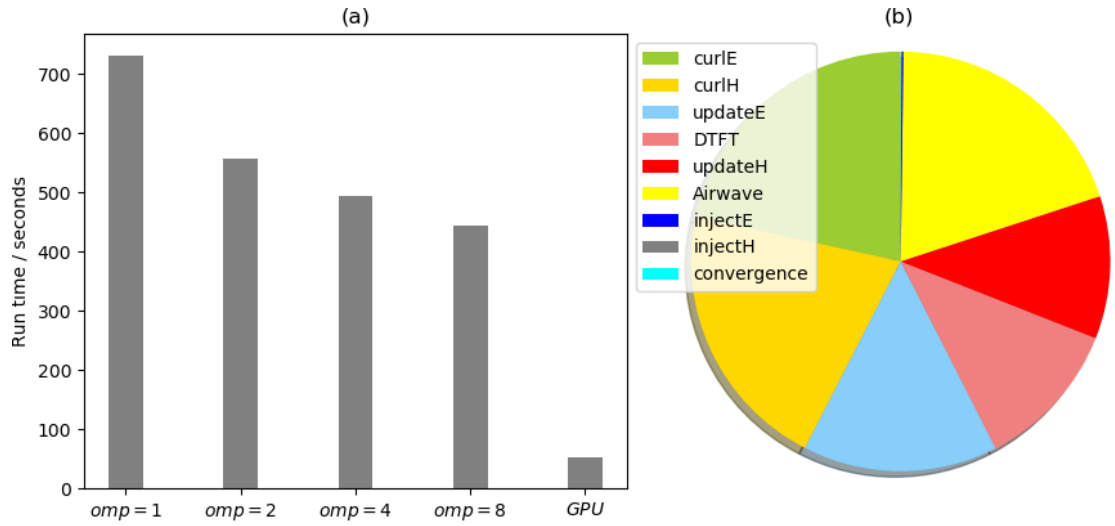

图 11：(a) 不同 OpenMP 线程数下中央处理器与 GPU 的总运行时间比较；(b) GPU 上各计算阶段所占的百分比

记

$$S := v\Delta t\sqrt{\frac{(\sum_{i=1}^{r} c_i \sin((i-\frac{1}{2})k_x\Delta x))^2}{\Delta x^2}+\frac{(\sum_{i=1}^{r} c_i \sin((i-\frac{1}{2})k_y\Delta y))^2}{\Delta y^2}+\frac{(\sum_{i=1}^{r} c_i \sin((i-\frac{1}{2})k_z\Delta z))^2}{\Delta z^2}} \tag{38}$$

以及波数的大小 $k := \sqrt{k_x^2+k_y^2+k_z^2}$。根据 $k = \omega/v$，方程 (37) 表明，FDTD 仿真中的数值相速度 v_{phase} 满足

$$\sin(\frac{k\Delta t v_{phase}}{2}) = S. \tag{39}$$

色散误差可定义为相对相速度（数值相速度与真实速度之比）偏离 1 的程度，

$$\delta := \frac{v_{phase}}{v} - 1 = \frac{2}{kv\Delta t}\arcsin S - 1. \tag{40}$$

均匀网格上波动方程的一维分析可参见 Taflove and Hagness（2005，第 2 章）。实际上，Dablain（1986）表明，采用高阶有限差分格式可降低色散误差，从而能够以较少的每波长点数进行仿真。对于三维麦克斯韦方程，上述数学推导为此类分析提供了一种定性方法。然而，在非均匀网格上进行此类分析颇具挑战性。

参考文献

Alumbaugh, D. 和 Newman, G.（1997）。三维大规模并行电磁反演——II：井间电磁实验分析。*Geophysical Journal International*，128(2)：355–363。

Bj¨orck, A. 和 Pereyra, V.（1970）。范德蒙方程组的求解。《计算数学》，24(112)：893–903。Castillo-Reyes, O.、de la Puente, J. 和 Cela, J. M.（2018）。PETGEM：一种使用边有限元进行三维CSEM正演模拟的并行代码。*Computers & Geosciences*，119：123–136。Chang-Chun, Y.、Xiu-Yan, R.、Yun-He, L.、Yan-Fu, Q.、Chang-Kai, Q. 和 Jing, C.（2015）。航空电磁反演理论及其应用综述。*Geophysics*，80(4)：W17–W31。Chave, A. D. 和 Cox, C. S.（1982）。用于测量海洋下方电导率的可控电磁源 1：正演问题与模型研究。《地球物理研究杂志》，87：5327–5338。Chew, W. C. 和 Weedon, W. H.（1994）。基于修正麦克斯韦方程组和拉伸坐标的三维完全匹配介质。*Microwave and Optical Technology Letters*，7：599–604。Cockett, R.、Kang, S.、Heagy, L. J.、Pidlisecky, A. 和 Oldenburg, D. W.（2015）。SimPEG：用于地球物理应用中模拟和基于梯度的参数估计的开源框架。*Computers& Geosciences*，85：142–154。Constable, S.（2010）。海洋可控源电磁法用于油气勘探的十年发展。*Geophysics*，75(5)：75A67–75A81。Constable, S. C.、Cox, C. S. 和 Chave, A. D.（1986）。海上电磁勘测技术。载于第*56*届*SEG*国际年会扩展摘要，第81–82页。勘探地球物理学家协会。da Silva, N. V.、Morgan, J. V.、MacGregor, L. 和 Warner, M.（2012）。一种用于频域三维CSEM建模的有限元多前沿法。*Geophysics*，77(2)：E101–E115。Dablain, M.（1986）。高阶差分在标量波动方程中的应用。*Geophysics*，51：54–66。Davydycheva, S.、Druskin, V. 和 Habashy, T.（2003）。一种适用于三维各向异性非均匀介质电磁测井的高效有限差分格式。*Geophysics*，68(5)：1525–1536。Eidesmo, T.、Ellingsrud, S.、MacGregor, L.、Constable, S.、Sinha, M.、Johansen, S.、Kong, F. 和 Westerdahl, H.（2002）。海底电缆测井（SBL）：一种在深水区远程识别和直接识别含油气层的新方法。《初至》，20(3)。Ellingsrud, S.、Eidesmo, T.、Johansen, S.、Sinha, M.、MacGregor, L. 和 Constable, S.（2002）。利用海底电缆测井（SBL）遥感油气层：安哥拉近海航次成果。《前沿》，21(10)：972–982。Fornberg, B.（1998）。课堂札记：有限差分公式中权重的计算。《*SIAM*评论》，40(3)：685–691。Golub, G. H.（1996）。《矩阵计算》第三版。约翰斯·霍普金斯数学科学研究丛书。Grayver, A. V.、Streich, R. 和 Ritter, O.（2014）。凯岑二氧化碳储存地层陆地可控源电磁数据的三维反演与分辨率分析。*Geophysics*，79(2)：E101–E114。Key, K.（2009）。多分量、多频率海洋CSEM数据的一维反演：用于识别薄电阻层的方法与合成研究。*Geophysics*，74(2)：F9–F20。Key, K.（2016）。MARE2DEM：用于可控源电磁和大地电磁数据的二维反演代码。*Geophysical Journal International*，207(1)：571–588。Komatitsch, D. 和 Martin, R.（2007）。针对地震波动方程、在掠入射条件下得到改进的非分裂卷积完全匹配层。*Geophysics*，72(5)：SM155–SM167。Li, Y. 和 Key, K.（2007）。二维海洋可控源电磁建模：第一部分——一种自适应有限元算法。*Geophysics*，72(2)：WA51–WA62。Maaø, F.（2007）。海洋地下电磁问题的快速时域有限差分建模。*Geophysics*，72：A19–A23。MacGregor, L.、Barker, N.、Overton, A.、Moody, S. 和 Bodecott, D.（2007）。利用地震与电磁综合数据降低勘探目标风险——福克兰群岛案例研究。《前沿》，26(3)：356–359。

MacGregor, L. and Tomlinson, J. (2014). 油气行业中的海洋可控源电磁法：方法与实践教程。*Interpretation*，2(3):SH13–SH32。Meju, M. A. (2019). 一种简易的地质风险定制三维可控源电磁多属性分析与定量解释方法。*Geophysics*，84(3):E155–E171。Micikevicius, P. (2009). 使用 CUDA 在 GPU 上进行三维有限差分计算。见第二届图形处理器通用计算研讨会论文集，第79–84页。Mittet, R. (2010). 利用波场与扩散场对应原理进行海洋可控源电磁勘测的高阶有限差分模拟。*Geophysics*，75(1):F33–F50。Mittet, R. (2021). 使用高阶有限差分和伪谱格式模拟小尺度介质变化。*Geophysics*，86(5):T387–T399。Mulder, W. (2006). 三维电磁扩散的多重网格求解器。*Geophysical Prospecting*，54(5):633–649。Newman, G. A. and Alumbaugh, D. L. (1995). 使用交错网格有限差分对航空电磁响应进行频率域建模。*Geophysical Prospecting*，43(8):1021–1042。Oristaglio, M. L. and Hohmann, G. W. (1984). 电磁场向二维地球中的扩散：一种有限差分方法。*Geophysics*，49(7):870–894。Rochlitz, R., Seidel, M., and B¨orner, R.-U. (2021). 三种三维时域电磁数据模拟方法的评估。*Geophysical Journal International*，227(3):1980–1995。Rochlitz, R., Skibbe, N., and G¨unther, T. (2019). custEM：复杂可控源电磁数据的可定制有限元模拟。*Geophysics*，84(2):F17–F33。Roden, J. A. and Gedney, S. D. (2000). 卷积完美匹配层（CPML）：一种针对任意介质的高效时域有限差分实现方案。*Microwave and Optical Technology Letters*，27(5):334–339。Smith, J. T. (1996). 三维电磁场的守恒建模，第一部分：性质与误差分析。*Geophysics*，61(5):1308–1318。Streich, R. (2009). 可控源电磁数据的三维有限差分频域建模：实现高精度的直接求解与优化。*Geophysics*，74(5):F95–F105。Taflove, A. and Hagness, S. C. (2005). 计算电动力学：时域有限差分法。Artech House，第3版。Wang, T. and Hohmann, G. W. (1993). 三维电磁建模的有限差分时域解。*Geophysics*，58(6):797–809。Ward, S. H. and Hohmann, G. W. (1988). 地球物理应用中的电磁理论。见应用地球物理中的电磁方法：第*1*卷，理论，第130–311页。勘探地球物理学家协会。Werthm¨uller, D., Mulder, W., and Slob, E. (2019). emg3d：三维电磁扩散的多重网格求解器。*Journal of Open Source Software*，4:1463。Yang, P. (2021). 利用图形处理器提升三维CSEM建模效率。见第*82*届*EAGE*年度会议暨展览会，编号1，第1–5页。欧洲地球科学家与工程师协会。Yang, P., Gao, J., and Wang, B. (2014). 使用有效边界存储的逆时偏移：一种交错网格GPU实现。*Computers & Geosciences*，68:64–72。Yang, P., Gao, J., and Wang, B. (2015). 时域全波形反演的图形处理器实现。*Geophysics*，80(3):F31–F39。Yang, P. and Mittet, R. (2023). 在非均匀网格上使用高阶时域有限差分法进行可控源电磁建模。*Geophysics*，88(2):E56–E67。Zhdanov, M. S. and Keller, G. V. (1994). 地球物理勘探中的地电方法，第31卷。