arXiv:2603.02871v1 [physics.comp-ph] 3 Mar 2026

用于可复现、可互操作且公平基准测试的 DDA 求解器浮点一致性交叉验证方法

Cl´ement Argentin[a,*], Patrick C. Chaumet[b], Michel Gross[c], Maxim A. Yurkin[a]

[a]鲁昂诺曼底大学´，鲁昂诺曼底国立应用科学学院，法国国家科学研究中心，*CORIA UMR 6614*，鲁昂，*76000*，法国[b]菲涅耳研究所，艾克斯-马赛大学，法国国家科学研究中心，马赛中央理工学院，马赛，*13013*，法国[c]蒙彼利埃第二大学´，法国国家科学研究中心，*L2C UMR 5221*，蒙彼利埃，*34090*，法国

摘要

离散偶极近似（离散偶极子近似）是一种广泛使用且通用的数值方法，用于求解任意形状物体的电磁散射。尽管其应用广泛，但由于线性系统约定、求解器设置和默认数值参数存在差异，不同独立实现之间的定量比较仍然具有挑战性。本文提出一种统一的软件辅助方法，用于对三种主要开源 DDA 求解器DDSCAT、 ADDA和 IFDDA进行交叉验证与基准测试。我们展示了如何通过统一所有自由参数，在不同实现之间达到机器精度一致性，并提供实用的等价表，从而实现可复现且可互操作的模拟。利用该方法，我们对涵盖OpenMP、MPI以及CUDA/OpenCL并行化的中央处理器和图形处理器性能进行了系统比较。除基准测试之外，我们的方法还可作为实用指南，用于配置一致的DDA模拟，并帮助理解精度、求解器选择和硬件架构如何影响计算光散射研究中的运行时间、可扩展性和精度。该软件包还支持面向未来代码版本的回归测试和逐位可复现性验证。

*通讯作者。电子邮箱：yurkin@gmail.com

程序摘要

程序名称：dda-bench*CPC*程序库程序文件链接：开发者存储库链接：
https://doi.org/10.5281/zenodo.18836855许可条款（请选择一项）：
GPLv3编程语言：Python问题性质：

离散偶极近似（DDA）的独立实现（如 DDSCAT、ADDA 和 IFDDA）广泛用于任意形状物体的电磁散射计算。然而，跨代码的定量比较和公平的性能基准测试仍然困难，因为各代码的差异不仅体现在线性系统约定上，还可能涉及默认数值参数、求解器设置、极化率和相互作用公式，以及输出约定或单位。细微的不匹配可能改变迭代收敛历史，并造成差异，从而掩盖真正的算法或硬件效应。因此，用户和开发者缺少一种实用且可复现的方法来：(i) 验证两个代码求解的是同一个数值问题；(ii) 评估特定参数不同时预期应达到的匹配位数；以及 (iii) 在不受精度差异干扰的情况下对运行时间进行基准测试。

解决方法：
本文介绍的软件提供了一个轻量级的基于命令行的框架，可通过可执行文件比较不同的离散偶极近似实现。用户在专用输入文件中指定每个代码或代码版本要测试的命令行。Python 封装程序会自动执行指定命令、收集生成的输出文件，并提取用户选定用于比较的物理量。必要时，提取的数据会自动转换为一致的定义和单位。模拟之间的一致性通过匹配位数指标进行量化，该指标可直接反映浮点一致性。此方法可以轻松比较不同的 DDA 代码，或同一求解器的多个版本，从而在确保配置等效后支持回归测试、交叉验证和公平的基准测试。

其他意见，包括限制条件和特殊情况：
该软件仅限于提供命令行界面的 DDA 实现，并假定用户已安装并编译相应的代码。框架本身充当 Python 封装程序，对生成的可执行文件进行操作。这一设计选择便于将该框架作为轻量级 Python 软件包实际使用，从而可以安装并导入

轻松地，同时允许用户跟踪并比较同一代码的不同发行版本，并自由选择在给定比较中使用哪些可执行文件版本。

关键词：光散射，离散偶极近似，可复现性，基准测试，交叉验证

1. 引言

粒子光散射是支配多种光学现象的基础过程 [1]。对散射、吸收和近场效应进行定量建模，对于解释实验结果和设计光学系统至关重要。麦克斯韦方程组的解析解仅适用于少数理想化几何形状（如球体或旋转椭球体），因此，对于形状、组成和周围环境各异的真实系统，数值方法不可或缺。

通用数值技术主要包括几大类方法：时域有限差分法（FDTD） [2], 有限元法（FEM） [3],边界元法或表面积分方程（BEM/SIE） [4], 以及体积分方程（VIE） [5]方法。在后一类方法中，离散偶极近似（DDA）因概念简单、模块化程度高，且随粒子尺寸和波长变化具有良好的扩展性而得到广泛应用 [6, 7]。DDA 于文献 [8], 中提出，并在文献 [9, 10], 中进一步发展。该方法将散射体离散为由可极化体素（偶极子）构成的阵列，并通过求解由麦克斯韦方程组的体积分形式导出的大型线性系统，计算其感应极化。得到体素的极化后，即可计算所有散射观测量，包括截面、穆勒矩阵以及近场区域和远场区域中的场分布。DDA 可用于任意形状的粒子及复杂光学环境，应用领域涵盖天体物理学 [9] 、大气科学 [11]、纳米技术 [12] 和生物医学应用 [13]。此外，由于 DDA 是一种“数值精确”方法，常被用作基准，以验证或修正近似解析模型 [14, 15]。

过去三十年来，人们开发了多种开源离散偶极子近似代码，各自针对不同的架构和应用进行了优化。使用最广泛的包括 `DDSCAT` [10]、`ADDA` [16] 和 `IFDDA` [17]。网上提供了大量公开可用的离散偶极子近似实现及相关工具的列表 [18, 19]。

尽管它们有着共同的理论基础，但比较不同离散偶极近似实现的数值性能和精度仍非易事。差异往往源于极化率公式、迭代求解器或线性系统约定上的细微不同。因此，跨代码基准测试 [20] 很难解读，因为这些代码的精度和计算性能各不相同。

因此，本文的目标有两方面。首先，我们提出一种统一的、软件辅助的方法，用于在独立的 DDA 实现之间进行浮点一致性交叉验证。通过协调所有自由参数与隐藏参数，我们证明求解器在双精度下可以达到机器精度的一致（即 14–15 位正确数字）。聚焦于 `DDSCAT`、`ADDA` 和 `IFDDA`，我们提供实用的等价表，以实现可复现且可互操作的 DDA模拟。由于这些实现依赖于不同的线性系统约定和 FFT后端，跨代码无法实现严格的逐位可复现性，因为浮点运算顺序中即使极微小的差异也会导致舍入误差发散。此类局限性是高性能科学计算中众所周知的问题，尤其是对于并行数值模拟 [21, 22]。

其次，利用这一一致的数值基线，我们在中央处理器和图形处理器上开展公平的基准测试研究，涵盖中央处理器上的 OpenMP 与 MPI 并行化，以及图形处理器上的 CUDA/OpenCL 加速。因此，本研究中观察到的性能差异反映的是算法效应与硬件架构效应，而非配置不一致。

除了交叉验证外，该软件包还支持比较同一实现的不同版本。在这种情况下，可以实现真正的逐位可复现性，从而支持持续集成/持续交付（CI/CD）回归测试以及版本间一致性检查。例如，这将补充 `ADDA` 工作流中已有的自动化测试。

最后，本文旨在作为参考，介绍建立DDA模拟的最佳实践，并帮助理解实现选择的影响。第2节对三种DDA 代码及其内部特性进行了比较概述；第3节作为方法指南，也是第4节和第5节所呈现的性能基准测试的技术前提。

2. DDA 实现与工作流程

本节的主要目标是对DDA工作流程提供简洁且自洽的描述，介绍其关键内部组件，以便于后续各节的讨论。我们还总结了所考虑的每种实现的独特特性，从而使本文成为三种主要开源离散偶极子近似代码的实用比较概述。

表 1：DDSCAT、ADDA 和 IFDDA 的部分特性概述。[a]

特性 / 代码	DDSCAT	ADDA	IFDDA
应用重点	通用	通用	多层基底与显微术
社区与贡献	开源（网站）	开源（GitHub）	开源（GitLab）
用户界面	面向命令行界面（配置文件-tion 文件）with 可选 GUI	面向命令行界面 with op-传统图形用户界面	面向图形用户界面 with op-传统命令行界面
语言	Fortran 90	C99	Fortran 77
硬件支持	仅中央处理器	中央处理器和图形处理器（OpenCL）	中央处理器和图形处理器（CUDA）
并行化	OpenMP + MPI（部分）[b]	MPI（完整）	OpenMP
取向平均	辛普森法则	龙贝格积分	–
精度	单精度（默认）或双精度	双精度（默认）	双精度（默认）或单精度
入射波类型	仅平面波	平面波及其他	平面波及其他
快速傅里叶变换算法	GPFA、英特尔 MKL	GPFA、FFTW	单例，FFTW
DDA 公式	POI, FCD	POI, FCD, IGT	POI, FCD, IGT
极化率模型	LDR、CLDR、FCD	CM、RR、LDR、CLDR, FCD, LAK, DGF	CM, RR, LDR, FCD, LAK, GB
初始场	仅为零	零及其他	零及其他
预条件器	左侧（雅可比）或无	none	左侧、右侧，两者，（Chan）或无
体积修正	yes	可选（是/否）	no
迭代求解器	BCGS2、BiCGStab、GP-双共轭梯度法、准最小残差法、PETRKP	BCGS2, 双共轭梯度法, BiCGStab, CGNR, CSYM、准最小残差法、QMR2	GPBiCG, GPBiCGStab, BiCGStab, CG、准最小残差法, IDRS及许多其他方法
独特特征	周期性目标	贝塞尔光束、电子能量损失谱/阴极发光, 稀疏模式、长方体体素-els, 均匀 sub-策略（采用三维快速傅里叶变换）	多层 sub-结构, 显微术, 完全各向异性目标体

[a]文中对所用缩写均有说明。[b]仅适用于不同粒子取向（取向平均），与 ADDA 中单次模拟的分布式内存并行化相对。

这三种实现： DDSCAT、 ADDA和 IFDDA，旨在模拟电磁波与任意形状的粒子或物体之间的相互作用。 DDSCAT 和 ADDA 主要用于一般光散射问题，尤其适用于数据库和大规模参数研究。相比之下， IFDDA专注于涉及复杂光学装置的应用，包括多层基底和显微成像配置。它能够生成强度图像等可观测量，因此特别适用于应用光学或实验光学中的建模与可视化任务 [23]。

这三种代码实现了类似的DDA工作流程，但数值细节和硬件优化策略各不相同。每种代码都支持多种基于Krylov子空间的迭代求解器（属于共轭梯度（CG）系列），用于求解大型稀疏线性方程组。正如文献 [24], 所指出的，通常无法确定一个在所有情形下都表现最佳的求解器，因为性能很大程度上取决于颗粒形貌和折射率。在文献 [24], 中，QMR求解器（用于 IFDDA）采用非对称化形式实现（与 DDSCAT中的情况相同）；而采用对称化形式时，如 ADDA所做的那样，矩阵-向量乘积的次数可减少一半，收敛速率和执行时间则与其他CG系列求解器相近。因此，比较不同代码的求解器性能时，必须考虑这些实现细节。为确保一致性，第4节和第5节中展示的所有性能结果均基于BiCGStab求解器。

离散偶极子近似在极化率模型和相互作用公式形式方面经历了多次改进 [25, 7]。经典方法，即点偶极子相互作用（POI）结合晶格色散关系（LDR）极化率 [10], ，仍被广泛使用；与此同时，滤波耦合偶极（FCD）[26, 27]和积分格林张量（IGT） [28, 29] 等较现代的公式形式也已开发出来，以提升精度和迭代收敛性 [7]。值得庆幸的是，所有三种代码都至少支持其中一种先进公式形式。FCD 和 IGT 各自配有专属的极化率方案，而 POI 支持许多其他方案 [25]：克劳修斯 - 莫索提（ CM ）、辐射反作用（RR）修正、数字化格林函数（DGF / GB）、Lakhtakia（LAK），以及晶格色散关系（LDR）或其修正版（CLDR）。

不同实现采用的入射场模型各不相同。 DDSCAT 主要支持平面波照明。 ADDA 还提供（近似）高斯光束和点偶极激发，支持从文件读取用户指定的任意场，并在其主分支中提供多种内置贝塞尔光束 [30]。开发分支进一步实现了任意宿主介质中粒子的电子能量损失谱（EELS）和阴极发光（CL） [31]。 IFDDA 支持物理高斯光束以及种类更丰富的结构化照明（圆偏振、多平面波、散斑、共焦和任意导入场），并允许将目标体嵌入多层衬底中。它还可以根据计算得到的场生成显微图像（例如明场、暗场、相衬和全息图像）。

迭代收敛可通过先进的初始场和适-

当的预条件器来加速。 IFDDA 实现了若干先进的初始猜测，例如可与 Chan 型预条件 [32] 相结合以减少迭代次数的 uGu [33]。最近，一种三级循环预条件器的实现扩展了这项工作，并已被证明可进一步减少迭代次数 [34]。 ADDA除经典的零场和入射场外，还提供 Wentzel–Kramers–Brillouin（WKB）初始场选项，而 DDSCAT则为 BiCGStab 求解器提供零场初始化，并可选用左雅可比预条件。应当指出，尽管这些初始场在某些情况下可能有益，但它们需要计算初始化时间，因此若在其适用范围之外使用，可能会增加计算时间。最近的一项研究 [35] 将初始 WKB 场与 uGu 形式 [36] 进行了比较，结果表明二者的性能大体相当。除严格但计算代价高昂的离散偶极子近似之外， IFDDA还提供对多层透射-反射问题有效的近似直接方法 [37]。

所有实现均可跨操作系统移植，且所需的外部依赖很少。默认情况下， DDSCAT 使用自包含的 GPFA 快速傅里叶变换算法 [38], ，仅需 Fortran 编译器（例如 gfortran），因此具有很高的可移植性。 ADDA 也可使用 GPFA（名为 Temperton）进行编译，无需外部库即可确保兼容性（gcc 和 gfortran 编译器除外）。然而，如第 4.2 节所示，与英特尔 MKL 的 DFTI 例程和 FFTW 等优化的快速傅里叶变换库相比，GPFA 的性能较差 [39]。使用 Single-ton 算法的 IFDDA 也存在同样的问题 [40]。因此，推荐的配置为 DDSCATMKL、 ADDAFFTW 和 IFDDAFFTW。GPU 模式还会引入额外依赖： ADDA 需要 OpenCL 和 clFFT，而 IFDDA 依赖 CUDA。ADDA 还支持诸如 Spack [41] 或 Nix之类的包管理器，便于在隔离环境和高性能计算（HPC）集群中进行可移植且可复现的安装。

高性能计算深刻影响了离散偶极子近似代码的发展。 DDSCAT和 IFDDA 使用 OpenMP 针对共享内存架构进行了优化，而 ADDA 则采用 MPI 来利用分布式内存集群。近年来，GPU 加速器已成为加速矩阵—向量运算的标准配置。IFDDA 使用 CUDA，这是最成熟、优化最广泛的 GPU 框架，但仅适用于英伟达硬件； ADDA则使用 OpenCL，以确保在英伟达、AMD 和英特尔 GPU 之间具备跨厂商可移植性，但性能较低。

浮点精度是另一项重要的设计选择。双精

度可确保更好的数值稳定性，但会增加内存流量和运行时间；而单精度具有更高的算术吞吐量和更低的内存需求，对图形处理器尤其有利，因为图形处理器的可用内存往往有限。如第 4.3 节和第 5.3 节所示，对于条件良好的问题，单精度可在误差水平可接受的情况下带来显著加速比。然而，在某些情况下，迭代求解器的收敛性可能会变差，这体现了现代离散偶极子近似实现中性能、可移植性与数值精度之间固有的权衡。

我们使用了各代码的最新版本（或提交）： `DDSCAT` v7.3.4、`IFDDA` v1.0.27 以及 `ADDA`v.1.5.0-alpha3（提交 `b03d648`）。为在基准测试中实现机器精度的交叉比较，我们进行了少量非侵入式修改；所有此类更改均记录于附录 A。构建配置（编译器、编译选项、库）与基准测试一并报告，以确保可复现性。

3. 精度、互操作性与指南

本节既是方法指南，也是性能基准测试的技术前提。我们首先分析DDA模拟中数值误差的主要来源，并阐明在此语境下“精度”的含义。随后，我们概述用于对齐不同 DDA 代码参数的互操作性方法，并提供实现一致、可复现且高精度结果的实用指导。

3.1. 理解 *DDA*模拟中的精度

DDA模拟的精度取决于若干相互作用的因素。DDA 是完全确定性的，因此其误差可以分解为明确界定的类别。理解这些误差来源，对于解释模拟结果之间观察到的差异，以及评估不同实现之间可达到的一致性水平至关重要。

- **迭代收敛误差**。 用于获得内部电场的迭代求解器必须达到指定容差 η，其定义为残差的相对范数 $\|\mathbf{r}\|/\|\mathbf{b}\|$。场及导出量（例如消光截面）中相应的相对误差通常为 η，与下文的方法误差相比通常可忽略不计（对于所使用的 $\eta = 10^{-4}$ 或 10^{-5}默认值）。不过，若为加快计算而增大 η，它仍可能成为问题 [31]。

• **方法误差**。 它表示离散偶极子近似相对于精确电磁解的固有精度。它由两个主要贡献组成:

– 离散化误差 源于用有限数量的体素 N近似目标体。通过增加离散网格的规模，例如增加 x 轴上的体素数量 n_x，或等效地减小体素间距 d [42]，可以提高精度。– 形状误差 源于离散化几何形状与实际粒子形状不匹配。除网格细化外，还可通过加权离散化 [43] 减小此误差。

所需或预期的方法误差在很大程度上取决于应用程序。通常，对于积分散射量，1% 的误差被认为可以接受；对于角分辨散射量，误差高达 10% 也可能可以接受。不过我们强调，完全可以达到更高的精度，而且已有研究展示了低至 10^{-7} 的相对误差 [44]。

• **舍入误差**。 有限精度算术会导致舍入误差在多次迭代中累积。当矩阵–向量乘积的次数和迭代次数 N_{it} 都很大时，这些微小的数值波动（单精度和双精度下分别约为 10^{-7} 和 10^{-16} ）会不断累积，最终使匹配位数受限于收敛阈值 η所设定的水平。

在实践中，比较不同实现或数值设置时，可以识别出两种典型的一致性水平:

• **机器精度一致性**。 当所有模拟参数在各代码之间保持一致时，通常可以达到接近机器精度的一致性，即双精度下匹配位数为 10–15 位。这构成了确定性离散偶极子近似求解器可复现性的上限。

• ***η*级匹配**。 比较物理设置相同、但求解所得线性系统的方法不同的各代码结果时，一致性会受到后者的限制。

了解这些不同的误差来源，为下一小节所述的跨代码验证方法奠定了基础。在实践中，只有使用相同的自由

参数（见第 3.4 节）、选择相同的精度模式，并且线性系统约定（见第 3.2 节）一致时，才能实现机器级一致性。如果其中任何条件不同，最多只能期望达到 η 级一致性（见第 3.3 节）；但在某些情况下，只能达到方法误差范围内的一致性。另请注意，对于大多数物理应用，η 级一致性已足够，也可视为满足软件测试的要求，因为 η 可以在很大的动态范围内变化。

3.2. 不同的线性方程组（*ADDA*、*IFDDA*、*DDSCAT*）

本节旨在确定这样的条件：不同 DDA 实现中使用的线性方程组不仅彼此等价（这一点始终成立），而且还会产生完全相同的标准迭代方法收敛轨迹。这体现了 η 级一致性与计算量达到机器精度一致性之间的区别。

下文中的所有量均采用国际单位制表示。所有介电常数和折射率均以真空背景为参照。相对介电常数张量定义为

$$\overline{\boldsymbol{\varepsilon}}(\mathbf{r}) \stackrel{\text{def}}{=} \begin{cases} \bar{\mathbf{I}}, & \mathbf{r} \in V_{\text{ext}}, \\ \overline{\boldsymbol{\varepsilon}}_{\text{p}}(\mathbf{r})/\varepsilon_0, & \mathbf{r} \in V_{\text{int}}, \end{cases} \tag{1}$$

其中，ε_0 是真空介电常数，$\boldsymbol{\varepsilon}_{\text{p}}(\mathbf{r})$是粒子的介电常数。复相对折射率和电极化率定义为

$$\overline{\mathbf{m}}(\mathbf{r}) = \sqrt{\overline{\boldsymbol{\varepsilon}}(\mathbf{r})}, \quad \overline{\boldsymbol{\chi}}(\mathbf{r}) \stackrel{\text{def}}{=} \overline{\boldsymbol{\varepsilon}}(\mathbf{r}) - \bar{\mathbf{I}}.$$

真空中的波数定义为 $k = \omega\sqrt{\varepsilon_0\mu_0}$，其中 μ_0 是真空磁导率。自由空间并矢格林函数为：

$$\overline{\mathbf{G}}(\mathbf{r},\mathbf{r}') = \frac{\exp(\mathrm{i}kR)}{4\pi R}\left[\left(\bar{\mathbf{I}} - \frac{\mathbf{R}\otimes\mathbf{R}}{R^2}\right) + \frac{\mathrm{i}kR-1}{k^2R^2}\left(\bar{\mathbf{I}} - 3\frac{\mathbf{R}\otimes\mathbf{R}}{R^2}\right)\right], \tag{2}$$

其中，$\mathbf{R} = \mathbf{r} - \mathbf{r}'$、$R = \|\mathbf{R}\|$，并矢为 $\mathbf{R}\otimes\mathbf{R}$。

在提及特定于代码的线性方程组时，我们使用上标来区分不同系统：$\mathbf{A}^p$ 表示标准极化形式 [9](DDSCAT [10])，$\mathbf{A}^x$ 表示对称化形式 [7] (ADDA [16])，以及$\mathbf{A}^E$ 表示内场形式 [6](IFDDA [17])。

3.2.1. 标准*DDA*（极化形式，例如 *DDSCAT*）

在国际单位制中，标准极化形式表述 [7], 经离散化后所求解的线性系统为：

$$\overline{\boldsymbol{\alpha}}_i^{-1}\mathbf{p}_i - k^2\sum_{j\neq i}\overline{\mathbf{G}}_{ij}\mathbf{p}_j = \mathbf{E}_i^{\mathrm{inc}} \tag{3}$$

其中，$\overline{\mathbf{G}}$ 为式（2）中定义的自由空间二阶格林张量，$\boldsymbol{\alpha}$ 为极化率张量。请注意，与常规 SI 定义相比，体素矩$\mathbf{p}_i$ 和极化率$\boldsymbol{\alpha}_i$ 均按 $1/\varepsilon_0$ 的因子进行缩放（参见参考文献 [7]中的式（10））， 以确保与 CSG 约定下的线性形式一致。后者与式（3）的区别仅在于缺少 k^2因子（参见参考文献 [25]中的式（20））。

该式可写成紧凑的矩阵形式：

$$\mathbf{A}^p\mathbf{p} = \mathbf{E}^{\mathrm{inc}} \tag{4}$$

其中，矩阵元素$\mathbf{A}_{ij}^p$ 定义如下：

$$\mathbf{A}_{ij}^p = \begin{cases} \overline{\boldsymbol{\alpha}}_i^{-1}, & i=j \\ -k^2\overline{\mathbf{G}}_{ij}, & i\neq j \end{cases} \tag{5}$$

对于任意复对称的 $\boldsymbol{\alpha}$，该矩阵$\mathbf{A}^p$ 均为复对称矩阵 —*i*.

3.2.2. 对称化形式（变量变换，例如 *ADDA*）

基于极化率张量平方根的变量变换 [7,45]:

$$\mathbf{b}_i = \overline{\boldsymbol{\beta}}_i\mathbf{E}_i^{\mathrm{inc}}, \quad \mathbf{x}_i = \overline{\boldsymbol{\beta}}_i^{-1}\mathbf{p}_i \tag{6}$$

其中 $\boldsymbol{\beta}_i = \boldsymbol{\alpha}_i^{1/2}$ 。代入标准DDA形式后，得到变换后的系统：

$$\mathbf{A}^x\mathbf{x} = \mathbf{b} \tag{7}$$

其矩阵元素如下：

$$\mathbf{A}_{ij}^x = \begin{cases} \mathbf{I}, & i=j \\ -k^2\overline{\boldsymbol{\beta}}_i\,\overline{\mathbf{G}}_{ij}\,\overline{\boldsymbol{\beta}}_j, & i\neq j \end{cases} \tag{8}$$

对于任意复对称 $\boldsymbol{\alpha}_i$，该矩阵$\mathbf{A}^x$ 均为复对称矩阵，且对角线元素均为 1（即经过雅可比预条件化）。

3.2.3. 内场形式（例如， *IFDDA*）

局域内电场在国际单位制下满足：

$$\mathbf{E}_i^\ell = \mathbf{E}_i^{\mathrm{inc}} + \sum_{j\neq i} k^2 \overline{\mathbf{G}}_{ij}\, \overline{\boldsymbol{\alpha}}_j\, \mathbf{E}_j^\ell \tag{9}$$

与参考文献 [6] 中的 CGS形式相比，线性系统的结构相差一个因子 k^2 ，这是由于 $\mathbf{G}$ 的变换所致。极化率 $\boldsymbol{\alpha}_i$相对于其严格的国际单位制定义按因子 $1/\varepsilon_0$ 缩放（如前文所述）。此外，参考文献 [6] 将电极化率定义为$\boldsymbol{\chi} = \frac{1}{4\pi}(\boldsymbol{\varepsilon}_r - \mathbf{I})$，并使用不含 $1/4\pi$ 前因子的格林张量；而我们采用标准的国际单位制定义 $\boldsymbol{\chi} = \overline{\boldsymbol{\varepsilon}}_r - \mathbf{I}$ ，并将 $1/4\pi$因子直接纳入格林张量（见公式（2））。

这里的$\mathbf{E}_i^\ell$ 是局域场，也称为微观场 [46] （根据 DDSCAT，参见手册 [47] 第25节），或激发场 [25] $\mathbf{E}^{\mathrm{exc}}$；它对应于总场 $\mathbf{E}_i$ ，但不包含体素对自身产生的场。对于非磁性散射体（$\mu = 1$），连接这两个场的一般公式为 [25]:

$$\mathbf{E}_i^\ell = [\overline{\mathbf{I}} + (\overline{\mathbf{L}}_i - \overline{\mathbf{M}}_i)\overline{\boldsymbol{\chi}}_i]\mathbf{E}_i. \tag{10}$$

这里$\mathbf{L}_i$ 是自项并矢；对于立方（或球形）体素，它等于$\mathbf{I}/3$ ；而$\mathbf{M}_i$ 是有限尺寸修正，其取值决定了具体的极化率方案。例如，假设$\mathbf{M}_i = 0$对应于所谓的 DDA弱形式 [48], ，由此得到：

$$\mathbf{E}_i^\ell = \mathbf{E}_i \frac{\overline{\boldsymbol{\varepsilon}} + 2\overline{\mathbf{I}}}{3}. \tag{11}$$

因此得到克劳修斯–莫索提极化率。

请注意，三种代码均输出总场，而 IFDDA还会提供局域场。不过，迭代求解器的初始化方式可能有所不同。 IFDDA 始终根据局域场定义初始场，而 ADDA 则可根据局域场或总场进行定义，具体取决于其-init_field 模式（参见其手册 [49]第 12.1 节）。此外， IFDDA会使用 Heged¨us 技巧 [50] 对该场进行缩放。

用矩阵记法表示为：

$$(\mathbf{I} - \mathbf{G}\mathbf{D}_\alpha)\, \mathbf{E}^\ell = \mathbf{E}^{\mathrm{inc}}. \tag{12}$$

He其中，$\mathbf{D}_\alpha = \mathrm{diag}(\boldsymbol{\alpha}_j)$ 是由极化率构成的块对角矩阵。为了

与前两种形式保持一致，我们可以定义矩阵 $\mathbf{A}^E = \mathbf{I} - \mathbf{G}\mathbf{D}_\alpha$，其中矩阵的元素为：

$$\overline{\mathbf{A}}_{ij}^E = \begin{cases} \mathbf{I}, & i = j \\ -k^2 \overline{\mathbf{G}}_{ij}\, \overline{\boldsymbol{\alpha}}_j, & i \neq j \end{cases} \tag{13}$$

该矩阵 $\mathbf{A}^E$ 始终具有单位对角线，但仅当所有 $\boldsymbol{\alpha}_i$ 均为相同的标量时才是复对称矩阵。

3.2.4. $\mathbf{A}^x$、$\mathbf{A}^E$与$\mathbf{A}^p$之间的关系

对于均匀各向同性目标体，极化率可简化为标量 $\boldsymbol{\alpha}_i = \alpha\mathbf{I}$，适用于所有体素。在这一特殊情况下，三个矩阵可简化为

$$\mathbf{A}^x = \mathbf{A}^E = \alpha \mathbf{A}^p, \tag{14}$$

它们仅相差一个常数缩放因子。然而，这些线性系统所采用的主未知量（$\mathbf{E}^\ell$、$\mathbf{p} = \boldsymbol{\alpha}\mathbf{E}^\ell$和$\mathbf{x} = \boldsymbol{\beta}^{-1}\mathbf{E}^\ell$）以及源项（$\mathbf{E}^{\mathrm{inc}}$和$\mathbf{b} = \boldsymbol{\beta}\mathbf{E}^{\mathrm{inc}}$）也各不相同，其中 $\boldsymbol{\beta} = \boldsymbol{\alpha}^{1/2}$。因此，在计算残差时：

$$\mathbf{r} = \frac{\|\mathbf{b} - \mathbf{A}\mathbf{x}\|}{\|\mathbf{b}\|} \tag{15}$$

对于这些系统，标量因子会相互抵消，因此解在机器精度范围内完全一致；仅因舍入误差和运算顺序而产生极小差异。

然而，对于各向异性或非均匀材料，极化率不再是标量，且可能随体素而异，这会破坏$\mathbf{G}$ 与$\mathbf{D}_\alpha$（或$\mathbf{D}_\beta$）的可交换性。因此，$\mathbf{A}^x$、$\mathbf{A}^E$和$\mathbf{A}^p$ 不再仅通过简单缩放相互关联，其残差也无法仅通过变换主未知量或源项来协调。尽管如此，所有系统的目标都是近似求解同一个物理散射问题；由于离散偶极子近似是一种确定性方法，精确解只有一个。因此，计算所得解之间的差异主要由求解器容差决定，一致性达到 η 级。

ADDA支持各向异性材料，但前提是目标体的介电张量在所选坐标系中为对角形式。另一方面， DDSCAT可通过构造旋转后的介电张量处理非对角分量：用户提供（在局部介电坐标系中的）对角元素以及旋转角，DDSCAT则计算

非对角分量（参见手册 [47]第28节）。不过，它假设介电张量是对称的。相比之下， IFDDA 支持完全一般的介电张量，包括非对角元素；但各向异性材料仅支持与克劳修斯–莫索蒂或辐射反作用修正 [9]极化率结合使用。当然，所有三种代码都支持非均匀材料。

3.3. 自由参数对模拟一致性的影响

尽管离散偶极子近似是一种确定性方法，但自由参数的细微差异也可能显著影响独立运行之间的数值一致性。为量化这种敏感性，我们在两个其他方面完全相同的离散偶极子近似模拟中分别改变单个参数，并比较所得的消光效率和内部场。两组结果之间的匹配位数可作为衡量一致性的实用指标；对于双精度数，10–15 位数字对应完全达到机器精度。表 2 汇总了当不同代码或配置中仅有某一参数不同时，各参数类别预期的一致性水平。

表 2：在不同离散偶极子近似代码中改变单个参数时，预期的一致性（以匹配位数表示）。

参数类别	描述		匹配位数
研究系统	粒子形状、折射率、波长		0
Grid	体素的数量和尺寸		1–2[a]
DDA公式	极化率模型和相互作用公式-化（例如，LDR、FCD、IGT）		1–2
求解器参数	迭代求解器的类型和阈值 η	收敛	η 级
初始场	迭代求解器的初始猜测（例如零，WKB 或预条件化）		η级

[a]这表示方法误差的典型值（见第 3.1 节），实际误差也可能显著更小。

如表 2 所示，所考虑的散射问题若存在差异，会立即破坏可复现性，通常导致匹配位数为零。DDA公式或计算网格的差异通常只能得到一到两个有效数字；位数的量化方式为：相对误差为 10^{-x} 时，对应 x 位一致性。求解器相关参数（算法选择、停止准则 η 或初始场）

通常会将一致性水平控制在 η 位左右。例如，使用 $\eta = 10^{-4}$ 时，一致性通常仅限于四位。

只要在双精度下将求解器容差收紧至 10^{-16}，即使在 η 量级也能达到完全的机器精度一致性。

3.4. 默认参数与统一命令行设置

为实现独立 DDA 代码之间的浮点一致性交叉验证，必须统一所有相关的数值参数和物理参数。尽管 IFDDA、ADDA 和 DDSCAT 具有相似的理论基础，但它们的默认配置在目标定义、偏振处理、数值约定和求解器参数方面存在显著差异。表3汇总了各代码的主要默认值。必须明确消除这些差异，才能实现一致且可复现的模拟。

尽管这三种实现默认都采用平面波入射场，但在偏振处理和坐标约定方面仍存在差异。 ADDA 和 DDSCAT 均可同时计算两种正交偏振，而 IFDDA 目前每次运行只处理一种偏振。传播方向和偏振方向在默认情况下也有所不同。此外，极化率模型和求解器选择差异很大，若不明确对齐，会导致迭代收敛和精度出现变化。

表3：各 DDA 代码的默认参数。必须消除差异，才能实现机器精度等价性。

参数	IFDDA	ADDA	DDSCAT
形状	球体	球体	16球体聚集体
Size	$R = 100$纳米	$D \approx 6.734552$ 微米[a]	$a_{\text{有效}} = 0.25198$微米
入射场	平面波	平面波	平面波
传播方向	$\hat{\mathbf{z}}$	$\hat{\mathbf{z}}$	$\hat{\mathbf{x}}$
偏振方向	$\hat{\mathbf{y}}$	$\hat{\mathbf{y}}$	$\hat{\mathbf{y}}$
目标取向	zxz-约定	zyz-约定	xzx-约定
波长 λ	632.8 nm	2π µm	0.6 µm
折射率	$\varepsilon = 1.1$	$m = 1.5$	$m = 1.33 + 0.01i$
极化率	RR	LDR	CLDR
网格分辨率 n x	10	16	24
求解器	GPBiCG1	QMR	PBCGS2
收敛阈值 η	10^{-4}	10^{-5}	10^{-5}
初始场	Born	自动	零场
体积修正	none	已启用	已启用
散射公式	Draine公式 [9]	Draine公式	Draine公式
单位制	SI输出（含代码方程-（CGS单位制中的.....）	CGS	CGS

[a]当 λ 和 m 取默认值时，等于 $(6N/\pi)^{1/3}\lambda/(10|m|)$；而对于 n_x 取默认值的球体，则为 $N = 2176$。

为便于开展系统性基准测试，我们开发了一个 Python 封装程序，使 DDSCAT 能够直接从命令行 [51]运行。借助

该程序，我们可以构造采用一致物理参数和数值参数的最简命令行示例，使三个代码之间达到机器精度级的一致性：

```
$./adda -lambda 632.8 -size 200 -init_field zero -grid 15 $./ifdda -
```

请注意，用户仍需手动创建或配置 DDSCAT 中用于折射率的输入文件。在上文及下文的命令行中，我们采用 DDSCAT 风格的约定，根据其数值为这些文件命名。

附录 C 提供了包含所有等效参数及其对应精度一致性的表格。对于所有命令行示例，我们利用 DDSCAT 和 ADDA 允许所有长度采用任意物理单位（对所有输入和输出保持一致）这一事实，将输入波长和粒子尺寸表示为纳米，以符合 IFDDA所提出的约定。此选择仅用于提高可读性和可用性，不会影响迭代收敛过程。此外，代码之间的等价性以及尺度不变规则 [52] 通常表明， IFDDA同样具有这种灵活性，但在解释输出时应当谨慎。例如，如果输入单位不是纳米，则所得截面的单位不会是 m^2 。

4. 性能比较：CPU 基准测试

本节介绍针对三个 DDA 实现开展的性能基准测试： IFDDA、 ADDA和 DDSCAT。三个代码均进行 CPU 基准测试，而 GPU 基准测试仅限于 ADDA 和 IFDDA。主要目标是在相同的物理和数值条件下评估运行时间、内存占用和精度影响。值得注意的是，DDSCAT和 IFDDA 分别支持在中央处理器和图形处理器上进行单精度运行，从而进一步揭示性能与精度权衡。

为确保比较公平，所有基准测试的配置都旨在使三种实现之间达到机器精度一致性。这一要求对模拟设置施加了严格限制，因为并非所有参数组合或求解器选项都能在每种代码中使用。

因此，我们聚焦于一个定义明确的单一案例：使用 BiCGStab迭代求解器和 FCD相互作用与极化率公式求解尺寸为$kD = 30$（D 为边长）的冰均匀立方体粒子。此配置具有典型的离散偶极子近似工作负载代表性，参考文献 [20] 中也采用了该配置。

如第2节所述，选择 BiCGStab 是为了避免不同代码中准最小残差法实现之间算法复杂度相差两倍。不过，BiCGStab 的实现也在停止流程上略有差异。原始算法 [53] 可以在迭代过程中途停止，从而省去一次矩阵向量乘积。 ADDA实现了这一点，但 PIMZ [54] 版本的 DDSCAT 和 IFDDA并未实现。是否会出现这种情况，取决于问题参数，且具有准随机性。以下测试案例经过专门调整，以避免这一问题。相比之下，其他散射问题可能只能达到 η级别的一致性，并因一次矩阵向量乘积而产生额外的计算时间差异。因此遗憾的是，目前还没有适用于所有三种代码、可进行稳健比较的理想迭代求解器。不过，成对比较时存在这样的选项（见表 C.1）。

由于浮点舍入误差，不同代码和硬件后端的收敛历史略有差异，因此对于固定的 η，迭代次数也会不同。这会降低 η级别的浮点一致性。为消除这一问题，我们针对每种实现专门调整了 η ，使所有中央处理器运行都在相同的迭代次数后终止（本例中为 54 次），从而使消光截面和效率达到 10–12 位匹配（见表 C.3）。

另一种更稳健的强制迭代次数一致的方法，是使用诸如 `-maxiter`（在 ADDA 中）、 `-MXITER`（在 DDSCAT中）或 `-nlim`（在 IFDDA 中）等选项限制最大迭代次数。然而，达到此限制时， ADDA 仍会计算物理可观测量，而 IFDDA 和 DDSCAT 都会终止执行，从而无法进行比较。因此，我们采用阈值调节作为一种实用的折中方案，但我们认为，迭代次数超过 100 次后，这种调节方式将变得不切实际。

由于我们的比较关注运行时间和求解器性能，而非角度量，因此我们减少了角度采样：在 DDSCAT 中，我们按手册建议将 `ETASCA = 10`设为相应值，并使用 `IORTH = 1` 计算单一偏振；对于 ADDA，这对应于 `-ntheta 10`。所使用的确切命令行如下：

```
$./ifdda -objectcube 2387.3241463784303 -lambda 500 -epsmulti 1.723969 0.0 -
```

内存占用随体素数量 N线性增长，因此可针对每个代码直接外推。相比之下，本章所报告的运行时间和加速比依赖于问题规模。如参考文献 [55, 7], 所述，每次迭代的时间并不随 N 严格线性缩放（例如，快速傅里叶变换会引入 $N \log N$ 行为），并且对快速傅里叶变换的实现、数值精度以及并行化范式高度敏感。这些效应也依赖于硬件。

对 N 进行全面扫描超出了我们目前的研究范围。因此，此处报告的加速比仅表征所选测试案例和硬件，在解释或外推到不同问题规模时应谨慎对待。

4.1. 计算配置

用于基准测试的CPU架构汇总于表4。它们分别对应于高性能Cray集群的单个节点和现代笔记本处理器。这一对比使我们能够说明在高性能计算硬件与日常桌面环境上可预期的性能潜力与实际限制。

所有集群任务均通过 SLURM 的 `SBATCH --exclusive`选项提交，以消除共享节点争用导致的性能波动。每项模拟均重复运行三次，以计算均值和样本标准差，评估各次运行之间的波动。

三种代码均使用相同的工具链进行编译，以尽量减少环境因素造成的偏差。编译器版本的影响可以忽略不计：从 `GCC` 11.2 切换到 `GCC` 12.2，所得结果在测量噪声范围内完全一致。在集群上，我们使用了以下软件栈： `GCC` 12.2、`OpenMP` 4.5、 `MPICH` 3.4a2、 `FFTW` 3.3.10 和 `Intel oneMKL` 2023.0；而在

表4：基准测试所用的CPU平台。缓存容量按每个插槽汇总。内存带宽为每个插槽 的理论峰值（十进制 GB/s）。

	AMD EPYC 9654（热那亚）	Intel Core Ultra 7 165H
插槽	2	1
核心 / 线程	96 / 192（每插槽）	16 / 22（8 个 E + 6P 核心）
基础频率 / 睿频（GHz）	2.4 / 3.7	1.4 / 5
一级指令缓存 / 一级数据缓存	3 MiB / 3 MiB	896KiB / 544 KiB
L2缓存 / L3缓存	96 MiB / 384 MiB	18MiB / 24 MiB
峰值内存带宽	460.8 GB/s (DDR5-4800)	120 GB/s（LPDDR5x-7467）
随机存取存储器（节点）	768吉比字节（24×32 吉比字节）	2×32吉比字节

笔记本电脑上： GCC 11.4、 OpenMP 4.5、 Open MPI 4.1.2、 FFTW 3.3.8 和 InteloneMKL 2025.0。 ADDA使用 -Ofast 优化标志编译， IFDDA使用 -O3，而 DDSCAT使用 -O2，均遵循各自默认的构建设置。

以下小节将介绍中央处理器的测试结果，而图形处理器的基准测试结果将在另一节中单独报告。每项测试均报告总运行时间、加速比、内存占用和精度效应。

4.2. 时间性能

我们测试了多种网格和 FFT 分解，发现网格尺寸 $n_x = 150$ 和 $n_x = 250$ 最能发挥所有代码的性能。对于所有 CPU 基准测试，MPI 进程数均按照 ADDA推荐的并行 FFT 实践进行选择，即该数值可整除 n_x。图 1 和图 2 分别报告了 AMD EPYC 9654 节点和 Intel Core Ultra 7 165H 处理器上的墙钟时间和加速比。误差棒表示样本标准差，仅当相应的变异系数超过 5% 时才显示。

本节仅报告总墙钟时间，因为它主要由迭代求解器阶段决定，而该阶段又主要由矩阵-向量乘积（即快速傅里叶变换）决定。相比之下，对于图形处理器基准测试，求解器阶段是卸载到设备上的部分，因此其耗时可能与留在中央处理器上执行的部分耗时相当。

与 DDSCAT手册 [47], 一致，MKL 在现代中央处理器上的表现优于旧版 GPFA 后端，尤其是在英特尔处理器上，因为 MKL 针对英特尔微架构进行了大量优化。因此， DDSCAT MKL在英特尔笔记本电脑上的性能约为其 GPFA 构建版本的两倍，而在 AMD EPYC 节点上的提升则较为有限。这使DDSCAT MKL 在单核运行中能够与 ADDA FFTW和 IFDDA FFTW 相媲美

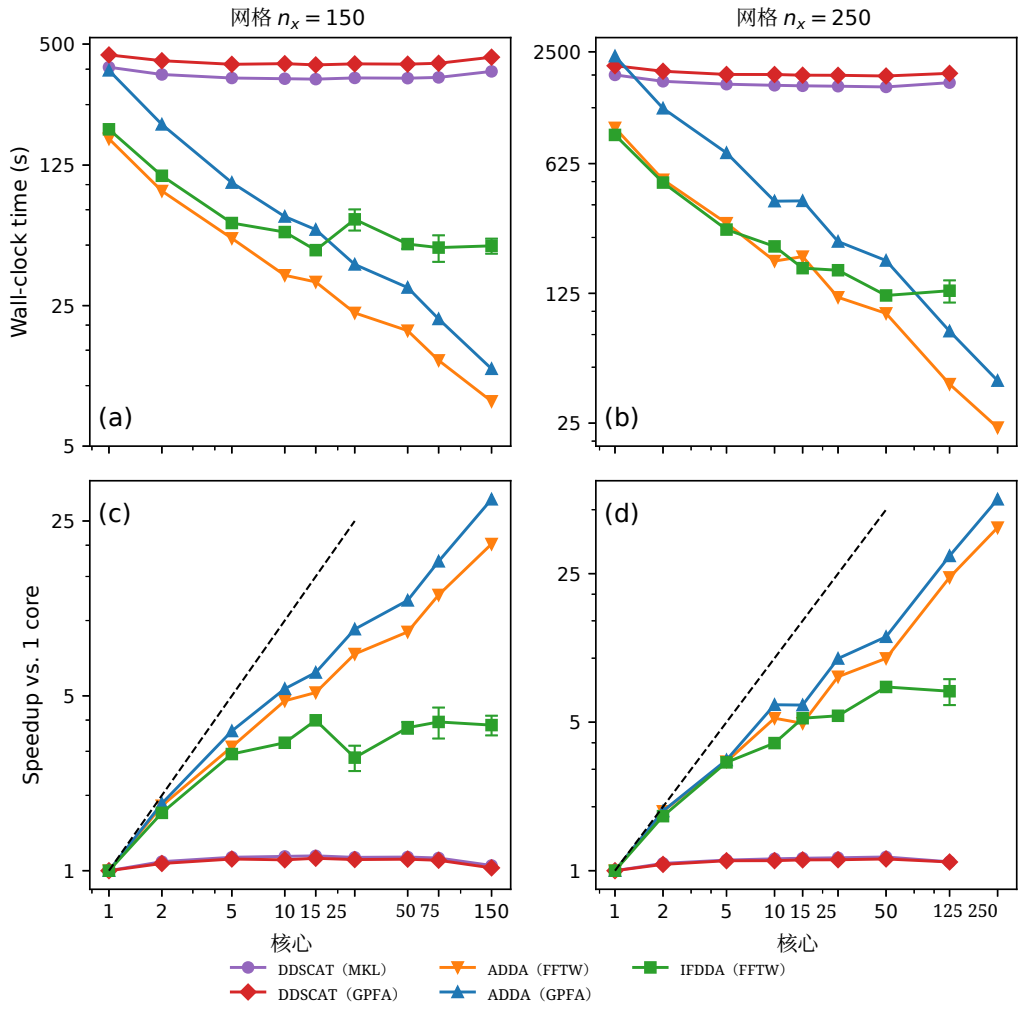


图1：AMD EPYC9654 节点上网格尺寸为 $n_x = 150$ （左）和 $n_x = 250$ （右）时的墙钟时间（上排）以及相对于单核基准的加速比（下排）。采用双对数坐标，虚线曲线表示理想的线性扩展。当样本标准差超过数值的 5% 时，图中会显示误差棒。

在较大网格的英特尔笔记本电脑上则是如此。不过，在 AMD 节点上， `ADDAFFTW` 和 `IFDDA FFTW` 仍然显著更快，分别以约 2.5 倍和 2.0 倍的速度领先于 `DDSCAT MKL`，对应 $n_x = 150$ 和 $n_x = 250$。这些差异在意料之中。事实上，虽然 `DDSCAT`在一次运行中执行完整的三维快速傅里叶变换 [56]，但 `ADDA` 通过……减少了快速傅里叶变换的总计算量

将三维卷积分解为一系列一维快速傅里叶变换，并在过程中进行中间转置。这种三维到一维策略最初见于参考文献 [57], ，通过显式利用卷积固有的零填充，将所需的一维快速傅里叶变换数量几乎减少了一半，同时保持较高的并行效率，并优化内存使用和执行时间。在本研究期间， IFDDA 也进行了修改，采用了与 ADDA类似的三维到一维实现，但主要针对执行时间进行了优化（关于内存差异的更多细节见第 4.4 节），因此在此处实现了约 10% 的性能提升。 OpenDDA [55]中也有相关改进。最后，具备硬件自适应规划功能的 FFTW库在所有受测硬件上均表现出最佳性能，这与此前的代码间比较结果一致 [20]。

对于多核CPU运行， DDSCAT 的扩展性能基本持平，因为其快速傅里叶变换例程占总运行时间的最大部分，却没有使用 OpenMP 进行并行化。 DDSCAT 的并行化主要针对取向平均（不在本研究范围内），在这种情况下可以观察到大得多的加速比。相比之下， ADDA 和 IFDDA在 AMD 节点上均表现出较好的扩展性，加速比最高分别达到 $24\times$ （使用 125 个核心）和 $7\times$（对于 $n_x = 250$）。在这一架构上，对于 $n_x = 150$ ， ADDA FFTW 的速度仍比 IFDDA FFTW 快 $5\times$ 以上；对于 $n_x = 250$ ，则快 $3\times$ 以上（使用 125 个核心）。此外，我们观察到，OpenMP 的性能变化比 MPI 更大。不过，在英特尔笔记本电脑上， IFDDA 的 $n_x = 250$ 运行时间更短、扩展性更好，加速比最高达到 $1.5\times$。

要解释这些行为，有必要讨论饱和效应。如图 1 所示， ADDA和 IFDDA 在核心数达到 2–3 个之前几乎呈线性扩展，此后便开始偏离理想扩展。这种饱和现象源于阿姆达尔定律和内存带宽限制：随着更多线程争用数量固定的内存通道，通信开销逐渐降低并行效率。近期一项使用 OpenMP 和 MPI 对 FFTW后端进行比较的研究也报告了类似趋势 [58]。 $n_x = 150$ 和 $n_x = 250$之间不同的饱和点凸显了三维快速傅里叶变换的网格相关扩展性。较大的网格能够在更长时间内保持较高的并行效率，因为其计算强度的增长速度快于同步和通信开销的增长速度。

在英特尔笔记本电脑上，这些影响出现得要早得多。可用内存带宽显著更低，缓存层次结构也更小（见表4），导致三维快速傅里叶变换的工作集溢出到主存。因此，尽管时钟频率更高使单核心性能略有提升，加速比仍在达到10个核心之前就已趋于饱和。这也解——

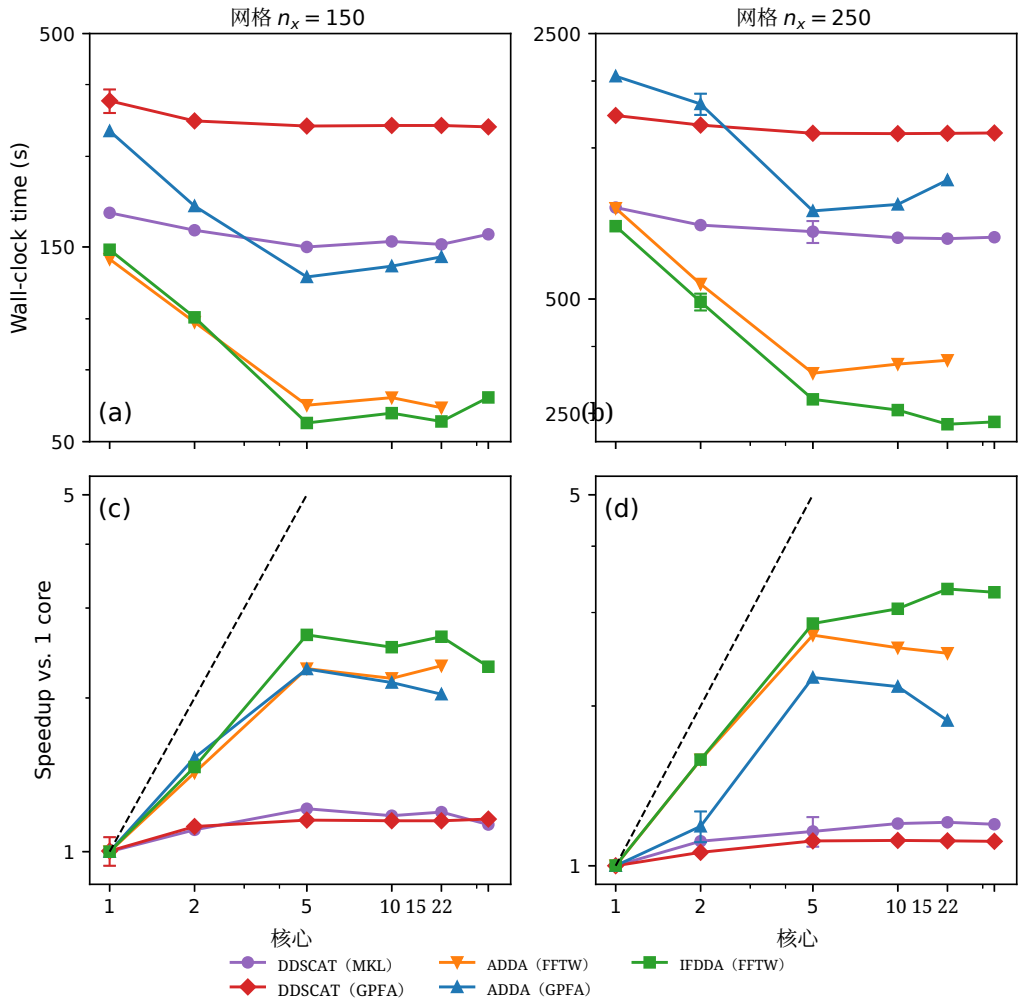


图2：在 Intel Core Ultra 7 165H 处理器上，网格规模为 $n_x = 150$（左）和 $n_x = 250$（右）时的墙钟时间（上排）以及相对于单核基准的加速比（下排）。采用双对数坐标，虚线曲线表示理想的线性扩展。当误差棒（表示样本标准差）大于数值的 5% 时，予以显示。

释了为什么在 AMD 节点上，`ADDA GPFA`的扩展性优于 `ADDA FFTW`；而在英特尔笔记本电脑上则观察到相反趋势（尽管 FFTW 在所有情况下的运行时间都更短）。具体而言，有限的内存带宽和缓存容量使 FFTW 的自适应特性在英特尔笔记本电脑上并行执行时更具优势。这凸显了

FFT 性能高度依赖底层硬件架构和算法优化策略。

最后需要指出， ADDA FFTW在 250 个核心上执行 $n_x = 250$的墙钟时间是使用两个 AMD 节点测得的，因为单个 EPYC 9654 节点仅提供 192 个核心。这样做是为了说明， ADDA 的 MPI 实现能够继续跨节点扩展。使用更多节点、且每个节点的核心数更少时，运行时间还可以进一步缩短，这表明延迟效应仍是一个限制因素。未来值得开展更系统的研究，包括基于网格尺寸和硬件特性的饱和度预测器。

总之，用户应优先选择 FFTW 或 MKL，而非 GPFA 后端；对于大规模问题，则应使用 ADDA的 MPI 执行模式。

4.3. 精度效应

DDSCAT 的一项实际优势是支持单精度，可将数组占用空间减少一半。对于 $n_x = 150$ 情况，由于迭代次数与双精度时相同，单精度也能缩短每次迭代的时间。

从双精度切换到单精度后，笔记本电脑上使用 GPFA 和 MKL 的运行时间分别缩短了约 30% 和 60%，集群上则分别缩短了 10% 和 40%（未展示数据）。此外，精度仍然良好，结果与 η 位有效数字相符。

因此，对于此网格，在笔记本电脑的单核上，单精度运行速度可以快于 IFDDA 和 ADDA （使用 MKL 时为 115 秒）。总体而言，在 DDSCAT 中，单精度可大幅节省内存（减少一半）并缩短运行时间（使用 MKL 时最多缩短 60%），而对该案例的精度影响可以忽略不计。然而，这一点并无保证，尤其是在迭代次数增加时。因此，已有研究提出了更复杂的方法，仅对部分变量使用双精度 [20]。因此，用户在权衡性能提升与精度可能降低之间时应保持谨慎。

4.4. 内存性能

表5报告的内存占用采用国际电工委员会（IEC）推荐的二进制前缀：GiB 表示吉比字节（1 GiB $= 2^{30}$ B）。唯一的例外是表4中的内存带宽，采用十进制 GB。我们仅报告物理内存，即常驻集大小（RSS）。根据我们的经验，虚拟内存大小（VM/VSZ）

并不是衡量这些工作负载容量的有效指标：VM 始终为 ≥RSS，因为它包含完整的地址空间（已保留但尚未触及的页面）以及映射到进程中的共享库的完整尺寸；相比之下，RSS 仅统计实际驻留在 RAM 中的页面。

表5：网格尺寸为 $n_x = 150$ 时的节点级物理内存 [吉比字节] 。总量为每个节点的 RSS；对于 MPI（ADDA），该值根据 sacct 数据计算：将各进程排名的 RSS 内存取平均值，再乘以进程排名数。

方法	1	25	50	75	150
ADDA	2.5	3.0	4.2	5.4	12.5
IFDDA	6.5	6.5	6.5	6.5	6.6
DDSCAT	6.3	6.3	6.3	6.3	6.3

对于使用 OpenMP（IFDDA和 DDSCAT）的代码，线程共享单个进程，因此大型工作数组只有一份副本；随着线程数增加，RSS 基本保持不变（见表5）。然而，对于 MPI 程序，内存占用会随进程排名数增加而上升：尽管ADDA 在单个进程排名时的内存占用约少 2.5 倍，但在核心数较多时，总内存占用可能超过使用 OpenMP 的代码。这主要是 MPI 导致的。与共享内存方式相比，MPI 会启动 P 个独立进程。每个进程排名都保留某些 MPI 辅助变量的独立副本，导致内存随进程排名数增加而显著上升。在中等尺寸的运行中，这种 MPI 相关开销十分明显；但对于更大的网格，由于合理分布的工作数组占据主导地位，该开销便可忽略不计。附录 D 对 MPI内存扩展进行了详细分析。

ADDA 初始内存占用较低，源于相互作用矩阵在傅里叶空间中的存储方式。实际上，在 FFT 加速的 DDA 公式中，由于对称性，只需使用并矢格林张量的六个独立分量；这些分量经变换后存储在零填充域中，该域在每个空间方向上的尺寸均为原来的两倍。在经典方法中， [56], 所有六个分量都存储在整个循环域中。 ADDA 利用两个方向上的镜像对称性降低了这一开销，同时不利用第三个轴上的对称性，以便进行 MPI 并行化；与经典公式相比，这使存储需求减少了四倍。更多详情见参考文献 [55], ，其中还介绍了 OpenDDA 如何利用所有三个空间方向上的对称性，达到理论上的最小值

存储量。

在完整节点上（192 个核心，714 吉比字节随机存取存储器），我们能够运行 ADDA，直至 $n_x = 960$。此外，减少每个节点使用的核心数并扩展到多个节点，可以处理更大的网格。例如，文献 [59], 报告了近期一次大规模 ADDA 运行，使用 44 个节点（3168 个中央处理器核心），峰值内存占用为 27.6 TiB，MPI 开销仅为 0.4 TiB（即 1.2%），在尺寸参数为 $kD = 2000$ 、折射率为 $m = 1.02$的立方体上达到了 $n_x = 3168$ 。

5. 性能对比：图形处理器基准测试

IFDDA （v1.0.27）和 ADDA（v.1.5.0-alpha3，提交 b03d648）的图形处理器模式仍在积极开发中。除了版本控制之外，公开的 ADDA 问题跟踪器以及官方讨论组中近期的基准测试讨论，也体现了持续进行的开发工作。还值得一提的是，其中一位作者最近创建了分支 ADDA-CXX[1]，将原始 C99 代码库转换为 C++，以便于未来集成 CUDA（本研究未使用）。因此，本文报告的运行时间和内存占用反映的是当前的实现和测试配置，而非硬性上限。未来版本的发布将显著改变这些结果。

5.1. 计算配置

GPU 基准测试在四种英伟达加速器设备上进行，涵盖集群和工作站架构，分别为 A100（安培）、H200（霍珀）、RTX 6000 Ada 和 RTX 2000 Ada。其主要架构特性汇总于表 6。

近期的 NVIDIA 图形处理器集成了专用于矩阵乘加（MMA）的张量核心单元，可加速通用矩阵乘法（GEMM）运算。它们在混合／低精度下可提供极高吞吐量，在近期的 A100/H100 图形处理器上，标准精度（FP64 张量核心）下也能达到极高吞吐量。不过，标准快速傅里叶变换库（FFTW、cuFFT、clFFT）默认不会使用这些核心。将快速傅里叶变换重新表述为适合张量核心的形式的研究原型 [60] 报告称，在 NVIDIA A100 上相比 cuFFT 实现了 1.1–3× 的加速比，但我们未将其纳入考量。

[1] https://github.com/michelgross34/adda-CXX

表 6：基准测试中使用的图形处理器平台。峰值浮点吞吐量和内存带宽数值由厂商提供，缓存容量则取自第三方硬件数据库。Ada 图形处理器上的双精度浮点吞吐量为 FP32 峰值的 $1/64$ 。SM 表示“流式多处理器”。

	A100（80 吉比字节 SXM4）	H200（141 吉比字节 SXM）	RTX 6000 Ada	RTX 2000 Ada
双精度浮点（每秒万亿次浮点运算次数）	9.7	34	1.4	0.2
FP32（每秒万亿次浮点运算）	19.5	67	91.1	12
内存带宽（GB/s）	2,039	4,890	960	256
一级缓存 / 流式多处理器（KiB）	192	256	128	128
L2缓存（MiB）	40	50	96	12
SM	108	132	142	22

每个图形处理器均搭配相应系统所配备的主机处理器：A100 配备双路 AMD EPYC 7543 节点，H200 配备双路 Intel Xeon 8558 节点，RTX 2000 Ada 配备 Intel Core Ultra 7 165H 笔记本电脑，RTX 6000 Ada 则配备 Intel Xeon w5–3525 工作站。理想情况下，所有图形处理器都应在相同的主机CPU上进行基准测试，以隔离加速器性能的影响。但由于硬件条件所限，这无法实现。尽管如此，这并不影响比较的有效性，因为我们会在每次运行中分别记录 GPU驻留求解器时间和总墙钟时间。这样便能区分图形处理器与主机处理器各自的贡献，并评估图形处理器和中央处理器硬件对整体运行时间的影响。

为确保直接的跨平台可比性，我们采用了与中央处理器部分相同的数值配置。图形处理器双精度运行同样在 54 次迭代后终止，而带有`IFDDA` 的图形处理器单精度运行则在 53 次迭代后收敛，而非 54 次。这一差异是因为相邻残差之间的步长变得过大，无法像双精度那样进行调整。尽管如此，这并不影响本文的比较，因为单精度的精度本身仅限于 7–8 位有效数字，而相差一次迭代对第 5 节报告的测量执行时间影响微乎其微。

与中央处理器部分一样，报告的计时结果为三次相同运行的平均值。此外，所有代码均使用相同版本的库和优化标志进行编译，其中 `CUDA` 13.0 用于 `IFDDA` ，`OpenCL` 3.0 用于 `ADDA`。

5.2. 时间性能

对于 DDA 求解器，每次迭代的主要开销来自快速傅里叶变换阶段。尽管多年来进行了许多优化尝试， [61, 55, 62], 快速傅里叶变换仍然是

迭代求解器的瓶颈，其余时间主要花在线性代数运算（BLAS（基本线性代数子程序））上。在 IFDDA 中，整个求解器循环都卸载到图形处理器上，从而最大限度地减少主机与设备通信。相比之下， ADDA 仅将快速傅里叶变换（矩阵-向量乘积）卸载到图形处理器，而线性代数运算仍由中央处理器执行，因此会产生额外的数据传输。 ADDA 还提供实验性的 OCL_BLAS 模式，可通过 clBLAS库将 BLAS 运算卸载到图形处理器，从而既加速 BLAS 运算，又消除相应的数据传输。截至当前的 ADDA 版本，此模式仅支持双共轭梯度法求解器。由于本研究重点关注跨代码比较和浮点一致性可复现性，因此我们未将其纳入主要基准测试。不过，我们在附录 E 中报告了该模式的性能；它实现了数倍加速，展现了完全驻留于 GPU 的实现所带来的优势。

图 3 报告了在四款 NVIDIA 图形处理器（见表 6）以及两种 CPU 配置（1 个和 10 个核心）上，网格尺寸为 $n_x = 150$ 和 $n_x = 250$ 时的仅求解器耗时（迭代阶段）和总墙钟时间（端到端）。但在 NVIDIA RTX 2000 Ada 上，IFDDA 计算因内存不足错误而失败。IFDDA 分别以双精度（DP）和单精度（SP）运行，而 ADDA 仅报告 1 个核心的结果，因为其默认 GPU 模式目前与 MPI 互斥。对于 ADDA，我们还展示了快速傅里叶变换的耗时，这是求解器的一部分。与图 2 不同，为清晰起见，此处未显示样本标准差。所有情况下，相应的变异系数均低于 5%（通常约为 1%）；唯一例外是$n_x = 150$ 在 NVIDIA RTX 6000 Ada 上运行 IFDDA 单精度版本时的求解器平均运行时间，其变异系数达到 6%。

IFDDA DP 的整体图形处理器行为可在屋顶线模型 [63], 中加以解释。该模型将算术强度（FLOP/B）与机器平衡（峰值 FLOP/s 除以峰值带宽）联系起来。三维快速傅里叶变换的算术强度较低，且取决于问题规模；因此，DDA 迭代主要处于内存受限区间 [64], ，但也可能转入计算受限或延迟受限区域 [63]。对于像 IFDDA 这样完全驻留于设备上的求解器，预计求解器时间大致与内存带宽成反比。实测运行时间在定性上符合这一趋势：带宽更高的图形处理器，其求解器时间更短，H200 $>$ A100 $>$ RTX 6000 Ada $>$ RTX 2000 Ada。不过，似乎难以进行定量描述，因为不同设备的同步开销、缓存与内存路径以及 FP64/FP32 比率差异显著

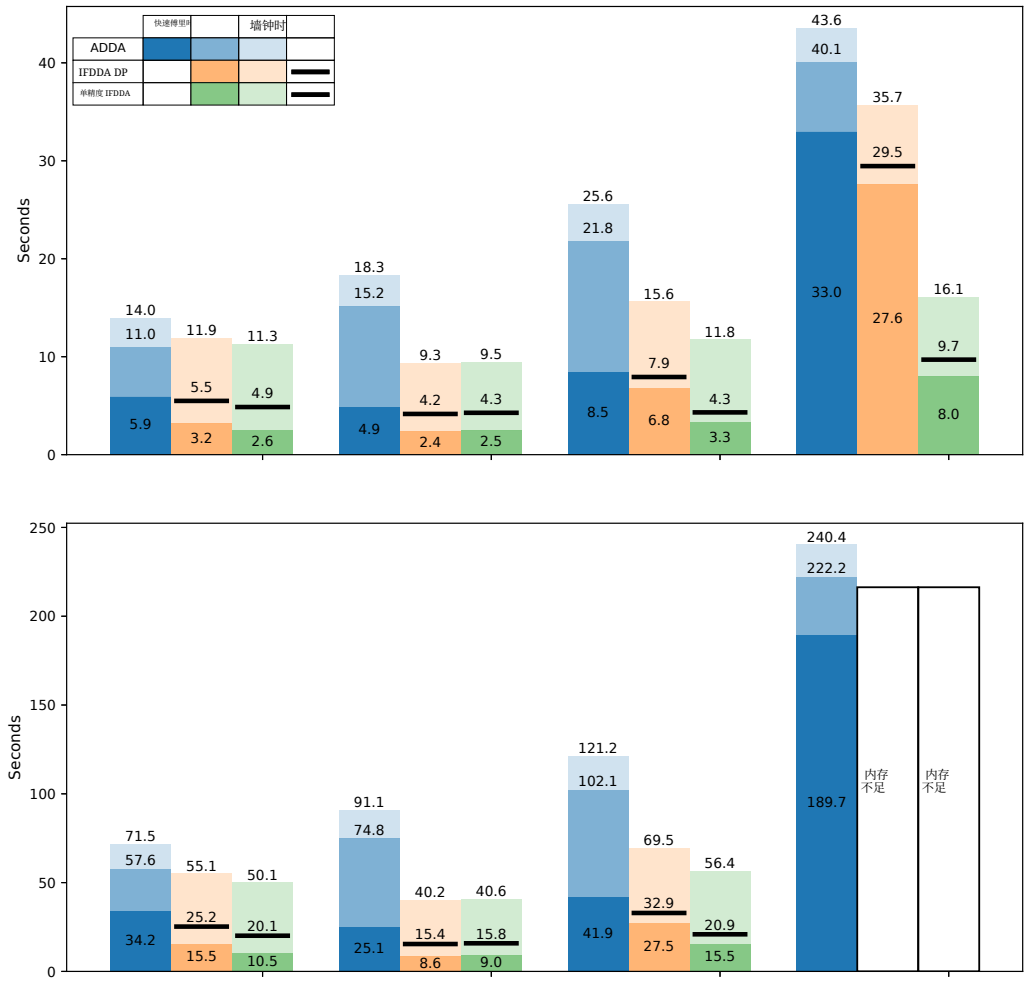


图 3：DDA 代码的图形处理器计时。柱形按代码分组，并以堆叠方式显示快速傅里叶变换时间、求解器时间和单核墙钟时间。黑色水平线表示 IFDDA（双精度/单精度）的 10 核墙钟时间。网格尺寸 $n_x = 150$ （上排）； $n_x = 250$ （下排）。

。这些因素会引入附加延迟项，导致性能下降。因此，H200 并未达到其带宽优势相较于 A100 或工作站 GPU 所预示的全部提升。

对于 `ADDA`，其趋势与 `IFDDA DP` 相似，但 H200 例外：其求解器和总运行时间均高于 A100。由于只有快速傅里叶变换在图形处理器上执行，每次求解器迭代都需要进行多次

主机与设备之间的数据传输和同步，因此性能对系统级延迟十分敏感。OCL_BLAS 模式（见图 E.1）提供了支持这一解释的证据：将整个求解器卸载到图形处理器后，趋势便恢复为预期情况。这表明， ADDA 中观察到的性能反转源于混合式中央处理器–图形处理器工作流中的延迟开销，而非图形处理器端的吞吐量。我们强调，这一现象无法用任何硬件规格来解释，因为与 H200 系统配套的内存层级和中央处理器在名义上都更优（见表 6）。这一现象的系统性研究留待未来开展。

在单核情况下， IFDDA DP 的性能始终优于 ADDA；即使只比较 GPU 卸载部分也是如此。原则上，由于仅执行快速傅里叶变换，这一部分理应更有利于 ADDA。这种现象可能由多种因素造成。首先，是前文讨论过的 CPU–GPU 延迟和同步开销。其次，在英伟达硬件上， cuFFT （用于 IFDDA）预计会优于 clFFT [65]。最后， ADDA 依赖于采用经验选定块尺寸的 FFT 计划转置，而这些块尺寸对于当前 GPU 架构而言可能并非最优。综合来看，仅凭绝对运行时间很难得出确切结论或计算比值。总墙钟时间也存在类似情况，因为 CPU 性能差异同样会产生影响。

不过，考虑到 BiCG 求解器中OCL_BLAS 模式在求解器部分实现的相对加速比（见附录 E），并将这一提升外推至 BiCGStab 情况，可以预期，当整个求解器循环都卸载到 GPU 时， ADDA 的性能可与 IFDDA 相当。独立实现也进一步支持了这一解释。一个 MATLAB 实现 MPDDA [62]将完整的迭代求解器卸载到 GPU，其运行时间优于默认的 ADDAOpenCL 模式，不过这一差异可能部分源于其使用了苹果公司的 OpenCL FFT 后端，而非标准的 clFFT 库。近期，一个采用类似策略并依赖 CuPy 的 Python 实现 CPDDA [66]报告称，在较小的晶格间距下，其性能超过了 MPDDA 。总而言之，这些结果支持将完整的迭代求解器卸载到 GPU 的优势。

最后， IFDDA 还可以通过增加 CPU 线程数来进一步缩短墙钟时间，即使在 GPU 模式下也是如此，因为其非快速傅里叶变换部分可受益于 OpenMP 并行。例如，使用 10 个 CPU 核心可使 A100 上的总运行时间减少 $2\times$ 。

5.3. 精度效应

对于快速傅里叶变换等带宽受限问题，从双精度切换到单精度理想情况下可将内存流量减半（从 8 字节降至 4 字节），有望获得 $2\times$ 加速比。对于计算受限区域，可达到的比例则受设备 FP64:FP32 吞吐量限制；而对于延迟受限区域，基本不会有加速效果，因为主要耗时在于等待数据，而非处理数据。

因此，测得的比值会因图形处理器在带宽、FP64 能力和开销之间的架构平衡不同而有所差异。在 A100 上，FP64 性能相对较强（为 FP32 的一半），且内存带宽较高。对于中等网格规模（$n_x = 150$–250），内核无法充分饱和内存子系统，因此观测到的 1.4–1.$5\times$ 加速比体现了带宽效应和延迟效应的综合影响。对于更大的网格（此处未展示），我们观测到的比值为 1.8。

在 RTX 6000 Ada 上，FP64 吞吐量极低（仅为 FP32 的 1/64），且内存带宽适中。因此，延迟造成的影响较小，这也解释了为何其 DP/SP 比值仍处于带宽受限区域，但达到 1.8–2.1，且对于 $n_x = 150$ 更接近计算受限区间。这一趋势在 RTX 2000 Ada 上更为明显，因为其 FP64 能力更低。在该设备上，DP 求解器显然更受计算限制，其表现为 DP/SP 比值达到 3.5，介于带宽受限和基于计算的数值之间。

H200 的表现有所不同。尽管它与 A100 的 FP64:FP32 峰值比相同（1/2），且 FP64 吞吐量和内存带宽都更高，但在 H200 上， `IFDDA DP` 和 `SP` 求解器时间几乎相同，甚至 SP 的求解器时间还更长。这表明系统处于延迟受限状态。与附录 E 中讨论的 `ADDA` 情况不同，这种表现不可能源于 CPU–GPU 传输，因为 `IFDDA` 完全驻留在设备上，而且比较的只是双精度与单精度。一种合理的解释是，在 H200 上，双精度求解器的性能已经接近实际性能上限，而这一上限并非由带宽决定，而是受快速傅里叶变换内核侧的内部延迟开销所限。

总体而言，结果表明，在 FP64 吞吐量有限且内存带宽较低的工作站 GPU 上，精度选择会显著影响求解器运行时间和总运行时间；只要迭代收敛允许，采用单精度即可获得显著提升。然而，在高带宽集群 GPU 上，如果求解器接近延迟受限区域，切换到单精度的收益可能会减少，甚至消失。

本节将 NVIDIA A100 上获得的最佳 GPU 运行时间与 AMD EPYC 节点上测得的最佳 CPU 结果进行比较。由于两种网格尺寸对应的可用核心数不同，CPU 参考结果定义为 CPU 基准测试中墙钟时间最短的最快配置：对于 `IFDDA`，$n_x = 150$ 使用 75 个核心；对于 `ADDA`，$n_x = 150$ 使用 150 个核心；对于 n_x =250，则分别使用 50 和 125 个核心，以避免利用 `ADDA` 多节点实现。

对于 $n_x = 150$，`ADDA` 的最佳中央处理器运行时间为 8.3 s，而 A100 需要 14s。因此，`ADDA` 在 OpenCL 模式下的速度比相应的多核 CPU 执行慢约 1.7× 。相反，`IFDDA`受益于 GPU 加速：双精度 GPU 耗时快 4.1× ，单精度则实现了 4.3×的加速比。

在 $n_x = 250$，`ADDA` 在 GPU 上的速度比 125 核中央处理器运行慢 1.8× ，而 `IFDDA DP` 和 `IFDDA SP` 分别实现了 2.2× 和 2.4× 的加速比。

对于 A100 上的 `ADDA OCL_BLAS` 模式，相较于 125 核中央处理器运行（针对 BiCG 求解器计算），我们观察到 1.3× 的加速比。然而，对于非常大的网格（例如 $n_x = 400$），纯 CPU MPI 运行会更快，尽管基于 GPU 的求解器本身快 15–20%。在这种情况下，性能受初始化阶段限制；该阶段在 OCL 模式下未并行化，因此主导了总体运行时间。在 OCL 初始化阶段引入 MPI 并行，或理想情况下将其卸载到 GPU，可消除这一限制，但会增加内存占用并降低可实现的最大网格尺寸。

最后，AMD 节点上的 SLURM 能耗统计显示，`ADDA` （75 个核心）的一次典型大型 MPI 运行会消耗约 700 kJ，而单精度图形处理器 A100 运行仅需 36 kJ。因此，即使图形处理器的运行时间更长，其能效仍然高得多。

总体而言，对于完全驻留于图形处理器的求解器（`IFDDA` 和 `ADDA OCL_BLAS`），图形处理器具有显著优势，尤其是在可使用单精度时。对于`ADDA`，GPU 模式在小型和中型网格上具有竞争力，但最大规模的算例仍有必要采用基于 CPU 的 MPI。对于后者，内存容量是关键因素：即使是现代加速器，其随机存取存储器也远少于多插槽 CPU 节点，从而限制了可行的网格规模。

5.5. 内存性能

根据表 7， IFDDA 的内存占用大于 ADDA，因为 IFDDA SP 和 IFDDA DP 所需的随机存取存储器分别约为 ADDA的 2 倍和 $4\times$ 。这一比例高于我们在 CPU 内存上观察到的情况（第 4.4 节报告，单个进程排名为 $2.5\times$ ），这是因为与 IFDDA相比， ADDA 仅将 FFT 数据卸载到图形处理器。

表 7：两种问题规模（即 $n_x = 150$ 和 250）下 GPU 设备内存峰值（吉比字节）。

方法	$n_x = 150$	$n_x = 250$
ADDA	1.5	6.5
IFDDA SP	2.9	13.4
IFDDA DP	5.8	28.5

总体而言，精度的影响符合预期，且呈一阶关系：从双精度切换为单精度，可使每个元素的存储量减半。因此，当迭代收敛允许使用单精度时，$2\times$ 内存用量的减少对于在 GPU 内存受限硬件上能否运行可能至关重要。

实际上，网格 $n_x = 250$ IFDDA DP 峰值（28.5 吉比字节）需要配备 40 吉比字节 GPU 内存的显卡（例如 RTX 6000 Ada）；而 IFDDA SP （13.4 吉比字节）可在 16 吉比字节工作站上运行， ADDA （6.5 吉比字节）则具有广泛的可移植性。在 A100 80 吉比字节上，我们的测量结果表明，可行的最大规模约为：IFDDA DP 下的 $n_x \approx 350$ 、 IFDDA SP 下的 $n_x \approx 450$ ，以及 ADDA 下的 $n_x \approx 530$ 。在 ADDAOCL_BLAS 模式下，限制略微降低至 $n_x \approx 510$ （针对双共轭梯度法）。请注意， ADDA 提供了 -opt mem 选项，可通过沿 x 方向逐层处理计算域，强制采用节省内存的执行模式。这能减少 GPU 内存占用，因此可能略微提高可行的最大网格尺寸。例如，对于 $n_x = 530$，GPU 内存最大使用量从 78.2 吉比字节降至 71.7 吉比字节。除了减少内存消耗外，此选项还可能提高运行时稳定性，因为在接近 GPU 内存上限时运行可能导致效率低下和性能下降。此权衡对整体性能的影响仍有待进一步研究。

6. 结论

我们提出了一种统一的基准测试与互操作性软件辅助方法，适用于三种最广泛使用的离散偶极子近似实现

（DDSCAT、 ADDA 和 IFDDA）。通过统一所有物理、数值和求解器参数，我们证明这三种代码能够在相同的测试案例上达到机器精度一致性。尽管此处仅应用于三种求解器，该软件包本身具有完全的通用性，因为任何离散偶极近似实现均可纳入同一交叉验证与基准测试方法。这表明现代 DDA 求解器在数值上既成熟又内部一致，并使性能比较能够与精度差异解耦。

我们的 CPU 基准测试表明，快速傅里叶变换策略是影响性能的主导因素，因为 MKL 和 FFTW 的表现显著优于旧版 GPFA 算法。在受测代码中， ADDA凭借其三维—一维快速傅里叶变换分解和优化的 MPI 实现，实现了最佳可扩展性。

GPU 基准测试证实，将整个求解器阶段卸载到设备上（如 IFDDA以及其 ADDA 实验性 OCL_BLAS 模式所实现的那样），能够带来最显著的加速。相比之下， ADDA 默认的 OpenCL 模式只将快速傅里叶变换卸载到设备上，而求解器循环仍在 CPU 上运行，因此其性能容易受到 CPU–GPU传输延迟的影响。GPU 性能也主要取决于架构因素，例如集群型显卡的内存带宽；而峰值每秒浮点运算次数则可能影响工作站显卡的性能。

总体而言，本研究表明，不同 DDA 代码之间的数值互操作性不仅能够实现，而且极具参考价值。在统一物理参数和数值参数后，剩余的性能差异便反映出真正的架构和实现效应。因此，我们的方法能够支持可复现的跨平台验证，并为相互独立的 DDA 实现提供公平的比较。

致谢

本研究得到诺曼底大区（RAD- DAERO 项目）的资助。部分模拟使用了 CRIANN（法国诺曼底）的计算资源。作者还感谢 Bruce Draine 和 Piotr Flatau 对 DDSCAT 代码提出的意见。

数据可用性

本研究中用于生成基准测试结果和图表的所有数据与脚本均已公开发布于 Zenodo： https://doi.org/10.5281/

zenodo.18847515。为复现软件环境，必须将补丁应用于代码仓库 README 中记录的确切基础版本。

附录 A. 代码的细微修改

为确保 DDSCAT、 ADDA 和 IFDDA 之间的数值一致性及性能比较的公平性，我们对 DDSCAT（版本 7.3.4）进行了几处细微修改。以下列出的这些修改改善了精度处理和求解器一致性，但不会影响物理结果。完整的补丁文件可在 Zenodo 仓库中获取（参见“数据可用性”中的文件夹 patches/）。我们还已将这些文件提交给 DDSCAT 作者，以供考虑纳入下一个版本。研究期间对 IFDDA 所做的类似细微修改未在此列出，因为它们已包含在最新版本中。

附录 *A.1.* 输出数值的精度

为了测试双精度下的浮点一致性，输出量必须以 16 位有效数字打印。为此，需要对 ADDA 和 DDSCAT 的输出例程进行细微修改。

附录 *A.2. DDSCAT* 的修改

LDR 系数的精度处理。为确保双精度下正确舍入，提高了晶格色散关系（LDR）系数的字面精度（参见附录 B）。这消除了 DDSCAT 与其他实现之间在 7–8 位有效数字层面观察到的舍入差异。

三角函数例程中的精度一致性。在余弦-正弦积分子程序中，数值常数被扩展到完整精度，容差阈值根据精度类型进行调整，并明确指定复变量的类型以防止降精度转换。这项更改不会影响单精度结果，但可避免双精度结果损失精度。

求解器重启策略。 禁用了 BiCGStab 重启，以便迭代次数和计时与 ADDA 和 IFDDA 保持一致，因为它们不使用重启。

预条件器和停止准则。默认情况下，DDSCAT 采用左 Jacobi 预条件器和真实残差停止准则，因此每次迭代都会额外进行一次矩阵-向量乘积。为确保基准测试等效，我们禁用了预条件处理，并采用相对残差准则，与其他离散偶极子近似代码保持一致。此更改避免了额外的矩阵-向量运算，并确保所有情况下求解的都是同一个线性算子。还请注意，由此得到的行为与 DDSCAT 中其他迭代求解器所采用的行为相同。

附录 B. LDR 系数的评估

以下总结了以任意数值精度计算晶格色散关系（LDR）系数所需的步骤。我们采用参考文献 [67],中公式 (3.19) 和 (A.4) 的等价表示

$$g(\beta,x)\stackrel{\text{def}}{=}\sum_{n=-\infty}^{\infty}e^{-x(\beta+n)^2}=e^{-x\beta^2}\,\theta_3\left(\mathrm{i}x\beta,e^{-x}\right)=\sqrt{\frac{\pi}{x}}\,\theta_3\Big(\pi\beta,e^{-\pi^2/x}\Big)\,, \tag{B.1}$$

其中，椭圆θ函数分别确保 $x\geq\pi$ 和$x\leq\pi$快速收敛 [68] 。具体而言，它们意味着

$$g(0,x)=\sqrt{\frac{t}{\pi}}\,g(0,t),\qquad t\stackrel{\text{def}}{=}\frac{\pi^2}{x}. \tag{B.2}$$

系数 c_1 由参考文献 [67]中的公式 (4.2) 定义：

$$c_1=\frac{2}{3}\int_0^\infty \mathrm{d}x\left[g^3(0,x)-\left(\frac{\pi}{x}\right)^{3/2}-1\right]. \tag{B.3}$$

该积分在 $x=\pi$处分段，并对区间 (0，π) 应用变量变换 $x\to t=\pi^2/x$ 。利用公式 (B.2)，可得

$$c_1=\frac{2}{3}\int_\pi^\infty \mathrm{d}x\left(1+\sqrt{\frac{\pi}{x}}\right)\left[g^3(0,x)-1\right]-2\pi. \tag{B.4}$$

g 的三次幂可表示为对所有非零整数向量的晶格求和（参考文献 [67] 的式 (A.3)），

$$g^3(0,x)-1=\sum_{\mathbf{n}\neq\mathbf{0}}e^{-x|\mathbf{n}|^2}. \tag{B.5}$$

因此，定义 $u=\pi|\mathbf{n}|^2$，即可得到

$$c_1=\pi\left\{\frac{2}{3}\sum_{\mathbf{n}\neq\mathbf{0}}\left[\frac{e^{-u}}{u}+\sqrt{\frac{\pi}{u}}\operatorname{erfc}(\sqrt{u})\right]-2\right\}. \tag{B.6}$$

这也可以用不完全伽马函数 $\Gamma(1,u)$ 和$\Gamma(1/2,u)$ 表示，从而使整个晶格和求值方法与参考文献 [69] 中的方法相似。

第二个系数由参考文献 [67] 的式 (4.3) 定义：

$$c_2=\frac{1}{2}\int_0^\infty \mathrm{d}x\, g(0,x)\frac{\partial g(0,x)}{\partial x}\left.\frac{\partial^2 g(\beta,x)}{\partial\beta^2}\right|_{\beta=0}. \tag{B.7}$$

利用式 (B.1) 和 (B.2) 计算导数，得到

$$\frac{\partial^2 g(\beta,x)}{\partial\beta^2}=-2x\left[g(\beta,x)+2x\frac{\partial g(\beta,x)}{\partial x}\right], \tag{B.8}$$

$$2x\frac{\partial g(0,x)}{\partial x}=-\sqrt{\frac{t}{\pi}}\left[g(0,t)+2t\frac{\partial g(0,t)}{\partial t}\right]. \tag{B.9}$$

利用这些恒等式，在 $x=\pi$ 处分割积分，并将 g 变回求和，式 (B.7) 可改写为

$$c_2=\int_\pi^\infty dx\left(1+\sqrt{\frac{\pi}{x}}\right)\sum_{\mathbf{n}\neq\mathbf{0}}e^{-x|\mathbf{n}|^2}\left[xn_z^2-2x^2n_y^2n_z^2\right]. \tag{B.10}$$

对求和中的各项分别积分，得到不完全伽马函数 $\Gamma(3/2,u)$、 $\Gamma(2,u)$、 $\Gamma(5/2,u)$ 和 $\Gamma(3,u)$；它们均可用 $\exp(-u)$ 和 $\operatorname{erfc}(\sqrt{u})$ 表示为闭式 [70]。由于具有立方对称性，n_z^2 可替换为 $|\mathbf{n}|^2/3$，而 $n_y^2n_z^2$ – 可替换为 $|\mathbf{n}|^4v/9$，其中

$$v\overset{\text{def}}{=}\frac{3\left(n_x^2n_y^2+n_y^2n_z^2+n_z^2n_x^2\right)}{|\mathbf{n}|^4},\qquad 0\le v\le 1. \tag{B.11}$$

最终得到

$$c_2=\frac{\pi}{3}\sum_{\mathbf{n}\neq\mathbf{0}}\left\{\frac{e^{-u}}{u}\left[1+2u-\frac{v}{3}\left(4+7u+4u^2\right)\right]+\left(\frac{1-v}{2}\right)\sqrt{\frac{\pi}{u}}\operatorname{erfc}(\sqrt{u})\right\}, \tag{B.12}$$

其中 $u = \pi|\mathbf{n}|^2$。

第三系数满足以下精确恒等式

$$c_1 - 4c_2 + 2c_3 = -\frac{4}{3}\int_0^\infty \mathrm{d}x \frac{\partial}{\partial x}\left\{x\left[g^3(0,x) - \left(\frac{\pi}{x}\right)^{3/2} - 1\right]\right\} = 0, \quad \text{(B.13)}$$

该恒等式可直接由参考文献 [67] 中的定义，以及被求导函数在积分域两端均为零这一事实推出。因此，无需独立计算 c_3 。参考文献 [71]的附录 A2 中也间接证明了这一恒等式。

在采用足够高的精度进行计算以确保达到四倍精度（128 位）运算的精度时（对 c_1，截断至 $|\mathbf{n}| \le 5$；对 c_2，截断至 $|\mathbf{n}| \le 5.2$），最终的 LDR 系数（CGS 单位制）为

$$b_1 = -\frac{c_1}{\pi} = 1.8915316529870796511106114030718259, \quad \text{(B.14)}$$

$$b_2 = -\frac{c_2}{\pi} = -0.16484691508771947306079362778185226, \quad \text{(B.15)}$$

$$b_3 = \frac{3c_2 + c_3}{\pi} = 1.77000040193213719085927384044517 42. \quad \text{(B.16)}$$

值得注意的是，为便于与其他 DDA 公式形式 [25]进行比较，我们采用 ADDA 代码 [16],中使用的符号约定，这与原论文 [67], 中的约定相反。以国际单位制表示时，系数 b_1 和 b_3还应分别除以 4π。自 2022年 起，这些数值已用于 `ADDA`（提交版本 `887a63c`），并且满足

$$b_1 - 10b_2 - 2b_3 = 0 \quad \text{(B.17)}$$

根据公式 (B.13)。

最后补充一点，上述方法也可用于计算长方体晶格的 LDR 系数 [72, 29]。

附录 C. 参数的等价性

使用 `ADDA`、 `IFDDA`和 `DDSCAT`进行等效模拟，需要按照表 2 中介绍的精度层级，逐层匹配命令行参数。实际操作中，首先从可执行文件`adda` 和 `ifdda`以及 Python 封装程序 `ddscatcli` [51]入手，然后逐步落实表 C.1 中列出的等效要求，并忽略所有不相关的行（任何代码都未使用的行）

。匹配位数由第一个不匹配的行决定。如果表中所有行都匹配，则达到机器精度级别（10–15 位）。

表 C.1 旨在展示参数之间的对应关系，而非提供具体数值列表来确保不同代码之间严格等效，因为要实现这一点还需格外谨慎，相关内容将在下文讨论。我们还在 Python 软件包 [73] （文件 DDA_commands 中）提供了一组跨代码等效命令行示例，作为可直接运行的实例。请注意，即使使用 ddscatcli包装器，如果 DDSCAT 示例文件夹中没有折射率文件，用户仍必须创建或修改该输入文件。

解读表 C.1 时，有几项约定至关重要。首先，不同代码对粒子尺寸和形状定义的处理各不相同。具体而言， DDSCAT通过等效半径 a_{eff}来定义目标尺寸。对于给定体积 V，这对应于

$$a_{\text{eff}} = \left(\frac{3V}{4\pi}\right)^{1/3}. \tag{C.1}$$

对于受体积修正影响的形状，标称半径或直径的直接一一映射无法在各代码之间保持精确的物理体积。因此，必须在IFDDA 或 DDSCAT 中调整粒子尺寸，以确保体积等效（ADDA可使用 -no_vol_cor进行切换）。对于非均匀材料（例如包覆球）、 IFDDA 中的任意形状（其被视为嵌入盒体），以及对角各向异性材料（IFDDA需要 CM 或 RR 极化率），即使所有其他参数均已匹配，可达到的最大一致性也仅限于 η级别。事实上，在这些情况下，线性方程组无法在各代码之间严格等效（见第 3.2 节）。双球体在 IFDDA 和 DDSCAT 中均被视为非均匀材料，因此需要相应的材料设定。在本工作中，为所有分量指定相同的介电函数，以表示均匀双球体，并在各代码之间实现全位数一致性。对于包覆球， DDSCAT 在使用 -SHPAR 1时还需要三个介电分量，并遵循各向异性壳层配置。请注意，对于任意形状， DDSCAT 和 ADDA 共享一种通用几何文件格式（源自 DDSCAT，基于整数体素位置），从而可在两种代码之间直接复用。相比之下， IFDDA采用基于实数体素坐标的不同格式。此外，与 DDSCAT 和 ADDA不同，

IFDDA 要求在此文件中明确给出网格尺寸，此外还需提供用于 IFDDA 和 DDSCAT内存分配的相应输入参数（-nnnr 和 -MEM_ALLOW）。此外，该网格参数在 IFDDA中对应于最大粒子范围，而 ADDA则期望沿 x方向的一个值（即 n_x）。最后，如第 3.4 节所述，只要保持相关比值并对输出进行一致的重新缩放，用于波长和粒子尺寸的物理单位可以是任意的。

此外，各代码对入射传播方向和目标取向的处理方式不同（尽管输入角度假定采用相同的度单位）。 DDSCAT和 ADDA 均在目标固定参考系（目标坐标系，TF）中构建体素网格，而入射光束则在实验室坐标系（实验室坐标系，LF）中描述。为了在不重建网格的情况下旋转粒子，两种代码都采用被动方法，即不是主动旋转偶极坐标，而是将入射传播方向和偏振矢量（以及散射方向）旋转到 TF 中。因此，旋转目标体与旋转入射光束完全等价——相应的矩阵互为逆矩阵。

尽管采用这一共同策略，两种代码在约定和默认坐标轴选择上仍有差异。 ADDA 采用 $\hat{}zyz$ 约定下的被动欧拉旋转矩阵，且默认$\mathbf{k}_0 = \hat{}\mathbf{z}$，而 DDSCAT— xzx 约定 [74] $\hat{}\,\hat{}$，并硬编码$\mathbf{k}_0 = \hat{}\mathbf{x}$。此处，$\mathbf{k}_0$ 表示 $\hat{}$ 实验室坐标系中的入射传播，以区别于 DDA 方程中所使用的目标坐标系内的$\mathbf{k}$ 。此外，对于某些形状（例如椭球体或 SPHERES_N）， DDSCAT 允许互换或旋转目标坐标轴（记为 $\hat{}\mathbf{a}_i$），这对应于粒子的预先主动旋转——我们不研究该功能，并假定默认 $\hat{}\mathbf{a}_i$。

$\hat{}$ 虽然 $\mathbf{k}_0$ 与欧拉角的任意组合都可以在 ADDA 中通过 DDSCAT 进行匹配，但只有在固定 $\mathbf{k}_0$ 时，才存在一一对应关系。对于后者，最自然的做法是使用 $\mathbf{k}_0 = \hat{}\mathbf{x}$，在ADDA中对应于 -prop 1 0 0 ，从而可以分别处理传播方向和取向的等价条件。后者归结为采用不同约定的欧拉旋转矩阵相等

$$\mathbf{R}_{zyz}(\alpha,\beta,\gamma) = \mathbf{R}_{xzx}(\Phi,\Theta,\beta_{\mathrm{D}}), \tag{C.2}$$

其中，我们将 DDSCAT 中的第三个欧拉角记为 β_{D} ，以区分其手册中使用的 β 与 ADDA 中的相应量。正如预期，默认取向（所有角度均为零）彼此对应，但在其他情况下，角度表达式的变换涉及正三角函数和反三角函数。

相比之下， IFDDA 采用主动旋转，即体素坐标在实验室坐标系中定义，然后旋转，以便在目标坐标系中对照粒子形状进行检验。该方法在物理上等价于被动表述，但不会对不同取向复用同一体素网格。实际上，DDA 方程是在实验室坐标系而非目标坐标系中求解的。因此，仅当旋转将网格晶格映射到自身时，即所有角度均为 90° 的整数倍时，才预期 IFDDA 与其他代码之间具有精确一致性。否则，一致性仅限于方法误差。

在 IFDDA 中，传播方向可以像在 ADDA中那样自由指定，但使用球面角而不是$\mathbf{k}_0$的笛卡尔分量。因此， -prop 中的^ 90 0对应于 IFDDA 中的$\mathbf{k}_0 = \hat{}\mathbf{x}$ ；在这种情况下，旋转矩阵$\mathbf{R}_{zxz}(\psi, \theta, \varphi)$应与式 C.2 中的矩阵相同（这三种表示采用的约定各不相同）。角度 θ, φ 采用 IFDDA 手册中的定义；请勿将其与定义入射方向或散射方向的角度混淆。

在实验室坐标系中， DDSCAT 使用$\hat{}\mathbf{y}$ 和$\hat{}\mathbf{z}$ 作为基本偏振矢量，而 ADDA 和 IFDDA 在默认$\mathbf{k}_0 = \hat{}\mathbf{z}$时使用$\hat{}\mathbf{y}$ 和$\hat{}\mathbf{x}$ 。当传播矢量旋转至$\hat{}\mathbf{x}$时，后两种偏振变为$\hat{}\mathbf{y}$ 和 $-\hat{}\mathbf{z}$，也就是说，除负号外与 DDSCAT 相同。对于大多数模拟量，这个负号没有影响，内场除外。如下文所述，对于入射场的两种讨论方向，$\hat{}\mathbf{y}$ 在 ADDA和 DDSCAT中都定义了默认散射平面。因此，我们将其称为该平面的$\parallel$ ，并将互补偏振称为 $\perp$。对于任意$\mathbf{k}_0$，这一命名仍然有意义。

虽然 DDSCAT 是通过完整复矢量指定单个入射偏振时最灵活的代码（可匹配 IFDDA和 ADDA的任意设置），但第二个偏振会根据第一个自动得出（默认情况下为 $\hat{}\mathbf{y} \rightarrow \hat{}\mathbf{z}$）。这意味着， -IORTH 2对于 DDSCAT 会使用与 ADDA相同的两个偏振，但第二个偏振的符号相反。或者，也可以使用 -IORTH 2 -POL_E01 "(0,0) (0,0)(-1,0)" ，以完全相同的偏振顺序相反地使用它们。

IFDDA 支持圆偏振以及任意线偏振。后者由单个实参数 pola （其中 $0 \leq \texttt{pola} \leq 1$）控制，该参数设定两种基本偏振的相对权重。通过指定复入射偏振单位矢量 $\hat{}\mathbf{e}_{01}$ ，即可将其映射到 DDSCAT，如下所示：

$$\hat{\mathbf{e}}_{01} = (0, s, -p),\ s \stackrel{\text{def}}{=} \sqrt{1 - \texttt{pola}},\ p \stackrel{\text{def}}{=} \sqrt{\texttt{pola}}, \tag{C.3}$$

ADDA主要用于模拟两种入射偏振，以获得穆勒矩阵。若要模拟单一偏振（而不浪费计算

资源），可以通过一种变通方法让 ADDA 将散射体视为对称体，但这样得到的穆勒矩阵通常不正确（最好舍弃）。因此，添加 `-sym enf -scat_matr none`是仅模拟 $\|$ 偏振的一种变通方法。要旋转这一单一线偏振，有两种可能的变通方法。第一种是通过额外旋转粒子来模拟。不过，对于任意的粒子取向和$\mathbf{k}_0$，不存在简单的表达式。尽管如此，仍可列举几种特殊情况：当$\mathbf{k}_0 = \hat{\mathbf{z}}$时，只需将欧拉角增加 $\gamma \to \gamma + \arcsin p$；而当$\mathbf{k}_0 = \hat{\mathbf{x}}$ 且粒子取向为默认值时，则有 $\alpha = -\pi/2$, $\beta = \arcsin p$, $\gamma = \pi/2$。第二种方法是使用 ADDA中实现的通用贝塞尔光束，将阶数和半锥角都设为零： `-beam besselM 0 0 00 <p> <s>`，但这与辐射力计算不兼容。上述两种变通方法都应与前述对称性技巧结合使用，否则其行为将与 DDSCAT中的 `-IORTH 2`类似。

对于圆偏振，我们采用 IFDDA 的定义，与参考文献 [75], 一致。具体而言，对于分别沿 $\hat{\mathbf{e}}_{\mathrm{R,L}} \overset{\text{def}}{=} (\hat{\mathbf{y}} \pm \mathrm{i}\hat{\mathbf{x}})/\sqrt{2}$ 传播的右旋和左旋圆偏振光束，其传播方向为 $\hat{\mathbf{z}}$。这基本上与 DDSCAT 手册中给出的定义相反；后者在将 $\mathbf{k}_0$ 旋转后，分别等价于 $\hat{\mathbf{e}}_{\mathrm{L}}$ 和 $\mathrm{i}\hat{\mathbf{e}}_{\mathrm{R}}$。在 ADDA 中，可以通过贝塞尔光束模拟此类偏振，例如，对于 $\hat{\mathbf{e}}_{\mathrm{R,L}}$，分别可用 `-beam besselM 0 0 0 0 0 <is2> 0 0 <±is2> 0`，其中 `is2` $= 1/\sqrt{2}$。与上文相同，这种模拟不适用于辐射力。此外，在这种情况下，用 ADDA 计算两种偏振完全没有意义，因为这两种偏振是等效的（与 DDSCAT 中的 `-IORTH 2` 不同）。最后，对于圆偏振，QMR 求解器（以及 QMR2 和 BiCG）会失效，因为入射场的伪范数为零 [49]；切换到 BiCGStab 即可解决此问题。

对于各向同性化 LDR 极化率， ADDA 和 IFDDA 采用不同的约定。ADDA仅对入射偏振下的项 S 取平均，同时保持传播方向不变（因此结果取决于传播方向，参见参考文献 [7]中的公式（46）），而 IFDDA则进行完全平均，得到 $\langle S \rangle = 1/5$。因此，必须谨慎选择入射传播角，以复现相同的 S值。例如，$\hat{\mathbf{k}} = \left(\sqrt{(5-\sqrt{5})/10},\ 0,\ \sqrt{(5+\sqrt{5})/10}\right)$，在 IFDDA 中对应角度$\theta = (1/2)\arctan 2$。

还有一些值得注意的数值方面。选项 `-scat`

`fin` 仅在某些其他参数组合下才会影响 C_{abs} 的值，例如采用 LDR 极化率的吸收粒子。相比之下，启用 `igt_so` 会使 C_{ext} 的精度降至 1–2 位有效数字，与完整 IGT 相比；其他公式形式的变化也有类似情况。不过，公式类别之间的大多数差异 $\mathcal{O}[(kd)^2]$, 会随着粒子尺寸减小（在离散化方式固定时）而减弱。值得注意的例外是相互作用公式，即使粒子远小于波长，其差异仍然显著。最后，BiCGStab 求解器是否等效取决于具体问题，这是由最后的残差停止准则导致的。

uracy-critical levels.

	Digits
IPSOID -SHPAR "<Nx> <Nx> <Nx>" -AEFF ff> 1 'LIN'" GLPRSM -SHPAR "<Nx> <Nx> <Nx>" -AEFF "..." GLPRSM -SHPAR "<Nx> <Ny> <Nz>" -AEFF "..." NDRPBC -SHPAR "<Nz> <Nx> 3 0 0 1" -AEFF	0
IPSOID -SHPAR "<Nx> <Ny> <Nz>" -AEFF "..." IPSO_2 -SHPAR "<Nx> <Ny> <Nz>" -AEFF "..."[a] ONSHEL -SHPAR "1 <D> <Din/D>" -AEFF "..." M_FILE -AEFF "..." <beta> <beta> 1" -ROT_THETA "<Theta> -ROT_PHI "<Phi> <Phi> 1"	
name" diels "filename1" "filename2" "filename3"	0
mp> --diels "..." (per material)	
OL_E01 "(0,0) (1,0) (0,0)" OL_E01 "(0,0) (0,0) (-1,0)" OL_E01 "(0,0) (<s>,0) (-<p>,0)" OL_E01 "(0,0) (1,0) (0,0)" OL_E01 "(0,0) (1,0) (0,-1)" OL_E01 "(0,0) (1,0) (0,1)"	0
S "<value> <value> 1 'LIN'"	
"<Nx> <Ny> <Nz>" -SHPAR "..."[c]	1–2
TDR RCD	1–2
CCG GST ICG GS2 >	η

accuracy associated with grid is relevant if only the

相比之下，表 C.2 描述了用于生成额外散射量的等效命令行参数。这些参数可以选择性地添加到表 C.1 所列的自由参数之上，因为它们不会影响代码的其他输出。当严格按照表中所列方式使用这些额外组合时，这些量的精度等级与表 C.1 中相应配置的精度等级相同。否则，预期精度等级将降至 0，因为这可能导致比较不同类型的场或力的不同分量。

ADDA 和 IFDDA 计算作用于每个偶极子上的辐射力（以及 IFDDA中的力矩） [76], ，而 DDSCAT 仅提供通过对散射方向进行数值积分得到的总积分值 [77]。此外，在 DDSCAT 中，辐射力 $\mathbf{Q}^{\mathrm{pr}}$ 必须通过动量守恒进行后处理 $Q_{\mathrm{ext}}\mathbf{k} - Q_{\mathrm{sca}}\mathbf{g}$，其中 $\mathbf{g}$ 是不对称参数）。因此，虽然 ADDA 和 IFDDA 之间可以达到机器精度一致性，但由于数值积分误差，涉及 DDSCAT 的比较通常只能达到约 4–5 位有效数字，即使使用数百万个散射方向也是如此。此外，上述关系仅适用于平面入射波，不过目前 ADDA 在逐偶极子计算力时也存在相同限制。

要比较 IFDDA和 ADDA之间的辐射力，需要进行额外的后处理，因为 IFDDA 输出时间平均力的总值 $\mathbf{F}_{\mathrm{rad}}$（单位为 N），而 ADDA 输出辐射压截面 $\mathbf{C}^{\mathrm{pr}}$ （在本例中单位为 nm^2）及其对应的效率 $\mathbf{Q}^{\mathrm{pr}}$。截面与总力之间的关系为

$$\mathbf{F}_{\mathrm{rad}} = \mathbf{C}_{\mathrm{pr}} \, |\mathbf{E}_0|^2 \, \frac{\varepsilon_0}{2}. \tag{C.4}$$

辐射力矩的处理过程类似，因为 IFDDA输出时间平均力矩总值 $\mathbf{\Gamma}_{\mathrm{rad}}$ （单位为 $\mathrm{N \cdot m}$），而 DDSCAT 输出辐射力矩效率因子$\mathbf{Q}^{\mathrm{trq}}$。两者之间的关系为

$$\mathbf{\Gamma}_{\mathrm{rad}} = \mathbf{Q}_{\mathrm{trq}} \pi a_{\mathrm{eff}}^2 \, |\mathbf{E}_0|^2 \, \frac{\varepsilon_0}{2k}. \tag{C.5}$$

关于内场， IFDDA和 DDSCAT 分别通过将局部场/激发场乘以辐射反作用修正和克劳修斯–莫索提因子来计算宏观场，而 ADDA则根据所选的极化率严格计算内场（见公式 (10)）。

因此，只有对于这两种极化率才能达到机器精度一致性（而不是三个代码之间都能达到）；对于其他选择，预期的一致性水平会下降到内部极化不匹配时所能达到的水平（1–2 位有效数字）。此外， DDSCAT 不支持克劳修斯–莫索蒂极化率（见表 1），因此对其而言同样无法达到完整的机器精度。原则上，也可以比较 DDSCAT 和 ADDA之间的体素极化，但这需要额外工作：必须先修改 DDSCAT的源代码，以保存乘以体素体积 d^3 的内部极化（以匹配 ADDA 约定）。

最后，也可以比较 ADDA 和DDSCAT之间的穆勒矩阵。在后一个代码中（使用 -CMDFRM LFRAME），散射方向以 $(\theta,\ \phi)$ 为参数，其中 $\phi = 0$ 对应于参考散射平面与 $(\hat{\mathbf{x}}_{\mathrm{LF}},\ \hat{\mathbf{y}}_{\mathrm{LF}})$ 平面重合的情形。请注意，这些 θ 和 ϕ 与粒子取向角 Θ 和 Φ不同。在 ADDA中，散射平面也由入射基矢确定，对应于 $(\mathbf{k},\ \hat{\mathbf{e}}_Y)$ 平面。因此，一旦在两种表述之间映射这些矢量（见上文），两个代码便采用相同的散射平面约定，从而可以直接比较穆勒矩阵。

振幅散射矩阵的比较会更加复杂，因为 DDSCAT 是相对于固定的入射偏振 [9]定义该矩阵的，而非相对于随散射平面一同旋转的偏振 [75]。此外， DDSCAT 不会输出该矩阵原始的复数元素，而是输出用于进一步计算穆勒矩阵的实数二次组合。

表 C.2：IFDDA、ADDA 和 DDSCAT 之间的命令行等价项，定义所计算的输出量。

参数	IFDDA	ADDA	DDSCAT
场	-near_field 0 1 0 -save_data 2	-store_int_field	-NRFLD 1
偏振	—	-store_dip_pol	-NRFLD 1
力	-nforce 1 1 0 0	-Cpr	-CMDTRQ DOTORQ
力矩	-nforce 1 1 1 1	—	-CMDTRQ DOTORQ
穆勒矩阵[a]	—	-scat_matr muel -ntheta 180	-IORTH 2 -CMDFRM LFRAME -NPLANES 1 -PLANE "0 0 180 1" -NSMELTS 16 -SMELTS_LIST "11 12 13 14 21 22 23 24 31 32 33 34 41 42 43 44"

[a]对应于 ADDA 的默认选项：单次散射平面，且粒子关于 xz 平面对称。

表 C.3 给出了本工作中获得的匹配位数示例（参见第 4 节中的命令行）。请注意， ADDA 报告的是消光

截面 C_{ext} ，单位为 nm^2 （当输入量采用 nm 时），因此，原本以国际单位制（m^2）表示的 IFDDA 值在比较前已转换为 nm^2 。相比之下， DDSCAT 仅通过无量纲效率 Q_{ext}进行比较。

表 C.3：ADDA、IFDDA 和 DDSCAT 之间 C_{ext} 和 Q_{ext} 的匹配位数。

网格尺寸	Code	C_{ext} [纳米^2]	Q_{ext}
$n_x = 150$	ADDA	**24729875.277** 382743	**3.58901283529** 54913
	IFDDA	**24729875.277** 422981	–
	DDSCAT	–	**3.58901283529** 78903
$n_x = 250$	ADDA	**24730049.76** 7323192	**3.5890381587** 809035
	IFDDA	**24730049.76** 6945954	–
	DDSCAT	–	**3.5890381587** 500482

附录 D. MPI 内存扩展模型

在 MPI 程序中，每个进程排名都有自己的进程空间和通信缓冲区，因此节点级内存开销会随进程排名数量呈线性和二次增长 P。我们将常驻集大小（RSS）总量建模为

$$\mathrm{RSS}(N, P) \approx D_{\mathrm{ADDA}}(N) + C_{\mathrm{RSS}} P + \alpha_{\mathrm{RSS}} P^2, \tag{D.1}$$

其中， $D_{\mathrm{ADDA}}(N)$ 是根据 ADDA 手册 [49]中的公式 (4) 预测的、依赖于应用程序的工作集， C_{RSS} 是所用的每进程恒定库运行时环境的常驻部分，而 α_{RSS} 量化了 MPI 通信池的内存使用量。在我们的集群上进行的初步拟合表明， $C_{\mathrm{RSS}} \approx 20\,\mathrm{MiB}$ 和 $\alpha_{\mathrm{RSS}} \approx 0.3\,\mathrm{MiB}$，但这些数值预计会因硬件和 MPI 实现而异。

二次项在进程等级较多的单节点运行中最为显著；对于每个节点进程等级较少的多节点配置，其增长则趋近于线性。在网格规模较大时， $D_{\mathrm{ADDA}}(N)$占据主导地位，与整体工作集相比，MPI 开销变得微不足道。这似乎符合 MPI 实现的自然设计目标，即节点的大部分内存应仍可供应用程序本身使用。

附录 E. 完全卸载的 ADDA 求解器（OCL_BLAS）

图 E.1 报告了在 A100、H200 和 RTX 2000 Ada 图形处理器上，针对 $n_x = 150$ 和 $n_x = 250$ 网格测得的快速傅里叶变换时间、求解器时间和单核墙钟时间，并比较了 ADDA 的 OCL_BLAS 模式（BiCG 求解器）与采用相同 BiCG 求解器的默认 OpenCL 后端。这确保了公平的

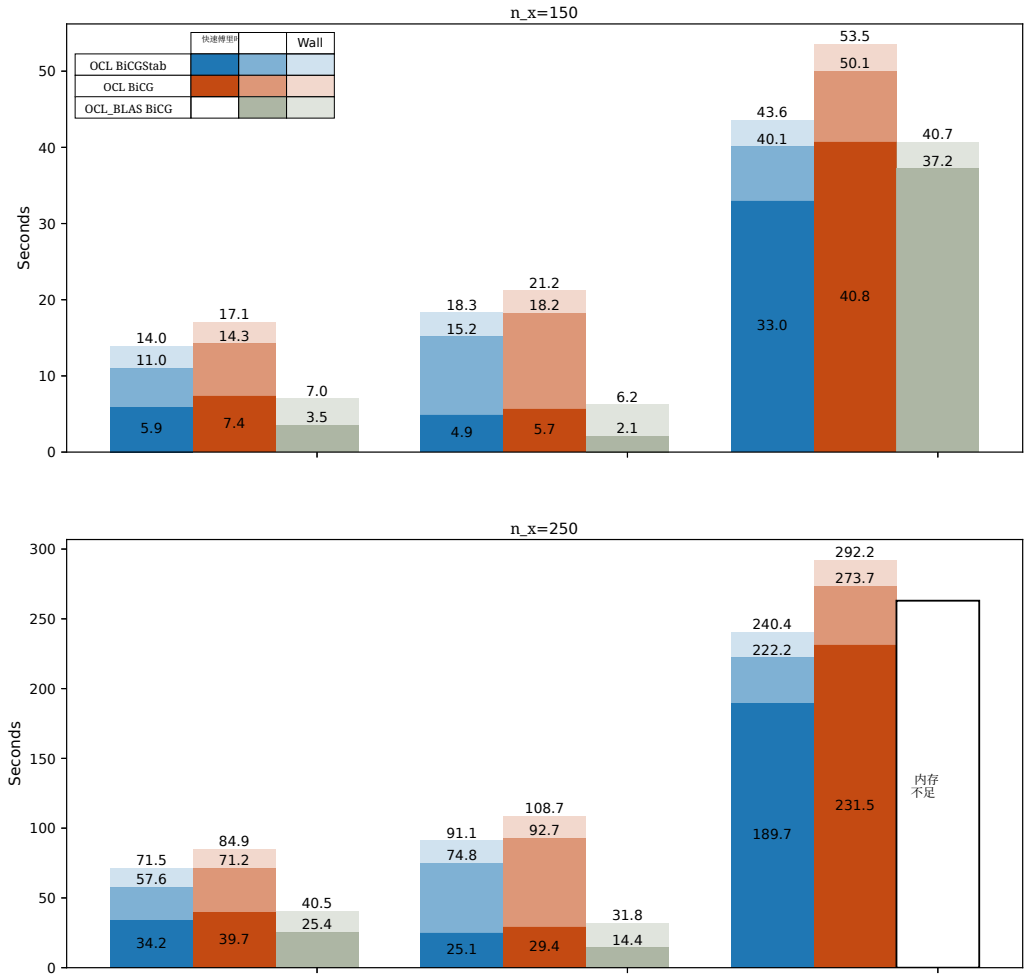


图 E.1：ADDA 和 ADDA OCL_BLAS 模式的图形处理器计时。柱形按模式和求解器类型分组，并以堆叠方式显示快速傅里叶变换时间、求解器时间和单核墙钟时间。网格尺寸 $n_x = 150$ （上排）； $n_x = 250$ （下排）。蓝色柱形结果（BiCGStab）与图 3 相同。

与 OCL_BLAS进行比较，并可与原始的ADDA BiCGStab结果进行有意义的对照（此处也从图 3 中重复列出）。遗憾的是，OpenCL 代码中尚未实现计时功能，因此无法获得 OCL_BLAS 模式下的快速傅里叶变换计时。

在三款图形处理器上，对于两种网格规模， OCL_BLAS 的性能均稳定优于默认的 ADDA OpenCL 模式；但在 NVIDIA RTX 2000 Ada 上运行 $n_x = 250$时例外，计算因内存不足错误而失败（限制约为 $n_x = 230$）。H200 上的提升最为显著，求解器时间分别缩短约 9 倍（$n_x = 150$）和 6.5 倍（$n_x = 250$）。这证实，在标准 OpenCL 模式下，性能瓶颈主要来自中央处理器与图形处理器之间的数据传输延迟，而非 GPU 计算吞吐量。

此处观察到的趋势与 IFDDA的结果非常相似：从 A100 切换到 H200 后， OCL_BLAS 求解器性能提升约 1.8 倍，这与根据 GPU 内存带宽预期的扩展趋势一致。

参考文献

[1] M. I. Mishchenko，《粒子及粒子群的电磁散射：导论》，剑桥大学出版社，英国剑桥，2014年。 doi:10.1017/CBO9781139019064.[2] A. Taflove、S. C. Hagness，《计算电磁学：有限差分时域法》，Artech House，2005年。[3] P. Monk，《麦克斯韦方程组的有限元法》，牛津大学出版社，2003年。 doi:10.1093/acprof:oso/9780198508885.001.0001.[4] R. F. Harrington，《矩量法场计算》，Wiley-IEEE出版社，1993年。[5] M. A. Yurkin、M. I. Mishchenko，电磁散射的体积分方程：对被动宿主介质中具有分段光滑边界的一组多层粒子的严格推导与分析，《Physical Review A》97 (2018年) 043824。 doi:10.1103/PhysRevA.97.043824。

[6] P. C. Chaumet，离散偶极近似：综述，《Mathematics》10 (2022年) 3049。 doi:10.3390/math10173049.[7]M. A. Yurkin，离散偶极近似，载于：M. P. Menguc、M. Francoeur（编），《Light, Plasmonics and Particles》，爱思唯尔，阿姆斯特丹，2023年，第167–198页。 doi:10.1016/B978-0-323-99901-4.00020-2.[8] E. M. Purcell、C. R. Pennypacker，非球形介电颗粒对光的散射与吸收，《天体物理学报》186 (1973年) 705–714。 doi:10.1086/152538.[9] B. T. Draine，离散偶极近似及其在星际石墨颗粒中的应用，《天体物理学报》333 (1988年) 848–872。doi:10.1086/166795.[10] B. T. Draine、P. J. Flatau，用于散射计算的离散偶极近似，Journal of the Optical Society of America A 11 (1994) 1491–1499。 doi:10.1364/JOSAA.11.001491.[11] C. M. Sorensen、J. Yon、F. Liu、J. Maughan、W. R. Heinson、M. J. Berg，包含烟炱在内的分形聚集体的光散射与吸收，Journal of Quantitative Spectroscopy and Radiative Transfer 217 (2018年) 459–473。 doi:10.1016/j.jqsrt.2018.05.016.[12] V. Amendola，采用离散偶极近似法研究石墨烯附近银和金纳米粒子的表面等离激元共振，《物理化学化学物理》18 (2016) 2230–2241。doi:10.1039/C5CP06121K.[13] M. A. Yurkin、K. A. Semyanov、P. A. Tarasov、A. V. Chernyshev、A. G. Hoekstra、V. P. Maltsev，利用扫描流式细胞术和离散偶极近似对成熟单个红细胞的光散射进行实验与理论研究，《应用光学》44 (2005年) 5249–5256。 doi:10.1364/AO.44.005249.[14] J. Yon、C. Roz´e、T. Girasole、A. Coppalle、L. M´e`es，扩展 RDG-FA 以预测烟炱聚集体的散射，并考虑大型单体之间的相互作用，《Part. Part. Syst. Charact》25 (2008) 54–67。 doi:10.1002/ppsc.200700011。

[15] C. Argentin, M. J. Berg, M. Mazur, R. Ceolato, 《分形聚集体的电磁耦合与结构因子确定》, Journal of Quantitative Spectroscopy and Radiative Transfer 296 (2023) 108451。 doi:10.1016/j.jqsrt.2022.108451。M. A. Yurkin, A. G. Hoekstra, 《离散偶极近似代码 ADDA：功能与已知局限》, Journal of Quantitative Spectroscopy and Radiative Transfer 112 (2011) 2234–2247。 doi:10.1016/j.jqsrt.2011.01.031。P. C. Chaumet, D. Sentenac, G. Maire, M. Rasedujjaman, T. Zhang, A. Sentenac, 《IFDDA：用于模拟分层介质中三维非均匀物体散射场的易用代码：教程》, Journal of the Optical Society of America A 38 (2021) 1841–1852。 doi:10.1364/JOSAA.432685。[18]ADDA-team, 其他相关 DDA 代码的 ADDA 维基页面, https://github.com/adda-team/adda/wiki/Links#other-related-dda-codes（2025年（访问于2025年11月25日））。[19] 离散偶极近似代码维基百科页面, https://en.wikipedia.org/wiki/Discrete_dipole_approximation_codes （2025年（访问于2025年11月25日））。[20] A. Penttila, E. Zubko, K. Lumme, K. Muinonen, M. A. Yurkin, B. T. Draine, J. Rahola, A. G. Hoekstra, Y. Shkuratov, 《离散偶极实现与精确技术的比较》, Journal of Quantitative Spectroscopy and Radiative Transfer 106 (2007) 417–436。doi:10.1016/j.jqsrt.2007.01.026。[21] M. Wiesenberger, L. Einkemmer, M. Held, A. Gutierrez-Milla, X. Saez, R. Iakymchuk, 《fel-tor 代码和库在并行计算机架构上的可复现性、精度与性能》, Computer Physics Communications 238 (2019) 145–156。 doi:10.1016/j.cpc.2018.12.006。[22] C. Collange, D. Defour, S. Graillat, R. Iakymchuk, 《多核和众核架构上的并行归约数值可复现性》, Parallel Computing 49 (2015) 83–97。 doi:https://doi.org/10.1016/j.parco.2015.09.001。[23] L. Durdevic, A. Rela˜no Gin´es, A. Roueff, G. Blivet, G. Baffou, 《利用光学波前显微镜测量体外单个神经突的生物量-

镜》，Biomedical Optics Express 13 (2022) 6550–6560。doi:10.1364/BOE.471284。P. C. Chaumet, 《离散偶极近似高效迭代求解器的比较研究》，Journal of Quantitative Spectroscopy and Radiative Transfer 312 (2024) 108816。doi:10.1016/j.jqsrt.2023.108816。M. A. Yurkin, A. G. Hoekstra, 《离散偶极近似：综述与最新进展》，Journal of Quantitative Spectroscopy and Radiative Transfer 106 (2007) 558–589。doi:10.1016/j.jqsrt.2007.01.034。N. B. Piller, O. J. Martin, 《提升耦合偶极近似的性能：一种谱方法》，IEEE Transactions on Antennas and Propagation 46 (1998) 1126–1137。doi:10.1109/8.718567。M. A. Yurkin, M. Min, A. G. Hoekstra, 《离散偶极近似在极高折射率下的应用：滤波耦合偶极方法重获新生》，Physical Review E 82 (2010) 036703。doi:10.1103/PhysRevE.82.036703。P. C. Chaumet, A. Sentenac, A. Rahmani, 《大介电常数散射体的耦合偶极子方法》，Physical Review E 70 (2004) 036606。doi:10.1103/PhysRevE.70.036606。[29] D. A. Smunev, P. C. Chaumet, M. A. Yurkin, 《离散偶极近似中的矩形偶极子》，Journal of Quantitative Spectroscopy and Radiative Transfer 156 (2015) 67–79。doi:10.1016/j.jqsrt.2015.01.019。[30] S. A. Glukhova, M. A. Yurkin, 《矢量贝塞尔光束：一般分类与散射模拟》，Physical Review A 106 (2022) 033508。doi:10.1103/PhysRevA.106.033508。[31] A. A. Kichigin, M. A. Yurkin, 《利用离散偶极近似模拟任意宿主介质中粒子的电子能量损失谱和阴极发光》，Journal of Physical Chemistry C 127 (2023) 4154–4167。doi:10.1021/acs.jpcc.2c06813。[32] T. F. Chan, 《托普利茨系统的最优循环预条件子》，《SIAM科学与统计计算杂志》9 (1988) 766–771。doi:10.1137/0909051。

[33] P. C. Chaumet, G. Maire, A. Sentenac，通过以标量解为初始值并采用循环预条件来加速离散偶极近似，Journal of Quantitative Spectroscopy and Radiative Transfer 298 (2023) 108505。 doi:10.1016/j.jqsrt.2023.108505.[34] S. Lanier，学习预条件处理：利用强化学习增强离散偶极近似的三级循环预条件，Journal of Quantitative Spectroscopy and Radiative Transfer 350 (2026) 109741。 doi:10.1016/j.jqsrt.2025.109741.[35] C. Argentin, M. A. Yurkin，利用专用初始猜测加速离散偶极近似中的迭代求解器，载于：第21届电磁与光散射会议，墨西拿大学，意大利米拉佐，2025年，第50页。网址 https://hal.science/hal-05142441[36] P. C. Chaumet, G. Maire, A. Sentenac，麦克斯韦方程的标量近似：推导与精度，Journal of the Optical Society of America A 39 (2022) 1462–1467。 doi:10.1364/JOSAA.462034.[37] P. C. Chaumet, G. Maire, A. Se ntenac，多层玻恩近似在透射与反射中的精度，Journal of the Optical Society of America A 42 (2025) 413–421。 doi:10.1364/JOSAA.549504.[38] C. Temperton，适用于任意 $n = 2^p 3^q 5^r$ 的广义素因子快速傅里叶变换算法，SIAM Journal on Scientific and Statistical Computing 13 (1992) 676–686。 doi:10.1137/0913039。[39] M. Frigo, S. Johnson, FFTW：一种适用于快速傅里叶变换的自适应软件架构，载于：1998年IEEE声学、语音与信号处理国际会议论文集，第3卷，1998年，第1381–1384页。 doi:10.1109/ICASSP.1998.681704。[40] R. C. Singleton，论快速傅里叶变换的计算，ACM通讯 10 (1967) 647–654。 doi:10.1145/363717.363771.[41] T. Gamblin, M. LeGendre, M. R. Collette, G. L. Lee, A. Moody, B. R. de Supinski, S. Futral，Spack软件包管理器：为混乱的高性能计算软件带来秩序，载于：高性能计算、网络、存储与分析国际会议论文集，2015年，第1–12页。 doi:10.1145/2807591.2807623。

[42] M. A. Yurkin, V. P. Maltsev, A. G. Hoekstra，离散偶极近似的收敛性。I. 理论分析，Journal of the Optical Society of America A 23 (2006) 2578–2591。 doi:10.1364/JOSAA.23.002578.[43] N. B. Piller，边缘网格对耦合偶极近似精度的影响，《光学快报》22 (1997) 1674–1676。 doi:10.1364/OL.22.001674.[44] M. A. Yurkin, M. Kahnert，立方体的光散射：离散偶极近似和T矩阵方法的精度极限，Journal of Quantitative Spectroscopy and Radiative Transfer 123 (2013) 176–183。 doi:10.1016/j.jqsrt.2012.10.001.[45] M. Yurkin，《离散偶极近似》勘误，[《光、等离激元与粒子》，编者：Menguc M. P. 和 Francoeur M.，(2025) 167–198doi:10.13140/RG.2.2.18469.61920.[46] J. D. Jackson，《经典电动力学》，第3版，Wiley，纽约，1998年。[47] B. T. Draine, P. J. Flatau，离散偶极近似代码DDSCAT 7.3用户指南 (2013)。 arXiv:1305.6497.[48]A. Lakhtakia，时谐电磁场散射的矩量法与耦合偶极子方法的强形式和弱形式，《国际现代物理学杂志C辑》3 (1992)。doi:10.1142/S0129183192000385.[49]M. A. Yurkin, A. G. Hoekstra，离散偶极近似代码ADDA 1.4.0用户手册 (2020)。网址 https://github.com/adda-team/adda/raw/v1.4.0/doc/manual.pdf[50] Z. Strakoˇs, J. Liesen，大规模线性代数计算中的数值稳定性，ZAMM 85 (2005) 307–325。 doi:10.1002/zamm.200410185。[51]C. Argentin, ddscatcli：v1.0.4，Zenodo（2026年2月）。doi:10.5281/zenodo.18619006。网址 https://github.com/Argencle/ddscatcli

[52] M. I. Mishchenko，电磁散射中的尺度不变规则，Journal of Quantitative Spectroscopy and Radiative Transfer 101（2006）411–415。 doi:10.1016/j.jqsrt.2006.02.047。[53] H. A. van der Vorst，Bi-cgstab：一种用于求解非对称线性系统的快速且平滑收敛的bi-cg变体，《SIAM科学与统计计算杂志》13（1992年）631–644。 doi:10.1137/0913035。[54] R. D. d. Cunha，T. Hopkins，用于在并行计算机上求解线性方程组的并行迭代方法（pim）软件包，《应用数值数学》19（1995）33–50。 doi:10.1016/0168-9274(95)00017-Q。[55]J. M. Donald，A. Golden，S. G. Jennings，OpenDDA：一种用于离散偶极近似的新型高性能计算框架，《国际高性能计算应用杂志》23（2009）42–61。 doi:10.1177/1094342008097。[56] J. J. Goodman，B. T. Draine，P. J. Flatau，快速傅里叶变换技术在离散偶极近似中的应用，《光学快报》16（1991）1198–1200。 doi:10.1364/OL.16.001198。[57] A. G. Hoekstra，M. D. Grimminck，P. M. A. Sloot，基于快速离散偶极近似的弹性光散射大规模模拟，《国际现代物理学杂志C辑》09（1998年）87–102。doi:10.1142/S012918319800008X。[58] A. Strack，C. Taylor，D. Pfl¨uger，RISC-V上的并行FFTW：一项包括OpenMP、MPI和HPX的比较研究（2025年）。 arXiv:2506.08653。[59] M. A. Yurkin，D. A. Smunev，S. A. Glukhova，A. A. Kichigin，A. E. Moskalensky，K. G. Inzhevatkin，C. Argentin，P. Bouillon，ADDA 代码的最新进展，载于：第21届电磁与光散射会议，墨西拿大学，米拉佐，意大利，2025年，第36页。URL https://hal.science/hal-05142352[60] B. Li，S. Cheng，J. Lin，tcfft：通过张量核心加速半精度快速傅里叶变换（2021年）。 arXiv:2104.11471。

[61] B. E. Barrowes，F. L. Teixeira，J. A. Kong，三维散射中非对称多层分块托普利茨矩阵的矩阵-向量乘快速算法，微波与光技术快报 31（2001）28–32。doi:10.1002/mop.1348。[62] M. Shabaninezhad，M. Awan，G. Ramakrishna，基于图形处理器的离散偶极近似MATLAB软件包：快速傅里叶变换与双共轭梯度，Journal of Quantitative Spectroscopy and Radiative Transfer 262（2021年）107501。doi:10.1016/j.jqsrt.2020.107501。[63] S. Williams，A. Waterman，D. Patterson，屋顶线模型：面向多核架构的富有洞察力的可视化性能模型，ACM通讯 52（2009）65–76。doi:10.1145/1498765.149878。[64] A. Ayala，S. Tomov，M. Stoyanov，A. Haidar，J. Dongarra，大型多GPU系统上并行快速傅里叶变换的性能分析，载于：2022年IEEE国际并行与分布式处理研讨会，IEEE，2022年，第372–381页。 doi:10.1109/IPDPSW55747.2022.00072。[65] P. Steinbach，M. Werner，gearshifft——面向异构平台的FFT基准测试套件，载于：高性能计算：第32届国际会议，施普林格出版社，柏林，海德堡，2017年，第199–216页。doi:10.1007/978-3-319-58667-0_11。[66] D. Xu，P. Tuersun，S. Li，M. Wang，L. Jiang，CPDDA：由CuPy加速的离散偶极近似Python 软件包，Nanomater. 15（2025年）500。 doi:10.3390/nano15070500。[67] B. T. Draine，J. Goodman，超越克劳修斯-莫索提：可极化点阵上的波传播与离散偶极近似，《天体物理学报》405（1993年）685–697。 doi:10.1086/172396。[68] J. D. Fenton，R. S. Gardiner-Garden，用于计算椭圆积分以及theta函数和椭圆函数的快速收敛方法，澳大利亚数学会杂志B辑 24（1982）47–58。doi:10.1017/S0334270000003301。[69] B. Nijboer，F. De Wette，偶极晶格中的内场，Physica 24（1958）422–431。doi:10.1016/S0031-8914(58)95803-8。[70] NIST数学函数数字图书馆，第8.4节：不完全伽马函数及相关函数， https://dlmf.nist.gov/8.4，F。

W. J. Olver, A. B. Olde Daalhuis, D. W. Lozier, B. I. Schneider, R. F. Boisvert, C. W. Clark, B. R. Miller, B. V. Saunders, H. S. Cohl, 和 M. A. McClain，编。1.0.24版（2019-09-15）。[71] D. W. Mackowski, 利用离散偶极矩方法计算非球形粒子的T矩阵，《Journal of the Optical Society of America A》19（2002）881–893。 doi:10.1364/JOSAA.19.000881.[72]D. Gutkowicz-Krusin，B. T. Draine，在可极化点构成的矩形晶格上传播电磁波（2004）。arXiv: astro-ph/0403082.[73] C. Argentin，dda-bench：v0.1.1，Zenodo（2026年3月）。 doi:https://doi.org/10.5281/zenodo.18836855。网址 https://github.com/Argencle/dda-bench[74]M. A. Yurkin，光散射模拟中取向平均技术的公平评估：评述 Fenni等人所著《评估高阶求积方案在提高非球形冰粒子取向平均单次散射特性计算效率方面的作用》，《地球物理研究杂志：大气》128（2023年）。 doi: 10.1029/2021JD036088.[75] C. F. Bohren，D. R. Huffman，《小粒子对光的吸收与散射》，John Wiley & Sons，1983年。[76] P. C. Chaumet, M. Nieto-Vesperinas，电磁场中偶极球所受的时间平均总力，《光学快报》25（2000）1065–1067。 doi:10.1364/OL.25.001065.[77]B. T. Draine，J. C. Weingartner，星际尘粒上的辐射力矩。I. 超热自旋加速，《天体物理学报》470（1996）551。 doi:10.1086/177887。