

文章编号: 2095-4980(2014)01-0067-04

## GPU 加速混合场积分方程求解导体目标散射问题

王 健, 顾长青, 李 苗, 段腾飞

(南京航空航天大学 电子信息工程学院, 江苏 南京 210016)

**摘要:** 利用图形处理单元(GPU)加速混合场积分方程(CFIE)分析导体目标电磁散射问题。较电场积分方程(EFIE)和磁场积分方程(MFIE), CFIE 消除了内谐振问题, 并且具有更好的条件数。求解的数值方法为基于 RWG 基函数的矩量法(MoM)。所有计算步骤均在 GPU 上实现, 包括: 阻抗元素填充、电压向量填充、矩阵方程的共轭梯度(CG)求解、雷达散射截面(RCS)计算。在保证数值精确定度的前提下获得了数十倍的速度提升。

**关键词:** 图形处理单元; 混合场积分方程; 矩量法; 共轭梯度法; 雷达散射截面

**中图分类号:** TN95

**文献标识码:** A

**doi:** 10.11805/TKYDA201401.0067

## Acceleration of CFIE for solving conductor target scattering problems by exploiting the GPU

WANG Jian, GU Chang-qing, LI Zhuo, DUAN Teng-fei

(College of Electronic and Information Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing Jiangsu 210016, China)

**Abstract:** The electromagnetic scattering analysis of conductor targets with Combined Field Integral Equation(CFIE) is accelerated by exploiting the Graphics Processing Unit(GPU). Compared with the Electric Field Integral Equation(EFIE) and Magnetic Field Integral Equation(MFIE), the CFIE could eliminate interior resonance problems, and has better condition number. The numerical technique is the Method of Moment(MoM) that based on RWG basis functions. And all the procedures are completed by the GPU, including impedance matrix filling, voltage vector filling, matrix equation solving with Conjugate Gradient(CG) algorithm, and the Radar Cross Section(RCS) calculation. More than ten times of speedups are achieved under the precondition to ensure the numerical accuracy.

**Key words:** Graphics Processing Unit; Combined Field Integral Equation; Method of Moment; Conjugate Gradient; Radar Cross Section

基于 RWG 基函数的矩量法被广泛用于求解诸如 EFIE, MFIE 等表面积分方程<sup>[1]</sup>。经典的矩量法简单实用, 然而阻抗元素填充和迭代求解的每一步时间复杂度却高达  $O(N^2)$ 。诸如快速多极子<sup>[2]</sup>、多层快速多极子<sup>[3]</sup>等多种快速算法已经被提出, 并且有效降低了计算的时间复杂度。

近年来, 硬件技术的发展在计算电磁学领域引发了许多关注, 其中之一便是面向 GPU 的加速技术。与 CPU 不同的是, GPU 被设计为处理图形渲染的并行流水线处理器, 称作流多处理器。在每个流多处理器中, 数百个独立的线程并发执行。得益于 GPU 的大规模并行处理能力, 其现在已可以处理各种非图形学的应用情景<sup>[4]</sup>。GPU 通用计算已经获得了长足进步, 并涌现出了许多支持此平台的编程开发环境, 例如 Cg, GLSL, Brook, ATI Stream SDK, CUDA 等<sup>[5]</sup>。

在 GPU 平台上使用矩量法求解表面积分方程已经成为当前的研究热点<sup>[6-7]</sup>, 主要集中在求解 EFIE 上, 然而其条件数较差, 迭代求解时收敛较慢, 且存在内谐振问题, 解决上述问题的有效办法是采用 CFIE<sup>[8]</sup>, 本文编写 GPU 程序便是面向 CFIE。并且, 整个求解过程均在 GPU 平台上完成, 当考虑到 GPU 与主机数据传输瓶颈时, 此方案具有更明显优势。

收稿日期: 2013-01-11; 修回日期: 2013-03-22

基金项目: 江苏高校优势学科建设工程资助项目; 国家自然科学基金资助项目(61071019); 东南大学基本科研业务费资助项目; 江苏省普通高校研究生科研创新计划资助项目(CXLX13\_092)

## 1 CFIE 及其矩量法解概述

文献[9]详述了导体目标表面积分方程的建立过程及其矩量法解。其中 EFIE 与 MFIE 可分别表示为：

$$\hat{\mathbf{i}}(r) \cdot \mathbf{E}^i(r) = jk\eta \hat{\mathbf{i}}(r) \cdot \iint_S [1 + \frac{1}{k^2} \nabla' \nabla' \cdot] \mathbf{J}(r') G(r, r') dr' \quad (1)$$

$$\hat{\mathbf{n}}(r) \times \mathbf{H}^i(r) = \frac{\mathbf{J}(r)}{2} - \hat{\mathbf{n}}(r) \times P.V. \iint_S \mathbf{J}(r') \times \nabla' G(r, r') dr' \quad (2)$$

式中： $\hat{\mathbf{i}}(r)$  与  $\hat{\mathbf{n}}(r)$  表示目标表面的单位切向量、单位法向量； $\mathbf{E}^i(r)$ 、 $\mathbf{H}^i(r)$  表示入射电场、入射磁场； $\mathbf{J}(r)$  与  $\mathbf{J}(r')$  表示场点与源点的表面电流； $G(r, r')$  表示自由空间格林函数； $k$  表示波数。

CFIE 由 EFIE 和 MFIE 线性组合而成，表示为： $\alpha EFIE + (1-\alpha)\eta MFIE$ ，其中  $\alpha$  为常数，典型值为  $0.2 \leq \alpha \leq 0.5$ ； $\eta$  为空间波阻抗。

使用矩量法求解 CFIE，首先对未知电流按 RWG 基函数  $f$  做加权展开，然后使用伽略金法对式(1)和式(2)中的等式两边分别做内积，得到矩阵方程：

$$\sum_{i=1}^N \mathbf{Z}_{mn} \mathbf{I}_n = \mathbf{V}_m \quad (3)$$

EFIE 与 MFIE 对应的阻抗矩阵元素、电压向量元素分别为：

$$\mathbf{Z}_{mn}^E = \frac{jk\eta}{4\pi} \iint_{f_m} \iint_{f_n} \{f_m(r) \cdot f_n(r') - \frac{1}{k^2} [\nabla \cdot f_m(r)] [\nabla' \cdot f_n(r')]\} \frac{e^{-jkr}}{r} dr' dr \quad (4)$$

$$\mathbf{V}_m^E = \iint_{f_m} f_m(r) \cdot \mathbf{E}^i(r) dr \quad (5)$$

$$\mathbf{Z}_{mn}^H = \frac{1}{2} \iint_{f_m=f_n} f_m(r) \cdot f_n(r) dr + \frac{1}{4\pi} \iint_{f_m} f_m(r) \cdot \iint_{f_n} \hat{\mathbf{n}}(r) \times [(r-r') \times f_n(r')] [1 + jkr] \frac{e^{-jkr}}{r^3} dr' dr \quad (6)$$

$$\mathbf{V}_m^H = \iint_{f_m} f_m(r) \cdot [\hat{\mathbf{n}}(r) \times \mathbf{H}^i(r)] dr \quad (7)$$

进而，可以按照 CFIE 的表达式求得 CFIE 的阻抗矩阵元素、电压向量元素。

电流向量通过式(3)求解得到，可以使用直接求解法(如 LU 分解)与迭代法(如 CG, GMRES)，由于在 CG 方法中的矩阵向量积具有良好的并行实现可行性，因此以此作为研究对象。

## 2 GPU 的加速实现

### 2.1 CUDA 编程模型

在底层，CUDA 采用线程束(warps)来创建、管理、调度和执行线程；在上层，CUDA C 屏蔽了底层实现细节，取而代之的是提供了丰富的使用接口，简化了操纵 GPU 计算实现并行方法。其定义 C 接口函数 kernel，每个 kernel 标识了唯一三维向量 ID，用以描述一维、二维或者三维的线程块(Block)。与此同时，若干个线程块构成了一个格(Grid)。

CUDA C 涉及 2 个平台：包含 CPU 的主机(host)，执行串行程序；包含 GPU 设备(device，通常为支持 CUDA 的图形显示卡)，执行并行程序，为典型的异构编程环境，如图 1 所示。

本文中，主机的 CPU 为 Intel CORE i3@2.1 GHz，其中串行程序用以执行 CFIE 计算所需的参数(如目标网格、入射频率、入射方向、散射方向等)。设备的 GPU 为 NVIDIA GT540M 型芯片，其中的并行程序用以求解 CFIE。

### 2.2 阻抗元素的填充

在填充阻抗矩阵时，阻抗元素(不论自阻抗还是互阻抗)的计算具备各自独立性：由式(4)和式(6)可知，其是耦合三角面元对通过基函数与检验函数反映出的互作用。若已知入射场及某三角面元对的几何信息，阻抗矩阵

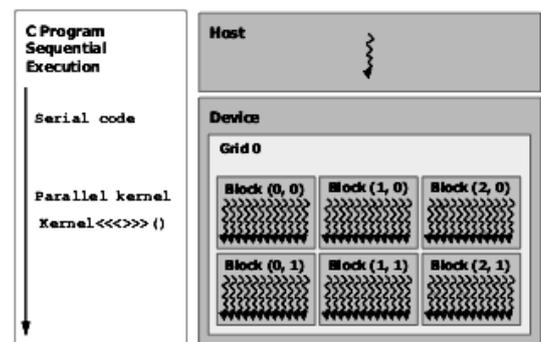


Fig.1 Heterogeneous programming model of CUDA C

图 1 CUDA C 异构编程模型

元素便可直接求出。在此,将阻抗元素矩阵  $Z$  映射到一个由线程块构成的二维 Grid 中,每个 Block 推荐的大小为  $16 \times 16$ ,而每个线程负责计算对应的阻抗矩阵元素,如图 2 所示。

Block 的个数由模型网格大小决定,如,某个导体球有 4 764 个未知量,因此阻抗矩阵大小为  $4\,764 \times 4\,764$ ,为覆盖所有阻抗元素,需要加载至少  $298 \times 298$  个线程块(每个线程块大小为  $16 \times 16$ )。

### 2.3 矩阵方程的求解

在电流向量求解阶段,使用 CG 算法求解矩阵方程  $Ax=b$ ( $A$  对应阻抗矩阵  $Z$ ;  $b$  对应电压向量  $V$ ;  $x$  对应电流向量  $I$ )的步骤如下:

Initial steps:

1) Guess  $x_0$ ; 2)  $r_0 = Ax_0 - b$ ; 3)  $P_1 = -A^+r_0$

Iterate( $i=1,2,\dots$ ,until convergence),do

$$1) \alpha_n = -\frac{\langle AP_n, r_{n-1} \rangle}{\|AP_n\|^2} = -\frac{\|A^+r_{n-1}\|^2}{\|AP_n\|^2};$$

$$2) x_n = x_{n-1} + \alpha_n P_n;$$

$$3) r_n = Ax_n - b = r_{n-1} + \alpha_n AP_n;$$

$$4) \beta_n = \frac{\|A^+r_n\|^2}{\|A^+r_{n-1}\|^2};$$

$$5) P_{n+1} = -A^+r_n + \beta_n P_n$$

其中,  $A^+$  表示  $A$  的厄米矩阵,CG 需要包含的矩阵向量积有:  $A^+r_{n-1}$ ,  $AP_n$  以及  $A^+r_n$ 。

如图 3 所示,提取出矩阵不同行向量 Row 与向量 Vector 同时做内积运算,结果存储在 Product 向量中。对应 GPU 中,某个线程完成了其对应 Row-Vector 的内积运算。

### 2.4 其余计算部分

电压向量填充与 RCS 计算的复杂度为  $O(N)$ ,并非主要耗时过程。然而,仿真计算数据通过连接主机与设备的 PCIe 总线实现交互,若频繁进行此操作,时间开销将相当可观<sup>[5]</sup>,即 CUDA C 程序的瓶颈所在。因此,为改善程序运行效率,数据应尽可能久地驻留在设备中。基于此,电压向量填充和 RCS 的计算同样在 GPU 中并行实现。

## 3 数值结果

为了验证本文 GPU 程序的正确性,图 4 给出了 Mie 级数法与本文 GPU 程序计算半径为 1 m 的理想导体球双站 RCS 比较结果。由图 4 可知,GPU 程序与 Mie 级数法结果吻合较好,具有可靠的精确度。

为获取 GPU 程序的加速效果,创建了不同未知量的模型用以比较:每个立方体边长 1 m,包含 900 个未知量,第  $n$  组包含  $n$  个立方体,沿着  $x$  轴排列,如图 5 所示。测试了最

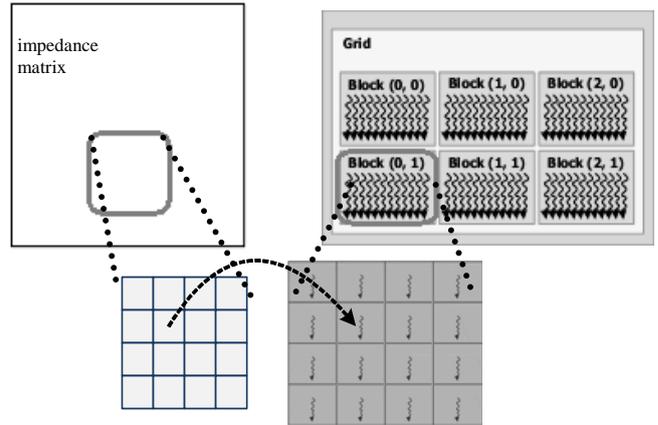


Fig.2 Mapping relationship between impedance elements and GPU threads  
图 2 阻抗元素与 GPU 线程的映射关系

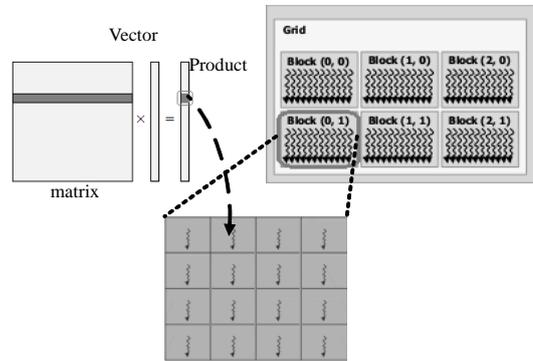


Fig.3 MVP parallel implementation on the GPU platform  
图 3 MVP 在 GPU 平台上的并行实现方法

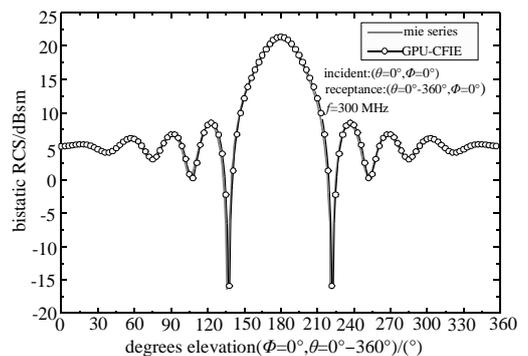


Fig.4 Bistatic RCS of sphere  
图 4 理想导体球的双站 RCS

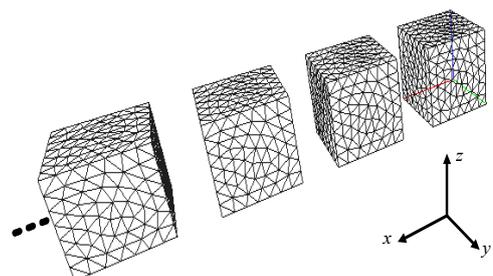


Fig.5 Cube array of perfect electric conductor  
图 5 理想导体立方体阵列

大  $n=9$  个立方体阵(包含 8 100 个未知量)。图 6 给出了不同未知量情况下的阻抗元素填充时间比较。由图 6 可知, GPU 程序使得阻抗元素填充加速了约 14 倍。表 1 给出了使用 CG 迭代算法的时间比较, 包含了每次求解的迭代步数。由表 1 可以看出 CFIE 算法收敛较快, 在此阶段, 获得了大约 33 倍的速度提升。

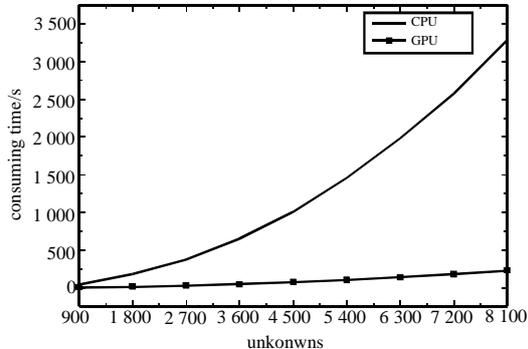


Fig.6 Consuming time of impedance filling of GPU vs CPU  
图 6 CPU 程序与 GPU 程序填充阻抗元素时间比较

表 1 CG 方法求解矩阵方程时间  
Table1 Consuming time of CG algorithm  
for solving matrix equation

unknowns	iteration steps	CG solving time/s	
		CPU	GPU
900	29	26.605	0.961
1 800	37	134.479	4.302
2 700	38	311.513	10.396
3 600	36	524.978	17.893
4 500	42	952.873	29.913
5 400	40	1 331.567	41.748
6 300	44	2 020.699	63.952
7 200	41	2 468.388	74.751
8 100	46	3 601.412	105.614

## 4 结论

本文论述了 GPU 加速 CFIE 求解导体目标的散射问题, 同时强调了主机与 GPU 传输数据带来的瓶颈问题。将 CFIE 的所有计算步骤应用在 GPU 平台上, 数值结果与经典的 CPU 程序吻合较好, 证明了 GPU 的计算密集性、高度并行特性带来的可观的加速效果。

## 参考文献:

- [1] 青滔, 聂在平, 宗显政. 波束波导电磁传输问题的积分方程数值解[J]. 太赫兹科学与电子信息学报, 2011, 9(5): 573-577. (QING Tao, NIE Zaiping, ZONG Xianzheng. Numerical solutions of the integral equation for the electromagnetic transmission in the beam waveguide[J]. Journal of Terahertz Science and Electronic Information Technology, 2011, 9(5): 573-577.)
- [2] Coifman R, Chew W C. The fast multipole method for the wave equation: a pedestrian prescription[J]. Antennas and Propagation Magazine, IEEE, 1993, 35(3): 7-12.
- [3] Song J, LU Caicheng, Weng ChoChew. Multilevel fast multipole algorithm for electromagnetic scattering by large complex objects[J]. IEEE transactions on antennas and propagation, 1997, 45(10): 1488-1493.
- [4] 俞惊雷, 柳彬, 王开志, 等. 一种基于 GPU 的高效合成孔径雷达信号处理器[J]. 太赫兹科学与电子信息学报, 2010, 8(4): 415-419. (YU Jinglei, LIU Bin, WANG Kaizhi, et al. A highly efficient GPU-based signal processor of Synthetic Aperture Radar[J]. Journal of Terahertz Science and Electronic Information Technology, 2010, 8(4): 415-419.)
- [5] NVIDIA Corporation. NVIDIA CUDA Programming Guide Version 4.2[EB/OL]. [2013-01-11]. <http://www.nvidia.com/cuda,2012>.
- [6] Peng S, Nie Z. Acceleration of the Method of Moments Calculations by Using Graphics Processing Units[J]. IEEE Transactions on Antennas and Propagation, 2008, 56(7): 2130-2133.
- [7] Lezar E. GPU-Accelerated Method of Moments by Example(Monostatic Scattering)[J]. IEEE Antennas and Propagation Magazine, 2010, 52: 120-135.
- [8] Yla-Oijala P, Taskinen M. Calculation of CFIE impedance matrix elements with RWG and  $n \times$  RWG functions[J]. IEEE Transactions on Antennas and Propagation, 2003, 51(8): 1837-1846.
- [9] Gibson W C. The Method of Moment in Electromagnetics[M]. New York: Wiley-Interscience, 2007.

## 作者简介:



王 健(1987-), 男, 南京市人, 硕士, 主要研究方向为计算电磁学、GPU 高性能计算。  
email:wangjian.nuaa@hotmail.com.

顾长青(1958-), 男, 南京市人, 博士生导师, 教授, 主要研究方向为计算电磁学、目标特性分析与控制、宽带天线、智能天线与天线小型化、电磁兼容性设计与电磁环境预估等。

李 茁(1979-), 男, 南京市人, 副教授, 主要研究方向为计算电磁学、电磁兼容等。

段腾飞(1988-), 男, 山东省济宁市人, 硕士, 主要研究方向为计算电磁学、电磁兼容。