arXiv:2301.01179v1 [math.NA] 3 Jan 2023

一种适用于轴对称域的快速多极子方法

Michael J. Carley

2023年1月4日

摘要

将泊松方程的快速多极子方法（FMM）扩展到轴对称区域中的非轴对称问题，这类问题由柱坐标描述。该方法基于对源进行傅里叶分解并将其表示为模态展开，再通过径向和轴向坐标上的二维树分解计算相应的场模态。场系数采用模态格林函数计算，该函数可利用第二类勒让德函数的已知递推公式求得，其导数则可通过柱坐标下的拉普拉斯方程递推求出。柱坐标问题与笛卡尔坐标问题的主要区别在于，计算局部相互作用时不具备平移不变性，因此轴对称区域的计算量有所增加。文中给出了解的精度、收敛情况以及与直接计算相比的计算时间。结果表明，该方法收敛良好，在所展示的测试案例中可达到十位有效数字的精度。计算时间取决于初始化与源点和场点之间局部相互作用计算之间的平衡；根据展开阶数不同，其计算时间约比直接计算少两个数量级。

1 引言

自从快速多极子方法（FMM） [8], 问世以来，它已成为解决一大类问题的首选算法。这类问题可以表述为：在大量场点处求大量点源产生的势。这包括由泊松方程和亥姆霍兹方程支配的问题，例如声学、电磁学和流体动力学中的边界积分问题，以及体积分问题，例如毕奥-萨伐尔

积分，该积分出现在电磁学和涡旋动力学中。如今，这种方法已得到充分发展，针对一系列问题，已有高效的二维 [4] 和三维 [10] 实现方案，并采用多种解析工具进行表述。

尽管快速多极子法功能强大且十分重要，但似乎尚未将其推广到非笛卡尔坐标系。尤其是，似乎还没有可用于以柱坐标描述的轴对称区域的方法公式。这类系统自然存在于多种应用中，而柱坐标是描述系统或施加边界条件的自然方式。本文旨在将用于泊松方程的快速多极子法推广到柱坐标，以流体动力学中的应用为动机；在这些应用中，边界积分问题和体积分问题均由拉普拉斯核支配。

据作者所知，此前有两项与该问题相关的研究。第一项是 Strickland 和 Amos [11,12]的工作，他们开发了一种加速方法，用于计算涡旋动力学中的轴对称流函数，这等价于求解泊松问题或计算毕奥-萨伐尔积分。作者采用单精度算术以及格林函数的五阶展开，以达到五位有效数字的精度，而所需计算量仅为直接计算势和速度场所需计算量的 1–3%。作者似乎并未将其方法推广到柱面域中的一般情形。

最近，Churchill [2] 在一篇未发表的论文中研究了一般的非轴对称问题，其动机是分析旋转曲面上的边界积分。他指出，柱坐标系中的特殊难点在于，格林函数在轴向坐标上具有平移不变性，但在径向上则没有，这使得快速多极子法中相互作用的计算变得复杂。他得出结论，该问题可以通过黑箱 [5]或广义 [6] 快速多极子法求解，但没有给出柱坐标问题的结果。

本文介绍了一种在柱状区域内快速计算势的方法，该势由一组方位角变化的环形源产生。其动机来自流体动力学中的泊松问题或毕奥-萨伐尔问题，预期该方法可用于边界积分求解器 [13, 例如], ，或用于计算由涡量分布 [12]产生的速度场。该方法本质上是标准二维快速多极子方法，并对相互作用的计算作了修改，以适应平移不变性的缺失。假设计算所需的任何输入傅里叶变换均已完成，因此计算的起点是一组圆形源的位置和半径，以及每个圆上源分布的傅里叶级数系数。本文简要介绍该方法的主要组成部分，并对其中

轴对称情形特有的部分作更详细的说明，即在柱坐标中计算格林函数及其导数。本文给出了源数量不断增加时某一问题的结果，以检验该方法的收敛性和速度表现。

2 分析

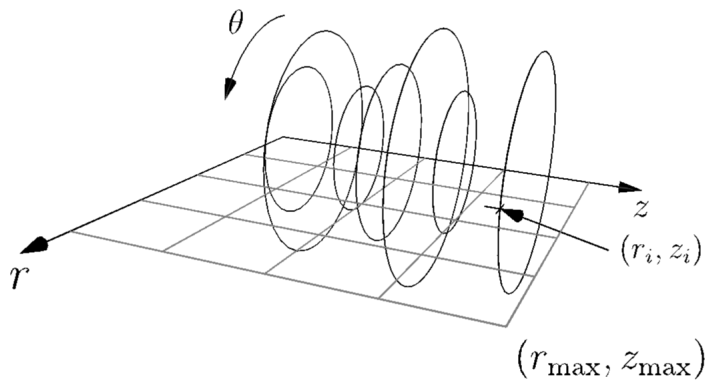


图 1：基本问题：半径为 r_i 的环形源分布在区域 $0 \le r \le r_{\max}$、$0 \le z \le z_{\max}$ 中的轴向位置 z_i。源由各个 (r_i, z_i) 处源强关于 θ 的傅里叶级数系数给出

待求解的问题如图 1 所示。我们假设已完成所有必要的预处理，将系统化为此处所示的形式。一组具有公共轴的环形源分布在整个区域 (r, z) 内。位置为 (r_1, z_1) 的环上的源强 $s(\theta_1)$ 由傅里叶级数给出，

$$s(\theta_1) = \sum_{n=-N}^{N} S^{(n)} \mathrm{e}^{\mathrm{i}n\theta_1}, \tag{1}$$

其中下标 1 表示源坐标。源产生的势 $\phi(r, \ z)$ 通过对 θ_1积分得到，

$$\begin{aligned} \phi(r, \theta, z) &= \int_0^{2\pi} \frac{s(\theta_1)}{4\pi R} \, \mathrm{d}\theta_1, \\ R^2 &= r^2 + r_i^2 - 2rr_1 \cos(\theta - \theta_1) + (z - z_1)^2. \end{aligned} \tag{2}$$

经过初等变换，并代入 $s(\theta_1)$的傅里叶级数 ，

$$\phi(r,\theta,z)=\sum_{n=-N}^{N}S^{(n)}\mathrm{e}^{-\mathrm{i}n\theta}\int_0^{2\pi}\frac{\mathrm{e}^{\mathrm{i}n\theta_1}}{4\pi R}\,\mathrm{d}\theta_1, \tag{3}$$
$$R^2=r^2+r_1^2-2rr_1\cos\theta_1+(z-z_1)^2.$$

势场 $\phi(r,\ z)$ 随后可表示为关于 θ 的傅里叶级数，

$$\phi(r,\theta,z)=\sum_{n=-N}^{N}\Phi^{(n)}(r,z)\mathrm{e}^{\mathrm{i}n\theta}, \tag{4}$$
$$\Phi^{(n)}=S^{(n)}G^{(n)}(r,r_1,z-z_1), \tag{5}$$
$$G^{(n)}(r,r_1,x)=\int_0^{2\pi}\frac{\mathrm{e}^{\mathrm{i}n\theta_1}}{4\pi R}\,\mathrm{d}\theta_1=\int_0^{2\pi}\frac{\cos n\theta_1}{4\pi R}\,\mathrm{d}\theta_1, \tag{6}$$
$$R^2=r^2+r_1^2-2rr_1\cos\theta_1+x^2.$$

假设源项 $s(\theta_1)$ 为实数，因此复傅里叶系数之间满足关系 $S^{(-n)}=\left(S^{(n)}\right)^*$。此后，所有计算均针对 $n\geq 0$进行，并假定共轭关系成立。

我们将 $G^{(n)}(r,\ r_1,\ x)$ 称为模态格林函数，它将源的傅里叶系数与其他位置处势的傅里叶系数联系起来。附录A详细介绍了 $G^{(n)}(r,\ r_1,\ x)$ 及其导数的计算。特别地，文中证明它可以精确表示为

$$G^{(n)}(r,r_1,x)=\frac{Q_{n-1/2}(\chi)}{2\pi\sqrt{rr_1}}, \tag{7}$$

其中 $Q_\nu(\chi)$ 是第二类勒让德函数 [3]。

给定分布在位置 $(r_i,\ z_i)$、 $i=1,\ \ldots,\ N_s$ 处的源，每个源都有一组傅里叶系数 $S_i^{(n)}$，$n=0,\ \ldots,\ N$，那么待求问题就是近似计算以下求和

$$\Phi^{(n)}(r_j,z_j)=\sum_{i=1}^{N_s}S_i^{(n)}G^{(n)}(r_j,r_i,z_j-z_i), \tag{8}$$

对于场点（$r_j,\ z_j$），$j=1,\ \ldots\ldots,\ N_f$。

2.1 快速多极子法概述

快速多极子方法已相当成熟，关于其算法和实现也有许多指南。这里我们概述该方法，以便

介绍必要的术语，并指出本文算法中与现有方法不同的部分。根据上一节，我们回顾一下，目标是近似计算总和

$$\Phi^{(n)}(r_j, z_j) = \sum_{i=1}^{N_s} S_i^{(n)} G^{(n)}(r_j, r_i, z_j - z_i), \quad n = 0, \ldots, N, \quad j = 1, \ldots, N_f,$$

给定一个源点和场点列表（r_i， z_i 和（r_j， z_j）。快速多极子法的第一步是对点进行排序，并将其表示为四叉树数据结构；该结构通过反复细分区域 $0 \le r_{i,\ j} \le r_{\max}$,$0 \le z_{i,\ j} \le z_{\max}$, $z_{\max} = r_{\max}$ 构建。图2展示了区间的反复二等分。在细分的每个层级 ℓ，区域被划分为 $2^\ell \times 2^\ell$ 个盒子，直到称为树的深度 d的最大层级。树中的盒子可以根据其在给定层级网格中的位置（i， j）进行索引，也可以根据其 Morton索引进行索引。

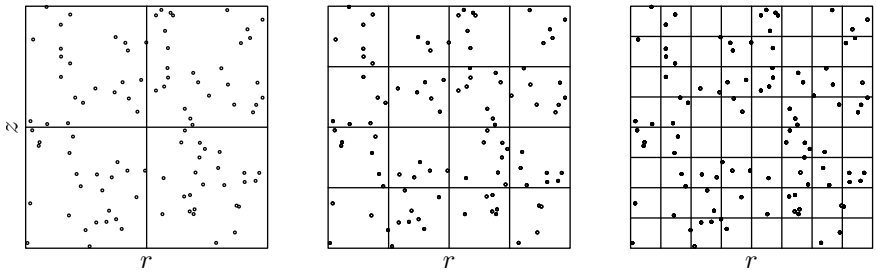


图2：将源区域递归细分为 4^ℓ 个盒子，对应 $\ell = 1$，2，3，从左到右

图3给出了盒子之间关系的术语。位于网格位置（i， j）、层级为 ℓ 的盒子，在层级 $\ell + 1$ 有四个子节点盒子，分别位于网格位置（$2i$， $2j$）、（$2i$， $2j+1$）、（$2i+1$， $2j$）、（$2i+1$， $2j+1$）；它在层级 $\ell - 1$ 的网格位置（$i/2$， $j/2$）有一个父节点盒子。位于最精细细分层级、深度为 d 的盒子没有子节点，称为叶盒子。

盒子的邻居是指与该盒子处于同一层级 ℓ 、且至少与该盒子共享一个顶点的盒子，包括盒子自身。其中，图3中编号为1–9的盒子都是盒子1的邻居。其他所有盒子均被称为盒子1的远场。快速多极子法的基本原理是将远场相互作用与邻居相互作用分离开，并使用加速求和计算任意盒子中的远场项。

首先，通过近似展开计算盒子内源所产生的场来实现这一点；这种方法比直接求和更快。

$$\Phi^{(n)}(r, z) = \sum_i S_i^{(n)} G^{(n)}(r, r_i, z - z_i),$$

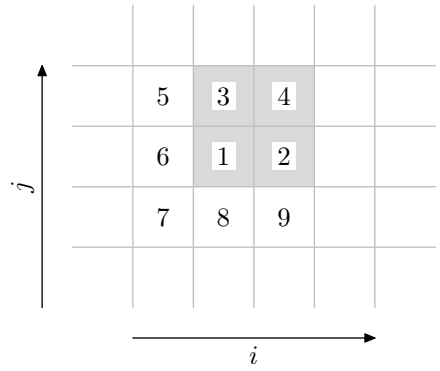


图3：盒子之间关系的术语：盒子1–4是阴影盒子的子节点；阴影盒子是盒子1–4的父节点；盒子1–9是盒子1的邻居

其中，求和遍历盒子内的所有源。对于距离盒子足够远的场点，源对应的格林函数可以用泰勒级数很好地近似，并截断至某个阶数 M，

$$G^{(n)}(r+\Delta r, r_1+\Delta r_1, x+\Delta x) \approx \sum_{m=0}^{M} \sum_{i+j+k=m} \overline{g}^{(n)}_{i,j,k} (\Delta r)^i (\Delta r_1)^j (\Delta x)^k, \quad (9)$$

$$\overline{g}^{(n)}_{i,j,k} = \frac{1}{i!}\frac{1}{j!}\frac{1}{k!} \frac{\partial^{i+j+k}}{\partial r^i \partial r_1^j \partial x^k} G^{(n)}(r, r_1, x).$$

关于 $G^{(n)}$ 的导数计算详情见附录A。

盒子中心为（r_1， z_1）的盒子中，位于（$r_1+\Delta r_1$， $z_1+\Delta z_1$）的源在点（r， z）产生的场为

$$\Phi^{(n)}(r,z) \approx S^{(n)} \sum_{m=0}^{M} \sum_{i+j=m} (-1)^j \overline{g}^{(n)}_{0,i,j} (\Delta r_1)^i (\Delta z_1)^j, \quad (10)$$

注意到 $x = z - z_1$ 和 $\partial/\partial z_1 = -\partial/\partial x$。对盒子中包含的所有源求和，

$$\Phi^{(n)}(r,z) \approx \sum_{m=0}^{M} \sum_{i+j=m} (-1)^j \overline{g}^{(n)}_{0,i,j} S^{(n)}_{ij}, \quad (11)$$

其中，矩 S_{ij} 定义为

$$S^{(n)}_{ij} = \sum_q S^{(n)}_q (\Delta r_q)^i (\Delta z_q)^j. \quad (12)$$

为在 FMM 的第一阶段初始化源数据，需要计算树中每个叶盒子的矩 $S_{ij}^{(n)}$。在向上遍历过程中，计算每一层盒子的矩 ℓ，其中 $\ell = d-1$，....., 1。无需从源数据直接计算矩，而是可以通过组合子盒子的矩来生成其父盒子中的矩，从而实现这一过程，见图 4。位移为（Δr， Δz）的子盒子中心处的矩，通过以下方式对其父盒子中心处的矩作出贡献:

$$S_{ij} = \sum_{q=0}^{i} \sum_{u=0}^{j} \binom{i}{q} \binom{j}{u} (-\Delta r)^q (-\Delta z)^u S'_{i-q,j-u}, \tag{13}$$

为清晰起见，省略了上标 (n)。向上遍历完成后， $\ell = 1, \ldots, d$ 各层的每个盒子都包含一组矩，可用于估算该盒子远场中的势。

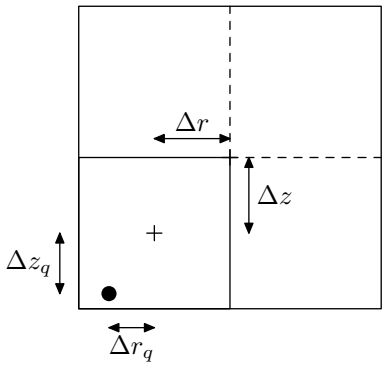


图 4：源矩的计算与平移：使用 (12) 计算叶盒子中心处的矩；使用 (13) 将子盒矩平移 $(\Delta r, \Delta z)$，从而计算父盒子中心处的矩。

在快速多极子法的下一阶段，即向下遍历中，为每个盒子赋予一个局部展开，用于计算远场中的源在盒子内部产生的势。快速多极子法的核心是在每一层采用尽可能高效的展开，来计算任意盒子中的远场项。中心位于 (r, z) 的盒子中的场为

$$\Phi^{(n)}(r + \Delta r, z + \Delta z) = \sum_{m=0}^{M} \sum_{k+\ell=m} \Phi_{k\ell}^{(n)} (\Delta r)^k (\Delta z)^\ell, \tag{14}$$

其中，展开系数 $\Phi_{k\ell}^{(n)}$ 是根据与场盒相互作用的盒子中的源所产生的贡献计算得出的。局部展开的阶数 M

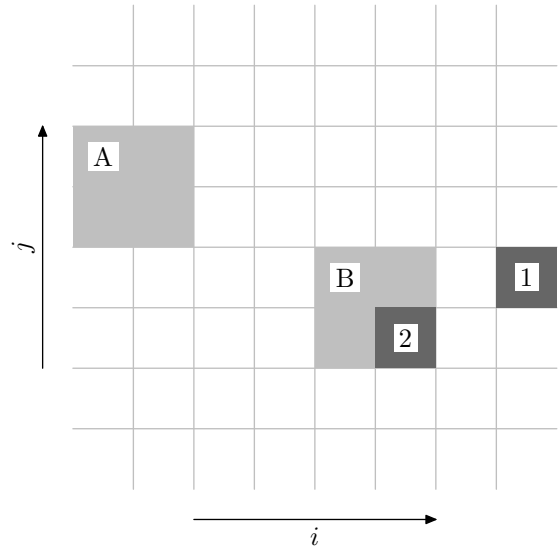


图5：盒子中局部场的生成：使用源到局部展开操作（15），计算盒子 B 中由盒子 A 内源产生的场的局部展开；将盒子 B 中的展开平移至盒子 2，并加上使用源到局部展开操作计算的盒子 1 的贡献，即可得到盒子 2 中的局部展开。

在任意层级都不必与源展开的阶数相同，但本文所展示的计算中将两者设为相同。

图5展示了计算盒子 2 中局部展开所涉及的主要操作，盒子 2 的父盒子为 B。在向下遍历过程中，将与 B 充分分离的盒子（例如盒子 A）中的源所产生的贡献相加，即可得到盒子 B 中的局部展开。该贡献通过平移到局部展开 或源到局部展开操作计算。利用 B 中的局部展开，可以生成其每个子盒子中的局部展开，包括盒子 2。随后，将与盒子 2 相互作用的盒子（例如盒子 1）所产生的贡献加到盒子 2 的局部展开中。向下遍历结束时，每个叶盒子都具有一个局部展开，其中计入了其邻居之外所有源的贡献。

这里要实现的两个操作是 S2L 和局部展开的父到子平移。S2L 平移可通过对式 (10) 求导得到，

$$\Phi_{k\ell}^{(n)} = \sum_{i,j} (-1)^j \binom{j+\ell}{j} S_{ij}^{(n)} \overline{g}_{k,i,j+\ell}^{(n)}, \tag{15}$$

它可以实现为基础线性代数子程序库的 2 级操作

位移为 $(\Delta r,\ \Delta z)$ 的子盒子中的局部展开由

父盒子的展开给出，

$$\Phi_{ij} = \sum_{q=0}\sum_{u=0} \binom{i+q}{q}\binom{j+u}{u}(-\Delta r)^q(-\Delta z)^u \Phi'_{i+q,j+u}, \tag{16}$$

其中，$\Phi'_{i,\ j}$ 是父盒局部展开的系数。

2.2 相互作用的计算

第2.1小节介绍的快速多极子方法概述包含了通用快速多极子法的主要要素，这些要素在现有实现中已广为人知。本节将介绍该算法中专用于柱面域的部分，即针对模态格林函数的源到局部展开操作。在现有的笛卡尔坐标方法中，平移算子不随坐标系的平移而改变。然而，正如Churchill [2], 所指出的，模态格林函数 $G^{(n)}$在轴向坐标 z 平移时保持不变，但在径向位移 r 时则不然。这增加了必须计算平移算子的方向数量，不过仍有一些对称性可用于减少计算量。

回顾一下，模态格林函数表示为

$$G^{(n)}(r, r_1, x) = \frac{Q_{n-1/2}(\chi)}{2\pi\sqrt{rr_1}},\ \chi = \frac{r^2 + r_1^2 + x^2}{2rr_1},\ x = z - z_1.$$

该函数在 z 方向的平移下保持不变，并且关于 r 和 r_1对称，这一性质被用于源坐标和场坐标中导数的递归关系 [11, 12]。为利用这一对称性，我们引入一些术语来描述平移操作。假设对于源到局部展开操作，我们计算 $G^{(n)}$ $(r,\ r_1,\ x)$关于 $r \geq r_1$、 $x \geq 0$的导数。那么，根据图6，我们可以推导出四种不同情形下的平移算子。它们分别对应沿轴向正向或负向（向前或向后）平移，以及从较大半径向较小半径平移（向外或向内）。

不作任何修改而直接使用格林函数导数的基本算子，是向前向外 或 FO 移位（15）。

$$\Phi^{(n)}_{k\ell} = \sum_{i,j}(-1)^j \binom{j+\ell}{j} S^{(n)}_{ij}\overline{g}^{(n)}_{k,i,j+\ell}.$$

对于向后向外移位（即 BO 移位），$x < 0$、$\partial/\partial z_1 = \partial/\partial x$和 $\partial/\partial z = -\partial/\partial x$，得到

$$\Phi^{(n)}_{k\ell} = \sum_{i,j}(-1)^\ell \binom{j+\ell}{j} S^{(n)}_{ij}\overline{g}^{(n)}_{k,i,j+\ell}.$$

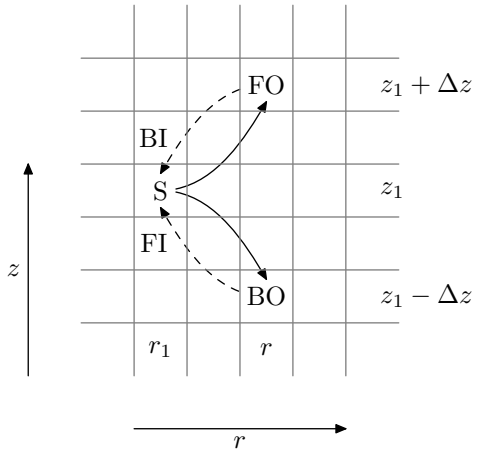


图6：正向/反向和向内/向外交互中格林函数导数的重复利用。源盒子 S 通过正向向外（FO）和反向向外（BO）运算，在 $r > r_1$处生成局部展开，图中以实线箭头表示。半径更大的盒子通过反向向内（BI）和正向向内（FI）运算，在盒子 S 中生成局部展开。这四种运算均基于格林函数的同一个展开计算。

为计算向内移位，我们交换 r 和 r_1，并交换相应的指标。对于向前向内源到局部展开操作，

$$\Phi_{k\ell}^{(n)} = \sum_{i,j} (-1)^j \binom{j+\ell}{j} S_{ij}^{(n)} \overline{g}_{i,k,j+\ell}^{(n)},$$

这给出了半径较大的盒子中的源 r 对半径较小的盒子中局部展开的贡献。最后，反向向内 算子定义为

$$\Phi_{k\ell}^{(n)} = \sum_{i,j} (-1)^\ell \binom{j+\ell}{j} S_{ij}^{(n)} \overline{g}_{i,k,j+\ell}^{(n)}.$$

为了执行平移操作，我们枚举可能对盒子中的场产生贡献的候选源盒，如图 7 所示。这由此产生两个相互作用列表：D列表包含通过对每个源点和场点评估场而产生贡献的盒子；S2L列表中的贡献则通过作用于源和局部展开系数的源到局部展开操作来计算。图 7 表明，D列表由盒子 B 的邻居组成，因此还有 27 个盒子可能通过 S2L 操作对 B 中的场产生贡献。所有其他盒子的贡献都在向下遍历过程中从其父盒传递到 B。利用轴向平移不变性以及 r 中的对称性



图 7：盒子 B 的相互作用列表。B 的父盒子以阴影标出；标记为 D 的盒子，其相互作用由式 (8) 直接计算；标记为 S2L 的盒子，其相互作用通过源到局部运算计算。未标记的盒子通过 B 的父盒子与 B 相互作用，因此无需显式计算这些相互作用。

和 r_1，需要计算的格林函数展开数量减少到十二个，分别对应位于 $z \geq z_1$ 和 $r \geq r_1$ 的盒子。对于 $z - z_1$ 的任意取值，这些展开都相同，但在向下遍历过程中，必须针对每个 r_1 更新展开。展开生成后，便用于更新外部盒子上的局部展开，并将这些盒子的源项对 B 上局部展开的贡献纳入其中。

最后，我们指出，模态格林函数可以用缩放坐标表示，因此

$$G^{(n)}(\sigma r, \sigma r_1, \sigma x) = \frac{1}{\sigma} G^{(n)}(r, r_1, x), \tag{17}$$

这样即可按需预先计算平移算子，并在每一层生成相应算子。我们已经实现了这一做法，但发现它并未节省时间，因为树的每一层所需的平移算子数量都是其父节点层的两倍，其中一半是新的。实际上，我们发现代码的瓶颈在于直接相互作用的计算，而不是源到局部算子的计算。

3 算法

结合前几节的内容，我们给出一种用于轴对称域中均匀快速多极子方法的算法。输入包括一个源点列表 N_s（r_i， z_i），$i = 1, \ldots, N_s$ 以及模态振幅 $S_i^{(n)}$，$n = 0, \ldots, N$，还有一个场点列表 N_f（r_j， z_j），$j = 1, \ldots, N_f$。

算法 1柱坐标系下的快速多极子方法

设置树深度 d按 Morton索引对源点（r_i, z_i）和场点（r_j, z_j）排序，并将其分配到第 d层的叶节点根据源振幅计算叶盒矩（12）{向上遍历}对于 $\ell = d-1, \ldots, 2$，根据第 $\ell+1$层的子盒矩计算第 ℓ 层的盒矩，（13）结束循环{向下遍历}对于 $\ell = 2, \ldots, d$，执行以下操作：对于 $i = 0, \ldots, 4^{\ell-1}-1$，将第 $\ell-1$ 层父盒子的局部展开平移到第 ℓ 层的盒子中结束循环对于 $i = 0, \ldots, 2^{\ell}-1$，计算径向位置 i处的格林函数展开系数对于 $j = 0, \ldots, 2^{\ell}-1$，对盒子（$i, j$）中的源应用 FO 和 BO 平移的源到局部算子对盒子（i, j）应用 FI 和 BI 平移的源到局部算子，以更新局部展开结束循环结束循环结束循环对于 $i = 0, \ldots, 4^{d}-1$，执行以下操作：对于盒子 i 中的场点，计算局部展开与邻盒直接贡献之和结束循环

算法 1 中的实现针对均匀 FMM，在将点分配到盒子时不会生成自适应分解。这样设计是为了减少计算盒间相互作用所需的平移算子数量。

4 结果

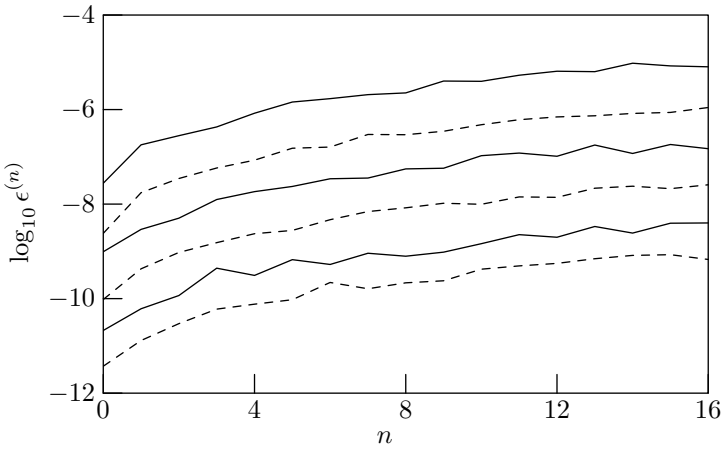


图 8：对于模态振幅 $N_s = 2^{16}$ 的实部，误差 ϵ 随模态阶数变化的关系；树深度 $d = 6$、展开阶数 $M = 6$，8，10，12，14，16 分别对应图中从上到下的曲线。

该算法通过将源点和场点随机分布在 $0 \le r,\ z \le 1$ 上，并赋予随机模态振幅 $0 < S^{(n)} < 1, n = 0$、.....、17，对其精度和计算时间进行了测试。在每种情况下，源数量 N_s 均设为场点数量 N_f，其中 $N_s = 2^q$，$q = 10$、.....、16，且源项和场项使用相同的展开阶数。结果展示了不同 N_s、最大展开阶数 M 和树深度 d 下的表现。代码使用 GNU C 编写，采用 gcc 的 -O3 优化，并通过 Goto BLAS 执行矩阵-向量运算。计算在一台运行频率为 2.3GHz 的 Intel i5-6200U 笔记本电脑上使用一个核心完成。用于作为误差参考值的直接计算采用了类似的代码和优化方式。

对场的每个模态振幅分别计算误差,

$$\epsilon^{(n)} = \frac{\max|\Phi^{(n)}_{FMM} - \Phi^{(n)}_{D}|}{\max|\Phi^{(n)}_{D}|}, \tag{18}$$

其中，$\Phi^{(n)}_{FMM}$ 是使用新算法计算得到的模态振幅，$\Phi^{(n)}_{D}$ 是通过直接计算得到的模态振幅。图 8 展示了 $N_s = 2^{16}$ 误差随模态编号和展开阶数变化的示例结果。该方法显然具有很高的精度，尤其是在展开阶数较高时，轴对称模态可达到十一位数字的精度。在较大的 n下，误差会增大，此时模态振幅的绝对值较小，从而使相对误差测度变大。

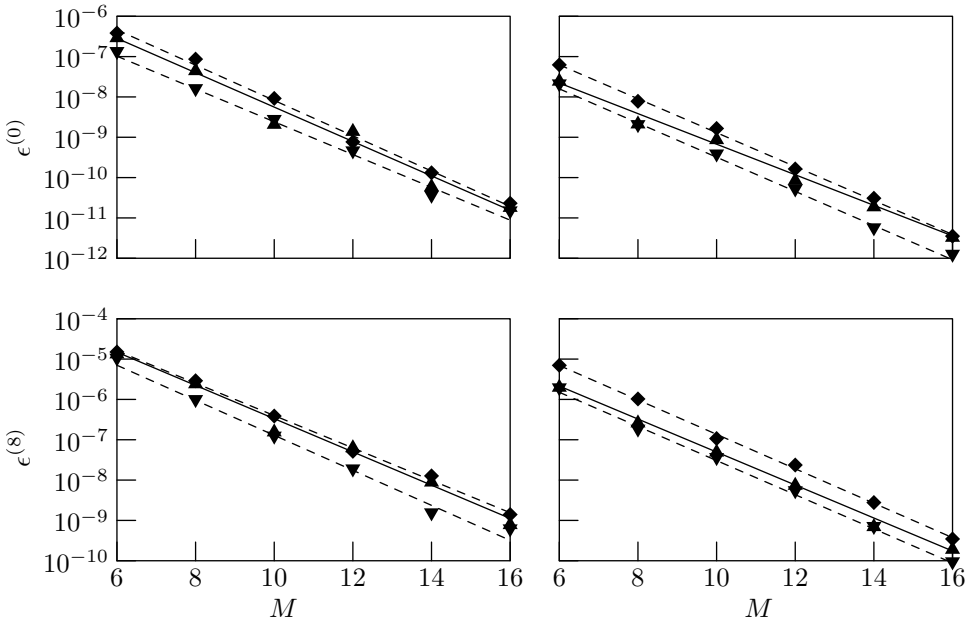


图9： $N_s = 2^{14}$ （左列）和 $N_s = 2^{16}$ （右列）的误差 ϵ： $n = 0$ （上排）和 $n = 8$ （下排）随展开阶数 M 的变化；菱形表示树深度 $d = 5$；上三角形表示 $d = 6$；下三角形表示 $d = 7$；拟合直线为$\epsilon \approx C^{-0.4M}$。

图9展示了轴对称 $n = 0$ 模式和 $n = 8$的误差随展开阶数的变化。误差近似按 $C^{-0.4M}$的比例缩放，对树深度的依赖较弱。在这里测试的问题规模范围内，该算法的收敛表现良好。

图10展示了计算时间随问题规模变化的基本数据。图中给出了直接计算的时间，其按预期以 $N_f N_s = N_s^2$的速率增长。对于低$M = 6$ 阶和高 $M = 16$ 阶两种情况，快速多极子算法的计算时间表现相似，只是随着展开阶数的变化，时间整体上移。对于较小的 N_s，两种情况下的计算时间大致恒定，此时求值时间主要由设置开销决定，而设置开销取决于树深度。随着问题规模增大，向下遍历的求值时间开始主导计算时间，其按与 N_s^2 成正比的方式增长，但其领先系数远小于直接计算。再次说明，这正是预期的行为，因为近场相互作用的直接计算会成为计算中占比最大的部分。随着树深度增加，盒间求值也会相应加快，因为每盒源点数会减少。

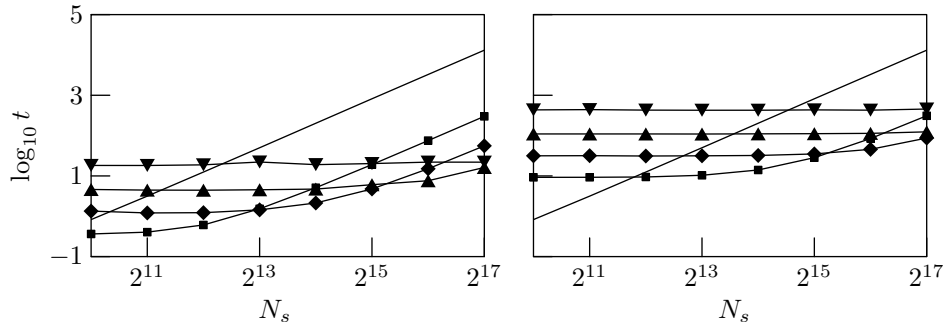


图10：直接计算（实线）以及树深度为4（方块）、5（菱形）、6（上三角形）和7（下三角形）时，计算时间随源点数和场点数的变化；左图：展开阶数为 $M = 6$；右图：展开阶数为 $M = 16$。

图11展示了计算的两个部分所耗计算时间的分布随树深度的变化。初始化阶段由向上遍历和向下遍历组成，其时间随问题规模近似线性增长，增长系数由树深度决定。随着盒子数量增加，初始化时间也随树深度增加。下图中的局部场求值对每盒平均源点数 $N_s/4^d$ 的扩展性良好，且随着树深度增加，所需时间减少。这意味着计算时间和精度取决于初始化与局部场求值之间的平衡，而这种平衡受源数量、展开阶数和树深度的影响

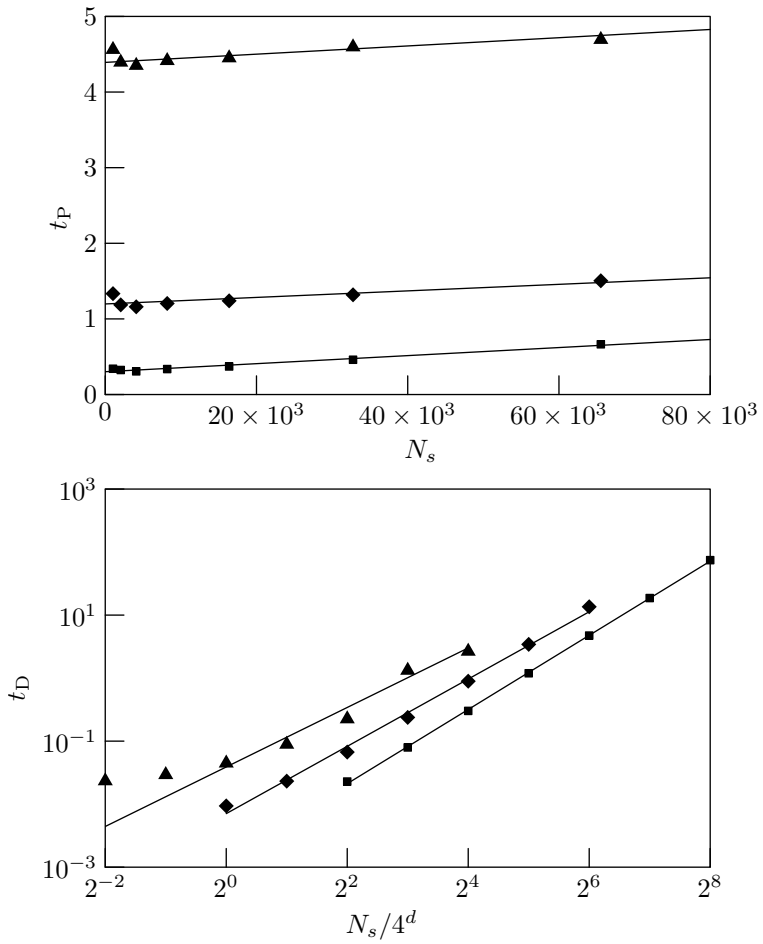


图11：展开阶数为 $M = 6$ 时各计算阶段的执行时间；盒子：树深度为 $d = 4$；菱形表示 $d = 5$，三角形表示 $d = 6$。上图为向上和向下遍历的时间及其线性拟合；下图为局部场求值的时间，以及对满足 $N_s/4^d \geq 1$ 的数据点进行的幂律拟合。

深度。

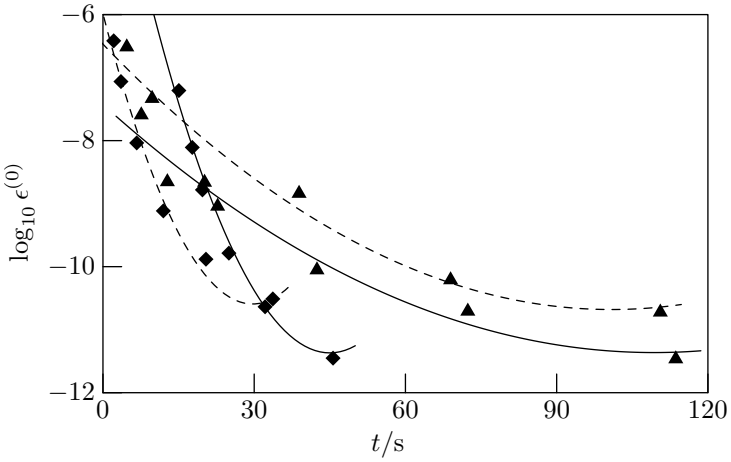


图12：误差 $\epsilon^{(0)}$ 与计算时间的关系：菱形表示深度 $d=5$，三角形$d=6$；实线表示 $N_s=2^{16}$；虚线表示 $N_s=2^{14}$；曲线为对数ϵ的二阶拟合。

图12展示了这种平衡关系：通过改变问题规模和展开阶数，并采用两种树深度，绘制了误差与计算时间的关系。对于每种树深度和源数量，误差均呈现出趋于最小值的趋势；若要达到更高的精度，较大的树深度则需要更长的总计算时间。

5 结论

快速多极子方法已扩展到圆柱区域中的非轴对称问题：通过计算泊松问题中源和场的傅里叶展开各模态的振幅来实现。与直接计算进行比较测试表明，该方法可收敛至高达十位数字的精度，并能显著提升速度，具体取决于展开阶数。仍有一些悬而未决的问题。首先是如何高效计算用于求得模态格林函数的勒让德函数；直接方法和快速方法都要用到这些函数，它们也是该方法中计算量最大的部分。其次是如何将该方法表述为能够使用BLAS 3级运算的形式，这应能进一步优化代码。最后，我们指出，本文采用的方法应同样适用于亥姆霍兹问题，尽管计算模态格林函数会更加困难。

A 格林函数及导数的求值

模态格林函数 $G^{(n)}(r, r_1, x)$ 定义如下:

$$G^{(n)}(r, r_1, x) = \int_0^{2\pi} \frac{\mathrm{e}^{\mathrm{i}n\theta_1}}{4\pi R}\,\mathrm{d}\theta_1, \tag{19}$$
$$R^2 = r^2 + r_1^2 - 2rr_1\cos\theta_1 + x^2.$$

Cohl 和 Tohline [3] 给出了 $1/R$的展开式,

$$\frac{1}{R} = \frac{1}{\pi\sqrt{rr_1}} \sum_{m=-\infty}^{\infty} \mathrm{e}^{\mathrm{i}m(\theta-\theta_1)} Q_{m-1/2}(\chi), \tag{20}$$
$$\chi = \frac{r^2 + r_1^2 + x^2}{2rr_1},$$

其中 $Q_\nu(\chi)$ 是第二类勒让德函数。对 θ_1积分可得

$$G^{(n)}(r, r_1, x) = \frac{Q_{n-1/2}(\chi)}{2\pi\sqrt{rr_1}}. \tag{21}$$

使用勒让德函数的递推 [7, 8.732.2], , 为简洁起见省略其对坐标的函数依赖,

$$(2n-3)G^{(n-2)} = 4(n-1)\chi G^{(n-1)} - (2n-1)G^{(n)}. \tag{22}$$

对于 $\chi > 1$, 向前递推不稳定, 而向后递推稳定但计算成本较高 [9]。为生成模态格林函数序列, 我们采用 Helsing 和 Karlsson [9] 的方法, 并对 $\chi < 1.008$使用向前递推, 初始值为 [3],

$$Q_{-1/2}(\chi) = \mu K(\mu), \tag{23}$$
$$Q_{1/2}(\chi) = \chi\mu K(\mu) - (1+\chi)\mu E(\mu), \tag{24}$$
$$\mu = \sqrt{\frac{2}{1+\chi}}.$$

这里, $K(\cdot)$ 和 $E(\cdot)$ 分别是第一类和第二类完全椭圆积分。这些积分使用 Carlson 的方法 [1]计算。

对于 $\chi \geq 1.008$, 从 $n = N + 80$ 的 $Q_{n-1/2}(\chi)$ 和 $Q_{n-3/2}(\chi)$ 任意取值开始, 采用向下递推直至 $n = 0$, 再利用已知的 $Q_{-1/2}(\chi)$ 值对序列进行缩放 (23)。使用此过程计算得到的 $G^{(n)}$ 值已与数值积分结果进行比较, 验证结果正确至机器精度。

已知 $G^{(n)}$、 $n=0$、.....、 N 的值后，可结合勒让德函数的递归关系和拉普拉斯方程求出 $G^{(n)}$ 的导数。为简洁起见，我们引入以下记号

$$g_{i,j,k}^{(n)}=\frac{\partial^{i+j+k}}{\partial r^i\partial r_1^j\partial x^k}G^{(n)}(r,r_1,x), \tag{25}$$

$$\overline{g}_{i,j,k}^{(n)}=\frac{1}{i!}\frac{1}{j!}\frac{1}{k!}\frac{\partial^{i+j+k}}{\partial r^i\partial r_1^j\partial x^k}G^{(n)}(r,r_1,x), \tag{26}$$

因此， $G^{(n)}$ 的泰勒级数为

$$G^{(n)}(r+\Delta r,r_1+\Delta r_1,x+\Delta x)=\sum_{m=0}^{\infty}\sum_{i+j+k=m}\overline{g}_{i,j,k}^{(n)}(\Delta r)^i(\Delta r_1)^j(\Delta x)^k. \tag{27}$$

导数采用基于拉普拉斯方程的递推方法计算，这与 Strickland 和 Amos 使用轴对称流函数方程 [11, 12] 的方法类似。这里，我们使用具有方位角依赖性 $\exp \mathrm{i}n\theta$ 的场的拉普拉斯方程。该递推需要初始值，这些值可利用勒让德函数的性质求得 [7, 8.732]。对于关于 x的导数，

$$\overline{g}_{0,0,1}^{(n)}=(n-1/2)\left[\left(\overline{g}_{0,0,0}^{(n)}+\overline{g}_{0,0,0}^{(n-1)}\right)\frac{x}{\rho_+^2}+\left(\overline{g}_{0,0,0}^{(n)}-\overline{g}_{0,0,0}^{(n-1)}\right)\frac{x}{\rho_-^2}\right], \tag{28}$$

$$\overline{g}_{0,0,k+1}^{(n)}=(n-1/2)\sum_{q=0}^{k}\frac{1}{q!(k+1)}\left[\left(\overline{g}_{0,0,k-q}^{(n)}+\overline{g}_{0,0,k-q}^{(n-1)}\right)\frac{\partial^q}{\partial x^q}\left(\frac{x}{\rho_+^2}\right)+\left(\overline{g}_{0,0,k-q}^{(n)}-\overline{g}_{0,0,k-q}^{(n-1)}\right)\frac{\partial^q}{\partial x^q}\left(\frac{x}{\rho_-^2}\right)\right], \tag{29}$$

$$\rho_\pm^2=(r\pm r_1)^2+x^2.$$

要计算 $n=0$ 的导数，可以使用关系式 $g_i^{(-1)}{}_{,j,k}\equiv g_i^{(1)}{}_{,j,k}$。对于关于 r的导数，

$$\overline{g}_{1,0,0}^{(n)}=-\frac{\overline{g}_{0,0,0}^{(n)}}{2r}+(n-1/2)\frac{r^2-r_1^2-x^2}{2r}\left[\left(\overline{g}_{0,0,0}^{(n)}+\overline{g}_{0,0,0}^{(n-1)}\right)\frac{1}{\rho_+^2}+\left(\overline{g}_{0,0,0}^{(n)}-\overline{g}_{0,0,0}^{(n-1)}\right)\frac{1}{\rho_-^2}\right], \tag{30}$$

$$\overline{g}_{1,0,k}^{(n)}=-\frac{\overline{g}_{0,0,k}^{(n)}}{2r}+\frac{n-1/2}{2r}\sum_{q=0}^{k}\frac{1}{q!}\left[\left(\overline{g}_{0,0,k-q}^{(n)}+\overline{g}_{0,0,k-q}^{(n-1)}\right)\frac{\partial^q}{\partial x^q}\frac{r^2-r_1^2-x^2}{\rho_+^2}+\left(\overline{g}_{0,0,k-q}^{(n)}-\overline{g}_{0,0,k-q}^{(n-1)}\right)\frac{\partial^q}{\partial x^q}\frac{r^2-r_1^2-x^2}{\rho_-^2}\right]. \tag{31}$$

导数 $g^{(n)}_{0\,1\,,,k}$ 可通过交换 r 和 r_1 求得，同时相应地交换导数中的指标。

最后，

$$\begin{aligned}\overline{g}^{(n)}_{1,1,k} = &-\frac{\overline{g}^{(n)}_{0,1,k}}{2r} \\ &+\frac{n-1/2}{2r}\sum_{q=0}^{k}\frac{1}{q!}\Bigg[\left(\overline{g}^{(n)}_{0,0,k-q}+\overline{g}^{(n-1)}_{0,0,k-q}\right)\frac{\partial^{q+1}}{\partial r_1\partial x^q}\frac{r^2-r_1^2-x^2}{\rho_+^2}+ \\ &\qquad\left(\overline{g}^{(n)}_{0,1,k-q}+\overline{g}^{(n-1)}_{0,1,k-q}\right)\frac{\partial^{q}}{\partial x^q}\frac{r^2-r_1^2-x^2}{\rho_+^2}+ \\ &\qquad\left(\overline{g}^{(n)}_{0,0,k-q}-\overline{g}^{(n-1)}_{0,0,k-q}\right)\frac{\partial^{q+1}}{\partial r_1\partial x^q}\frac{r^2-r_1^2-x^2}{\rho_-^2}+ \\ &\qquad\left(\overline{g}^{(n)}_{0,1,k-q}-\overline{g}^{(n-1)}_{0,1,k-q}\right)\frac{\partial^{q}}{\partial x^q}\frac{r^2-r_1^2-x^2}{\rho_-^2}\Bigg].\end{aligned} \tag{32}$$

为计算其余导数，我们利用基于柱坐标下拉普拉斯方程的递推。注意到 $\partial/\partial\theta \to \mathrm{i}n$,

$$\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial G^{(n)}}{\partial r}\right)-\frac{n^2}{r^2}G^{(n)}+\frac{\partial^2 G^{(n)}}{\partial x^2}=0. \tag{33}$$

由此得到关系式，

$$\overline{g}^{(n)}_{i+2,j,k}=\sum_{u=0}^{i}\frac{(-1)^u}{r^{u+1}}\frac{1}{(i+1)(i+2)}\left[\frac{(u+1)n^2}{r}\overline{g}^{(n)}_{i-u,j,k}-(i-u+1)\overline{g}^{(n)}_{i-u+1,j,k}\right], \tag{34}$$

该关系式可用于递推生成关于 r的高阶导数。交换 r 和 r_1，可得到相应的关于 r_1 的高阶导数关系式，并由此计算任意所需阶数的完整导数集。

参考文献

[1] B. C. Carlson。《实数或复数椭圆积分的数值计算》。《*SIAM* 数值分析杂志》，10:13–26，1995。

[2] VictorChurchill。《轴对称几何的快速多极子方法》。数学硕士，纽约大学库朗数学科学研究所，2016年5月。

[3] Howard S. Cohl 和 Joel E. Tohline。用于求解位势问题的紧凑柱坐标格林函数展开。《天体物理学杂志》，527：86–101，1999年。[4] Frank Ethridge 和 Leslie Greengard。二维快速多极子加速泊松求解器的新方法。《*SIAM*科学计算杂志》，23(3)：741–760，2001年。[5] William Fong 和 Eric Darve。黑箱快速多极子法。《计算物理学杂志》，228：8712–8725，2009年。[6] Zydrunas Gimbutas 和 Vladimir Rokhlin。非振荡核的广义快速多极子法。《*SIAM*科学计算杂志》，24(3)：796–817，2003年。[7] I. Gradshteyn 和 I. M. Ryzhik。积分、级数与乘积表。Academic出版社，伦敦，第5版，1980年。[8] L. Greengard 和 V. Rokhlin。粒子模拟的快速算法。《计算物理学杂志》，73：325–348，1987年。[9] Johan Helsing 和 Anders Karlsson。用于轴对称曲面积分方程的显式核分裂分片式 Nystr¨om 格式。《计算物理学杂志》，272：686–703，2014年。[10] M. Harper Langston、Leslie Greengard 和 Denis Zorin。基于快速多极子法的自由空间自适应三维偏微分方程求解器。《应用数学与计算科学通讯》，6(1)：79–122，2011年。[11] James H. Strickland 和 Donald E. Amos。轴对称环形涡旋系统的快速求解器。技术报告SAND90-1925，桑迪亚国家实验室，阿尔伯克基，新墨西哥州87185，美利坚合众国，1990年。[12] James H. Strickland 和 Donald E. Amos。轴对称环形涡旋系统的快速求解器。《美国航空航天学会杂志》，30(3)：737–748，1992年。[13] P. Young、S. Hao 和 P. G. Martinsson。用于定义在旋转对称曲面上的边界积分方程的高阶 Nystr¨om 离散格式。《计算物理学杂志》，231：4142–4159，2012年。