

一种基于MPI的并行三维WCS-FDTD方法*

南秦 陈娟[†] 牟春晖 范凯航

(西安交通大学信息与通信工程学院, 西安 710049)

针对两个方向存在精细结构的大规模三维电磁问题, 弱条件稳定时域有限差分方法虽能放宽传统时域有限差分方法的时间步长限制, 但需反复求解大规模三对角矩阵方程, 单机计算能力和存储空间难以满足求解需求. 为此, 提出一种基于消息传递接口的并行弱条件稳定时域有限差分方法. 该方法在精细网格方向采用隐式迭代, 使时间步长主要由粗网格方向决定. 同时, 沿粗网格方向对计算区域进行一维划分, 使各子区域内的三对角方程可独立求解, 并通过相邻进程交换边界场量实现并行计算. 以金属超光栅为算例, 对近场探针结果, 远场雷达散射截面和计算效率进行验证. 结果表明, 所提方法与传统FDTD方法及串行WCS-FDTD方法的计算结果吻合良好. 在相同12进程条件下, 并行WCS-FDTD的计算时间约为并行FDTD的1/4.2. 在32进程下, 加速比达到30.5, 并行效率为95.3%. 进一步的多节点测试表明, 当每节点固定使用16个MPI进程, 计算节点由1个增加至3个时, 计算时间由223.2 s降低至89.4 s, 相对于单节点16进程结果, 加速比为2.5, 跨节点并行效率为83.2%. 该方法能够同时利用多个计算节点的计算与存储资源, 具有较好的跨节点扩展能力, 适用于两个方向含精细结构的大规模三维电磁问题.

关键词: 时域有限差分方法, 弱条件稳定时域有限差分方法, 区域分解, 并行计算

PACS: 02.70.-c, 47.11.Bc, 41.20.Jb

* 国家自然科学基金青年科学基金项目(A类) (批准号: 62525114), 国家自然科学基金青年基金 (批准号: 62401455), 陕西省创新能力支撑计划 (批准号: 2024RS-CXTD-07) 和陕西省重点研发计划 (批准号: 2024GX-ZDCYL-05-04, 2025PT-ZCK-04)资助的课题.

[†] 通信作者. E-mail: chen.juan.0201@mail.xjtu.edu.cn

第一作者. E-mail: qin.nan@stu.xjtu.edu.cn

1 引言

时域有限差分 (finite-difference time-domain, FDTD) 方法通过对 Maxwell 方程进行时间和空间离散, 采用显式迭代格式在时域内求解电磁场 [1,2]. 由于其算法形式简洁, 适用性强, FDTD 方法已广泛应用于天线、散射和复杂介质电磁分析等问题中 [3-8]. 然而, 传统 FDTD 方法的时间步长必须满足 Courant-Friedrichs-Lewy (CFL) 稳定性条件, 当计算区域中存在精细结构时, 时间步长会显著减小, 从而导致时间迭代步数增加, 计算效率严重下降 [9].

为克服 CFL 条件对 FDTD 时间步长的限制, 研究者提出了无条件稳定的 FDTD 方法, 如交替方向隐式 (alternating direction implicit, ADI) FDTD [10,11], 局部一维 (locally one-dimensional, LOD) FDTD [12-15] 和遵从散度条件的隐式 (complying-divergence implicit, CDI) FDTD [16] 等. 近年来, LOD-FDTD 已被进一步用于复杂色散介质建模, CDI-FDTD 则通过满足散度约束, 避免在无源区域产生非物理的虚假电荷, 从而改善计算结果的物理一致性和数值精度. 这些方法通过引入隐式差分格式放宽了时间步长限制, 从而在一定程度上提高了计算效率. 然而, 无条件稳定方法在每个时间步通常要求解 6 个相互耦合的隐式方程, 由此增加了计算复杂度. 此外, 过大的时间步长可能引入显著的数值色散和截断误差, 进而影响计算结果的准确性.

在实际电磁问题中, 精细结构往往并不同时存在于三个空间方向 [17-20]. 针对这一特点, 有研究人员提出了几种方法, 仅在精细网格方向采用隐式离散, 其余方向则保持显式迭代, 在放宽时间步长限制的同时在一定程度上保证了精度并降低了计算复杂度. 其中, 混合显-隐式 (hybrid implicit-explicit, HIE) FDTD 方法适用于仅一个方向存在精细网格的问题 [21-23]. 近期研究还将满足散度约束的 HIE-FDTD 方法与复频移完全匹配层相结合, 进一步提高了其求解开放域精细结构问题的适用性. 当目标在两个方向上存在精细结构时, 弱条件稳定 (weakly conditionally stable, WCS) FDTD 方法更为适用 [24,25]. WCS-FDTD 通过对两个精细网格方向进行隐式离散, 使时间步长主要由余下的粗网格所在方向的网格大小决定. 相较于无条件稳定方法, WCS-FDTD 仅需求解 4 个隐式方程, 在计算效率和数值精度方面具有一定优势.

然而, 在求解大规模三维精细结构问题时, WCS-FDTD 仍面临较大的计算和存储开销. 一方面, 大规模网格需要存储多个电磁场分量, 介质参数以及隐式求解所需的迭代系数. 另一方面, WCS-FDTD 在每个时间步内需要求解多个三对角矩阵方程, 增加了单步计算量. Moradi 等 [26] 通过并行求解同一时间步内相互独立的更新方程, 实现了单节点共享内存环境下的多线程并行. 该方法能够提高单节点内的计算效率, 但未涉及计算区域的 MPI 分解, 进程间边界通信及跨节点扩展, 其可处理的计算规模和存储容量仍受单节点硬件

资源限制. 因此, 有必要研究面向分布式内存平台并具有跨节点计算和存储扩展能力的并行WCS-FDTD方法.

已有研究基于消息传递接口(message passing interface, MPI), 采用区域分解和边界场量通信, 实现了传统FDTD及HIE-FDTD方法的并行计算, 为大规模电磁问题的高效求解提供了有效途径^[27-29]. 与HIE-FDTD类似, WCS-FDTD仅在精细网格方向采用隐式迭代, 而在粗网格方向保留显式更新特性, 因而具备通过区域分解实现MPI并行的条件. 考虑到本文采用的WCS-FDTD需要沿 x 和 z 两个精细网格方向求解三对角方程, 本文方法沿粗网格所在的 y 方向进行一维区域分解, 使每个MPI进程均保留完整的 x 和 z 方向网格, 并能够在本地独立完成三对角方程求解. 相邻进程仅在子区域交界面交换必要的边界场量. 与已有单节点多线程并行方法不同, 本文实现的是基于区域分解的MPI进程级并行. 最后, 通过同等进程数下的MPI并行FDTD对比, 单节点强扩展测试及多节点测试, 验证所提方法的计算精度, 并行效率和跨节点扩展能力.

本文其余部分安排如下, 第2节介绍 WCS-FDTD 方法的基本公式及其基于 MPI 的并行实现过程, 第3节以金属超光栅为算例, 验证所提方法的计算准确性和并行效率, 第4节总结全文.

2 理论与实现

2.1 通用WCS-FDTD基本公式

文献^[24]给出了WCS-FDTD的基本更新公式, 其中迭代系数为固定值. 本文将其推广为空间位置相关的迭代系数, 使其能够处理包含复杂介质的目标. 为避免重复, 下面仅以第一个子步骤中 E_x 的更新为例说明其实现过程.

第一个子步骤中, E_x 和 H_y 的更新公式分别为

$$\begin{aligned}
 E_x^{n+\frac{1}{2}}(i+\frac{1}{2}, j, k) = & CA(i+\frac{1}{2}, j, k)E_x^n(i+\frac{1}{2}, j, k) \\
 & + CB(i+\frac{1}{2}, j, k) \frac{H_z^n(i+\frac{1}{2}, j+\frac{1}{2}, k) - H_z^n(i+\frac{1}{2}, j-\frac{1}{2}, k)}{\Delta y} \\
 & - \frac{CB(i+\frac{1}{2}, j, k)}{2} \left[\frac{H_y^{n+\frac{1}{2}}(i+\frac{1}{2}, j, k+\frac{1}{2}) - H_y^{n+\frac{1}{2}}(i+\frac{1}{2}, j, k-\frac{1}{2})}{\Delta z} \right. \\
 & \left. + \frac{H_y^n(i+\frac{1}{2}, j, k+\frac{1}{2}) - H_y^n(i+\frac{1}{2}, j, k-\frac{1}{2})}{\Delta z} \right]. \quad (1)
 \end{aligned}$$

$$\begin{aligned}
H_y^{n+\frac{1}{2}}\left(i+\frac{1}{2}, j, k+\frac{1}{2}\right) &= CP\left(i+\frac{1}{2}, j, k+\frac{1}{2}\right) H_y^n\left(i+\frac{1}{2}, j, k+\frac{1}{2}\right) \\
&\quad - \frac{CQ\left(i+\frac{1}{2}, j, k+\frac{1}{2}\right)}{2} \left[\frac{E_x^{n+\frac{1}{2}}\left(i+\frac{1}{2}, j, k+1\right) - E_x^{n+\frac{1}{2}}\left(i+\frac{1}{2}, j, k\right)}{\Delta z} \right. \\
&\quad \left. + \frac{E_x^n\left(i+\frac{1}{2}, j, k+1\right) - E_x^n\left(i+\frac{1}{2}, j, k\right)}{\Delta z} \right].
\end{aligned} \tag{2}$$

将任一场分量所在的Yee节点位置记作 m , 则空间位置相关的迭代系数为

$$\begin{aligned}
CA(m) &= \frac{1 - \frac{\sigma(m)\Delta t}{2\varepsilon(m)}}{1 + \frac{\sigma(m)\Delta t}{2\varepsilon(m)}}, & CB(m) &= \frac{\frac{\Delta t}{\varepsilon(m)}}{1 + \frac{\sigma(m)\Delta t}{2\varepsilon(m)}}, \\
CP(m) &= \frac{1 - \frac{\sigma_m(m)\Delta t}{2\mu(m)}}{1 + \frac{\sigma_m(m)\Delta t}{2\mu(m)}}, & CQ(m) &= \frac{\frac{\Delta t}{\mu(m)}}{1 + \frac{\sigma_m(m)\Delta t}{2\mu(m)}}.
\end{aligned}$$

其中, ε , μ , σ 和 σ_m 分别为介电常数, 磁导率, 电导率和磁导率损耗参数.

将式 (2)代入式 (1), 并将待求的 $E_x^{n+1/2}$ 移至等式左端, 可得到沿 z 方向的三对角矩阵方程. 对于每个固定的 (i, j) , 该方程仅包含同一网格线上相邻三个节点的 $E_x^{n+1/2}$, 不同 (i, j) 网格线对应的三对角方程相互独立.

在第一个子步骤中, 首先沿 z 方向求解 E_x , 再沿 x 方向求解 E_y , 随后显式更新 H_y 和 H_z , 而 E_z 和 H_x 保持不变. 在第二个子步骤中, 首先沿 x 方向求解 E_z , 再沿 z 方向求解 E_y , 随后显式更新 H_x 和 H_y , 而 E_x 和 H_z 保持不变. 至此完成从 n 到 $n+1$ 时刻的一次迭代. 其他场分量的具体更新公式可由文献^[24]中的方法得到, 这里不再重复列出.

对于FDTD而言, 时间步长受限于CFL条件, 其范围为:

$$\Delta t_{\text{FDTD}} = \frac{1}{c \sqrt{\left(\frac{1}{\Delta x}\right)^2 + \left(\frac{1}{\Delta y}\right)^2 + \left(\frac{1}{\Delta z}\right)^2}} \tag{3}$$

而对于WCS-FDTD的时间步长范围可取为:

$$\Delta t_{\text{WCS-FDTD}} \leq \frac{\Delta y}{c} \tag{4}$$

其中 $c = 1/\sqrt{\varepsilon\mu}$ 为波在介质中的传播速度. 可以看出对于粗网格仅存在于 y 方向的WCS-FDTD, 由于不受 x , z 方向精细网格的影响, 相对于FDTD, 能够在一定程度上放宽CFL条件对于时间步长大小的限制.

2.2 基于MPI的WCS-FDTD并行化

从上一节可知, 对于在 x 和 z 方向存在精细结构的目标, WCS-FDTD需要分别沿这两个方向求解三对角矩阵方程. 为保证各进程能够独立完成隐式方程的求解, 计算区域不能沿 x 或 z 方向进行划分. 因此, 本文沿粗网格所在的 y 方向对计算区域进行一维区域分解, 并将划分后的子区域分配给不同的MPI进程. 由于每个子区域均包含完整的 x 和 z 方向网格, WCS-FDTD中的三对角矩阵方程可以在各进程内部独立求解, 无需在隐式求解过程中进行跨进程通信.

设计算区域在 y 方向包含 L_y 个网格, 参与计算的进程数为 P_y . 为使各进程承担的计算量尽可能均衡, 本文采用按余数均匀分配的方式, 确定各子区域在 y 方向上的网格数量. 首先计算整数商 q 和余数 r :

$$q = \left\lfloor \frac{L_y}{P_y} \right\rfloor, \quad (5)$$

$$r = L_y \bmod P_y. \quad (6)$$

对于进程编号 $\text{rank} < r$ 的进程, 在 y 方向分配 $q + 1$ 个网格. 对于 $\text{rank} \geq r$ 的进程, 在 y 方向分配 q 个网格. 因此, 第 rank 个进程在 y 方向上的本地网格数 $N_y(\text{rank})$ 为

$$N_y(\text{rank}) = \begin{cases} q + 1, & \text{rank} < r, \\ q, & \text{rank} \geq r. \end{cases} \quad (7)$$

第 rank 个进程所对应的全局起始网格编号 $j_s(\text{rank})$ 为

$$j_s(\text{rank}) = \begin{cases} \text{rank}(q + 1), & \text{rank} < r, \\ r(q + 1) + (\text{rank} - r)q, & \text{rank} \geq r. \end{cases} \quad (8)$$

相应的全局终止网格编号 $j_e(\text{rank})$ 为

$$j_e(\text{rank}) = j_s(\text{rank}) + N_y(\text{rank}) - 1. \quad (9)$$

上述分配方式能够保证任意两个进程在 y 方向上的网格数量之差不超过1, 从而实现近似均衡的计算负载分配. 对于位于两个子区域交界面上的场量, 本文将其分配至右侧进程. 整个计算区域的划分方式如图1所示. 该划分方式避免了相邻进程重复存储和更新同一交界面上的场量, 明确了各场量的更新归属, 从

而消除因相邻子区域均包含交界面电场分量而可能引发的更新歧义, 便于统一局部索引及边界通信规则. 需要说明的是, 这种划分仅涉及并程序中的数据分配方式, 不会改变 Yee 网络的离散形式, 也不影响 WCS-FDTD 方法原有的场量更新关系.

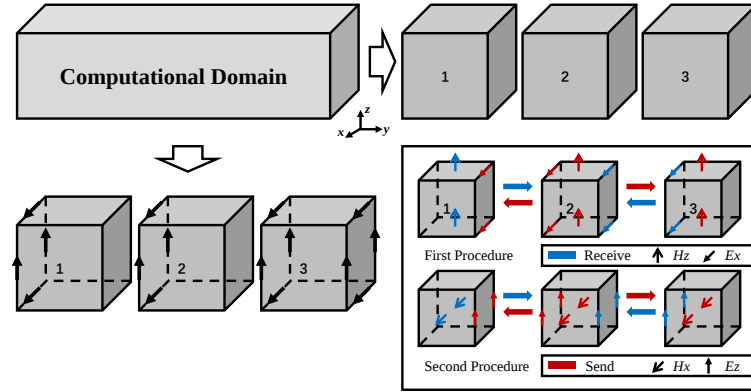


图 1 计算区域划分以及场值通信

Fig. 1. Computational domain partitioning and field-value communication.

完成区域划分后, 每个进程仅分配并存储本地子区域中的电磁场分量, 介质参数以及WCS-FDTD隐式更新所需的系数. 为完成子区域边界处的空间差分计算, 各场量数组在 y 方向的两端设置通信缓冲区, 用于存储相邻进程传递的边界场量. 位于全局计算区域两端的进程仅需设置与其相邻进程对应的有效缓冲区. 吸收边界如卷积完全匹配层 (convolutional perfectly matched layer, CPML) [30], 连接边界, 外推边界等根据本地网格对应的全局编号进行设置.

并行WCS-FDTD的场量更新顺序与串行算法一致. 每个完整时间步由两个子步骤组成. 各进程首先在本地区域内完成相应电场分量的隐式更新, 随后交换边界场量, 再完成其他电磁场分量的更新. 由于计算区域沿 y 方向进行分解, 进程间传递子区域交界面上的二维场量数据, 通信数据为 xoz 截面上的场量.

在第一个子步骤中, 各进程首先完成本地 E_x 的更新. 随后, 除最左侧进程外, 每个进程将其左边界处的 E_x 传递给左侧相邻进程. 左侧进程接收到该边界场量后, 完成子区域右边界附近 E_y 和 H_z 的更新. 在电场更新完成后, 各进程在本地计算 H_y 和 H_z . 随后, 除最右侧进程外, 每个进程将其右边界处的 H_z 传递给右侧相邻进程, 为下一时间步中边界位置处 E_x 的更新提供所需数据.

在第二个子步骤中, 各进程首先完成本地 E_z 的更新. 随后, 除最左侧进程外, 每个进程将其左边界处的 E_z 传递给左侧相邻进程. 左侧进程利用接收到的 E_z 完成子区域右边界附近 E_y 和 H_x 的更新. 在电场更新完成后, 各进程在本地计算 H_x 和 H_y . 最后, 除最右侧进程外, 每个进程将其右边界处的 H_x 传递给右侧相邻

进程, 为下一时间步中边界位置处 E_z 的更新提供所需数据.

程序采用MPI_Sendrecv完成相邻进程之间的边界场量通信. 在每次通信中, 各进程将本地边界场量发送至相邻进程, 同时接收另一相邻进程传递的场量并存入缓冲区. 该操作将发送与接收合并在同一通信调用中, 可避免因阻塞式发送和接收顺序不当而产生进程间相互等待. 程序分别为 E_x , H_z , E_z 和 H_x 设置独立的消息标签, 以保证不同场分量能够被正确识别和接收. 通信完成后, 各进程利用缓冲区中的边界场量更新子区域交界面附近的电磁场.

WCS-FDTD的隐式更新需要求解三对角矩阵方程. 本文采用Thomas算法^[31]完成求解. 以第一子步骤中的 E_x 为例, 对于每个固定的 (i, j) , 求解关于 E_x 的三对角方程组时, 沿 z 方向的三对角方程组可表示为:

$$a_k E_x^{n+\frac{1}{2}}(k-1) + b_k E_x^{n+\frac{1}{2}}(k) + c_k E_x^{n+\frac{1}{2}}(k+1) = d_k \quad (k = 1, 2, \dots, N_z - 1) \quad (10)$$

通过将式(2)代入式(1)并合并同类项, 可得系数 a_k, b_k, c_k 和右端项 d_k 具体形式, 此处不作赘述, 边界处有 $a_1 = 0$ 和 $c_{N_z-1} = 0$. Thomas算法分为前向消元和后向回代两个步骤:

前向消元时, 对系数进行归一化处理, 消去下对角线元素. 定义变量 c'_k 和 d'_k :

$$\begin{aligned} c'_1 &= \frac{c_1}{b_1}, \quad d'_1 = \frac{d_1}{b_1} \\ c'_k &= \frac{c_k}{b_k - a_k c'_{k-1}}, \\ d'_k &= \frac{d_k - a_k d'_{k-1}}{b_k - a_k c'_{k-1}}, \quad (k = 2, 3, \dots, N_z - 1) \end{aligned} \quad (11)$$

后向回代从最后一个网格点开始, 依次求解 $E_x^{n+\frac{1}{2}}(k)$:

$$\begin{aligned} E_x^{n+\frac{1}{2}}(N_z - 1) &= d'_{N_z-1} \\ E_x^{n+\frac{1}{2}}(k) &= d'_k - c'_k E_x^{n+\frac{1}{2}}(k+1) \quad (k = N_z - 2, N_z - 3, \dots, 1) \end{aligned} \quad (12)$$

通过上述两步, 即可高效求解出当前时间步的 E_x 场分量. 不同 (i, j) 网格线对应的三对角方程组相互独立. 由于计算区域仅沿粗网格所在的 y 方向划分, 每个MPI进程均保留完整的 x 和 z 方向网格. 因此, Thomas算法的前向消元和后向回代均在各进程的本地子区域内完成, 求解过程中不需要进行进程间通信. 仅在相应场分量求解完成后, 交换子区域边界上的场量. 其他隐式电场分量采用相同的处理方法.

并程序的主要执行流程如下:

1. 初始化 MPI 并行环境, 获取进程总数和当前进程编号.
2. 根进程导入网格及介质参数, 设置吸收边界, 连接边界和外推边界等, 从而确定完整的计算区域.

3. 沿 y 方向划分计算区域, 并将相应的网格及介质参数分配至各进程.
4. 各进程分配本地场量数组, 计算隐式迭代系数.
5. 在第一个子步骤中, 各进程首先采用Thomas算法求解 E_x 的三对角方程, 然后交换边界处的 E_x 。随后求解 E_y 的三对角方程, 更新 H_y 和 H_z , 最后交换边界处的 H_z 。
6. 在第二个子步骤中, 各进程首先采用Thomas算法求解 E_z 的三对角方程, 然后交换边界处的 E_z 。随后求解 E_y 的三对角方程, 更新 H_x 和 H_y , 最后交换边界处的 H_x 。
7. 重复执行上述两个子步骤, 直至达到设定的时间迭代步数.
8. 汇总并后处理各进程的计算结果, 释放内存, 结束 MPI 并行计算.

3 数值算例

采用金属超光栅算例验证所提算法的计算精度和效率^[18]. 其结构及尺寸如图2所示. 金属板的尺寸为 $415.68 \text{ mm} \times 300 \text{ mm} \times 12 \text{ mm}$. 金属板上周期排布着一些凹槽, 每个周期内包含两种尺寸不同的槽: 第一种槽宽为 8.25 mm , 长为 300 mm , 深为 9 mm . 第二种槽宽为 4.8 mm , 长为 300 mm , 深为 4.8 mm . 沿 x 方向, 第一种槽与第二种槽之间的间距为 6.48 mm , 第二种槽与下一周期第一种槽之间的间距为 15.12 mm . 因此, 该结构沿 x 方向的周期为 34.65 mm , 共包含12个周期. 目标在 y 方向上是均匀的, 通过非均匀网格剖分, $\Delta x = 0.4 \text{ mm}$, $\Delta z = 0.25 \text{ mm}$, $\Delta y = 2 \text{ mm}$, FDTD的时间步长为:

$$\Delta t_{\text{FDTD}} = \frac{1}{c\sqrt{(1/\Delta x)^2 + (1/\Delta y)^2 + (1/\Delta z)^2}} = 0.70 \text{ ps},$$

WCS-FDTD的时间步长为

$$\Delta t_{\text{WCS-FDTD}} = \frac{\Delta y}{c} = 6.67 \text{ ps}.$$

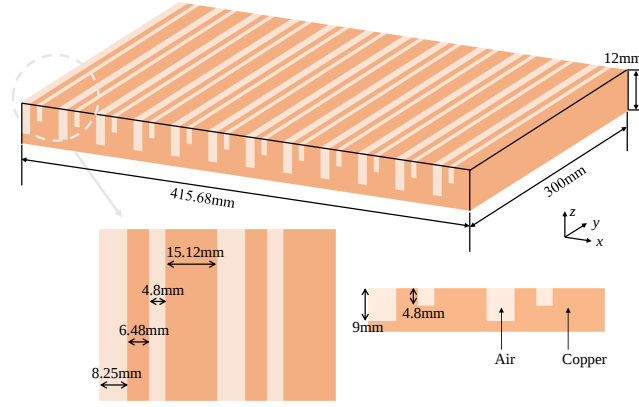


图 2 金属超光栅结构和尺寸示意图

Fig. 2. Schematic diagram of the structure and dimensions of the metallic metagrating.

整个计算区域大小为 $204 \times 294 \times 62$, 总网格数约为 371.9 万. 该算例具有较大的计算规模, 对计算时间和内存资源均提出了较高要求, 可用于检验所提并行算法处理大规模电磁仿真问题时的计算效率和并行扩展能力. 边界用CPML进行截断, 吸收层厚度为10个网格, 用沿 $-z$ 方向传播, x 方向极化的平面波进行照射, 激励波形公式为:

$$s(t) = \exp \left[- \left(\frac{t - t_0}{t_w} \right)^2 \right],$$

其中 $t_w = \frac{\sqrt{\ln 10}}{\pi f}$, $t_0 = 3.68t_w$, 此处令 $f = 12 \text{ GHz}$. 记金属板上边界面为 $z = 0$ 面, 金属板中心为 xoy 面原点, 在 $(0, 0, 20 \text{ mm})$ 处放置一近场探针, 近场探针结果和远场雷达散射截面 (radar cross section, RCS) 结果由图3和图4给出, 能够看出超光栅在10GHz下有使波束发生偏转的特性. 由表1可知, 串行WCS-FDTD和并行WCS-FDTD的近场结果相对于FDTD的误差为1.4%, 近场结果的相对误差由式 (13) 给出, 其中, N_t 为时间采样点总数, m 为时间采样点序号. 约在 $\theta = -60^\circ$ 处, FDTD的远场RCS最大值为23.54 dB, WCS-FDTD的远场RCS最大值为23.63 dB, 二者相差0.09 dB. 结果表明, 串行和并行WCS-FDTD均与FDTD结果吻合良好, 能够准确表征超光栅的近场和远场散射特性, 验证了所提方法的准确性与可靠性. 为统一表述, 本文正文采用“并行WCS-FDTD”, 图表中将其简记为P-WCS-FDTD. 相应地, P-FDTD表示并行FDTD.

$$\text{Err}_{\text{near}} = \frac{1}{N_t} \sum_{m=1}^{N_t} \frac{|E_{\text{FDTD}}(m) - E_{\text{WCS-FDTD}}(m)|}{\max |E_{\text{FDTD}}|} \times 100\%. \quad (13)$$

传统FDTD, WCS-FDTD, 并行FDTD和并行WCS-FDTD四种算法的仿真时间对比如表1所示, 其中P表示MPI并行方法. 并行FDTD和并行WCS-FDTD均采用12个进程. 串行WCS-FDTD通过放宽时间步长, 将计算时间缩短至传统FDTD的1/3.8, 在相同进程数下, 并行WCS-FDTD的计算时间为并行FDTD的1/4.2.

此外, 并行WCS-FDTD的计算时间为串行WCS-FDTD的1/11.5, 表明该方法能够在保持计算精度的同时进一步提高计算效率.

算法并行效率由表2给出, 并行效率定义为:

$$E_p = \frac{S_p}{p} \times 100\% = \frac{T_1}{pT_p} \times 100\%, \quad (14)$$

其中, E_p 表示使用 p 个进程时的并行效率. 图5给出了对应的加速比, 其定义为:

$$S_p = \frac{T_1}{T_p}, \quad (15)$$

其中, S_p 表示使用 p 个进程时的并行加速比, T_1 表示单进程计算时间, T_p 表示使用 p 个进程时的计算时间. 能够看出针对超光栅算例, 算法在32进程时并行效率保持在95%以上, 加速比约为30.5, 体现了算法良好的扩展性. 表1和表2的计算在课题组Linux服务器上完成. 该服务器配置两颗Intel Xeon Gold 6248R 3.00 GHz处理器, 共48个物理核心(96个逻辑核心), 内存1 TiB, 操作系统为Ubuntu 22.04.3 LTS, 采用GCC 11.4.0和MPICH/HYDRA 4.0. 每个MPI进程使用一个物理核心.

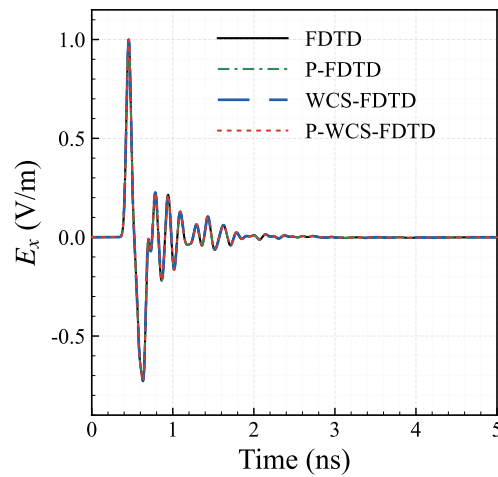


图3 金属超光栅算例的近场探针结果对比

Fig. 3. Comparison of near-field probe results for the metallic metagrating case.

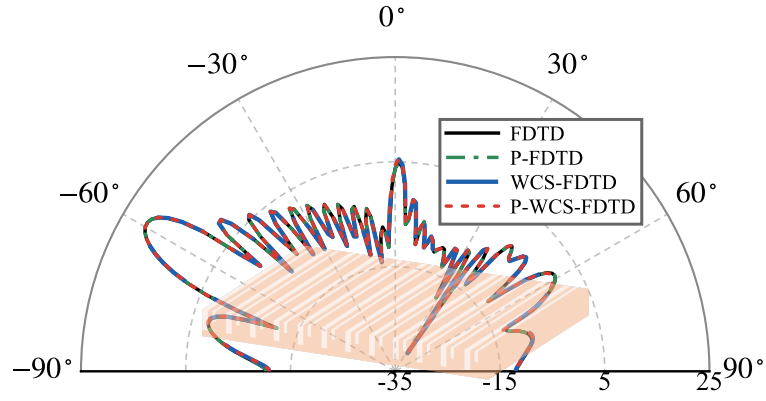


图 4 金属超光栅算例在10 GHz下的RCS结果对比

Fig. 4. Comparison of RCS results for the metallic metagrating case.

表 1 超光栅算例中四种方法的仿真时间对比

Table 1. Comparison of simulation time for the four methods in the metagrating case.

方法	时间步长 (ps)	计算时间 (s)	误差 (%)
FDTD	0.70	13031.1	—
P-FDTD	0.70	1248.2	—
WCS-FDTD	6.67	3410.0	1.4
P-WCS-FDTD	6.67	295.7	1.4

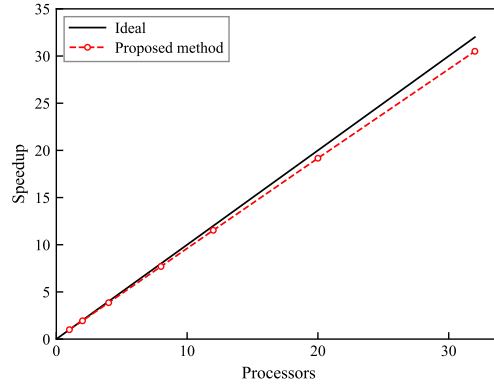


图 5 算法在金属超光栅算例中的加速比

Fig. 5. Speedup of the method for the metagrating case.

为进一步验证所提方法的跨节点扩展能力, 在每个节点固定使用16个MPI进程, 分别采用1, 2和3个计算节点进行测试. 由表 3可知, 节点数由1增加至3时, 计算时间由223.2 s降低至89.4 s. 相对于单节点16进程

表 2 算法在超光栅算例中的并行效率

Table 2. Parallel efficiency of the method for the metagrating case.

进程数	计算时间 (s)	并行效率 E_p
1	3410.0	100%
2	1758.1	97.0%
4	883.6	96.5%
8	443.1	96.2%
12	295.7	96.1%
20	177.9	95.8%
32	111.8	95.3%

结果, 3节点48进程的加速比为2.5, 跨节点效率为83.2%. 跨节点加速比和并行效率均以单节点16进程结果为基准, 并行效率为相对加速比与节点数之比. 结果表明, 所提方法能够利用多个计算节点的计算和存储资源, 具有不同于单节点线程级并行的跨节点扩展能力. 表 3所示的跨节点数值计算在合肥先进计算中心完成. 每个计算节点配置一颗Hygon C86 7185处理器, 包含32个物理核心(64个逻辑核心) 和251 GiB内存, 节点间采用ConnectX-6 InfiniBand网络互连. 操作系统为CentOS Linux 7, 采用GCC 7.3.1和Open MPI 4.1.4. 测试时每个节点固定使用16个MPI进程.

表 3 不同节点数下的并行计算性能

Table 3. Parallel performance with different numbers of nodes.

节点数	进程数	每节点进程数	计算时间 (s)	相对16进程加速比	跨节点效率
1	16	16	223.2	1.0	100.0%
2	32	16	125.2	1.8	89.1%
3	48	16	89.4	2.5	83.2%

4 总结

本文针对两个空间方向存在精细结构的大规模三维电磁仿真问题, 提出并实现了一种基于 MPI 的并行 WCS-FDTD 方法. 根据 WCS-FDTD 的求解特点, 本文采用沿粗网格方向的一维区域划分策略, 将整体计算区域划分为多个子区域并分配给不同 MPI 进程. 各进程在本地子区域内独立完成场量更新, 仅在相邻子区域边界处进行必要场量通信, 从而提高并行计算效率. 以金属超光栅结构为例, 对所提出方法的精度和效率进行了验证. 近场探针结果和远场 RCS 结果表明, 并行 WCS-FDTD 方法与传统 FDTD 方法及串行 WCS-FDTD 方法吻合良好, 说明该并行算法能够保持原有 WCS-FDTD 方法的计算精度. 计算时间对比表明, WCS-FDTD 通过放宽时间步长显著减少了仿真耗时, 而并行 WCS-FDTD 在此基础上进一步提升了计算效率. 在相同12进程条件下, 并行WCS-FDTD的计算时间为295.7 s, 约为并行FDTD计算时间的1/4.2, 表明其效率优势并非仅来自MPI并行, 还来自WCS-FDTD对精细网格时间步长限制的放宽. 在32进程下, 所提方法的加速比达到30.5, 并行效率为95.3%. 进一步的跨节点测试中, 每节点固定使用16个MPI进程, 当计算节点由1个增加至3个时, 计算时间由223.2 s降低至89.4 s, 相对加速比为2.5, 跨节点效率为83.2%. 上述结果表明, 所提方法能够将计算量和场量数据分布至多个计算节点, 具有区别于单节点线程级并行的跨节点计算与存储扩展能力.

参考文献

- [1] Yee K 1966 *IEEE Trans. Antennas Propag.* **14** 302
- [2] Chew W C, Weedon W H 1994 *Microw. Opt. Technol. Lett.* **7** 599
- [3] Guo X M, Deng H C, Man L, Li H Y, Chen W, Yang L X, Bo Y 2024 *Chin. J. Radio Sci.* **39** 80 (in Chinese) [郭先敏, 邓浩川, 满良, 李海燕, 陈伟, 杨利霞, 薄勇 2024 电波科学学报 **39** 80]
- [4] Chen B Y, Zhang Y R, Wang L, Wang F F 2016 *Acta Phys. Sin.* **65** 144101 (in Chinese) [陈碧云, 张业荣, 王磊, 王芳芳 2016 物理学报 **65** 144101]
- [5] Li Y C, Chen F, Yang W X 2025 *Opt. Commun.* **586** 131878
- [6] Warren C, Giannopoulos A, Giannakis I 2016 *Comput. Phys. Commun.* **209** 163
- [7] Ma Z H, Zou B, Zhang L M 2025 *IEEE Trans. Antennas Propag.* **73** 3863
- [8] Yao Z, Wang Y E, Keller S, Carman G P 2015 *IEEE Trans. Antennas Propag.* **65** 3335

- [9] Taflov A, Hagness S C 2005 *Computational Electrodynamics: The Finite-Difference Time-Domain Method* (3rd Ed.) (Norwood: Artech House) pp107–167
- [10] Namiki T 1999 *IEEE Trans. Microw. Theory Tech.***47** 2003
- [11] Namiki T 2000 *IEEE Trans. Microw. Theory Tech.***48** 1743
- [12] Shibayama J, Muraki M, Yamauchi J, Nakano H 2005 *Electron. Lett.***41** 1046
- [13] Ahmed I, Chua E K, Li E P, Chen Z 2008 *IEEE Trans. Antennas Propag.***56** 3596
- [14] Liu Q F, Chen Z, Yin W Y 2009 *IEEE Trans. Antennas Propag.***57** 2409
- [15] Xie G D, Hou G L, Xu K, Niu K K, Feng N X, Li Y S, Huang Z X, Elsherbeni A Z 2023 *IEEE Trans. Antennas Propag.***71** 8830
- [16] Tan E L 2021 *IEEE Antennas Wirel. Propag. Lett.***20** 853
- [17] Chen J, Li J X, Liu Q H 2017 *IEEE Trans. Microw. Theory Tech.***65** 3689
- [18] Rahmanzadeh M, Khavasi A 2020 *Opt. Express***28** 16439
- [19] Chen J, Xu N, Zhang A X, Guo J Y 2016 *IEEE Trans. Antennas Propag.***64** 3011
- [20] Chen J, Wang J G 2007 *IEEE Trans. Electromagn. Compat.***49** 354
- [21] Huang B K, Wang G, Jiang Y S, Wang W B 2003 *Microw. Opt. Technol. Lett.***39** 97
- [22] Chen J, Wang J G 2006 *Microw. Opt. Technol. Lett.***48** 2291
- [23] Hou G L, Chen Y, Xie G D, Ding W J, Li Y S, Huang Z X 2025 *IEEE Trans. Antennas Propag.***73** 6670
- [24] Chen J, Wang J G 2007 *IEEE Trans. Electromagn. Compat.***49** 419
- [25] Liang W B, Zhang K L, Wu Y, Wang M J, Fan C, Zheng H X, Li E P 2023 *IEEE Microw. Wirel. Tech. Lett.***33** 951
- [26] Moradi M, Nayyeri V, Sadrpour S M, Soleimani M, Ramahi O M 2020 *IEEE Trans. Electromagn. Compat.***62** 498
- [27] Guiffaut C, Mahdjoubi K 2001 *IEEE Antennas Propag. Mag.***43** 94
- [28] Nan Q, Mou C H, Chen J 2023 *Appl. Comput. Electromagn. Soc. J.***38** 849
- [29] Wang L X, Chen J, Fan K H 2026 *IEEE Trans. Antennas Propag.***74** 2632

- [30] Wang J G, Wang Y, Zhang D H 2006 *IEEE Trans. Plasma Sci.***34** 681
- [31] Thomas J W 2013 *Numerical Partial Differential Equations: Finite Difference Methods* (Vol. 22)
(New York: Springer) p87

录用稿件，非最终出版稿

An MPI-Based Parallel Three-Dimensional WCS-FDTD Method*

NAN Qin CHEN Juan[†] MOU Chunhui FAN Kaihang

(*School of Information and Communications Engineering, Xi'an Jiaotong University, Xi'an
710049, China*)

Abstract

For large-scale three-dimensional electromagnetic problems with fine structures in two spatial directions, the Courant–Friedrichs–Lewy (CFL) condition imposes a severe restriction on the finite-difference time-domain (FDTD) method. Its time step is limited by the smallest grid size, resulting in low computational efficiency. The weakly conditionally stable FDTD (WCS-FDTD) method relaxes this restriction through implicit field updates. However, large tridiagonal systems must be solved repeatedly at each time step. These solutions impose substantial computational and memory demands on a single computing node. To overcome these limitations, an MPI-based parallel three-dimensional WCS-FDTD method is proposed for distributed-memory platforms.

The field components along the two finely discretized directions are updated implicitly. Therefore, the time step is determined by the grid size in the coarsely discretized direction. The computational domain is decomposed one-dimensionally along this coarse-grid direction. As a result, each MPI process retains complete grid lines in both implicit directions. It independently solves the local tridiagonal systems using

* Project supported by the National Natural Science Foundation of China Youth Science Fund Project (Category A) (Grant No. 62525114), the National Natural Science Foundations of China (Grant No. 62401455), Shaanxi Province Innovation Capability Support Plan (Grant No. 2024RS-CXTD-07) and Key R&D Program of Shaanxi Province (No. 2024GX-ZDCYL-05-04 and No. 2025PT-ZCK-04).

[†] Corresponding author. E-mail: chen.juan.0201@mail.xjtu.edu.cn

The First Author. E-mail: qin.nan@stu.xjtu.edu.cn

the Thomas algorithm. No interprocess communication is required during the implicit solutions. After the corresponding field updates, neighboring processes exchange only the required boundary field components through `MPI_Sendrecv`. This strategy combines the enlarged time step of WCS-FDTD with process-level parallelism. It also distributes the computational workload and field data across multiple nodes.

A metallic metagrating is used to evaluate the accuracy, computational efficiency, and parallel scalability of the proposed method. The calculated near-field probe responses agree closely with those obtained using the conventional FDTD and serial WCS-FDTD methods. Close agreement is also observed for the far-field radar cross sections. For this example, WCS-FDTD increases the time step from 0.70 ps to 6.67 ps. It reduces the serial runtime by a factor of 3.8 relative to conventional FDTD. When the same 12 MPI processes are used, parallel WCS-FDTD is 4.2 times faster than parallel FDTD. It is also 11.5 times faster than serial WCS-FDTD. With 32 processes, the parallel WCS-FDTD method achieves a speedup of 30.5 and a parallel efficiency of 95.3%. Multi-node tests are also performed using 16 MPI processes per node. Increasing the number of nodes from one to three reduces the runtime from 223.2 s to 89.4 s. This corresponds to a relative speedup of 2.5 and a cross-node parallel efficiency of 83.2%.

These results demonstrate that the proposed method preserves the accuracy of WCS-FDTD and substantially improves computational efficiency. It also effectively exploits the aggregated computing and memory resources of multiple nodes. Therefore, the method is well suited to large-scale three-dimensional electromagnetic simulations involving fine structures in two spatial directions.

Keywords: finite-difference time-domain method, weakly conditionally stable finite-difference time-domain method, domain decomposition, parallel computing