

分类号_____ 密级_____

UDC _____ 编号_____

大连理工大学

博士后研究工作报告

Particle-In-Cell 模拟的发展：物理考虑 和计算技术

王虹宇

指 导 教 师： 王友年

流动站（一级学科）名称： 物理学

专业（二级学科）名 称： 等离子体物理

报 告 完 成 日 期： 2010-7-9

Particle-In-Cell 模拟的发展：物理考虑和计算技术

**The Developments of Particle-In-Cell
Simulation: Physical consideration and
Computational technology**

计：研究报告 80 页
表 格 3 个
插 图 5 幅

研究工作起始时间 2007-9

研究工作期满时间 2010-6

单 位 名 称 大连理工大学

大连理工大学
Dalian University of Technology

摘 要

粒子云网格(Particle In cell)方法是研究等离子体行为的关键性理论工具之一。它可以看成方便而粗糙的流体力学模拟和严格但困难的动理学方程求解的一种有效的折中。从七十年代建立了PIC技术的基本框架以来,这种技术被应用到不断扩展的领域中,从机理研究到具体设备的设计都从中受益。

尽管PIC方法的基本数学框架并没有实质性的变化,但随着使用领域的推广,问题背景的变化对PIC模拟技术的细节提出了越来越多的要求:实际工程模拟和设备设计要求模拟方法提供更快的速度和更好的适应性;新物理现象的机理研究要求模拟必须可以考虑更多的物理效应甚至化学问题;极端条件下的物理问题的理解要求模拟具有更高的精度或者使用全新的物理理论;多尺度问题的存在要求模拟中使用更强有力的数学方法;最后,部分问题的极端复杂性需要使用混合模拟来克服,而这要求PIC方法和其他模拟手段的无缝对接。在现在的模拟技术中,PIC模拟方法向上连接到连续模拟(流体力学等),而向下重叠了纯粹的分子动力学模拟(MD),在这个庞大的领域中,提供了各种复杂问题的分析方法。

在另外一方面,从上世纪末期开始,简单而廉价的并行技术,如集群(cluster)成为计算机技术的亮点之一。集群技术的出现使得高性能计算机变成了能够普及的产品。传统上巨大计算量的PIC模拟技术从中获得了极大的帮助。在近几年,“桌面上的超级计算”进一步地提供了PIC模拟方法的发展希望。相应地,这类非传统的高性能计算技术常常意味着以前的模拟程序框架被彻底地重建(即使数学上没有本质区别)以便发挥硬件的性能。这也对PIC模拟技术提出了新的挑战。

本文将概述近年来PIC技术的主要进展,包括我们在上述两方面的若干工作。在论文最后,将分析现阶段存在的关键性问题及可能的突破口。

关键词: 粒子云网格方法 等离子体模拟 高性能计算

Abstract

The Particle-In-Cell (PIC) Method is a key tool for the investigation of plasma theory. It can be regarded as a tradeoff between the hyperdynamic simulations (convenience but inaccuracy) and the complete kinetic solving (strict but difficult). Since the construction of PIC framework in 1970s, the PIC technologies have been applied to increasing regions. Both the investigations of physical mechanism and device design gain benefits from it.

Although the mathematic frameworks of PIC have not be changed substantially, more and more demands of the technological details are introduced because of the varied problem's background. The device design and real project's simulating need higher speed and more adjustability; the analysis of physical problems in extreme conditions need higher precision simulation or fresh new theory being applied; the multiple scale problems need more powerful mathematical methods be applied. Finally, investigation of some very complex problems need (general) hybrid simulation. Which need the composition of PIC method and other simulating methods. In present simulating technologies, PIC methods attached the continue simulation (hyperdynamic simulation) upward and linked to Molecular Dynamics (MD) simulation downward. In the all regions, PIC methods gave out methods for complex problems.

On the other hand, since the ending of the last century, some simple and cheap parallel computing technologies as cluster system become the high point of the computational technology. The invention of cluster technology make the High Performance Computing (HPC) turn to popularized products. The PIC methods gains much benefits from it. In the recent years, "desktop-scale super-computing" provide new hopes of the development of the PIC methods. These non-traditional HPC technology means the code frameworks must be reconstruct completely to take advantage of the hardware capabilities even without mathematical change. These problems raised new challenge to the standard PIC technology.

This article will review the process on the both sides and analyse the key problems to find possible breakthroughs.

Key words: Particle-In-Cell Plasma Simulation High Performance Computing

目 录

1	概述	1
1.1	基本概念	1
1.2	物理背景和进展	3
2	Weighting和插值	7
2.1	有限大小粒子和weighting	7
2.2	空间网格宽度: 动量守恒和能量守恒	8
2.3	变宽度网格和可变权重粒子	10
3	场建模及其求解	13
3.1	电磁场模型的分类	13
3.2	电磁模型: Maxwell方程的求解	14
3.3	静电模型:Poisson方程	19
3.4	鞘层自偏压和外电路建模	25
3.5	频率分离:有质动力和低频场	27
4	尺度问题: 数学考虑	31
4.1	时间和空间尺度问题: 定标	31
4.2	直接隐式模拟	33
4.3	矩方法	38
4.4	多尺度和网格细化模拟	41
5	粒子-粒子耦合: 物理考虑	43
5.1	蒙特卡洛耦合: 带电粒子-中性粒子碰撞	43
5.2	库仑碰撞	45
6	计算机技术问题	48
6.1	性能优化: 算法和体系结构	48
6.2	并行处理: 基本考虑	52
6.3	非局域求解器问题	54
6.4	GPU处理和其他硬件体系	59

7	实例和面向问题的建模	61
7.1	基本的PIC模拟程序.....	61
7.2	非结构网格上的放电模拟.....	63
7.3	QuickPIC	65
8	结论：问题和展望	68
	参考文献	71
	致谢与声明	81

主要符号和缩写

PIC	粒子云网格方法 (Particle-In-Cell)
cluster	集群
PPC	每网格粒子数 (Particle per Cell)
FPE	福克-普朗克方程 (Fokker-Planck Equation)
$S_m(x)$	m阶插值函数
ρ	电荷密度
λ_D	德拜长度
ω_p	等离子体频率
e	电子电荷
m	电子质量
c	光速
ϕ	电势
ϵ_0	真空介电常数
μ_0	真空磁导率
J, j	电流密度
Ψ	辅助标量场
LWFA	激光尾场加速
PWFA	等离子体尾场加速

1 概述

1.1 基本概念

数值模拟是研究等离子体演化问题的重要手段^{[1][2][3][4][5][6][7]}。通常的等离子体数值模拟技术可以按照其数学模型分成两类，一类是基于等离子体的流体模型，通过等离子体的密度，流速，温度等局部平均量的演化方程进行的数值研究，这种模拟称为流体模拟。另一类是基于动理学模型，即直接在相空间中研究等离子体分布函数的演化。对于等离子体来说，流体力学方程基本上是Vlasov方程的头几阶矩方程，是Vlasov方程对于速度空间做平均的结果，因此必须适当地假定速度空间分布函数以及局域输运系数才能建立。这使得流体力学模拟中不能包含各种动理学和非局域效应。动理学研究是严格和自洽的，然而由于Vlasov方程是(3+3)维度的，其直接数值求解远远超出现阶段计算机的能力。

此外，无论是流体力学模拟还是直接处理Vlasov方程，还存在一些和数值方法相关的困难。其中之一和所谓数值耗散效应有关：非线性双曲型方程的数值求解很容易带来数值振荡，在间断面附近尤其明显；为了消除这种振荡，维持计算的稳定性，通常需要在求解中引入人为阻尼(或限幅)，而这种阻尼常常会影响到一些小尺度过程的正确建模。

作为Vlasov问题的一种折衷处理手段，等离子体的Particle-In-Cell(PIC)模拟^{[1][2]}是较为成熟的手段之一。PIC模拟本身基于简单的“宏粒子”概念，即用一些宏粒子（云）的运动来代替实际等离子体带电粒子的运动，每个宏粒子可以看成一堆一起运动的等离子体粒子，其荷质比等于真实粒子的荷质比，同时具有一个权重参数，表明每个宏粒子代表多少个真实粒子。模拟流程如图(1.1)，首先定义模拟所用的空间网格和宏粒子，然后执行下述模拟循环：

(1)将网格内粒子的电荷分配到网格格点上来获得电荷电流分布。

(2)通过在网格上求解电磁场方程来计算粒子之间的电磁场。

(3)得到电磁场分布后，用经典或者相对论粒子的运动方程来求解宏粒子的运动。

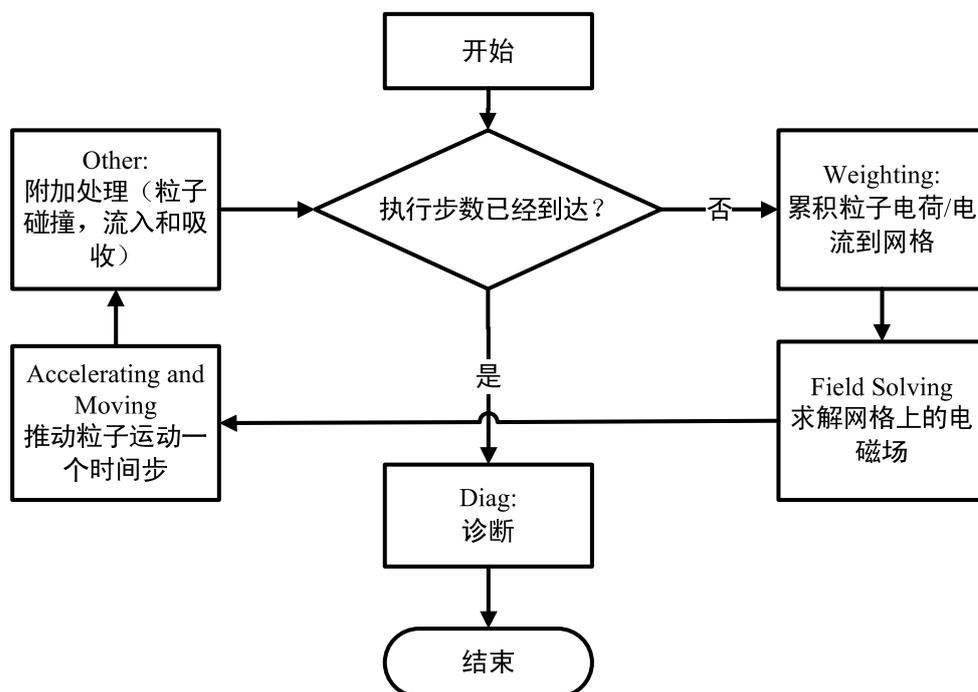


图 1.1 PIC模拟流程

(4)执行其他操作，例如电离和碰撞等等。

如此循环，就得出对等离子体运动模式的描述。这种模拟方法自然的包含了动理学效应，原则上可以处理几乎所有的等离子体问题，只要粒子和网格的数目足够多。在涉及到快速电子生成的问题中，PIC模拟是一种常用的研究手段。相对于直接求解Vlasov方程，PIC模拟的最明显优点可以通过相空间的“相流体”分布看出^[8]：当等离子体运动的时候，和坐标空间的流动不同，相流体并不会保持在大致不变的区域中；相反，相流体会剧烈地变形，导致相空间的总体积变得非常庞大，而其中绝大部分是空的，只有很少的部分才包含相流体，因此大部分计算资源都被浪费来处理相空间中的空区域。相反，PIC模拟跟踪的永远是存在等离子体的相空间部分。这种优点意味着PIC方法非常适合（常常是唯一的方法）模拟射束问题，也就是少数具有高动量和射程的粒子在基本静止的背景中运动的情形。

PIC模拟也存在许多方面的缺点，其中最令人熟知的问题在于，粒子是一种离散的目标，因此在计算粒子的相空间密度的时候必然存在散粒噪声。这种噪声的强度反比于使用的宏粒子的数目平方根，因此为了获得比较平滑的结

果，必须使用非常大数目的模拟宏粒子。这个问题在粒子运动比较局域的时候变得尤其严重，因为这时候需要了解每个局部的具体分布函数。在这种情况下，所关心的区域中，每个网格的粒子数(Particles per Cell, PPC)成为严重的限制因素。

另外一个典型的问题是，PIC模拟基本上截断了粒子之间的短程相互作用。在PIC方法中，宏粒子之间的相互作用是软的，也就是没有短程库仑碰撞。对于一般等离子体问题，这个特性是必须的；但当密度和自由程改变导致大动量的库仑碰撞占据主导的时候，这个问题变得很难处理，并且导致模拟结果的失真。在某些情况下，可以通过在PIC模拟中添加随机的短程碰撞过程来解决这个问题，但随机模拟会进一步加大噪声，于是需要更多的模拟粒子。一些类似的问题，例如原子的碰撞电离和激发等等，也依赖于这种PIC+随机模拟^[9]，这类模拟手段统称为PIC/MC模拟。

PIC模拟的第三个也是最根本的困难和数学方法有关：不同于流体力学模拟，标准的PIC模拟基本上是无阻尼的，特别是时间-空间精细行为并不会自动地被数值方法阻尼掉。这在物理上是大家所需要的，但同时却意味着实际的模拟时间空间步长必须能够分辨这些精细行为，以免出现数值不稳定性。于是，PIC模拟的时间和空间步长往往都比其他模拟方法小得多。

所有这些问题最后都归结为计算资源的开销。对于PIC模拟来说，计算量基本上可以理解为时间步数 \times 空间网格数目 \times PPC。通常情况下，这个开销远远高于流体力学模拟。进一步地，由于1-D或者2-D模拟常常不能充分的揭示问题的物理特性，有时候必须进行全3维模拟^[8]，而3D模拟的计算量非常高，即使对于超级计算机，也常常是非常困难的。不断添加的物理模型进一步增加了计算量：例如，引入碰撞电离的情况下，每网格需要的粒子数常常数十倍于纯粹的PIC模拟。

1.2 物理背景和进展

尽管有上述的缺陷，PIC方法仍然被广泛的使用，一方面，PIC方法是一种第一原理出发的模拟，几乎不需要唯像参数，因此比较适合研究物理机理问题；另一方面，对于很多问题，特别是粒子自由程和整个系统的尺度可以相比的情

形，PIC仍然是最有效（并且常常是唯一可以使用）的研究方法。这类问题的典型有下面一些（但不止这些）：

高功率微波器件，例如磁控管，行波管等。在这类器件中，具有决定性意义的是宏观电磁场和电子的相空间行为。在腔内的电磁波和带电粒子运动都必须被建模，同时碰撞过程基本可以忽略（因为几乎是在真空背景中），于是，可以用PIC进行相当有效的分析模拟^{[10][11][12]}。直线加速器等设备也属于这一类^[13]。

低气压下的放电过程，例如在微电子制造工艺中经常碰到的射频等离子体源^[14]。当背景气压非常低时，电子自由程接近甚至大于整个放电器件的尺寸；同时电子具有一些和轨道相关的能量获取机制（例如费米加热），因此电子分布会显著偏离麦克斯韦分布；最后，鞘层行为常常很重要，于是必须考虑局部电荷分离效应。这时，流体力学模拟难以跟踪其中发生的各种物理过程，而传统的扩散近似FPE模拟方法由于电子自由程太长也无法使用，通常都需要用PIC方法建模。

极强外场（激光）和稀薄等离子体的相互作用^{[8][15][16]}。由于外场极强，电子具有很高的颤动能量，并且会被有质动力以及尾场加速到极高的能量，电子的能量分布是高度非各向同性的，甚至难以使用球谐函数展开；由于能量非常高，碰撞截面很低，以至于可以完全忽略电子-电子碰撞效应；最后，极少数被加速粒子是物理上主要关心的目标，大部分背景粒子甚至可以当成冷等离子体描述。这样的问题也非常适合用PIC方法处理。

在另一个方面，也存在传统上不适合用PIC方法研究的过程，这主要发生在等离子体密度非常高，和/或碰撞截面非常大的情况下。这类情况下，一方面PIC模拟的计算量会成数量级地上升，另一方面由于固有的噪声问题，模拟结果会变得相当不可靠。应该指出的是密度和碰撞截面参数都是相对于系统的实际尺度的，在激光加速问题的数十微米尺寸内， $10^{19}cm^{-3}$ 的电子密度仍然可以认为相当稀薄，而在放电问题中，由于特征长度达到几个厘米，即使 $10^{11}cm^{-3}$ 的电子密度就可以看做非常高了。

无论如何，理论上只要空间网格够细，每个网格内的粒子数够多，那么PIC模拟完全等同于直接求解Vlasov方程。换句话说，PIC模拟能考虑的物理效应等价于能写进Vlasov方程中的项目。而Vlasov方程包含了弱碰撞（或者无

碰撞)区几乎所有重要的等离子体物理。因此,只要克服了PIC方法的巨大计算量困难,PIC方法原则上可以解决我们关心的绝大部分等离子体问题。但在实际中,如同前述的情况,多维的PIC模拟,其计算量过于庞大,基本不可能有效地处理。因此,几十年来,很多人都致力于对PIC方法的改进:对于那些适合PIC方法处理的问题,希望能得到更准确和适应更复杂环境的模拟程序;而对传统上不适合PIC方法的问题,则致力于绕过PIC方法处理这类问题的困难。这种改进分为几种不同的类型。

第一种是通过修改PIC的基本框架和物理原理,使得PIC方法可以研究其它物理效应。这方面的代表性工作就是利用附加的Monte-Carlo过程来引入粒子-粒子短程耦合^{[9][17]},从而使得PIC方法可以处理碰撞和电离、合并等等过程。这种方法已经被大量应用在低温等离子体物理研究中,特别是关于放电过程的模拟。通过引入库仑碰撞模型或者直接静电相互作用求和法,这种方法也被引入惯性约束核聚变等具有高密度的等离子体问题中^[18]。

第二种类型的改进是在PIC方法中引入近似,得到部分第一原理和部分简化模型(或唯像模型)的合并结果。这常常通过把理论(解析)研究中使用的某些近似移植到数值方法中实现。例如,在强激光和稀薄等离子体的相互作用中,激光场可以被近似地当成有质动力源,而等离子体中的电磁场就可以描述为有质动力和等离子体自生电磁场的组合^{[19][20]}。另一种典型的例子是放电模拟中的Boltzmann近似,即只处理离子的运动,而电子被当成包围在离子附近的无惯性电荷云。利用这种方法,往往可以在不损失关键性物理效应的同时,极大地降低计算资源的开销。

第三种类型的改进是通过数学上的强有力技巧来化简研究问题。在这方面有两个有趣的例子。其中之一是所谓“隐格式”模拟方法^{[21][22]},通过和计算流体力学中相似的阻尼技巧,这种方法可以克服PIC模拟固有的时空网格尺寸问题,对大尺寸和高密度等离子体这类本来完全无法处理的问题进行快速的定性分析。另一个例子是在激光加速问题中发展的“Boost Frame”技巧^[23],通过简单的洛伦兹变换来提高计算效率。这方面的改进还有另外一个部分,即通过改变数学模型来提高模拟的精度。在涉及到极端情况的模拟问题中,个别数学模型的准确度可能会带来结果的严重影响,甚至定性上的差别。

最后一种类型的改进是计算技术上的。90年代之后,Beowulf系统的出现使

得cluster体系的并行机开始普及。这种并行系统具有低得多的成本和很高的可操纵性，以前难以实现的大资源量模拟现在可以通过廉价的并行集群完成。于是，如何将各种具体的算法并行实现成为PIC技术研究的重要领域。最近几年，一些非传统的计算技术，特别是GPU计算或者FPGA定制计算之类的方法，有望成为克服计算资源/成本问题的一条出路。这方面的研究，也是当前PIC技术的重要部分之一。

和一般的数值模拟方法的发展不同，在PIC技术中，后三种改进的基本目标都是加快求解问题的速度。在许多模拟方法中，大家集中注意力于如何在问题中包含更多的物理内容。而PIC技术的进步更多的是在改进求解问题的速度。这是因为通常所说的PIC方法“不能做某件事”并不是真的不能做，而只是不能在有意义的时间内完成而已。速度问题是PIC模拟的重要问题（重要程度远远超过其他计算物理领域），未来可能也是唯一的根本问题。

本文将介绍PIC模拟技术的一些重要进展，并分析当前研究中急需的一些突破。由于一些比较成熟的进展（特别在放电领域）已经很好地描述在J.P.Verboncoeur^[24]和D.Tskhakaya^[25]的评述文章中，因此下面将主要介绍相对不那么熟知的一些进展。从前面的流程分析中可以看到，PIC模拟技术的主要步骤包括（1）累积电荷，即Weighting过程；（2）求解电磁场；（3）推动粒子；（4）附加处理。下面将分别研究这几个过程，概述其主要困难所在和技术改进。

2 Weighting和插值

2.1 有限大小粒子和weighting

Particle-in-Cell方法的基础，在于使用有限大小粒子，或者说从粒子到网格的插值。这个过程形式上是一个从宏粒子坐标 x_p 映射到网格点上的密度函数 $\rho(x_g)$ 的过程，其中 x_g 是网格点的坐标。引入形状函数 $S(x)$ ，可以把这个插值过程写成

$$\rho(x_g) = \sum_p q_p S(x_g - x_p) \quad (2-1)$$

很明显， $S(x)$ 必须满足归一化条件： $\int_{-\infty}^{\infty} S(x - x_p) dx = 1$ 。此外，因为粒子大小是有限的， $S(x)$ 函数应该具有紧支集，即 $S(x) \neq 0$ 的 x 范围应该足够小。为了避免出现可疑的各向异性问题，要求 $S(-x) = S(x)$ 。

满足这些要求的函数非常多，最简单的选择是利用多项式形式的样条插值。头几阶样条插值函数是

$$S_1(x) = \begin{cases} 1, & |x| < \frac{\Delta x}{2} \\ 0, & |x| > \frac{\Delta x}{2} \end{cases} \quad (2-2)$$

$$S_2(x) = \begin{cases} 1 - \frac{|x|}{\Delta x}, & |x| < \Delta x \\ 0, & |x| > \Delta x \end{cases} \quad (2-3)$$

$$S_3(x) = \begin{cases} \frac{1}{\Delta x} \left[\frac{3}{4} - \left(\frac{x}{\Delta x} \right)^2 \right], & |x| < \frac{\Delta x}{2} \\ \frac{1}{2\Delta x} \left[\frac{3}{2} - \frac{|x|}{\Delta x} \right]^2, & \frac{\Delta x}{2} < |x| < \frac{3\Delta x}{2} \\ 0, & |x| > \frac{3\Delta x}{2} \end{cases} \quad (2-4)$$

使用 S_1, S_2, S_3 的插值分别称为NGP, PIC(CIC)和QS插值方法。从数学上看，不同的 $S_n(x)$ 之间差别在于其光滑性不同： k 空间的大 k 值处 $S(k)$ 衰减速度不同。

除了在累积电荷过程中需要进行点变量到网格变量的插值之外，在计算宏粒子所受的作用力的时候也需要进行插值，其计算过程与上述插值几乎完全相

同，即

$$F(x_p) = \sum_g F_g S(x_p - x_g) \quad (2-5)$$

由于PIC方法模拟的是粒子的集体行为，因此使用不同阶插值在原理上并没有实质的区别，它们的差别主要来自计算误差：首先，使用NGP方法的时候，宏粒子在运动中会导致网格量的跃变，因此会导致很大的随机噪声。对于CIC和QS插值，虽然看起来格点量变化是连续的，但因为网格具有周期 Δx ，所以网格量的计算中存在“假名”(aliasing)问题^{[1][6]}：空间波数为 k 和 $k - mk_g$, $m = 1, 2, 3, \dots$, $k_g = \frac{2\pi}{\Delta x}$ 的模式在这样一个网格上是不可区分的。在计算粒子所受的作用力的时候，需要电场进行插值，得到的 $F(k) = qE(k)S(-k)$ 。于是，在计算中总是存在由于S的高波数模导致的干扰。而这种干扰的大小，基本决定于 $S(k)$ 在大波数处的衰减速度，即在上述 $S_n(x)$ 中 n 的阶次。

为了降低这种干扰，应该选取较大的 n 值。另一方面， n 越大，意味着需要越复杂的插值运算，通常很少有人使用QS或更高阶的插值。另一方面，NGP的噪声太大，所以，绝大部分的模拟代码都选择PIC(CIC)插值方法。

2.2 空间网格宽度：动量守恒和能量守恒

在稠密等离子体的情况下，由于假名问题的影响，在等离子体的Langmuir振荡过程中，相差若干个 k_g 的模式会耦合起来，导致非物理的数值振荡。通常情况下，粒子具有热运动，在分布接近Maxwellian的情况下，波长短于德拜长度的等离子体振荡会被阻尼，因此上述干扰并不是很大的问题。但如果网格的宽度设置得很宽（远远大于德拜长度），于是 $k \pm mk_g$ 的假名模式中存在波长明显大于德拜长度的模式，热运动不足以对这种模式提供足够的阻尼，那么数值振荡就可能发展起来，导致所谓精细网格不稳定性“Fine Grid instability”或者“自加热”现象。对这一过程的分析表明，要避免这种现象，网格宽度限制必须满足^{[1][6]}

$$\Delta x < \zeta \lambda_D \quad (2-6)$$

其中 ζ 是依赖于具体插值函数的一个数，其大小在1左右的量级（对于标准CIC插值， $\zeta \simeq 3$ ）。因为德拜长度 $\lambda_D \propto \sqrt{T/n}$ ，所以对于越稠密和越冷的等离子体，这

个限制影响越大。实际模拟表明，当 $\Delta x \sim 10\lambda_D$ 的时候，自加热导致的不稳定性增长率大概是在 $0.02\omega_p$ 的水准。当 Δx 增大的时候，增长率随之增加。如果模拟的时间很短（几个到几十个等离子体周期），通常可不考虑这种不稳定性。对于较长时间的模拟，则需要具体分析。对于许多束流问题，这个限制基本无需考虑。但在固体密度甚至更高程度（如ICF）等离子体中，这个限制意味着在模拟中使用完全不现实地多的网格数目，这将导致PIC方法基本没有应用的价值。对于某些问题，虽然等离子体密度很低，但问题的时空尺度太大，这个限制也会严重影响PIC方法的实用性。例如，对于典型的CCP放电过程，德拜长度大约是 0.1mm ，而整个设备尺寸尺度大约在几十厘米的水平，整个模拟需要执行上百万步到上千万步，这也是难以完成的。更加极端的例子出现在电离层等空间物理问题之中，这时候，整个问题的尺寸达到数千公里的程度。

从前面的分析看到加热是由于假名效应引起的，因此可以使用高阶的 $S_n(x)$ 来抑制假名耦合。Abe等人测试了 $S_6(x)$ 的性能^[26]，结果表明，在 $\Delta x = 100\lambda_d$ 的条件下， $\omega_p t = 500$ 的时间内，不稳定性几乎观察不到。

尽管如此，仅仅修改 $S(x)$ 的方法只能推迟自加热的的时间而不能根本消除自加热问题。Sentoku的结果^[18]表明，即使使用高阶插值方法，对于ICF这类问题，仍然需要不切实际的高密度网格。此外，使用高阶的插值函数会降低PIC代码的性能，对于每网格粒子数很多的问题尤其如此。

除了提高插值方法的阶次，还可以采用人为的方法抑制假名耦合，即在电荷分配和电场力计算中使用不同阶的Weighting函数 $S(x)$ 。这就是所谓的“能量守恒”格式^{[1][6][27]}：在计算网格电荷密度的时候使用 $S_m(x)$ 插值，而在计算电场的时候使用 $S_{m-1}(x)$ 插值：

$$\rho(x_g) = \sum_p q_p S_m(x_g - x_p) \quad (2-7)$$

$$F(x_p) = \sum_g F_q S_{m-1}(x_p - x_{g+1/2}) \quad (2-8)$$

其中场的插值格点位于电荷密度插值网格的边界上，在均匀网格就是放在半点。这时，在 $\Delta t \rightarrow 0$ 的极限下，系统的能量守恒。通常情况下，这一方法是不切实际的，因为 $m = 2$ 意味着电场力要使用NGP方式插值，这带来严重的随机噪声问题。但如果使用 $m = 3$ 或更高阶的插值，那么可以得到很好的效果。Pointon的

工作^[28]表明,这一方法在 $\Delta x \sim 10^3 \lambda_D$ 的时候,即使执行数万个等离子体周期,仍然表现出很好的稳定性和能量守恒性质。对于微波放电等离子体设备,这可能是解决自加热问题的一个有效途径。

尽管如此,能量守恒格式也存在一些问题。首先,使用不同阶的电荷分配和电场力插值意味着系统的动量不守恒,或者存在“自力”效应^[1],这对粒子的相空间分布有一定影响;自力效应还会影响粒子轨道的细致结构,例如,在PIC/NGP的情况,自力会导致粒子以频率 $\omega = \sqrt{\frac{e^2}{\epsilon_0 m \Delta x^2}}$ 振荡^[29]。在存在鞘层随机加热(例如容性耦合放电)的情况,这种自振荡会影响粒子在鞘层附近的轨道,可能会对加热效果产生影响。其次,这种格式的能量守恒只是在静止等离子体和 $\Delta t \rightarrow 0$ 的条件下得出的。当等离子体整体漂移速度和热运动速度相当的时候,能量守恒格式有可能导致数值不稳定性。第三个问题是,假名耦合中决定性的因素是电荷密度和电磁场的插值。对于电磁问题的模拟(见关于电磁场模型的部分),被分配到格点上的量是电流密度而非电荷密度,因此需要根据电荷的插值函数正确设计高级电流插值函数^[28]。

绕过自加热和德拜长度限制的最后的也是最有效的手段是所谓“隐式”方法,由于这种方法涉及到总体框架的修改,我们在尺度分析的部分再介绍。

2.3 变宽度网格和可变权重粒子

在某些情况下,需要使用变宽度的网格。一个典型的例子是激波问题,为了获得间断面附近行为的良好描述,在激波面附近必须加密网格^[30]。或者虽然网格宽度不变,网格单元的体积却可以变化,极坐标系中的模拟就属于这种情况^[31]。

在不均匀网格的情况下,由于每个网格的大小不同,上述插值算法中必须包含体积因子,以便使得均匀的宏粒子数密度分布产生出均匀的电荷密度。一维下,如果宏粒子密度函数为 $f(x)$,于是插值后的密度为

$$n_j = \frac{\int_{-\infty}^{\infty} f(x)S(x)dx}{\int_{-\infty}^{\infty} S(x)dx} \quad (2-9)$$

代入 $f(x) = \sum_p q_p \delta(x - x_p)$ 后,分子就变成前面的插值公式(2-1),而分母称为体积因子。

对于曲线坐标系，上述积分需要转换为体积积分。在球坐标系和圆柱坐标系的情况，结果已经由Verboncoeur算出^[32]。近来，这结果被推广到一般的非结构化网格情况^[33]。

由于网格单元的体积发生了变化，如果使用相同权重的宏粒子，在小体积的网格单元内的粒子数会变得很少，导致极大的噪声；同时大体积的网格单元内的粒子数又会变得很多，导致总的计算量增大。解决方法是采用可变权重的模拟粒子^{[31][34]}，即每个宏粒子代表真实粒子的数目（权重）是不同的，在网格体积较大的部分使用大权重的粒子，在网格变密的地方使用小权重粒子，兼顾分辨率和总粒子数两方面的要求。

这种方法对流注等问题非常有效^[34]，但当不同体积的网格单元之间存在较大的整体漂移率的时候会出现粒子交换的问题：随着粒子的漂移，平均权重会趋向于常数，最后导致方法失效。此外，大权重粒子进入密网格后，会导致很强的噪声，对于激波问题尤其如此。

解决问题的思路有两种，一种是使用基于局部网格定义的权重^[35]，即粒子权重按照网格的体积定义，当粒子在网格间漂移的时候，权重会不断变化。为了保持连续性方程成立，当一个粒子由体积为 ΔV_1 的网格进入体积为 ΔV_2 的网格的时候，假如 $\Delta V_1 > \Delta V_2$ ，那么只有 $P = \frac{\Delta V_2}{\Delta V_1}$ 的几率粒子会顺利进入，另外 $1-P$ 几率粒子会被简单地抛弃。这一方法在某些情况下是有效的，但在高维条件下容易导致流密度的严重误差。此外，由于这个过程电荷不守恒，会由于随机扰动导致长程静电振荡模式的激发，只有每个网格内的粒子数非常多(> 400)的时候才能给出比较合理的结果。

另外一种方式是使用粒子的分裂和合并^[36]：每隔若干模拟步，对所有网格进行一次“rezone”处理：对于处在小体积网格内的大权重粒子，将这个粒子分裂成若干个小权重粒子；而对处在大体积网格内的小权重粒子，将同一网格内的几个粒子合并成一个。

为了避免过度扰动粒子的分布函数 $f(x, v, t)$ ，分裂和合并都必须考虑守恒量问题。分裂是比较容易处理的，只要分裂后的粒子和原来的粒子的速度相同，分裂中保持质量和电荷守恒，分裂后的粒子位置可以和原来粒子相同或者彼此分开一段距离，正确设置分离距离可以使这样分裂后的粒子分布函数不变。粒子的合并要复杂得多，因为最简单的两粒子合并成一个的算法，不可能保证这

个合并过程中动量和动能都守恒。通常为了降低这个问题的影响，可以选择总是合并动量比较接近的两个粒子，然后取合并后粒子的动量等于合并之前的总动量。此外，还有一个解决方法是每次从网格中提取三个粒子进行合并，而合并结果为两个粒子，调节合并后两个粒子的质量和电荷，可以使得电荷-动量-能量守恒同时成立，只要求解方程

$$\begin{aligned}
 m_1 + m_2 + m_3 &= m_A + m_B \\
 m_1 v_{1\alpha} + m_2 v_{2\alpha} + m_3 v_{3\alpha} &= m_A v_{A\alpha} + m_B v_{B\alpha} \\
 m_1 v_{1\alpha}^2 + m_2 v_{2\alpha}^2 + m_3 v_{3\alpha}^2 &= m_A v_{A\alpha}^2 + m_B v_{B\alpha}^2
 \end{aligned} \tag{2-10}$$

其中 $\alpha = x, y, z$ 代表三个方向。但Lapenta的测试表明^[36]，这一方法因为影响速度分布函数的全局行为，可能产生比两粒子合并方法更不可靠的结果。

3 场建模及其求解

3.1 电磁场模型的分类

通常情况下，电磁场模型决定了PIC方法的基本框架。在PIC方法中，可以严格求解格点上的电磁场，也可以通过模型选择一个近似的电磁场表达式。比较常见的电磁场模型有下面几种：

(1)全电磁场模型，即在计算电磁场中使用完整的麦克斯韦方程组（或者达兰贝尔方程式，依赖于使用的是 \mathbf{E}, \mathbf{B} 表述还是 \mathbf{A}, ϕ 表述）。这个模型考虑了所有可能的电磁现象，但同时也意味着在求解中需要最多的时间和空间开销（见后面的分析）。

(2)Darwin模型：当电磁场的频率不是很高的时候，某些情况下可以完全忽略辐射效应，也就是在麦克斯韦方程组中忽略位移电流项，这样，方程式变成

$$\begin{aligned}\nabla \times \vec{B} &= \mathbf{J} \\ \nabla \times \vec{E} &= -\frac{\partial \mathbf{B}}{\partial t} \\ \nabla \cdot \vec{B} &= 0 \\ \nabla \cdot \vec{D} &= \rho\end{aligned}$$

这个模型在某些天体物理问题中应用很广泛。

(3)静电模型。假设电磁场的演化速度远远低于等离子体特征时间（ $1/\omega_p$ 以及 L/v_{max} ），且自生磁场可以忽略，那么电磁场可以看成是一个自生的静电场和一个外磁场的叠加。于是在计算电磁场的时候，只需要求解静电部分

$$\nabla^2 \phi = -\rho/\epsilon_0$$

对于射频放电等离子体，这个模型最为常用。

具体选择哪种模型需要根据物理问题的性质而定。但从应用角度来看，上述三种场模型之间最大的差别，在于计算上的开销：全电磁模型可以用电磁场的时域有限差分方法(FDTD)简单地建模，但因为FDTD方法的CFL限制问题，需要非常短的时间步长来消除不稳定性。而后两种模型，特别是静电模型，不

受电磁波CFL条件的限制，但一方面后两种模型物理上忽略了电磁辐射，因此对于高频电磁波主导的问题没有意义；另一方面，在后两种模型中往往必须处理二阶椭圆型微分方程（例如泊松方程或者亥姆霍兹方程），其差分法构造和求解都要比标准的FDTD复杂得多。

3.2 电磁模型：Maxwell方程的求解

对于不同的模型，需要求解不同的电磁场方程。在最一般的电磁模型中，需要求解的是麦克斯韦方程组。假设粒子的电荷和电流已经被正确地分配到计算网格的各点上：

$$\rho = \rho_i, \vec{J} = \vec{J}_i, i = 1, \dots, N$$

接下来的问题是解出电磁场。通常情况下，这个问题可以用Yee的电磁场差分方法完成^[37]，即离散麦克斯韦方程组的两个旋度方程：

$$\begin{aligned} \frac{1}{c^2} \frac{\partial \vec{E}}{\partial t} &= \nabla \times \vec{B} - \mu_0 \vec{J} \\ \frac{\partial \vec{B}}{\partial t} &= -\nabla \times \vec{E} \end{aligned}$$

这一方程可以用简单的时间中心差分格式来差分，即将电场定义在整数时间步 t^n 上而将磁场定义在半整数时间步 $t^{n+1/2}$ 上，从而上述方程就变为一组蛙跳格式的时间差分方程。

对于空间导数的离散化，Yee格点的定义方式，是将电磁场交错排布，即将电磁场分量定义在下述格点上：

$$\begin{aligned} E_x &= e_x(i + 0.5, j, k) \\ E_y &= e_y(i, j + 0.5, k) \\ E_z &= e_z(i, j, k + 0.5) \\ B_x &= b_x(i, j + 0.5, k + 0.5) \\ B_y &= b_y(i + 0.5, j, k + 0.5) \\ B_z &= b_z(i + 0.5, j + 0.5, k) \end{aligned}$$

(3-1)

这样，电磁场的差分方程就可以写成

$$\begin{aligned}
 & \frac{b_z^{n+1/2}(i+0.5, j+0.5, k) - b_z^{n-1/2}(i+0.5, j+0.5, k)}{\Delta t} \\
 = & \left[\frac{(e_x^n(i+0.5, j+1, k) - e_x^n(i+0.5, j, k))}{\Delta y} \right. \\
 & \left. - \frac{(e_y^n(i+1, j+0.5, k) - e_y^n(i, j+0.5, k))}{\Delta x} \right] \\
 & \frac{1}{c^2} \frac{e_z^{n+1}(i, j, k+0.5) - e_z^n(i, j, k+0.5)}{\Delta t} \\
 = & \left[\frac{(b_y^{n+1/2}(i+0.5, j, k+0.5) - b_x^{n+1/2}(i-0.5, j, k+0.5))}{\Delta x} \right. \\
 & \left. - \frac{(b_x^{n+1/2}(i, j+0.5, k+0.5) - b_x^{n+1/2}(i, j-0.5, k+0.5))}{\Delta y} \right] \\
 & - \mu_0 j_z^{n+1/2}(i, j, k+0.5) \\
 & \dots\dots
 \end{aligned} \tag{3-2}$$

上述方程是六个交错的耦合差分方程，在典型的PIC方法中，这组方程被用来在时间域推进电磁场^[38]：利用整点时刻的**B**和电流密度**J**求出半点时刻的电场强度**E**，或者用半点时刻的**E**求出下一个整点时刻的磁场**B**。由于这样求出的电磁场在空间上总是定义在“半”网格点上，因此应用于粒子推进之前还需要再插值到正确的网格点以消除自力效应。

关于这个算法有一系列众所周知的结论，其中之一是关于电荷守恒的：上述计算只使用了麦克斯韦方程组的两个旋度方程，计算结果并不能保证散度方程的成立性。只有当计算中始终保持电荷守恒方程成立时，才能避免出现一些可疑的解。但格点上的电流密度**J**^{n+1/2}来自乘积**ρv**的插值，或者说利用半整数时间点时粒子的位置**x_p^{n+1/2}**将每个粒子的速度**v_p**累积到格点上。如果对速度累积使用简单的**S_m(x) Weighting**方法，累积出的电流密度并不能保证电荷守恒方程在格点上严格成立。解决这个问题基本方法是在每步求解电场之后，减去多余的纵向电场，即使用一个电场修正^[39]：

$$\nabla^2 \Psi = \frac{\rho}{\epsilon_0} - \nabla \cdot \mathbf{E}_{Yee} \tag{3-3}$$

$$\mathbf{E} = \mathbf{E}_{Yee} + \nabla \Psi \tag{3-4}$$

由于电场修正需要存储粒子的电荷密度并且求解一个Poisson方程，后者是

个二阶椭圆型方程，一般来说需要运用迭代法求解，比上述简单的直接推进慢得多并且难于并行化，所以一般都考虑使用变通的方法，即通过保持电荷守恒来消除上述要求。这就要求在计算粒子的电流密度的时候，必须使得累积的电荷和电流密度满足连续性方程。于是，对于网格点的电流累积，必须使用和电荷累积不同的方式完成，这个方法称为“电荷守恒分配”。最基本的电荷守恒分配方案由Villasenor和Buneman提出^[40]，基本思路是假设粒子在 Δt 时刻内进行直线运动，然后顺着粒子运动的连线利用有限尺寸的电荷云来估计通过网格界面的流量 $\mathbf{j}\Delta t$ 。Eastwood提出了另外一个思路^[41]，即根据粒子运动的起点和终点来计算电流流量，如果粒子跑动超过一个网格，则把整个运动分解成一系列网格内的运动。由于粒子位置是定义在整数时间点上，因此这两种方法算出来的电流流量自然位于半整数时间点上，无需再额外求 $x^{n+1/2}$ 。

利用这类电流分配方法，可以避免求解Poisson方程。但同时Weighting过程变得很复杂，并且包含大量的IF THEN ELSE判断分支。此外，Villasenor的原始方法只对简单的线性Weighting有效。因此又出现了各种优化方法，例如Esirkepov的密度分解法^[42]和Umeda的“zigzag line”方法^[43]等。Eastwood的流量算法也被推广到高阶插值模式^[28]

对于非结构网格的情况，电荷守恒流分配变得非常困难。这种情况下可以使用辅助的场演化方程^[44]。考虑任意线性微分算符 \hat{D} 和辅助标量场 Ψ ，写出下述方程组

$$\begin{aligned}\frac{\partial \mathbf{E}}{\partial t} &= c^2 \nabla \times \mathbf{B} - c^2 \nabla \Psi - \frac{\mathbf{j}}{\epsilon_0} \\ \frac{\partial \mathbf{B}}{\partial t} &= -\nabla \times \mathbf{E} \\ \hat{D}\Psi + \nabla \cdot \mathbf{E} &= \frac{\rho}{\epsilon_0} \\ \nabla \cdot \mathbf{B} &= 0\end{aligned}$$

可以证明这组方程在合适的初边值条件下具有唯一解。此外，对于正确的初始条件（满足散度方程）有 $\hat{D}\Psi = 0$ 。不难看出，这里的 \hat{D} 就是散度方程的误差。

进一步地，容易得出

$$\frac{\partial \hat{D}\Psi}{\partial t} - c^2 \nabla^2 \Psi = \frac{1}{\epsilon_0} \left(\frac{\partial \rho}{\partial t} + \nabla \cdot \mathbf{j} \right) \quad (3-5)$$

对这个方程的分析表明对适当选择的 \hat{D} ， $\hat{D}\Phi$ 不会随时间增加，即如果给出正确的初始条件并按照上述方程(3-5)演化电磁场，则以后散度方程可以近似保持正确。这种方法实际是一种拉格朗日乘法。

选择不同的算符 \hat{D} 得到各种不同的方法。例如如果取 $\hat{D}\Psi \equiv 0$ ，就退回到前面的Poisson方程修正方法。一个比较常用的方法是取 $\hat{D}\Psi = \frac{\Psi}{\chi}$ ，于是关于 \mathbf{E} 的推进方程变为

$$\frac{\partial \mathbf{E}}{\partial t} = c^2 \nabla \times \mathbf{B} - \chi c^2 \nabla \left(\frac{\rho}{\epsilon_0} - \nabla \cdot \mathbf{E} \right) - \frac{\mathbf{j}}{\epsilon_0} \quad (3-6)$$

这方程可以用简单的时间分裂近似求解，也就是

$$\bar{\mathbf{E}}^{n+1} = \mathbf{E}^n + \Delta t \left(c^2 \mathbf{B}^{n+1/2} - \frac{\mathbf{j}^{n+1/2}}{\epsilon_0} \right) \quad (3-7)$$

$$\mathbf{E}^{n+1} = \bar{\mathbf{E}}^{n+1} + \chi c^2 \nabla \left(\frac{\rho^n}{\epsilon_0} - \nabla \cdot \bar{\mathbf{E}}^{n+1} \right) \quad (3-8)$$

上述方程是个抛物型方程，正规求解法应该使用隐格式，显式时间分裂近似虽然简单但存在误差问题。也可以使得演化方程(3-5)成为纯双曲型方程，即取 $\hat{D}\Psi = \frac{1}{\chi^2} \frac{\partial \Psi}{\partial t}$ ，这时可以使用标准的双曲型方程显式解法进行处理，具体方法可以参考文献^{[44][45]}。

无论如何，应该指出的是直接Poisson校正是准确的，而抛物-双曲方法在 $\chi \rightarrow \infty$ 的时候退回到Poisson方法。问题在于这种条件下显式方法不再适用，隐格式的解法就回到求解Poisson方程。所以抛物-双曲方法一般来说都要在给出准确的结果（大 χ ）和便于求解（小 χ ）之间进行均衡。

Yee方法求解Maxwell方程的另外一个熟知的限制是数值上的，也就是CFL条件：

$$c\Delta t/\Delta x < 1$$

这个限制意味着，在上述计算中，时间分辨率必须足以追踪以光速传递的信号。在很多情况下，这是个过于严厉的限制。例如在射频腔放电的模拟中，决定性的模式是低频模，其频率不超过1GHZ；但因为PIC方法的空间分辨率问题，通常需要取空间网格宽度在零点一毫米的水平；如果要使用上述FDTD方法处理

电磁场，那么时间步长将短到 10^{-13} s，这是缺乏实践意义的。这个问题并不仅限于PIC方法，实际是计算电磁学FDTD方法固有的问题。在计算电磁学中，可以使用交替方向隐式方法来绕过这个困难^{[46][47]}，其基本思路是将电磁场定义在同样的时间点（即：在半整数和整数时间步上都定义E和B）。然后，注意到在 t^n 到 t^{n+1} 步的推进中，右端项必然总是两个空间导数的差，例如

$$\frac{\partial E_x}{\partial t} = \left(\frac{\partial H_z}{\partial y} - \frac{\partial H_y}{\partial z} \right) - J_x$$

把这个推进拆成两步，即

$$\frac{E_x^{n+1/2} - E_x^n}{\Delta t/2} = \left(\frac{\Delta H_z^{n+1/2}}{\Delta y} - \frac{\Delta H_y^n}{\Delta z} \right) - J_x^{n+1/2}$$

和

$$\frac{E_x^{n+1} - E_x^{n+1/2}}{\Delta t/2} = \left(\frac{\Delta H_z^{n+1/2}}{\Delta y} - \frac{\Delta H_y^{n+1}}{\Delta z} \right) - J_x^{n+1/2}$$

同样拆开相应于 E_z 和 E_y 的方程，联立后，关于 H_x 的方程化为一组三对角方程，可以用简单的追赶法解出。

这套方法在计算电磁学中是熟知的，在处理低频电磁波占据主导地位的等离子体问题中，可以在PIC模拟中引入这个方法。

FDTD方法（包括ADI-FDTD）的第三个问题是所谓数值色散效应。因为空间网格效应的存在，有限差分方法的波速并不等于解析上求出的波速。对于某个特定波长的行波， $k\Delta x$ 越大，数值波速和真实波速的差距越大。对于标准的Yee网格显式FDTD，数值色散关系是^[48]

$$\frac{1}{c^2} \sin^2 \frac{\omega \Delta t}{2} = \left(\frac{\Delta t}{\Delta x} \right)^2 \sin^2 \frac{k_x \Delta x}{2} + \left(\frac{\Delta t}{\Delta y} \right)^2 \sin^2 \frac{k_y \Delta y}{2} + \left(\frac{\Delta t}{\Delta z} \right)^2 \sin^2 \frac{k_z \Delta z}{2}$$

对于ADI-FDTD，也有类似的结果。通常情况下 $\Omega \Delta t$ 和 $k\Delta x$ 都不大，因此数值色散误差并不太大，对于一般的问题往往可以忽略。但如果研究的是激光尾场加速（LWFA）之类问题，激光需要传播很长的距离，同时束流电子的速度会非常接近光速。这时候，电磁波速度稍有一点误差就会带来束流电子能量的很大误差（因为束流电子能量的上限是由激光群速度决定的）；此外，一旦电子速度超过数值光速，还会引发虚假的切伦可夫辐射^[48]。这种情况下，就需要用专门的高阶差分格式来压低数值色散^{[49][50][51]} 或者使用滤波。

3.3 静电模型:Poisson方程

当辐射和电磁感应都可以忽略的时候，电磁场简化为静电问题，也就是需要求解泊松方程

$$\nabla^2 \phi = -\frac{\rho}{\epsilon_0}$$

此外，对于Darwin近似，电磁场求解也常常归结为求解Poisson方程。

尽管Poisson方程的求解是偏微分方程数值方法中研究得最透彻的部分，但应用中往往仍然不能令人满意。这主要是Poisson方程的全域性质导致的：不可能把泊松方程的求解器写成类似前述Maxwell方程那样，一点的场量仅决定于附近点的场量。相反，任何解法都需要使用全部求解域内的源函数。特别是，要依赖于边界点的场量，无论边界点有多远。结果是求解一次Poisson方程的计算量总是远远大于Maxwell方程的一步推进所需要的计算量。对于N个网格的系统，几乎也总是大于推进N个宏粒子一步所需要的计算量（依赖于具体解法，可以在一个到三个数量级的差距）。以至于涉及到静电场求解的问题，如果不小心地处理Poisson求解器，场方程的求解常常会压倒粒子推进变成模拟速度的主要限制。

如同熟知的那样，Poisson问题的第一个困难来自唯一性定理：给定了正确的边界条件之后，电势才有唯一解。而通常的“开放边界条件”在数值上是难以给定的^{[1][52]}。相对来说，标准的Dirichlet或者Neumann边界条件比较容易实现也容易求解，但却未必对应正确的物理问题。

在均匀网格构造的规则区域上，给定Dirichlet或者Neumann边界条件的Poisson方程可以用相当快的方法解出，即所谓“快速泊松求解器”^{[53][54][55][56]}。这种求解器基于傅里叶分解法：导数算子 $\hat{L}_x = \frac{\partial^2}{\partial x^2}$ 的离散形式可以表达为 $(\hat{L}_x u)_n = \frac{1}{h^2}(u_{n-1} - 2u_n + u_{n+1}) = \frac{1}{h^2}\{1 - 2 \mathbf{1}\}$ ，依赖于边界条件，可以写出它的本征矢量是 $u^k = \{\sin \frac{k\pi}{N+1}, \sin \frac{2k\pi}{N+1}, \dots, \sin \frac{Nk\pi}{N+1}\}$ 或者对应的余弦形式，本征值 $\lambda_k = 2(\cos \frac{k\pi}{N+1} - 1)$ 。于是将待求的解分解成这些本征矢量的叠加， \hat{L}_x^2 就变成了一个简单的乘法算符，不同的本征矢之间被脱耦。消去两个方向的 \hat{L}^2 算符后，第三个方向的离散方程转化为标准的三对角方程，可以用追赶法解出。在求解中需要把源项 ρ 分解为上述 u^k 的叠加，求解后还需要把得到的结果转换回坐标空间表示，通常这需要进行矩阵乘法，但由于上述计算中的 u^k 是标准的三角级数，

因此这两次乘法可以用快速傅里叶变换法(FFT)完成, 于是总的方程求解可以在 $O(N^d \log N)$ 计算量下完成, d 是空间维度。

我们可以用圆柱形区域上的2维Poisson方程来说明这一解法^[57]。考虑一个旋转对称的封闭圆柱形空腔, 在极坐标 r - z 系统中, 有限体积离散后的Poisson方程是

$$\begin{aligned}
 -\rho_{j,k} &= \frac{2r_{j+1/2}}{(\Delta r^2)_j \Delta r_{j+1/2}} (\phi_{j+1,k} - \phi_{j,k}) - \frac{2r_{j-1/2}}{(\Delta r^2)_j \Delta r_{j-1/2}} (\phi_{j-1,k} - \phi_{j,k}) \\
 &\quad + \frac{1}{\Delta z^2} (\phi_{j,k+1} - 2\phi_{j,k} + \phi_{j,k-1}) \\
 -\rho_{0,k} &= \frac{2r_{1/2}}{\Delta r^2_0} \frac{\phi_{1,k} - \phi_{0,k}}{\delta r_{1/2}} + \frac{\phi_{0,k+1} - 2\phi_{0,k} + \phi_{0,k-1}}{\Delta z^2}
 \end{aligned} \tag{3-9}$$

可以将其简写为

$$\begin{aligned}
 -a_j X_{j-1} + b_j X_j - c_j X_{j+1} + A X_j &= r_j \\
 X_j &= \{\phi_{j1}, \phi_{j2}, \dots, \phi_{jm}\}^T \\
 r_j &= -\Delta z^2 \{\rho_{j1}, \rho_{j2}, \dots, \rho_{jm}\}^T
 \end{aligned} \tag{3-10}$$

z 方向的差分是标准的 $\frac{1}{\Delta z^2} \{1 \ -2 \ 1\}$ 差分算子, 其矩阵表示为

$$A = \begin{pmatrix} 2 & -1 & 0 & \dots & \dots & \dots \\ -1 & 2 & -1 & 0 & \dots & \dots \\ 0 & -1 & 2 & -1 & 0 & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & 0 & -1 & 2 & -1 \\ \dots & \dots & \dots & 0 & -1 & 2 \end{pmatrix}$$

于是本征值和本征矢量是

$$\begin{aligned}
 A z_i &= \lambda_i z_i \\
 z_i &= \sqrt{\frac{2}{N+1}} \left\{ \sin \frac{i\pi}{N+1}, \sin \frac{2i\pi}{N+1}, \dots, \sin \frac{Ni\pi}{N+1} \right\} \\
 \lambda_i &= 2 - 2 \cos \frac{i\pi}{N+1}
 \end{aligned} \tag{3-11}$$

这样，矩阵A可以对角化，也就是

$$\begin{aligned} QAQ^T &= \Lambda \\ \Lambda &= \text{diag}\{\lambda_i\} \\ Q &= \{z_i\} = Q^T \end{aligned} \quad (3-12)$$

在差分方程两边乘以Q，得到一组三对角方程：

$$\begin{aligned} -a_j y_{i,j-1} + b_j x_{ij} - c_j y_{i,j+1} + \lambda_i y_{ij} &= s_{ij} \\ y_{ij} &= (Y_j)_i, \quad s_{ij} = (S_j)_i \\ Y_j &= QX_j, \quad S_j = QR_j \end{aligned}$$

这组三对角方程可以针对不同的i分别解出，得到Y_j后，乘以Q⁻¹就得到所求的解X_j，而用Q和Q⁻¹相乘正好是傅里叶正弦变换。因此整个求解过程包括两次傅里叶变换和一次三对角方程求解。

这种傅里叶展开法可以用于直角坐标或者柱坐标。在目标为球状的情况下，使用球坐标，Poisson方程变为^[58]

$$\frac{\partial^2 \Phi}{\partial r^2} + \frac{2}{r} \frac{\partial \Phi}{\partial r} + \frac{1}{r^2} \frac{\partial^2 \Phi}{\partial \theta^2} + \frac{\cot \theta}{r^2} \frac{\partial \Phi}{\partial \theta} + \frac{1}{r^2 \sin^2 \theta} \frac{\partial^2 \Phi}{\partial \phi^2} = -\frac{\rho}{\epsilon_0} \quad (3-13)$$

首先将Φ展开为

$$\Phi(r, \theta, \phi) = \sum \Phi_n(r, \theta) \exp\{in\phi\}$$

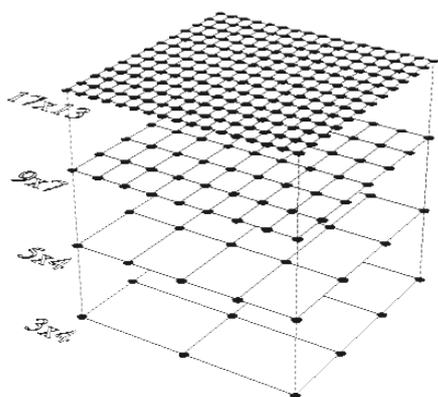
然后对θ和r用有限差分方法建立差分方程，得到的方程和上述方程很相似：

$$\begin{aligned} \frac{\Phi_{i+1,j} - 2\Phi_{ij} + \Phi_{i-1,j}}{\Delta r^2} + \frac{2}{r_i} \frac{\Phi_{i+1,j} - \Phi_{i-1,j}}{2\Delta r} + \frac{1}{r_i^2} \frac{\Phi_{i,j+1} - 2\Phi_{i,j} + \Phi_{i,j-1}}{\Delta \theta^2} \\ + \frac{\cot \phi}{r_i^2} \frac{\Phi_{i,j+1} - \Phi_{i,j-1}}{2\Delta \theta} - \frac{n^2}{r_i^2 \sin^2 \phi_j} \Phi_{ij} = -\frac{1}{\epsilon_0} \rho_{ij} \end{aligned} \quad (3-14)$$

但中央矩阵不是标准的{1 - 2 1}形式。这方程可以用循环消去法解出^[59]。

上述两种方法只能用于规则区域。对于简单的规则区域的组合（或者被挖去一块的情况），可以用电容矩阵法（Capacitance Matrix Method）^[60]把问题化归为规则区域上的问题。这时，除了规则区域上的Poisson问题求解，还需要做

图 3.1 Concepts of coarsening



一个电容矩阵的LU分解，矩阵的行数基本等于不规则部分的边界长度。如果这部分边界长度不大，CMM方法是很快的。

如果不规则部分的边界变得很长（很复杂），或者问题性质决定需要使用非均匀网格，那么就只能使用迭代法了。Poisson方程的迭代求解数学上也有很多结果。经典的点迭代法（GS，SOR等等），交替方向隐式（ADI）^{[61][62]}方法以及预处理共轭梯度迭代(PCG)^[63]都曾经广泛地用于泊松方程求解。八十年代后，多重网格方法(Multigrid Method)^{[64][65]}成为主流的求解手段。多重网格方法在不同尺度和分辨率的网格上逐渐进行“松弛”操作，对于标准的泊松问题具有常数收敛率。

多重网格的基本思路来自观察不同分辨率网格上的残差。考虑对一个区域使用不同分辨率的网格进行描述，如图(3.1)所示。对差分方程 $\hat{A}\Phi = f$ 定义残差 $r^n = \hat{A}\Phi^n - f^n$ 。其中 \hat{A} 是差分矩阵， Φ^n 是迭代中的近似解。任何迭代法相当于逐步修改 Φ 使得所有 $r^n \rightarrow 0$ 的过程。传统的迭代法（如Gauss-Seidel）收敛缓慢，原因是在精细网格上执行的迭代往往对长波模式（粗网格）的 r_n 影响甚微（从另一个视点上来说，就是每次迭代的影响距离由网格宽度决定，于是越细的网格，迭代越难影响到较远的网格）。于是，如果能逐层进行迭代，可以让所有不同模式的残差同步收敛到0。具体实现如下：

引入算子 I_k^k ，它将第 k' 层上的网格数据/残差/差分算符通过插值映射到大

小不同的第 k 层，这个过程称为“限制”（对于 $k = k' - 1$ ）或者“延伸”（对于 $k = k' + 1$ ）。需要注意的是对于直角坐标下的Poisson方程，不同网格上的Poisson差分算子是相似的。

再引入任何一种独立使用的时候能够收敛的迭代法，例如Gauss-Seidel迭代或者ADI迭代，定义smooth过程为执行一步迭代的操作，称为“光滑化”过程。那么，多重网格方法可以用算法(1)描述：

Algorithm 1 Multigrid Algorithm

```

1:  $u^L \leftarrow w_0$ 
2:  $A^L \leftarrow A$ 
3: while  $\|Au^L - f\| > \epsilon$  do
4:    $f^L \leftarrow f$ 
5:   for  $l=L-1$  to 2 do
6:      $u^l \leftarrow 0$ 
7:      $f^{l-1} \leftarrow I_l^{l-1}(f^l - A^l u^l)$ 
8:      $A^{l-1} \leftarrow I_l^{l-1} A^l$ 
9:   end for
10:   $u^1 \leftarrow \text{solve}(A^1 u^1 = f^1)$ 
11:  for  $l=2, L$  do
12:     $u^l \leftarrow u^l + I_{l-1}^l(u^l)$ 
13:     $u^l \leftarrow \text{smooth}(u^l, f^l)$ 
14:  end for
15: end while

```

对于标准的笛卡尔坐标网格，每次“松弛”都由一次红-黑高斯-塞德尔迭代构成，其计算量正比于网格点数 N^d ，于是，总的计算量正比于 $N^d \log \epsilon$ ， ϵ 是误差。

对于不均匀网格或者曲线坐标系，需要使用一些变通方法，并选择合适的松弛手段。例如，对于极坐标系，一个有效的变形是使用 z 方向的semi-coarsening，并使用逐行的线光滑化^[66]。semi-coarsening的概念可以参考图(3.2)。适当选择的 semi-coarsening方法使得计算量仍然正比于 $N^d \log \epsilon$ 。

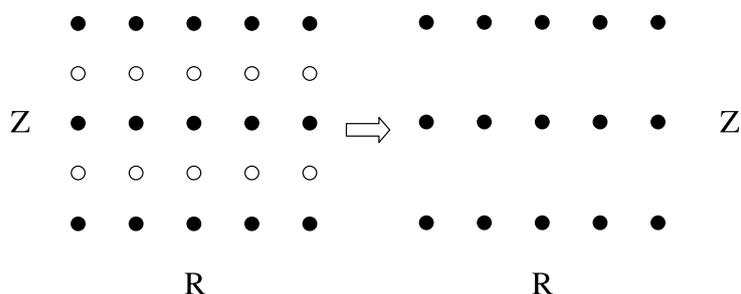


图 3.2 The concepts of semi-coarsening

尽管如此，多重网格需要的迭代次数还很多，对于典型的网格尺度，其速度仍然远低于傅里叶方法。同等优化条件下，使用傅里叶方法， N 格点泊松问题的求解比推动 N 个粒子一步的开销高几倍（ ~ 10 ），而对于多重网格方法，一般要比推动对应数目的粒子慢几十倍。如果由于某种原因使用了共轭梯度或者ADI等方法，其速度比多重网格方法还要再慢一个量级。^①

因此，除非问题中使用了非常多的粒子，否则Poisson方程求解通常都慢于粒子推动步，在网格规模较大的时候，由于并行上的困难，困难常常更加严重。这是静电PIC问题的一个特征。

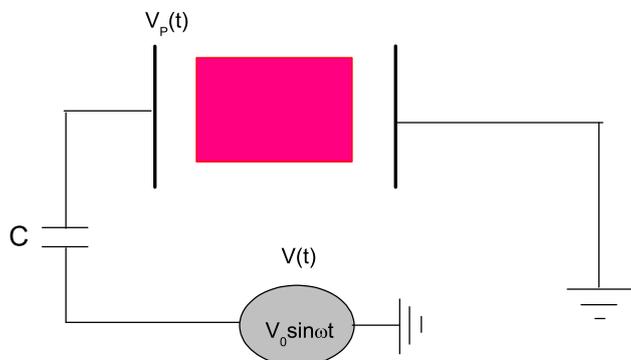
在某些特殊情况下（例如考虑中间介质或者隐格式），Poisson问题中会包含可变的介电常数，也就是

$$\nabla \cdot [(1 + \chi)\nabla\phi] = \rho$$

其中 χ 是极化率张量。这时的泊松方程求解变得更加困难，主要因为当 χ 发生快速变化（甚至间断）的时候多重网格方法的收敛率会受到严重影响。这时需要使用一些复杂的技巧，例如基于间断迦廖尔金方法或者矩阵分解法的插值和限制等等^[67]。

① 我们在1.8GHZ core2系统的一个CPU核上分别使用使用傅里叶本征值方法，多重网格方法和预条件共轭梯度法（使用一个类似多重网格的Precondition）求解1024*1024矩形网格上的标准2维泊松方程，其中预条件共轭梯度代码使用来自Trilinos中的数学库。测得每次求解的平均执行时间分别为 0.20秒，1.77秒和10.3秒。至于使用其他预条件的共轭梯度方法，需要的时间超出了我们的耐心。虽然理论上，当网格尺寸非常大时，多重网格的速度会超过傅里叶方法，但从目前来看，那是下一个计算时代的事情。

图 3.3 The solving of circuit effects



3.4 鞘层自偏压和外电路建模

对于由电极驱动的等离子体，为电极提供电压的外电路也必须被建模。这主要是由于自偏压效应：当电极暴露在等离子体中的时候，电荷会沉积到极板上，于是极板不再是电中性的，沉积电荷改变了极板电压。对于射频或者直流驱动的放电，这个效应会对等离子体的鞘层行为产生很大的影响。

自偏压建模的核心是如何自洽地求出极板的电压。使用静电模型，有几种方法处理自偏压效应，例如人工调节极板电势使得到达电极的正负电荷平均流量相等^[68]；或者在电极处修改Poisson方程的边界条件包含极板电荷密度等等(Verboncoeur)^[69]。后者是包含外电路的放电系统建模的基本方法：将电极分割为若干单元，记录沉积到单元上的电荷量，并对电极单元使用高斯定理，建立包含电极上累积电荷在内的电场方程。最后，和外电路方程耦合求解。

在频率不是很高且极板为金属的情况下，可以利用极板电势处处相等的假设来简化这个方法，这就是Vahedi的场分解方法^[70]。由于简化中只是消除了极板内部耦合的建模，其他方面与Verboncoeur方法没有本质区别，因此这里只简要介绍简化后的方法：

考虑最简单的一维容性耦合等离子体，一个电极接地，另一个电极施加射频电压，如图3.3。在接地部分，电极电势为0；而在驱动电极上使用高斯定理得出

$$\epsilon_0 E_{\Delta x/2} * S_{power} = Q$$

也就是

$$\epsilon_0 \frac{\phi_0 - \phi_1}{\Delta x} S_{power} = Q \quad (3-15)$$

其中电势 ϕ 具有网格值 $\phi_0, \phi_1, \dots, \phi_N$ 。 ϕ 本身满足Poisson方程:

$$\begin{aligned} \nabla^2 \phi &= -\frac{\rho}{\epsilon_0} \\ \phi_0 &= V_P(t) \\ \phi_N &= 0 \end{aligned} \quad (3-16)$$

$V_P(t)$ 是驱动电极的电压。

利用Poisson方程的线性性质，将 ϕ 分解成两个解的线性叠加，即

$$\phi = \phi_P + V_P(t)\phi_L \quad (3-17)$$

其中 ϕ_P 和 ϕ_L 分别是齐次边界条件下的Poisson方程解以及归一化边界条件下Laplace方程的解:

$$\begin{aligned} \nabla^2 P &= \frac{\rho}{\epsilon_0} \\ P(0) &= 0 \\ P(N) &= 0 \end{aligned} \quad (3-18)$$

$$\begin{aligned} \nabla^2 L &= 0 \\ L(0) &= 1 \\ L(N) &= 0 \end{aligned} \quad (3-19)$$

这样，Poisson方程的解中自然包含了 $V_P(t)$ 。接下来只要对外电路方程使用简单的时间差分建模就可以得到决定 $V_P(t)$ 的方程。例如，外电路由电压源 $V(t)$ 和电容器 C 构成，外电路方程是

$$\frac{dQ}{dt} = I(t) + S J_{conv}$$

积分，并且注意到

$$\int I(t)dt = Q_C(t) = CV_C(t)$$

且

$$V_C(t) + V_P(t) = V(t)$$

就得到从 $t-1$ 时刻到 t 时刻的演化方程

$$Q^t = Q^{t-1} + CV(t) - CV_P(t) - Q_C^{t-1} + Q_{conv}^t$$

将前面得到的 Q 值表达式 3-15 和 ϕ 表达式 3-17 代入，得到

$$[\epsilon_0 S_{power} \frac{(1-L_1)}{\Delta z} + C]V_P(t) = Q^{t-1} + \frac{\epsilon_0 S}{\Delta z} P_1 + CV(t) - Q_C^{t-1} + Q_{conv}^t \quad (3-20)$$

这就是确定电极电压和自偏压的方程式。这样就得到模拟流程：

(1) 设置初值。

(2) 在时刻 $t-1$ ，已知极板电压 ϕ_0^{t-1} 和电容上存储的电荷 Q_C^{t-1} ，模拟等离子体运动。

(3) 累积 Δt 时刻内到达极板的离子/电子通量，分别求解 P 和 L 的 Poisson 方程；利用方程 (3-20) 求出 t 时刻的极板电势。求出极板电势 $V_P(t)$ 后，电容上的电压 $V_C(t) = V(t) - V_P(t)$ ，从而得到电容上存储的电荷 Q_C^t 。

(4) 循环。

这套方法可以毫无困难地推广到两维以上的情况。对于极板等势条件不再适用的情况（如极板存在电阻或者驱动频率非常高出现了驻波效应），需要回到标准的 Verboncoeur 电荷累积方法。

3.5 频率分离:有质动力和低频场

针对具体的物理问题。可以构造不同的场模型，即从物理的角度将电磁场分成不同组成部分，对其中行为复杂的部分使用完整的电磁场描述，而对行为比较简单可以做半解析处理的部分使用约化公式。这种方法对于激光等离子体相互作用问题最为常用^{[19][20]}。

考虑激光在稀薄等离子体中传播的问题，等离子体频率远远低于激光频率，可以将系统中的电磁场分成两部分：激光的快速振荡电磁场 \tilde{a}_L 和等离子体的自生电磁场 \tilde{a}_W ：

$$a = \tilde{a}_L + \tilde{a}_W$$

这样，电磁场方程被拆成

$$\left(\Delta - \frac{1}{c^2}\partial_{tt}\right)\tilde{a}_{L,W} = -j_{L,W} + \nabla\partial_t\phi_{L,W} \quad (3-21)$$

对激光场中的电子运动使用冷等离子体和横场近似，并且利用包络近似 $\tilde{a}_L = \frac{a}{2}\exp[i(kx - \omega t)]$ ，可以得出激光的传播方程

$$\left(\Delta_T - \frac{2i\omega}{c^2}\partial_\tau + \frac{2}{c}\partial_{\tau\xi}\right)a = -\mu_0\chi a \quad (3-22)$$

其中 χ 是非线性折射率，而 $\xi = x - ct$ ，于是只需要求解这个包络方程和等离子体慢变电磁场的方程。

直接求解前述的慢变场 \tilde{a}_W 的方程是困难的，这主要是因为因果律问题，在库仑规范下无论 \tilde{a} 还是 ϕ 都不是局域的。在洛伦兹规范下虽然方程形式上是因果律的，但又会出现一些其他的问题（如求解电磁场 \mathbf{E} 和 \mathbf{B} , \mathbf{A} 的不确定性等等）。注意到快变和慢变项在此近似下已经脱耦，慢变场完全由等离子体本身的运动产生，所以可以对慢变场使用 \mathbf{E} , \mathbf{B} 的一阶麦克斯韦形式处理，也就是

$$\begin{aligned} \epsilon_0 \frac{\partial \tilde{\mathbf{E}}}{\partial t} &= (\nabla \times \tilde{\mathbf{H}}) - \tilde{\mathbf{J}} \\ \mu_0 \frac{\partial \tilde{\mathbf{H}}}{\partial t} &= -(\nabla \times \tilde{\mathbf{E}}) \end{aligned} \quad (3-23)$$

这样问题就归化为解一个激光的包络方程和一个麦克斯韦方程。

慢变场麦克斯韦方程的源项是等离子体运动方程中的慢变电流项，激光对等离子体的影响主要是通过颤动项的时间平均。在激光等离子体相互作用理论中知道这个平均的效果是激光对等离子体粒子产生有质动力 $\nabla\gamma \simeq \nabla\frac{|a|^2}{2}$ ，于是得出在慢变部分中等离子体中粒子的运动方程：

$$\partial_t P_i = q_i(E + v_i \times B - \frac{q_i}{4m_i}\nabla|a|^2) \quad (3-24)$$

利用上述三个近似方程，在模拟中可以忽略激光传播中的精细效应，而只分析其包络行为，这个近似称为有质动力导向中心近似。

物理上，这个近似大幅度地降低了研究所需要的空间分辨率，提供了较大尺度模拟的可行性，在涉及到激光加速和尾场生成的问题中，很多模拟都使用这个方法。作为代价的是，电磁波的包络方程(3-22)变得很复杂，因为出现了时间和空间的混合导数项 ∂_{xt}^2 。理论上，混合导数项和尾场激发有关，直接忽略这一项在物理上将影响到激光场-尾场耦合的可靠性。而保留这一项，包络方程不再是标准的抛物型微分方程，给数值处理带来了很大的困难。解决方法有几种，从物理上看，这个方程包括三个物理效应：纵向的传播；横向的衍射效应（ ∇_T^2 ）以及非线性折射率效应。于是，直接的考虑是将横向效应和折射-传播效应分离开，利用分步方法进行有限差分求解^[71]：

$$\left(-\frac{2i\omega}{c^2} + \frac{2}{c}\partial_\xi\right)\frac{a_1^{n+1/2} - a_1^{n-1/2}}{\Delta\tau} = -\frac{1}{2}\mu_0\chi(a_1^{n+1/2} + a_1^{n-1/2}) \quad (3-25)$$

$$\left(-\frac{2i\omega}{c^2} + \frac{2}{c}\partial_\xi\right)\frac{a_1^{n+1/2} - a_1^{n-1/2}}{\Delta\tau} = -\frac{1}{2}\nabla^2(a_1^{n+1/2} - a_1^{n-1/2}) \quad (3-26)$$

然后再将上述两步分别写成有限差分形式。第一步可以直接用三对角形式求解，而第二步中横向只有微分算符，于是可以对横向做快速傅里叶变换后再化为三对角方程求解。

另外一个方法是去求算符 $1 - \frac{1}{ik}\partial_\xi$ 的倒数^{[19][20]}。例如，使用算符级数，不难写出 $(1 - \frac{1}{ik}\partial_x)^{-1} = 1 + \frac{1}{ik}\partial_x - \dots$ ，于是如果只保留到一阶项，那么方程变成了

$$(\partial_x + c\partial_t)a = \frac{c^2}{2i\omega}\left(1 + \frac{1}{ik}\partial_x\right)(\mu_0\chi a + \nabla_T^2)a$$

这个方程不难用Crank-Nelson方法处理。

当 ω/ω_p 不够大的时候，上述级数展开法会出现较大的误差，从数值结果可以看到，在等离子体密度不是非常低的时候，这解法得到的自调制效应常常和精确结果出现很大的偏差。

此外，不难证明由于使用的是有限差分方法，因为有限网格宽度效应，上述两个方法总是存在数值色散，（即使在级数展开法中对展开式保留到任意

阶), 当激光传播到调制效应发展起来之后, 上述解法给出的调制结果会严重失真。

一个变通手段是使用谱方法给出算符的倒数^[72], 也就是在纵向进行傅里叶变换, 直接用谱空间除法来代替算符求逆。由于谱空间除法可以消除掉有限空间网格效应, 这一方法可以给出比较可靠的激光传播演化结果。如果计算区间足够大可以包含整个激光包络, 这种方法非常有效。

尽管如此, 由于忽略了激光频率的改变, 有质动力导向中心近似法只在衰变不稳定性不明显的环境下才可以得出可靠的结果。幸好, LWFA的绝大部分实例都属于这种情况。在某些不能忽略拉曼过程的问题中, 可以通过加入拍频效应, 修改有质动力表达式和激光传播的包络方程(例如换成三波耦合方程), 得出定性上正确的结果^[73]。

4 尺度问题：数学考虑

4.1 时间和空间尺度问题：定标

完全的PIC模拟存在一系列稳定性约束。这些约束的存在是因为为了产生正确的物理描述，模拟过程必须能够分辨最快的重要物理过程。对于电磁PIC模拟，稳定性约束可以归纳为如下限制：

$$\begin{aligned}\Delta x &< \lambda_D \\ \Delta t &< \frac{2}{\omega_p} \\ \Delta t &< \frac{\Delta x}{c}\end{aligned}$$

这三个限制分别来自精细网格不稳定性，蛙跳格式的时间步长约束以及电磁场求解的CFL约束条件。对于纯静电模拟，第三个限制可以消除，但因为等离子体漂移扩散的问题，需要增加一个约束即

$$\Delta t < \frac{\Delta x}{v_{thermal}}$$

其中 $v_{thermal}$ 是等离子体中电子的平均热运动速度。

除了上述因为PIC方法本身的数学性质导致的限制之外，如果使用了Monte-Carlo耦合来模拟粒子之间的碰撞，那么还必须约束时间步长使得每个推进步内不要发生太多次碰撞，也就是

$$v\Delta t \ll 1 \tag{4-2}$$

其中 v 是每个粒子的碰撞几率。

由于上述时间空间限制的存在，正统PIC方法的时间步长和空间网格宽度相对于流体力学或者其他典型的模拟技术来说总是很小的。事实上，绝大部分问题都由于需要的网格过于精细及时间步数太多而无法用PIC方法进行全尺寸模拟。举例来说，在涉及到ICF等问题的研究中，需要处理固体密度($n \sim 10^{23} \text{cm}^{-3}$)的等离子体的运动，而即使在充分加热的情况下($T \sim 1 \text{Kev}$)，

德拜长度也不到一个纳米。相应的时间步长限制大概在 $0.1fs$ 的量级。对于尺度达到毫米量级，时间持续至少若干纳秒的作用过程，以目前的计算机性能显然不存在任何完全模拟的可能性。

类似的情况还发生在其他很多模拟过程中，一个相当极端的情况是电离层中的等离子体运动模拟。尽管这时的等离子体极其稀薄，但模拟问题的尺度实在太太，进行标准格式等离子体模拟的可能性为零。

这种和时空尺度相关的困难，是PIC模拟的根本问题，并且从来没有得到过根本解决。我们首先考虑一种比较简单的处理方式，即所谓标度缩放技术。所谓标度缩放技术，就是通过修改某些常数，计算一个物理上相似但尺度大大缩小的体系，最后再尺度放大到实践的尺寸。例如，在放电模拟中修改碰撞截面调整其演化时间^[74]，或者修改介电常数（有效电荷）改变尺寸等等。Sentoku等的“Reduced PIC”方法^[18]也是典型的标度缩放技术：在高密度区域，等离子体动力学效应没有碰撞效应重要，因此一旦实际密度超过某个阈值，就在PIC步骤中冻结等离子体密度为阈值，即在动力学过程中使用阈值密度，而在碰撞过程中使用真实密度对应的碰撞率。

最典型的标度缩放例子出现在霍尔发动机的模拟中^[75]，为了节省计算时间，将设备尺寸直接缩放到一个很小的尺度。霍尔发动机中的空间特征尺度主要包括粒子的自由程（决定于电子-中性气体碰撞自由程）和电子在磁场中的回旋半径，因此，为了保持相似性，需要增加碰撞截面和磁场使得电子自由程和回旋半径都按照同样比例降低。于是，模拟中使用的空间网格数可以大大降低。

更精细的处理需要保持鞘层分辨率。为此，可以同步提升系统的等离子体频率^[76]。整个标度关系是

$$\frac{\lambda_{eN}}{L} = const \quad \frac{\omega_p}{\nu_{eN}} = const \quad \frac{r_e}{L} = const \quad \frac{\omega_p}{\Omega_e} = const \quad (4-3)$$

根据标度关系给出主要参数的缩放规则，其中 $\zeta = (0, 1)$ 是相似比

物理量	缩放规则
尺寸	$L' = \zeta L$
等离子体密度	$n' = \zeta^{-1} n$
电场	$E' = E$
电势	$V' = \zeta V$
温度（能量）	$T' = \zeta T$
速度	$u' = \zeta u$
流量	$\dot{m}' = \zeta \dot{m}$
磁场	$B' = \zeta^{-1/2} B$
电流	$I' = \zeta^{3/2} I$

注意到 $\frac{L'}{L} = \frac{L'}{L}$ ，模拟的空间尺度是不变的，但由于系统的特征时间发生了改变，趋于平衡的时间缩短到了 $\zeta^{1/2}$ ，于是总的计算量削减到 $\zeta^{1/2}$ 倍。

标度方法简单易行，但因为不可能把所有重要物理量全都按比例缩放，缩放中忽视的物理量越多，越能够降低计算开销，同时导致模拟的可靠性越差。仅当相似律能够主宰系统的主要行为时，才能给出物理上有意义的结果。因此，这种方法只能在特定的条件下使用。

4.2 直接隐式模拟

解决尺度问题的根本手段是隐格式模拟。传统上有两种隐式模拟，分别称为直接隐式模拟和矩方法模拟。首先研究直接隐式模拟。

考虑前述粒子推进的蛙跳格式，为了简单，我们用静电问题来说明。在每一步推进中，是用已知的 t^n 时刻电场值 E^n 来计算下一个半步时刻 $t^{n+1/2}$ 的粒子速度 $v^{n+1/2}$ ，再用这个速度来求解 t^{n+1} 时刻的位置（电场） x^{n+1} 。每一步的计算都是显式给出的，所以称为显格式。

如果我们可以使用未来时刻的电场 E^{n+1} 来计算粒子速度 v^{n+1} ，也就是

$$\begin{aligned}
 \frac{v^{n+1/2} - v^{n-1/2}}{\Delta t} &= \bar{a}^n \\
 \frac{x^{n+1} - x^n}{\Delta t} &= v^{n+1/2} \\
 \bar{a}^n &= \frac{1}{2} [\bar{a}^{n-1} + \frac{q}{m} \mathbf{E}^{n+1}(x^{n+1})] \quad (4-4)
 \end{aligned}$$

利用导出前面时间步长限制同样的方法，可以发现这个算法对于谐振子方程是无条件稳定的。于是，可以使用非常大的时间步长。同样，利用网格色散关系的分析，可以看到这种方法对于精细网格不稳定性（自加热效应）也有良好的阻断效果。因为在计算 $v^{n+1/2}$ 的时候，电场 E^{n+1} 是未知的，它隐含的依赖于 x^{n+1} 和 v^{n+1} ，所以这类格式称为“隐格式”。有很多种可能的隐格式实现，从色散关系的分析中不难看到，这种格式的稳定性依赖于隐式电场计算和对于高频振荡的阻尼（ $\bar{\mathbf{a}}^n$ 计算中的时间累积平均）。通常可以通过改变格式算法来改变这种阻尼的大小和性质，从而得到不同的隐格式。这里给出的最简单格式称为“D1”格式。

直接求解上述隐格式运动方程是不可能的，因为 E^{n+1} 中包含了所有粒子位置（从而，所有粒子的速度）的贡献。于是这是一个 $6N$ 维的耦合方程组，完全没有求解的希望。隐格式模拟的要点，就是用某种方法写出 E^{n+1} 的近似估计式，从而能够有效地解出上述粒子运动方程。数学上，这个方法类似解常微分方程时的半隐方法。由于估 E^{n+1} 的方法不同，有两大类隐格式，第一种是所谓“Direct”隐格式^{[77][78][79]}。这种隐式方法的要点，是利用前述粒子运动方程的求和来给出决定 E^{n+1} 的方程。因为是直接从粒子运动方程开始，所以称为“直接”隐式方法。

直接隐式方法的关键点是将 $\Delta x = x^{n+1} - x^n$ 分解成两部分，其中一部分正比于未来时刻的电场 E^{n+1} ，例如

$$\begin{aligned} x^{n+1} &= x^n + v^{n+1/2} \delta t \\ &= v^{n-1/2} \delta t + \frac{1}{2} \bar{a}^{n-1} \delta t^2 + \frac{1}{2} a^{n+1} \delta t^2 \\ &= \tilde{x}^{n+1} + \frac{1}{2} a^{n+1} \delta t^2 \\ v^{n+1/2} &= v^{n-1/2} + \frac{1}{2} \bar{a}^{n-1} \delta t + \frac{1}{2} a^{n+1} \delta t \\ &= \tilde{v}^{n+1/2} + \frac{1}{2} a^{n+1} \delta t \end{aligned}$$

其中 $a^{n+1} = \frac{q}{m} E^{n+1}(x^{n+1})$ 。这样，就引入了带的变量 \tilde{x} 和 \tilde{v} 。而相应于 $\delta x = x^{n+1} - \tilde{x}^{n+1}$ ，可以把 ρ 类似地分解成

$$\rho^{n+1} = \tilde{\rho}^{n+1} + \delta \rho^{n+1} \quad (4-5)$$

$$\delta\rho^{n+1} = -\nabla \cdot P = -\nabla \cdot [\tilde{\rho}\delta x] \quad (4-6)$$

与Poisson方程 $\nabla \cdot \nabla\phi^{n+1} = \frac{\rho^{n+1}}{\epsilon_0}$ 合并给出

$$\nabla \cdot [(1 + \chi(\tilde{x}^{n+1}))\nabla\phi^{n+1}] = \frac{\tilde{\rho}^{n+1}}{\epsilon_0} \quad (4-7)$$

其中

$$\chi(\tilde{x}) = \sum_v \frac{1}{2} \tilde{\rho}_v^{n+1} \frac{q_v}{\epsilon_0 m_v} \delta t^2$$

这就是用来估计 E^{n+1} 的方程。于是，模拟算法如下：(1)从 x^n, v^n 推进到 $\tilde{x}^{n+1}, \tilde{v}^{n+1}$ 。

- (2)使用 \tilde{x}^{n+1} 累积电荷密度 $\tilde{\rho}^{n+1}$ 。
- (3)解带有极化率的泊松方程求出 E^{n+1} 。
- (4)利用 E^{n+1} 从 $\tilde{x}^{n+1}, \tilde{v}^{n+1}$ 推动到 x^{n+1}, v^{n+1} 。

不难看出在步骤(4)中需要使用的加速度是 $\frac{q}{m}E^{n+1}(x^{n+1})$ ，而此时已知的仅有格点上的电磁场 $E^{n+1}(x_g)$ 以及粒子的中间位置 \tilde{x}^{n+1} 。如果要求不高的话，可以直接用 \tilde{x}^{n+1} 插值出的加速度 $\frac{q}{m}E^{n+1}(\tilde{x}^{n+1})$ 进行推动。如果对近似精度要求较高，就需要使用迭代法。通常情况下，由于最后一步的推动的 δx 并不大，可以用牛顿迭代快速地给出结果。

可以看到，建造直接隐式方法的要点，一是分解粒子推动过程为两步，一步只和过去时刻的参数有关，另一步正比于未来时刻的电磁场。通过这种分解，可以构造出用来估计未来时刻电磁场的方程；二是需要求解这样得到的电磁场方程，由于存在极化率（在电磁情况下，还包括磁化率），这样的方程求解要困难得多得多。

上述方法可以推广到电磁情况。Hewett曾经给出过基本的电磁隐格式框架^[80]，基于横纵场的分离和麦克斯韦方程的二阶形式。但最简单的电磁直接隐式模拟方法由Welch等给出^[81]，基本过程如下：

$$p^{n+1/2} = p^{n-1/2} + \Delta t [m a^n + (p^{n-1/2} + p^{n+1/2}) \times q B(x_n) / (2\gamma^n m c)]$$

移动 $p^{n+1/2}$ 到左侧，解出这个方程就是

$$p^{n+1/2} = \langle T \rangle \cdot A$$

其中 $\langle T \rangle$ 是一个磁场旋转张量

$$\langle T \rangle = \frac{1}{1 + \Omega^2} \begin{bmatrix} 1 + \Omega_1^2 & \Omega_1\Omega_2 + \Omega_3 & \Omega_1\Omega_3 - \Omega_2 \\ \Omega_1\Omega_2 - \Omega_3 & 1 + \Omega_2^2 & \Omega_2\Omega_3 + \Omega_1 \\ \Omega_1\Omega_3 + \Omega_2 & \Omega_2\Omega_3 - \Omega_1 & 1 + \Omega_3^2 \end{bmatrix}$$

其中

$$\Omega = \frac{qB\Delta t}{2\gamma mc}$$

而A是源矢量

$$A = p^{n-1/2} + \Delta t a^n + p^{n-1/2} \times \Omega$$

如同前面的静电情况，直接隐式方法的基本思路是将这个推进步分解为两个push过程，即first push和final push。不同的分解方式带来不同的隐式格式，同时产生不同的场方程。

Welch etc的分解方式为

$$A = A_1 + A_2$$

A_1 和 A_2 分别是firstpush和finalpush，利用 $a_n = \frac{1}{2}(a^{n+1} + \bar{a}^{n-1})$ 其表达式是

$$A_1 = p^{n-1/2} + \frac{1}{2}\bar{a}^{n-1}\Delta t + p^{n-1/2} \times \Omega$$

$$A_2 = \frac{q}{m} E^{n+1}(x^{n+1})\Delta t/2$$

关于 A_2 的表达式忽略了纵横质量问题。

将执行firstpush之后的粒子速度记为 \tilde{v} ，写出 $v_{n+1/2} = \tilde{v} + \delta v$ ，利用能量关系式，

$$\Delta\gamma = \frac{qv \cdot E^{n+1}\Delta t}{mc^2}$$

其中对于 E 的取值从时刻 n 移动到 $n+1$ 。

于是finalpush的形式改成了

$$\delta v = \frac{q\Delta t}{\gamma^{n+1/2}m} \left[\langle T \rangle \cdot E^{n+1} - \frac{v(v \cdot E^{n+1})}{c^2} \right]$$

相应的，对于 x 的两次push是

$$\begin{aligned} \tilde{x}^{n+1} &= x^n + v^{n-1/2}\delta t + \frac{1}{2}a^{n-1}\Delta t^2 \\ &= x^n + \tilde{v}^{n+1/2}\Delta t \end{aligned} \quad (4-8)$$

和

$$x^{n+1} = \tilde{x}^{n+1} + \frac{1}{2}a^{n+1}\Delta t^2$$

令 $x^{n+1} = \tilde{x} + \delta x$ ，需要建立关于 E^{n+1} 的电磁场方程。Welch使用一阶旋度-旋度方法，因此首先要得出 $n + 1/2$ 时刻的电流密度，每个粒子提供的电流贡献，可以写成 $J_i = q \frac{x_i^{n+1} - x_i^n}{\Delta t} = qv_i^{n+1/2}\Delta t$ ，首先在firstpush结束的时候累加所有粒子的电流贡献 \tilde{J} ，于是剩余电流是

$$\Delta J = J^{n+1} - \tilde{J}^{n+1} = \rho\Delta x/\Delta t = \langle S \rangle \cdot E^{n+1}$$

其中的极化项

$$\langle S \rangle = \frac{\rho q \Delta t}{2\gamma^{n+1/2}m} \left(\langle T \rangle - \frac{\vec{v}_{n+1/2}\vec{v}_{n+1/2}}{c^2} \right)$$

于是麦克斯韦旋度方程组可以用隐式差分写成

$$\frac{E^{n+1} - E^n}{\Delta t} = \nabla \times B^{n+1/2} - \tilde{J} - \langle S \rangle \cdot E^{n+1} \quad (4-9)$$

$$\frac{B^{n+1} - B^n}{\Delta t} = \nabla \times E^{n+1/2} \quad (4-10)$$

这组方程是线性的，可以用ADI方法求解。

上述模拟方法中存在几个基本的问题，首先， δJ 是个近似的表达式（因为 δx 是近似的），因此， δJ 并不真正保持电荷守恒。而在Welch方法中，求解的是电磁场的两个旋度方程，散度方程的正确性依赖于电荷守恒的成立。所以，随着计算推进，会累积起额外的电势，并且逐步破坏模拟的可靠性。

解决方法是使用减除，也就是求解剩余方程

$$\nabla^2 \psi = -(\nabla \cdot E - \frac{\rho}{\epsilon_0})$$

从 E 中减去解出的剩余场 $E' = -\nabla \psi$ ，就得到正确的电磁场。

另一个问题是，尽管上述方程可以用ADI(交替方向隐式)求解，但求解效率并不高。如果磁场很弱，可以忽略在磁场中的旋转，上述求解可以显式完成^[82]：在非相对论条件下

$$\langle S \rangle = \frac{\rho q \Delta t}{2m} \langle (T) \rangle$$

其中 $\langle T \rangle$ 张量代表在磁场中的运动。忽略其非对角项，于是 $\langle T \rangle$ 转化为一个对角张量，这时，电磁场方程可以直接求解，无需迭代操作。

此外，也可以用二阶模式求解上述变形的Maxwell方程，例如Drouin等人的工作^[83]。

无论如何，直接隐式方法提供了令人满意的收敛特性，同时对于PIC核心的快粒子射束问题仍能给出物理上合理的结果。主要的误差在于由于使用了时间阻尼，在模拟中消除高频振荡的同时，会导致等离子体的自冷却行为^[77]。典型的模拟表明，对于温度敏感的问题，这一方法带来的误差仍然比较大。对于具体问题，这一自冷却效应的后果需要具体分析。

4.3 矩方法

矩方法是另外一种隐式PIC模拟技术。它的基本思路是利用矩方程的展开来获得对于未来时刻电磁场的估计^{[84][85][86]}。和直接隐式方法类似，首先要把速度分解为一个中间速度和一个正比于电场的修正，也就是对于粒子 P ，使用

$$\mathbf{x}_p^{n+1} = \mathbf{x}_p^n + \bar{\mathbf{u}}_p \Delta t$$

$$\bar{\mathbf{u}}_p = \hat{\mathbf{u}}_p + \beta_s \hat{E}_s^{\frac{1}{2}}(\bar{x}_p)$$

这里的 \hat{E}_s 是有效电场强度，也就是考虑到了磁场导致的旋转过程后，电场对粒子的加速作用，而 $\beta_s = \frac{q_s \Delta t}{2m}$ ， $\bar{\mathbf{u}}_p = \frac{1}{2} \mathbf{u}_p^n + \mathbf{u}_p^{n+1}$ 。用直接展开给出

$$\hat{\mathbf{E}}_s^{n+1/2} = \alpha_s^n \cdot \hat{\mathbf{E}}^{n+1/2} \quad (4-11)$$

$$\hat{\mathbf{u}}_p = \alpha_s^n \cdot \mathbf{v}_p^n \quad (4-12)$$

$$\alpha_s^n = \frac{1}{1 + (\beta_s \mathbf{B}^n / c)^2} (\mathbf{I} - \beta_s \mathbf{I} \times \mathbf{B}^n / c + \beta_s^2 \mathbf{B}^n \mathbf{B}^n / c^2) \quad (4-13)$$

考虑电荷密度和电流密度，可以将这两个量的格点值写成

$$\rho_s = \sum_p q_p S(x - x_p) \quad (4-14)$$

$$\mathbf{J}_s = \sum_p q_p \mathbf{u}_p S(x - x_p) \quad (4-15)$$

展开插值函数S得到

$$S(\mathbf{x} - \mathbf{x}_p^{n+1}) = S(\mathbf{x} - \mathbf{x}_p^n) - (\mathbf{x}_p^{n+1} - \bar{\mathbf{x}}_p^n) \cdot \nabla S(x - x_p^n) + \frac{1}{2} (\mathbf{x}_p^{n+1} - \mathbf{x}_p^n)(\mathbf{x}_p^{n+1} - \mathbf{x}_p^n) :: \nabla \nabla S(x - x_p^n)$$

代入到 ρ 和 \mathbf{J} 的表达式，就得出

$$\rho_s^{n+1} = \rho_s^n - \Delta t \nabla \cdot \mathbf{J}_s^{n+1/2} \quad (4-16)$$

$$\mathbf{J}_s^{n+1/2} = \hat{\mathbf{J}}_s - \frac{\Delta t}{2} \mu_s \cdot \mathbf{E}_\theta - \frac{\Delta t}{2} \nabla \cdot \hat{\mathbf{\Pi}}_s \quad (4-17)$$

其中的源项是

$$\hat{\mathbf{J}}_s = \sum_p q_p \hat{\mathbf{v}}_p S(\hat{\mathbf{x}} - \hat{\mathbf{x}}_p) \quad (4-18)$$

$$\hat{\mathbf{\Pi}}_s = \sum_p q_p \hat{\mathbf{v}}_p \hat{\mathbf{v}}_p S(\hat{\mathbf{x}} - \hat{\mathbf{x}}_p) \quad (4-19)$$

而唯像极化率张量是

$$\mu_s^n = -\frac{q_s \rho_s^n}{m_s} \alpha_s^n \quad (4-20)$$

这样得到了麦克斯韦方程中的电流密度和电荷密度项。由于极化率张量可能很大，因此直接用ADI方法求解旋度-旋度方程会碰到很严重的散度问题。可

以用近似的投影分解处理这个问题，但更一般的方法是使用二阶模式，也就是将Maxwell方程组写成时间域的有限差分形式，并且代入得到

$$\frac{(c\Delta t)^2}{4}[-\nabla^2 \mathbf{E}^{n+1/2} - \nabla \nabla \cdot (\mu_s^n \cdot \mathbf{E}^{n+1/2})] + \epsilon^n \cdot \mathbf{E}^{n+1/2} = \mathbf{E}^n + \frac{c\Delta t}{2}(\nabla \times \mathbf{B}^n - \frac{4\pi}{c} \hat{\mathbf{J}}_n) - \frac{(c\Delta t)^2}{4} \nabla(4\pi \hat{\rho}^n) \quad (4-21)$$

$$\nabla \cdot [\epsilon^n \cdot \mathbf{E}^{n+1/2}] = 4\pi \hat{\rho}^n \quad (4-22)$$

原则上，这对方程需要同时求解，但Ricci等人证明^[87]，实际上散度方程可以只在边界上求解，在模拟区域内部的网格点上，只要求解第一个方程就可以给出正确的结果。于是，每一步模拟可以通过求解上述关于 \mathbf{E} 的二阶方程来完成。求出 \mathbf{E} 之后，使用 $\nabla \cdot \mathbf{E}^{n+1} = 4\pi \rho^{n+1}$ 消去由于电荷守恒问题导致的误差。得出 $n+1$ 时刻的电场后，磁场可以用

$$\frac{B^{n+1} - B^n}{\Delta t} = -\nabla \times E^{n+1/2}$$

求出。

最后的问题是推动粒子。这个步骤数学上的问题相当于直接隐式模拟中的final push，但因为矩方法中的推动包含了一个时间步的全部推进，因此一般来说为了提高精度总是迭代解出粒子的运动方程。此时的推动中包含了磁场，于是对于每个粒子，其运动方程组有六个非线性方程，需要使用一些求解技巧，例如Newton-Krylov迭代等等^[86]、

由于矩方法中没有使用时间域的累积阻尼，其稳定性行为比较复杂。标准的三个稳定性约束被改为^[22]

$$O(0.1) < \frac{v_{thermal} \Delta t}{\Delta x} < 1 \quad (4-23)$$

和直接隐式模拟一样，矩方法的主要困难也在于求解场方程。电场方程形式上是一个亥姆霍兹方程和一个Poisson方程，可以用有限体积或者有限差分离散化后转化为线性方程组，由于矩阵是非对称非正定的，通常用ADI或者GMRES方法求解^[88]。对于这样的问题，如何设计高效能的迭代法是现阶段的主要数学困难之一。通常在Poisson类型问题中使用的多重网格方法用在Helmholtz问题上效果常常不够好，需要选择特殊的光滑化方法^[89]或者作为预条件使用^[90]。

最后，矩方法也可以推广到相对论形式^[91]，但其做法非常复杂，并且如同在直接隐式方法中一样需要相当多的近似。其具体结果还需要验证。

4.4 多尺度和网格细化模拟

隐格式模拟提供了对于稠密等离子体模拟需要的稳定性，同时也意味着随时间步增大的阻尼。正常情况下，这是研究所需要的。但在涉及到鞘层问题的建模中，过宽的空间网格会影响鞘层的正确描述，因为等离子体鞘层厚度基本上就在德拜长度的量级。为了正确描述鞘层行为，在靠近边界处必须使用分辨率较高的网格，而在等离子体的准中性区域可以使用比较宽的网格。此外，由于等离子体德拜屏蔽的问题，在鞘层内的电子响应也比内部要复杂，需要使用和网格宽度匹配的时间步长。于是当等离子体尺寸较大的时候，网格宽度之比可以达到很大，这就形成了多尺度模拟^{[92][93]}。

使用隐格式，在不同尺度的网格上可以使用基本相同的模拟算法。由于不同网格对应的时间步长是不同的，粒子需要在网格中分配（参考关于“Pull Mode”的内容），然后在推动粒子时，根据当前执行步数只推动相应的网格内的粒子。例如，设置一组彼此之间尺寸成比例的网格1,2,3,...，其中1-4号网格时空步长是 $(\Delta t, \Delta x)$ ，而5-8号网格时空步长则是 $(2\Delta t, 2\Delta x)$ ，9-10号网格时空步长是 $(4\Delta t, 4\Delta x)$ ，.....。然后网格1到网格4中的粒子在每个时间步都被推动，而网格5到网格8中的粒子两个时间步才被推动一次，依次类推。利用这种方法，每个 Δt 时间步内需要被推进的粒子数可以大大降低。

对于静电问题，这种多尺度方法是非常行之有效的。由于Poisson方程的全局特性，每次细网格推动后，都要根据被推动的粒子来重新累积电荷（未被推动的粒子无需重新计算，可以简单地在原有的网格电荷上加减由于推动导致的电荷改变），并重新求解Poisson方程。在这种情况下，求解Poisson方程成了主要的时间消耗（除非问题是一维的）。为此，需要开发高效的非规则网格Poisson求解器。基本的思路类似于前面使用的多重网格方法，即首先在粗网格上近似求解方程，然后逐级精确化^[94]。为了提高解的可靠性，往往需要多次在粗细网格之间迭代。由于在网格宽度发生剧烈变化的时候，差分矩阵的对称正定性可能遭到破坏，矩阵的迭代求解需要特别小心。

对于密度发生剧烈变化或者有复杂边界结构的目标，存在另外一种进行多尺度模拟的动力。可以在高密度区域加密网格^{[94][95][96][97]}，或是在复杂形状区域调整网格获得对外形的更精确逼近。和前面讨论的问题不同的是，这时时空步长的差距往往并不会太大（在前面的鞘问题中，空间步长可以相差上百倍）。很多情况下，这种模拟可以用显格式完成，称为网格细化技术。

网格细化技术（包括自适应网格细化）因为使用的是显格式处理，其数学形式上比隐式模拟要简单一些，但也会碰到一些显格式的固有问题。其中之一是前面分析过的粒子分裂和合并问题。另一个问题出在粒子漂移过粗细网格边界时碰到的伪加速的问题：当粒子越过不同形状和大小的网格边界的时候，作用力会发生跃变，从而产生虚假的作用力。为此，需要用比较复杂的插值方法来降低这种噪声。

在在电磁PIC模拟中会碰到额外的困难：由于不同宽度的网格具有不同的电磁波速度，于是电磁波穿过不同网格分界面的时候会发生反射，特别是，短波长的电磁波无法在宽网格中传播，因此在界面上会被反射回来；如果细网格被粗网格所包围，电磁波会来回反射，最后导致严重的不稳定性。

为此，可以使用隐格式（并添加适当阻尼）吸收掉高频电磁波^{[96][80]}，但由于在非均匀网格上ADI-FDTD方法不能应用，于是每一步都要求解一个亥姆霍兹方程，这是相当麻烦的。也可以修改差分格式来处理掉这种虚假反射^[98]。另一类解决方法是分别在粗细网格上分别求解麦克斯韦方程组，并且对结果进行减除来消除可疑的振荡项^[99]。

5 粒子-粒子耦合：物理考虑

5.1 蒙特卡洛耦合：带电粒子-中性粒子碰撞

在放电问题中，等离子体需要靠电离碰撞或者二次电子发射来进行维持。为此，在PIC建模中必须引入描述带电粒子-中性粒子碰撞的过程。这可以通过Monte-Carlo方法获得，这种用PIC方法描述粒子运动，用蒙特卡洛方法描述粒子之间的近距离碰撞过程的方法称为PIC/MCC模拟^[9]。

考虑一个相对于靶以速度 V 运动的入射粒子，粒子动能为 E ；靶粒子数密度为 n ，而入射粒子和靶之间碰撞的总截面为 $\sigma(E) = \sigma_1(E) + \sigma_2(E) + \dots$ ，下标代表各种不同类型的碰撞。那么，发生第 i 种碰撞的频率是

$$v_{coll} = n(x)\sigma_i V$$

在时间 Δt 内，发生碰撞的总几率是

$$P = 1 - \exp[-V\Delta t\sigma(E)n(x)] \quad (5-1)$$

原则上，这种过程可以通过对每个粒子执行一次蒙特卡洛过程来完成：遍历全部个宏粒子，对于每个宏粒子，生成一个随机数 r ，如果 $r < P$ ，就根据随机的碰撞参数模拟一次碰撞。例如，对于弹性碰撞，改变粒子的动能和动量；对于电离过程，按照能量关系产生一个新粒子等等。但遍历所有粒子并计算碰撞几率是一种开销非常大的行为。为了提高性能，带电粒子和中性粒子之间的碰撞通常使用“空碰撞”方法描述^[100]。

空碰撞方法的基本概念是首先设置最大碰撞几率，即

$$v_{max} = \max[n(x)] \max \left[\sigma(E) \sqrt{\frac{2E}{m}} \right] \quad (5-2)$$

$$P_{null} = 1 - \exp(-v_{max}\Delta t) \quad (5-3)$$

设总的模拟粒子数为 N ，计算出一个时间步内发生的最大可能碰撞数

$$N_{coll} = NP_{null} = N[1 - \exp(-v_{max}\Delta t)] \quad (5-4)$$

然后在总粒子中随机抽取 N_{coll} 个粒子，然后遍历抽取的样本粒子，对每个样本粒子，生成随机数 R ，并根据 r 的具体大小执行如下操作：

$$R < \frac{v_1(E)}{v_{max}} \quad \text{执行1类型的碰撞}$$

$$\frac{v_1(E)}{v_{max}} < R < \frac{v_1(E) + v_2(E)}{v_{max}} \quad \text{执行2类型的碰撞}$$

.....

$$\frac{\sum_{i=1}^N v_j(E)}{v_{max}} < R \quad \text{无操作}$$

在碰撞中碰撞参数（瞄准距离）的设置是任意的，末速度利用出射角度计算(see, e.g. [101])。由于散射几率存在一定的方向性，需要根据微分截面计算散射到各个方向的几率并按照这个几率抽样出射角度。在质心系中，出射到某一角度的几率是

$$P(V, \chi) = \frac{\sigma(V, \chi)}{\sigma_{total}} \sin \chi d\chi d\phi \quad (5-5)$$

因此选择出射角度的方法是产生一个随机数 r ，并且解方程

$$\frac{2\pi}{\sigma_T} \int_0^{\chi} \sigma(V, \chi') \sin \chi' d\chi' = r \quad (5-6)$$

对于散射截面不是球对称的情况，这个方程的求解是相当复杂的。

由于在上述PIC/MCC的每一步推进中，假定粒子都是在电磁场作用下无碰撞地运动 Δt 再发生碰撞，因此建模的可靠性要求在每个推进步中，每个粒子的碰撞可能远远低于1，实际上，一般取

$$v\Delta t < 0.1$$

因此，上述方法一般只能用于背景气压较低的情况。

这种包含碰撞电离的模拟方法在放电问题中已经广泛应用。和单纯PIC方法的主要区别，也是这模拟的根本问题，在于为了消除电离碰撞导致的电荷随机涨落噪声，每个网格内的粒子数必须相当大 (> 100)。因此，其模拟计算量一般都比相同尺度的无碰撞问题大两个数量级。

5.2 库仑碰撞

在PIC模拟中，粒子之间相互作用只考虑其集体部分，从而截断了粒子之间的短程库仑碰撞。通常情况下这个问题不大，但对于密度较高的的等离子体，这种截断意味着宏粒子达到热平衡态的速度要比真实粒子的驰豫过程慢很多。在热运动比较重要的场合，这可能会带来物理上的问题。

在PIC模拟中加入库仑碰撞的思路很早就有，基本方法是所谓“binary collisions”^[102]：首先将粒子按照网格分组，然后每个网格内的粒子随机配对，最后对于每一对粒子，进行一次随机碰撞。为了保证统计特性，碰撞出射角度分布要满足碰撞算子的结果：

$$P(\chi) = \frac{\chi}{\langle \chi^2 \rangle_{\Delta t}} \exp\left(-\frac{\chi^2}{2\langle \chi^2 \rangle_{\Delta t}}\right)$$

其中出射角方差是

$$\langle \chi^2 \rangle_{\Delta t} = \frac{q_\alpha^2 q_\beta^2 n \Delta t \ln \Lambda_{\alpha\beta}}{2\pi\epsilon_0^2 \mu^2 V_{\alpha\beta}^3}$$

μ 是折合质量， $V_{\alpha\beta}$ 是相对速度，而 $\ln \Lambda_{\alpha\beta}$ 是库仑对数，即 $\Lambda_{\alpha\beta} = \frac{b_{max}}{b_{min}}$ ， b_{max} 和 b_{min} 分别是最大和最小的瞄准距离（碰撞参数）。一般来说， b_{max} 可以设置为 λ_D ，即只需要考虑德拜球内的碰撞；而最近的瞄准距离可以根据平均相对速度估计，即^[103]

$$b_{min} = \frac{|q_\alpha q_\beta|}{2\pi\epsilon_0 \mu \langle g_{\alpha\beta}^2 \rangle}$$

相对速度的平方均值 $\langle g_{\alpha\beta}^2 \rangle$ 是

$$\langle g_{\alpha\beta}^2 \rangle = \frac{3kT}{m_\alpha} + \frac{3kT}{m_\beta} + (\bar{v}_\alpha - \bar{v}_\beta)^2$$

其中前两项是热运动导致的平均速度，而 \bar{v}_α 、 \bar{v}_β 分别是两种粒子的整体漂移速度。

近来，这个模型被推广到相对论情形^[18]。

尽管这个思路并不复杂，但和空碰撞方法不同，在实践中，配对碰撞方法不太具备可应用性。原因是为了使得这个计算有意义，需要满足碰撞频率条件

$$\nu_{coulomb} \Delta t \ll 1$$

其中

$$v_{coloumb} = \frac{q_\alpha^2 q_\beta^2 n \ln \Lambda_{\alpha\beta}}{2\pi\epsilon_0^2 \mu^2 V^3}$$

从物理上讲，当库仑碰撞起到较为明显的作用的时候，等离子体密度一般来说比较高，于是模拟的时间步长会很短，开销变得非常大。另一方面，和带电粒子-中性原子碰撞不同的是，这种配对碰撞模型，需要对每个粒子都进行一次计算，其总的计算量大大高于使用空碰撞方法。此外，编程方面的某些设计进一步使得问题复杂化：很多代码中，粒子数据以平直数组模式存储，于是为了确定粒子所在的网格，不得不在每次碰撞之前进行粒子排序^[104]，这时的性能会恶化到难以接受的程度。

Nanbu提出的累积碰撞模型稍稍降低了所需的计算量^{[105][106][103]}，思路是将多次小角度库仑碰撞归并为一次大角度的碰撞。由于这些次碰撞是随机的，可以按照随机过程累积为一次大角度碰撞。于是不再需要限制 Δt 的长度。相反， Δt 用来计算库仑散射的分布，也就是散射几率 $f(\chi)d\Omega$ 满足

$$f(\chi) = \frac{A}{4\pi \sinh A} \exp(A \cos \chi) \quad (5-7)$$

$$\coth A - A^{-1} = \exp[-S_{\alpha\beta}(\Delta t)] \quad (5-8)$$

$$S_{\alpha\beta}(\Delta t) = \frac{\ln \Lambda_{\alpha\beta}}{4\pi} \left(\frac{q_\alpha q_\beta}{\epsilon_0 \mu_{\alpha\beta}} \right)^2 n_\beta V_{\alpha\beta}^{-3} \Delta t \quad (5-9)$$

针对这个分布，Nanbu给出抽样公式为

$$\cos \chi = \frac{1}{A} \ln[\exp(-A) + 2R \sinh A] \quad (5-10)$$

Nanbu方法中的主要问题是在不同权重的粒子之间碰撞时，单个碰撞过程并不是能量动量守恒的，只有大量粒子的平均结果才守恒。因此，当每个网格内的宏粒子数很少且需要使用变权重粒子的时候，这一方法可能会扰动粒子分布。这是许多高能密度物理方面的问题所在。但如同我们指出的那样，对于这类问题，运用简单的显式PIC方法本来就是完全不现实的。

即使使用了Nanbu累积方法，库仑碰撞的计算量仍然是相当大的，首先粒子必须被分划到网格，这可以使用粒子排序，也可以使用所谓“Pull”模式。其次，网格内的粒子必须被随机配对，这可以通过随机改变粒子顺序（洗牌）之

类的算法实现。最后，对每个配对抽样 χ 仍然需要很大的计算开销。总体上，计算量大致可以估计为 $O(N \log N)$ ， N 是使用的总粒子数。考虑到空间网格分辨率问题，除非使用隐格式，否则在多维PIC代码中考虑库仑碰撞并不是十分现实的。此外，还需要指出的是，使用库仑碰撞模型对于放电问题的动理学行为有一定影响。

另一种思路是使用所谓“Tree code”^[107]。即将粒子组织成不同的簇，将簇用树状数据结构表达，然后按照树的结构递归地计算粒子之间的作用力。对于远处的粒子，其效果可以通过直接计算簇的集体作用求得；而对近处的粒子，可以在计算中直接计算两体库仑力。这种方法将库仑力求和的计算量从 $O(N^2)$ 缩减到 $O(N \log N)$ 。进一步地，使用多级矩展开法可以将这计算的计算量削减到 $O(N)$ ^[108]。

Tree code在引力问题的研究如星系动力学中很常用，而在等离子体问题中的应用还需要进一步验证。此外，原则上Tree方法只能用于静电问题，这限制了它的前景。

6 计算机技术问题

6.1 性能优化：算法和体系结构

前面讨论的主要是物理和数学方面的一些进展。在另一方面，近20年来计算机技术的发展，主要是先进微处理器架构和并行体系结构的发展，为PIC技术提供了新的进步点。同时，PIC模拟的巨大计算量又为发展高性能计算技术提供了新的动力。近些年的PIC程序除了使用越来越复杂的数学技巧之外，也开始引入来自计算机科学（如体系结构）方面的技术，甚至包含了软件工程等学科的新进展。这些技术在计算机科学中通常不算十分新颖，但在科学计算领域中的大规模使用却是很近期的事情。另一方面，这些技术对模拟程序性能的提升是立杆见影的，这使得开发PIC模拟程序的研究者们很可能需要了解通常只有计算机科学技术方面的研究者才需要了解的一些知识。

这类技巧的引入主要是为了提高代码的性能。关于如何提高数值计算程序的性能，研究者已经有了很多的经验，作为这些经验的总结，lapack代码的产生和广泛应用，将科学计算程序引入了软件工程的时代。但PIC程序相比于传统的数值程序，具有一些不寻常的特性。从代码组成来看，PIC代码的非同一般之处在于：

(1)通常情况下，PIC代码的“time consume core”非常简单。在理想的情况下，PIC算法的最大执行开销来自Weighting和Pushing/Moving步。这两个步骤的运算量正比于模拟中使用的粒子数。而在另一方面，除非使用非结构化网格，否则这两个步骤的内容又是非常简单的，通常仅几十行C/FORTRAN代码。

(2)很少有适用于PIC核心代码的通用程序框架。尽管从算法角度看PIC的粒子推动代码非常简单，但因为存在太多可能的具体实现，而且不同实现常常对应完全不同的粒子和电磁场模型，因此直接组合现成的子程序库的方法在PIC代码开发方面往往得不偿失，或者导致程序的极大复杂化。

(3)在目前阶段，PIC代码的基准性能普遍不高。这里“基准性能”指在单个CPU内核上运行的PIC代码的有效性能与该CPU的理论性能之比。所谓有效性能，指的主要是代码中执行了多少必须的操作。以粒子推动步为例，非相

表 6.1 PIC代码执行效能

程序	粒子推动时间	测试设备	CPU峰值	效率
VLPL	$3\mu s$	Alpha 21264 300MHZ	600MFLOPS	~ 5.5%
OSIRIS	$7.6\mu s$	Alpha 21264 450MHZ	900 MFLOPS	~ 1.6%
VORPAL(实测结果)	$5 * 10^{-7} s$	Core 2 1.8 GHZ	7.2 GFLOPS	~ 2.8%
PIC on JASMIN	$\sim 10\mu s$	Itanium 1.3 GHZ	5.2 G FLOPS	< 0.2%

对论Boris旋动中推动一个粒子包含39个浮点操作（包括18个加法和21个乘法）。计入Load/Store指令，连同Weighting在内，所需要的浮点操作可估计在100个浮点操作(FLOP)。一个理论峰值性能为 k MFLOPS的CPU，每个粒子的推动时间是 $\frac{2}{k} * 10^{-4}$ 。但实际上几乎所有的现有PIC代码。其性能都与这个理论值相距甚远，举例来说，一些典型的PIC代码其推动性能如下(这些结果来源于文献，其中可能包含由于场求解器导致的误差，但通常影响不大) [50][109][110][111]：

当然，上述峰值都是按照理想情况下的CPU峰值计算，由于使用如有效电流分配（电荷守恒格式）等复杂算法，在相对论下存在除法/开方等特殊的长延迟指令，实际PIC计算程序性能必然低于理论峰值。但即使如此，上述性能仍然是非常低的，原因主要是目标代码中存在大量附加部分（例如OOP中的对象维护代码等等），以及浮点流水线不能满载运行。后者是现代微处理器中出现的新问题：许多高性能处理器具有每周期执行两个甚至四个双精度浮点操作的能力，而编程人员和编译器没有能力生成足够多的互不关联的浮点指令。在以前的单浮点部件处理器上，并不需要考虑这种问题，于是相对效率要高些[112]。

上述特性的存在，使得PIC代码的性能优化常常和软件工程的基本思想相抵触。在性能较为敏感的情况下，常常要首先根据性能决定步骤来设计子程序，再由子程序来确定数据结构和整体程序框架。于是PIC的性能优化往往是“Case by Case”的，通用的优化技巧并不多。应该指出的是，这种方法在大部分数值计算程序中是不受欢迎的，原因是这设计方式导致软件维护和扩展的困难。

最常见的通用优化技巧是关于存储器访问方面的优化[113][114][115]。现代微处理器普遍使用分级存储结构，即速度很快但容量非常小的cache和速度较慢但容量大的内存。当处理器试图访问某一数据的时候，它首先在Cache中寻找，如果Cache中存在这一数据的拷贝，则可以直接返回；否则，系统必须从内存将数据读入Cache。后一种情况存在一个非常长的等待时间（在现代微处理器上常

常需要数百个CPU执行周期),在此过程内CPU只能等待。被读入的数据将保存在Cache里面直到由于Cache容量原因被换出。为了使得内存的访问有效化,每次读入Cache的数据除了CPU请求的内容之外,还会包含这一内容附近的一段(称为一个“Cache Line”)数据。

在PIC的Weighting步和Pushing步中,需要访问两种数据:粒子的位置、速度以及网格上的场量。考虑最常见的PIC算法实现,程序遍历所有粒子,对于每个粒子,取得其坐标信息,然后根据坐标信息找到它所在的网格,再将其电荷插值到网格或者根据网格量求出它所感应到的电磁力。如果粒子被设计为连续存储,那么无论如何粒子数据的读入都是高效的,而对网格数据访问并不是,因为依次读入两个粒子不一定在同一个网格中。于是,如果能在执行中让连续读入的两个粒子总是处在同一个网格中,那么存储器访问可以处于最高效率下。实现这一点的最简单做法是对粒子按照其网格排序,即对网格编号,然后在内存中把网格号相同的粒子排列在一起存储。当然,粒子会运动,于是网格次序会逐步失去。但这没有关系,因为只要每隔若干步进行一次排序就可以保证粒子大部分情况下都处于近似排序的状态。依赖于快速排序算法, N 个粒子的排序可以在 $O(N \log N)$ 操作内完成,同时,实践证明,一般情况下每几百个粒子推动步进行一次粒子排序,已经能够得到相当满意的性能提升。在现代的PC架构上,排序方法可以提高插值操作两到三倍执行效率。这方法的主要缺陷在于粒子数据必须是连续存储的,当出现粒子在边界上吸收之类的情况时,需要填补由于粒子消失产生的数组空隙。当每一步消灭的粒子数不多时,这可以通过在消除粒子之前直接交换数组末端的粒子和被消除的粒子来完成^[57]。如果涉及到的粒子数很多,一般采用对粒子进行标记并在若干步推动之后整理数组消除空隙的方法。

容易想到的另一个办法是使用所谓“Pull Mode”^[116]:为网格编号,然后每个网格拥有自己所包含的粒子。当进行插值的时候,遍历所有网格,对每个网格,取出其中包含的粒子数据并进行插值。这个方法在需要考虑两体近程作用的MD模拟中是非常常用的,对于需要考虑库仑碰撞和电子-离子合并过程的PIC/MCC模型也有很多应用。但单就PIC算法来说这个方案并不像表面上看来那么有效,除非程序中还包含两体碰撞部分。原因有两个,明显的第一个原因是当粒子运动的时候,粒子必须从一个网格对应的存储区移动到另一个,这

产生额外的开销；另一个原因是算法上的：由于每个网格包含的粒子数可能会剧烈地变化（从 $O(1)$ 到 $O(1000)$ ），为了避免出现内存耗尽问题，常常需要用链表之类的数据结构存储每个网格的粒子。由于这类数据结构是间接寻址的（比如，链表的每个数据都是通过访问一个指针值取得地址，再通过地址找到对应的数据），并不能保证数据存储的连续性，因此，其性能往往明显低于平直数组访问。为此，需要使用一些更高级的技术，如C++的Vector容器^[110]等等。

Pull mode反映了目前PIC代码设计的一个问题，即为了实现较为高级的算法，不得不大量引入软件工程思想。从OOPIC引入面向对象开始，PIC程序的基本特色已经发生了明显的变化。例如，Vorpal代码大量使用了模板编程来处理维度相关代码。未来，这种变化还将进一步增加扩展。

另一方面，对于代码本身的优化虽然较为少见，但也可能具有极大的作用。一个典型的例子是SIMD代码的使用^[117]。很多现代微处理器引入了单指令多数据(SIMD)扩展指令，允许在同一条指令中对多个浮点数据进行操作。当程序中含有大量对成组数据的相同操作的时候，使用SIMD扩展指令（矢量化）比使用标量运算指令可以成倍地提高性能。这正是PIC粒子推动代码的情况。这种矢量化代码和传统的向量机上的执行代码类似，不同的是，由于PIC推动代码通常比较灵活，编译器还不能有效地辨认其中的数据相关并且将其转化为矢量代码。目前利用SIMD特性主要的需要使用汇编语言。对于一般的数值模拟程序，用汇编语言改写需要太大的工作量，但对于PIC代码，执行时间主体是简单的插值代码，这种做法是行之有效的。此外，高级语言特别是面向对象程序中总是存在大量的多余指令，用汇编语言改写后除了能利用SIMD特性外，还可以大幅度削减这种多余指令，进一步提升性能。实践表明，使用汇编语言或者C语言内嵌指令一般可以提高代码性能2-3倍（在极端的情况下，用汇编语言直接操作底层数据的代码可以比C++的面向对象代码快10倍以上）^[57]。未来微处理器的执行宽度和SIMD宽度都可能被进一步提升，而自动矢量化编译技术仍然不够成熟，汇编语言和高级语言的性能差距可能会进一步加大。问题在于，这种性能提升是要以放弃代码的可移植性和数据结构的可维护性为代价的。这是未来的PIC代码需要面对的一个困境：代码的维护性和性能之间如何权衡？

6.2 并行处理：基本考虑

并行计算是提高数值模拟程序性能的基本手段之一。从上世纪90年代开始，研究者在PIC程序的并行化方面已经有了大量的工作。和优化中碰到的问题一样，PIC模拟程序的并行化也常常是具体问题相关的。

进行并行化处理的基础是并行计算模型^[118]。对于大部分数值程序，脱离使用的并行计算模型来讨论并行化方法是毫无意义的。近15年来，随着低成本的集群（Cluster）计算机的普及^[119]，在科学计算领域，使用进程消息传递和分布存储模型的消息传递接口(MPI)^[120]已经成为并行化的标准。即使微处理器架构不断发展提供了SIMD或者CMT等技术的廉价实现，但这些技术更多地用来进行节点上代码的优化，而程序的整体结构并未发生变化。

消息传递并行模型和Cluster硬件结构对于程序结构的主要限制在于数据视图。在这种并行体系中，每个进程只能看到自己私有的数据，要访问其他进程（其他节点）的数据，只能通过消息传递函数。由于硬件和软件实现的问题，消息传递通常很慢并且具有很长的延迟：即使使用Infiniband连接技术，两个节点之间的数据通讯带宽也不会超过20Gbits/s (2.5GB/s)^[121]，而通讯延迟时间在数微秒的水准；如果是用廉价的千兆以太网连接，对应参数会下降到250MB/s和50 μ s。相比于节点内大约10GB/s的数据带宽和几十纳秒的延迟，节点间通讯很容易成为并行化的瓶颈。为此，尽量减少通讯开销是必要的，其中减少小数据量多次访问（这种访问带来很大延迟）行为尤其重要。

PIC算法的工作数据包括场量和粒子状态数据。粒子之间的相互作用由场量决定，一旦完成场量计算，粒子就可以独立地被推动（包括网格电磁场到粒子的插值），其并行化是平凡的。并行化需要考虑的非平凡内容包括：如何进行粒子电荷到网格电荷密度的插值；以及如何通过网格电荷/电流密度计算电磁场。在这两步之中，网格上的场量之间存在耦合，因此需要考虑其数据视图。

考虑电磁场求解是局域化的情况，例如显式麦克斯韦求解。这时，场量耦合仅仅发生在相邻格点上；粒子只和所在网格的场量耦合。于是，网格和粒子可以同步划分。为每个进程指派其所有的网格以及网格中包含的粒子，PIC工作数据就被分割为按照进程的结构。在每步推进之前，每个进程需要取得其临近进程的边界网格场量来进行自己的推进；每次Weighting操作后，相邻进程也需要交换自己的信息，使得自己的边界网格得到正确的Weighting。最后，由于

完成，其他大部分节点在推进粒子的时候处于空等状态；其次在完成一个推进步之后交换粒子的时候，由于过大的数据通量使得通讯开销大大增加，限制了模拟的效率。

在这种情况下，需要将场和粒子进行非同步地分割，即场和粒子所用指派策略不同。当需要推动粒子的时候，每个计算进程按照负责的粒子去申请对应的场量信息；而在求解电磁场的时候，每个进程按照自己被指派的场区域进行计算。负载平衡通过动态调度每个计算进程上负责的粒子数来完成。例如，可以将粒子分划为若干群，每步推动之前，将对应粒子分配给相关进程^[124]。这种方法会大大增加区域映射和数据传送的复杂性，但由于提供了较好的负载平衡性，对于强的束流问题可以得到很好的效果。(虽然如此，这一方法会极其严重地降低代码的基准性能，除非问题具有极端的不平衡性，否则从负载平衡带来的受益往往不足以抵消基准性能的下降)

最后，对于涉及到鞘层和宏观运动的问题，每个网格内含的粒子数可能会发生长时间的变化。例如在有鞘层生成的放电问题中，鞘层形成后，鞘区的电子密度下降到接近于0，于是包含鞘层的计算程序的执行负载会随时间而大大降低。为了提高并行效率，可以在运行期间调节区域划分的策略和尺寸形状，以实现每个进程上的计算量大体相等，这是动态负载均衡技术的另一种典型例子，也是更为常见和实用的情况^[125]。

6.3 非局域求解器问题

在显式电磁PIC模拟的各个步骤中，主要的开销来自粒子推动过程，这对于静电问题或者隐格式模拟中可能是不正确的，因为这时需要解二阶椭圆型方程，典型的如Poisson方程和Helmholtz方程。在二维或者三维条件下，N点Poisson问题的求解可以比N粒子的推进开销大10-100倍。此外，不仅方程的求解变得比较复杂，其并行化也变得大为困难。考虑Poisson方程的情况，原则上，其求解器总是全域的，每一点的场量依赖于所有点的源项。于是，在求解中，每一点的源项数据都必须被传递到全部计算进程。在消息传递系统中，这意味着一个“All to All”（多对多）通讯。于是，在并行机上，场方程求解可以压倒粒子推进步成为决定性步骤。

通讯问题的严重性随问题尺寸和互联硬件结构而定。在典型的cluster体系结构上，这种多对多通讯的时间不仅随通讯数据量的增加而增加，同时还会随总通讯节点数目的增加而增加。于是，随着并行程度的增加，通讯时间也会逐步增加。最后，进一步增加计算节点数反而会导致程序性能的下降。

容易想到的方法是改变通讯方式。快速Poisson求解器是这种情况的一个很好例子。容易看到，在进行快速傅里叶变换和三对角方程求解时，每个计算进程只需要了解所在的格点行和列的数据。于是，如果把数据按列划分，在计算傅里叶变换时，每个进程可以独立地完成；在完成了快速傅里叶变换之后，对网格数据进行转置，然后又又可以独立进行三对角求解或者另一次傅里叶变换^{[13][126][127][128][129]}。虽然全局数据转置仍然是一个开销非常大的操作，但在三维情况下，这操作可以被分解成小得多的子块内通信来完成，于是，其并行化效率可以有效地提升^[130]。对于二维的情况下，也有类似的分解法^{[131][127]}，但效果受到数据延迟的限制很容易饱和。不过，这种方法总的通讯量非常高，对于慢速网络，带来的开销仍然是致命的。另外，当网格被分解得很细的时候，其并行效率还是会饱和。

早期对于Poisson问题求解常常使用一些准局域的迭代法，如古典SOR方法等等。但不难看到，Poisson问题的全域本性决定，每一迭代步导致的影响域越大，迭代收敛速度越快。因此，使用这类迭代法在容易实现很高的表面并行效率的同时，会因为基准效率的下降导致实际效果不升反降。相反，类似于Multigrid方法的半全局方法和快速Poisson求解器的全局方法，可以获得非常高的基准效率，同时却给并行化带来了巨大的麻烦。这是并行计算技术中的典型困境之一。

工程上也常用代数层面的并行方法。这是由于二阶椭圆型问题的求解普遍使用的是迭代法，如共轭梯度(CG)，广义最小残差(GMRES)等等。这类迭代法一般是由较为基本的矩阵操作（如矩阵和向量的乘法，向量的内积等等）按照一定的顺序组合而来。因此，如果把系数矩阵和对应的解都按照格点分布在对应的计算节点上，并且建立了底层的并行计算模块（特别是并行的向量乘法），那么上述方法可以直接并行化^{[132][133]}。这种方法的优点主要是抽象了算法实现，比较适用于各种非结构网格问题，同时往往作为标准程序库提供，如Trilinos^[134]和Petsc^[135]等等，这也使得程序易于维护。但如同大家熟知的那

样，由于典型的场方程条件数随问题尺度快速地增长，这类迭代法的速度比针对问题特点设计的解法要慢得多。例如，在典型尺寸的网格上，普通的不完全楚列斯基分解预条件共轭梯度(ICCG)比多重网格法要慢一到两个数量级。此外，因为这种分解法的粒度比较细，在慢速网络上效率一般也不能令人满意。^①

由于多重网格法兼顾了适用性和速度，其并行化是近年来一个研究热点。主要难点在于多重网格方法实际上是一种“半”全局方法，粗网格的并行很容易导致小数据量通讯问题。近年来的一些研究利用了网格分解映射和鬼网格技术^[136]，可以达到一定的并行效率。但这种方法普遍来说都非常复杂，而且，即使使用多重网格方法，其基准效率通常也无法和快速傅里叶方法相比。

回到快速Poisson求解器，其中包含两种主要的计算步骤：傅里叶变换和三对角方程求解。可以通过对这两个算法的分布并行化实现求解器的并行。快速傅里叶变换的并行算法已经有很成熟的结果，但其并行效率通常不高^[118]，于是问题归结为分布式三对角方程的求解。后者也有很多结果，其中，最重要的是Wang的带状矩阵分裂算法及其变形^{[137][138]}，利用这类分裂算法，可以提供理论上8/17的并行效率。主要的缺陷是在这个算法中仍然需要多对一通讯（虽然通讯量已经大大降低了）。因为三对角方程求解的极端重要性，这一方法一直在被改进中。

一个有趣的思路是使用软件流水线方法求解三对角方程^{[57][139]}。这种思想本质上并不是新的，但由于历史的问题长期被忽略，直到近期才被用于Poisson方程求解中。首先考虑三对角方程组的Tomas消去法^[140]：

Listing 6.1 sample TDMA code

```

1  SUBROUTINE tridiag (a, b, c, r, u, n)
2
3  ! ..... omitted the variable declaration
4
5      ! first stage
6      bet=b(1)
7      u(1)=r(1)/bet
8      do j=2,n
9          gam(j)=c(j-1)/bet
10         bet=b(j)-a(j)*gam(j)
11         u(j)=(r(j)-a(j)*u(j-1))/bet

```

① 参考第三章脚注1

```

12     enddo
13     !second stage
14     do j=n-1,1,-1
15         u(j)=u(j)-gam(j+1)*u(j+1)
16     enddo
17 END subroutine

```

这算法可以看成两趟求解方法的组合，首先是从 u_1 递推到 u_n ，然后再从 u_n 到 u_1 。每一步递推都是精确地前后相继的，或者说在计算第 $j+1$ 个值的时候，必须已知第 j 个值。当存在 $m \gg 1$ 个三对角方程组需要求解的时候，需要连续运用这个算法若干次，而这个过程可以用流水线的方法实现：

假设 m 非常大，于是需要求解许多个 u_{ij} , $i = 1, \dots, m$ and $j = 1, \dots, n$ 。对于每个 i ，都需要求解一个三对角方程组。把 n 分解成为 l 段，每段长为 k ，这样第一段是 $u_{i1}, u_{i2}, \dots, u_{ik}$ ，第二段为 $u_{i,k+1}, u_{i,k+2}, \dots, u_{i,2k}$ ，依此类推。用 l 个进程来进行计算，每个进程拥有对应段的 u 和 r 。

把上述Tomas算法拆成 l 段来进行，对于第一次递推，每一段从 $u_{i,jk}$ 递推到 $u_{i,(j+1)k}$ 。当程序刚开始执行的时候，第一个进程首先计算 $u_{1,1}$ 到 $u_{1,k}$ ，然后将 $u_{1,k}$ 和 bet_k 发送到第二个进程，自己再回头开始从 $u_{2,1}$ 到 $u_{2,k}$ 的递推过程。如此类推，每当一个进程算完了对应一个 j 的递推后，就把算出的 $u_{i,(j+1)k}$ 和 bet_k 传递到 $j+1$ 段对应的进程。当第一次递推全部完毕后，从第 l 个进程开始反向执行第二次递推。

整个执行顺序可以用下面的执行状态表6.2描述: (let the iteration time on one processors is t_c , time for send data to next processor is t_s)

processor	1st	2nd	3rd	...
time $t_c + t_s$	calculated $u_{1,k}$ and sent	received u_{1k}	blocked	...
time $2(t_c + t_s)$	calculated $u_{2,k}$ and sent	calculated $u_{1,2k}$ and sent, received $u_{2,k}$	received $u_{1,2k}$...
time $3(t_c + t_s)$	calculated $u_{3,k}$ and sent	calculated $u_{2,2k}$ and sent, receive $u_{3,k}$	calculated $u_{1,3k}$ and sent, received $u_{2,2k}$...
...

表 6.2 流水线时序

回代操作的时序与之完全相反。总的执行时间 $(l + m - 1)(t_s + t_c)$ ，而串行执行的时间是 $m * l * t_c$ ，于是并行效率是 $\frac{m * t_c}{(l + m - 1)(t_s + t_c)} \rightarrow \frac{t_c}{t_s + t_c}$ 。此外，容易看到这个算

法的通信开销是 $2m(l-1)$ 个浮点数，而且全部属于一对一通讯（除此之外，对于合适的硬件， T_s 和 T_c 可以重合，因此理论效率能够达到100%）。为了避免过多小数据量通讯导致的延迟问题，可以把 i 分组，每个进程在一次计算中多产生一些 u 和 bet 值，然后一次发送给下一个进程。整个算法如2。

Algorithm 2 Pipeline Algorithm for Tridiagonal Solver

```

1:  $j \leftarrow 1$ 
2: while  $j < m$  do
3:    $l \leftarrow \min(m, j + gpm - 1)$ 
4:   call receive_from_left(leftin)
5:    $k \leftarrow j$ 
6:   while  $k < l$  do
7:     call iteration_section(leftin(k),rightout(k))
8:      $k \leftarrow k + 1$ 
9:   end while
10:  call send_to_right(rightout)
11:   $j \leftarrow j + gpm$ 
12: end while

```

在这个计算过程中，需要把Tomas算法分裂成段落来处理，下面是对第一趟迭代的分裂算法例子：

Listing 6.2 sample of the splitting of TDMA code

```

1 subroutine mid_stage1(a,b,c,r,u,n,cin,&
2   betin ,uin ,cout ,betout ,uout ,gam)
3   ! ..... omitted the variable declaration
4   bet=betin
5   gam(1)=cin/bet
6   bet=b(1)-a(1)*gam(1)
7   u(1)=(r(1)-a(1)*uin)/bet
8   do j=2,n
9     gam(j)=c(j-1)/bet
10    bet=b(j)-a(j)*gam(j)
11    u(j)=(r(j)-a(j)*u(j-1))/bet
12  enddo
13  betout=bet

```

```
14     uout=u(n)
15     cout=c(n)
16
17     end subroutine
```

由于除了快速Poisson求解器之外，ADI方法^[62]和半粗化多重网格方法^[66]都使用大量的三对角方程求解，并且如果只在三对角求解方向分割计算区域，则另外一个（两个）方向的计算将是独立的，因此上述方法可以毫无困难的应用到这些方法中。

这种利用软件流水线方法进行三对角方程求解的并行方法，是大规模并行计算中一种新的尝试。在微处理器设计中，流水线方法是基本的内容，而从逻辑上将并行机映射为一台“超级”处理器以使用微处理器设计中的方法，是一种很有潜力的发展方向。

6.4 GPU处理和其他硬件体系

近年来PC技术的进展之一是具有通用计算能力的GPU的出现。由于3D显示技术的发展，需要进行的计算越来越复杂，许多相关工作被转移到图形卡上实现，于是产生了功能越来越强大的图形处理器(GPU)^[141]。目前，GPU的浮点峰值性能已经远远超过CPU，为了使这种计算能力在除了3D图形之外的领域得到应用，发展出了各种具有通用计算能力的GPU和编程接口。

GPU用于除了3D图形之外的其他浮点密集型任务的主要困难在于提供一个易于使用的编程模型。为了达到非常高的峰值，GPU普遍使用了高带宽多核心的结构。相比CPU来说，GPU具有非常多的计算单元，而每个单元的计算能力和复杂性都明显低于CPU内核，必须靠多线程的并行执行才能发挥其峰值性能。由于需要执行的线程数非常多，现代GPU厂商通常会为程序员提供专用的编程模型，例如流式编程或者轻量级进程模型等等。目前，最常用的编程模型是Nvidia提供的CUDA^[142]，这个模型允许用户将程序实现为大量的轻量级进程（线程），在进程之间的通信依靠同步指令和共享内存来实现。

目前已经有很多计算代码被移植到GPU上，特别是分子动力学和生物信息学相关的代码。PIC代码和分子动力学代码非常相似，因此也有研究者在进行PIC代码的GPU移植工作。对于PIC代码来说，主要的困难在于两个：一

是PIC代码涉及到非常大的存储量，常常会超过图形卡所配备的内存；二是图形存储器访问的效率问题。

图形存储器的访问是移植PIC代码中面对的最大问题^[143]。第一个问题是共享冲突：当进行电荷和电流的weighting时，可能会有来自不同线程的粒子电荷被Weighting到同一个网格的情况出现。这种情况下，如果不提供对目标数据的锁定，就会导致熟知的“Dirty Write”错误。问题在于，典型的GPU编程模型（例如CUDA）并不提供高效率的原子化机制。为了解决这个问题，可以使用前述的“Pull Mode”将粒子分配到网格，然后每个线程负责若干个网格。

另一个问题，是GPU的内存访问机制中存在一些和数据对齐相关的问题。通常GPU的主要存储器是不使用Cache的，并且其峰值带宽是在多个线程按照一定规则同步访问时才能获得。例如，在早期的CUDA中，只有第k个线程的访问正好是一段内存的连续第K个字时，这组线程才能达到最高效率。才能达到最高的访问效率。为了解决这个问题，常常需要非常小心地设计数据结构，并且使用前述的粒子排序。

由于有上述的问题，PIC代码的GPU移植还有很多工作需要做。此外，大部分GPU只有对于单精度计算才能达到最高性能，而PIC中部分算法必须适当修改才能保证在单精度计算中收敛。这也是限制PIC代码移植的问题之一。

除了GPU加速以外，还可以使用其他一些方法达到足够的性能，如Cell甚至FPGA定制等等。这类方法的共同问题，都是直接依赖于目标机器的体系结构，并且往往需要在数据结构上从头设计程序，于是不具备任何移植可能。

7 实例和面向问题的建模

7.1 基本的PIC模拟程序

从上世纪90年代开始，已经有很多研究组开发了适用于各种情况的PIC模拟程序。到上世纪末期，主要的显格式模拟程序普遍转向了三维并行化和面向对象程序设计框架。由于显格式电磁PIC模拟的主要技术已经相当成熟，因此和一般的数值模拟软件的发展不同，像并行化和三维代码，图形化输出，前后处理等方面的技术含量并不多。即使对于较大规模(> 1000进程)的并行，实践中也没有发现未预测到的技术困难。这样，主要的技术侧重点，是程序的可扩展性和引入较新的算法。

由于这种原因，PIC程序公布源代码的并不多。2D的XOOPIC^[144]，3D的TRISTAN^[3]和Celeste3D^[86]是少数开源代码的例子。此外，部分模拟程序曾经提供过测试代码，如Vorpal等。不过对研究来说，公布源代码的压力也不大：即使获得了源代码，真正会分析和研究其内容的研究人员也是非常少的（可参考VORPAL的情况，尽管2004年就已经提供了测试版源代码，但对其进行分析研究的很少）。相反，公布使用的技术（算法进展）是非常重要的，因为对于具体问题，其模拟结果的可靠性，非常强烈地依赖于使用的算法是否适用于这个问题。此外，许多PIC算法的进展也会在相应程序的更新中发布。通常，一个PIC程序，如果其使用的主要算法已经被公布，就可以视为是基本公布的。

近10年来发展最快的是3维电磁显式相对论PIC程序，特别是一些用于激光等离子体相互作用研究的程序，如UPIC^[128]，VLPL^[50]，OSIRIS^[109]，VORPAL^[110]等。出于方便性和可维护性的考虑，这类代码普遍使用通过面向对象方式设计，将程序对象实现为各种类结构，于是可以对各种不同性质的问题使用类似的抽象算法，同时隐藏其具体实现。OSIRIS基本上使用的是FORTRAN 90加若干OOP扩展，UPIC使用了FORTRAN 95，而VLPL则用C++实现。VORPAL的做法尤其激进，使用了大量的模板和泛型编程技术，从而可以得到更大程度的可扩展和可维护性。在VORPAL中，粒子和场的模型都是可以

修改和替换的，各种不同的粒子、场模型也可以在不修改程序框架的情况下组合起来。于是，可以很容易地引入新的算法实现。Decyk近年来也在努力发展代码引入更多的设计模式概念^[145]。由于大规模并行计算机系统硬件的问题，代码的诊断和前后处理一般被作为独立的工具实现，这方面也有一些开放的实现。

随着上述程序设计方法的进步，各种较新的技术也可以实现在PIC代码中。配对碰撞模型和PIC-MCC方法在OSIRIS和VORPAL中都已经广泛地引入，VORPAL还引入了R-K甚至DSMC推动代码。场算法的引入要少一些，VLPL首先注意到电磁场色散关系对于高能电子束运动的影响，并引入了NDFX (Numerical Dispersion Free in X direction) 算法。Vorpall 在2004年之后引入了4阶的电磁场差分方法。由于高阶电磁场差分对网格分割方式有一定的影响，很多开发者对此兴趣缺缺。但随着LWFA和PBFA的发展，这一问题会变得越来越严重。近来，VORPAL引入电磁场时间隐式差分求解用来克服CFL限制，用于处理低频电磁问题。

在大规模并行方面，多种PIC代码都得到了验证^{[146][147]}。除了UPIC的FFT算法在非常高的并行路数下出现了预计的性能饱和以外，OSIRIS和VORPAL表现都符合估计。考虑到这两种程序在模拟中并未使用不常见的算法，可以估计显式电磁PIC模拟的并行并不存在严重的技术困难。对于扩展到10000进程以上的模拟，FFT等全局求解器的并行需要特殊处理，如使用前述的流水线方法。至于100,000进程以上的模拟，由于此时区域分割过细，具体性能还无法预料。

除了上述标准的PIC程序之外，也存在一些使用上述各种简化或改进型算法的PIC程序，如使用有质动力导向中心近似的TurboWake和使用boost frame的Aladyn等。

在高功率微波器件和直线加速器方面，也存在很多模拟程序，并且很多已经商业化。由于尺寸问题，直线加速器的模拟一直是这方面具有挑战性的问题之一。

射频放电模拟的情况完全不同。直到不久以前，2D的XOOPIC还是常用的放电模拟程序。这部分是因为放电问题的MCC模型本身就晚于PIC技术，部分是因为包含碰撞模型时巨大的粒子数。此外，如同前面指出的，静电问题的难

以并行化也带来了许多困难。即使 VORPAL，在静电模型中仍然使用的是 Aztec (in Trilinos) 函数库。此外，商业上的问题也限制了专业放电模拟程序的推广。

隐式模拟方面，主要的模拟程序包括 XPDP2^[70]，LSP^[81]和 Celeste3D^[86]。其中 XPDP2 和 Celeste 3D 已经公开了源代码，分别用于放电过程和电离层的模拟。而 LSP 则是商业软件，用于重离子聚变问题的研究。和静电模型的情况类似，隐格式的并行化目前还存在很大的困难。此外，LANL 和 LLNL 对这方面的支持减少也导致了发展上的困难。

7.2 非结构网格上的放电模拟

低气压下的放电模拟具有很多工程应用价值，特别是射频场驱动的气体放电等离子体源，其数值模拟技术在微电子制造和材料处理中都具有很多应用。而对于这类问题，最有效的模拟手段是 PIC/MCC 技术。由于研究者的努力，这种模拟的物理背景已经研究得比较充分，遗留的主要困难出在算法方面。

第一个问题是放电模拟中鞘层行为常常是非常重要的。于是，时空尺度困难会变得非常突出。此外，鞘层总是位于等离子体和电极接触面上，电极形状决定了鞘的形状。而工程中大部分设备的形状是不规则的。

另一个问题是，气体放电通常是靠碰撞或者二次电子发射维持的。而大部分气体都具有非常复杂的碰撞截面-能量关系，特别在低能（零点几到几个电子伏特）区。因此，这种 PIC/MCC 模拟对能谱结构相当敏感，同时，典型情况下，碰撞率很低，意味着必须用相当多的模拟宏粒子才能产生出比较可信的 MC 碰撞模拟。在典型的放电模拟中，通常需要每个网格数百个宏粒子来抑制散粒噪声带来的扰动。一般来说，在气体放电的 PIC/MCC 模拟中，即使二维的问题，也需要用并行机才能进行有意义的模拟。

第一个问题的解决需要使用不规则的网格，甚至是非结构化的网格。非结构化网格技术在计算流体力学中已经是非常常见的技术了，但在 PIC 模拟中，还存在很多的算法问题。最主要的问题出在不规则网格上的粒子推动常常会包含“自力”。幸运的是放电模拟通常是静电模型的，可以设置电荷累积和场插值使用相同的插值函数，这时，自力问题会被大大缓解。

非结构四面体网格上的三维静电模型放电模拟已经有了一些程序实现，

主要包括Gatsonis^[148]和J.S.Wu^{[149][150]}的工作。从物理角度，模拟的过程和一般PIC方法并无区别，但由于网格的特殊性，每一步的具体算法都需要设计。我们将按照Spirkin的实现介绍和分析一下主要的实现算法。

对于给定的问题，生成非结构四面体网格的技术在弹性力学和计算流体力学中是熟知的，可以直接移植过来。这样，需要实现的算法包括：Weighting/场插值，场方程的建模和求解，粒子推动。

CIC形式的Weighting和插值可以通过基本的考虑给出。考虑一个四面体，四个顶点为 $\{r_1, r_2, r_3, r_4\}$ ，待插值的点为 r_p ，所需要的插值函数是 $W_j(r_p)$ ，那么，按照CIC的精神，有

$$W_1 r_p(x_p - x_1) + W_2 r_p(x_p - x_2) + W_3 r_p(x_p - x_3) + W_4 r_p(x_p - x_4) = 0 \quad (7-1)$$

$$W_1 r_p(y_p - y_1) + W_2 r_p(y_p - y_2) + W_3 r_p(y_p - y_3) + W_4 r_p(y_p - y_4) = 0 \quad (7-2)$$

$$W_1 r_p(z_p - z_1) + W_2 r_p(z_p - z_2) + W_3 r_p(z_p - z_3) + W_4 r_p(z_p - z_4) = 0 \quad (7-3)$$

解出这个方程给出

$$W_1(r_p) = \frac{\Omega_{p234}}{\Omega_{1234}} \quad (7-4)$$

$$W_2(r_p) = \frac{\Omega_{p134}}{\Omega_{1234}} \quad (7-5)$$

$$W_3(r_p) = \frac{\Omega_{p124}}{\Omega_{1234}} \quad (7-6)$$

$$W_4(r_p) = \frac{\Omega_{p123}}{\Omega_{1234}} \quad (7-7)$$

其中 Ω_{pijk} 是 r_p 和 r_i, r_j, r_k 三个点围成的四面体体积，而 Ω_{1234} 是整个四面体的体积。

在四面体网格上，Poisson方程可以用有限体积法或者有限元方法离散化。有限体积法的离散相当简单，对于任何一个节点 p ，连接所有邻点 i 得到一组从 p 点出发的有向线段，而这些线段中点法平面围成的多面体就是这一节点的体积 V_p ；多面体的电场通量是

$$\sum_i \frac{\Phi_i - \Phi_p}{\Delta_{pi}} S_i = \frac{1}{\epsilon_0} \rho_p V_p$$

求解这个稀疏矩阵方程可以得出电势分布，注意到上述方程组实际上已经确定

了多面体的各个面上的法向电场，即相当于知道 $\vec{E} \cdot \vec{r}_i$ 的近似值，于是对于 \vec{E} 的估计可以用最小二乘法获得。这只要求解一个 m 阶线性方程组， m 是每个节点的邻点数。

粒子的推动是最大的问题所在。在上面的计算中已经可以得到直角坐标系下每个节点的电磁场，利用插值公式给出粒子所在点的电磁场，然后可以在直角坐标系下面计算粒子的推动。这种方法唯一需要额外处理的是确定粒子处于哪个网格中。由于网格的非结构性，像标准均匀网格那样快速计算粒子所在网格是不现实的。为此，需要记录粒子所在网格，当粒子被推动一步后，计算粒子与该网格顶点构成的四面体的面积。当四个四面体体积相加等于原来四面体体积的时候，说明这个粒子处于原来的网格之内，否则，需要搜索该网格的邻居确认粒子所在网格。为了简化计算，通常直接计算粒子运动轨迹和四面体表面的交点。

尽管这些代码所涉及到技术都是比较成熟的，但很容易看到，相关的计算量远远超过了规则网格上的PIC代码。此外，由于场方程需要用共轭梯度或者类似的迭代法求解，不仅速度非常慢，其并行化也相当困难，并且其并行加速比尚不能完全令人满意。

7.3 QuickPIC

高能带电粒子束和稀薄等离子体的相互作用是加速器物理中的重要问题之一。理论和实验都表明，这是等离子体中可以激发出大振幅的静电振荡，形成很强的纵向电场，并导致极高能量粒子射束的形成，这就是所谓的等离子体尾场加速(Plasma Wake Field Accelerator, PWFA)。类似地，也可以用超快超强激光驱动等离子体振荡，这就是激光尾场加速器(LWFA)。

这种问题的模拟，主要困难在两个方面，一是问题通常是强相对论的，因此粒子的运动需要高精度和快速的推进算法；另一个问题是时空尺度。在典型的问题中，束流需要在等离子体中传播厘米量级的距离，同时，即使使用移动窗口，计算区域仍然要达到一百微米左右的尺度，而等离子体密度大约在 10^{17}cm^{-3} 的水平，三维下这意味着 10^8 个网格和 10^6 个时间步。在目前，这种模拟还是相当困难的。

如同前面分析的那样，这时的模拟需要使用各种强有力的技巧来简化问题，例如，在激光加速问题中，往往可以用有质动力近似来化简问题。前面讨论的“Boost Frame”也是一种很有效的手段。除此之外，从上个世纪中期开始，解析研究中使用的准静态近似(Quasi-Static Approximation)也被引入PIC模拟中，产生了如Wake^[151]和QuickPIC^[71]等若干约化的PIC模拟代码。其中，QuickPIC是这类约化建模的成功代表。

考虑移动坐标框架 $\xi = ct - z, s = z$ ，对于PWFA和LWFA问题，静电振荡是由高能束流驱动，假设束流能量很高且等离子体很稀薄，那么在驱动束流向前运动的同时，束流本身的变化是由于其能量被传递给了等离子体，束流形态的变化远远慢于驱动出来的振荡。利用这种近似，使用Lorentz规范，可以写出电磁场方程为

$$-\nabla_{\perp}^2 \phi(x, y, s, \xi) = -\frac{1}{\epsilon_0} \rho(x, y, z, \xi) \quad (7-8)$$

$$-\nabla_{\perp}^2 A(x, y, s, \xi) = -\mu_0 J(x, y, z, \xi) \quad (7-9)$$

$$\nabla_{\perp} \cdot A_{\perp}(x, y, s, \xi) = -\frac{\partial}{\partial \xi} (\phi - A_z) = -\frac{\partial \psi}{\partial \xi} \quad (7-10)$$

分别对束流粒子和背景粒子使用准静态近似可以写出其运动方程，对于束流粒子是

$$\frac{dP_{\perp}}{ds} = -q \nabla_{\perp} \psi \quad (7-11)$$

$$\frac{dP_z}{ds} = q \frac{\partial \psi}{\partial \xi} \quad (7-12)$$

对于背景是

$$\frac{dS}{d\xi} = \frac{1}{1 - v/c} \quad (7-13)$$

$$\frac{dP_{\perp}}{d\xi} = \frac{q}{1 - \beta_b} \left[E_{\perp} + \left(\frac{v}{c} B \right)_{\perp} \right] \quad (7-14)$$

$$\gamma_p - P_z/mc = 1 - \frac{q\psi}{mc^2} \quad (7-15)$$

其中 $\gamma_p = \frac{1}{\sqrt{1 - \beta_p^2}}$, β_p 是束流的速度，S和P分别是粒子的纵向位置和动量。

计算电荷和电流密度的时候，需要使用运动框架下的形式，即对于一个静止条件下电荷为q的宏粒子，当它以纵向速度为 v_{zi} 运动时，有效电荷

是 $Q_i = \frac{q}{1-v_i/c}$ ，有效电流是 $Q_i V_i$ 。

上述基本方程可以按照PIC的逻辑结构组成有效的模拟程序。主要的困难在于 J_{\perp} 和 A_{\perp} 的关系并不是局域的：粒子运动方程必须在每一个 ξ 点上得到满足，于是各点的场会耦合起来。这是前面提到的因果律问题的一个例子。为此，这组方程必须以迭代方式求解。在实践中，一组预测-校正过程用来处理粒子群沿着 ξ 方向的“演化”，通常只要迭代2-4次就可以给出基本满意的结果。

由于使用了准静态近似，模拟可以使用更大的时间步长和比较少的粒子。计算表明，Quick PIC准静态模拟在PBFA问题中可以使问题的规模降低几个数量级，同时并不带来过大的误差。主要的缺陷在于因为忽略了粒子的细致振荡，无法有效地模拟注入过程，对于LWFA问题尤其如此。

最近期的一个进展是对于Quick PIC的并行化。由于PWFA模拟的计算量过于庞大，即使使用QuickPIC也需要并行处理。Quick PIC的场方程基本是个Poisson方程，其高路数并行并不容易。但如果把计算区域沿着 ξ 方向划分，那么可以注意到模拟过程是顺序过程：对于任何一个 ξ 平面，需要从 $\xi - \Delta\xi$ 取得场信息，然后求解本平面上的场，然后推动本平面上的粒子，累积本平面的电流和电荷密度，最后再将场信息传递给 $\xi + \Delta\xi$ 平面。当所有 ξ 平面的粒子都被推动后，一个计算循环完成并进行下一个。这个计算可以用流水线方式完成^[152]：将计算进程分组，每个组负责一些 ξ 平面上的计算，这可以用任何并行Poisson solver完成；组之间使用流水线方式并行。这种并行手法可以有效地实现上万进程的并行。

8 结论：问题和展望

本文概述了PIC模拟方法的基本框架和近年来的主要进展，分析了各种数学和算法优化技巧及其适用性。随着等离子体研究的进展，多维PIC模拟方法正在不断地扩展其应用领域，并且开始进入传统上较少应用PIC模拟方法的一些领域。在这种情况下，PIC模拟方法的一些固有问题表现得更加尖锐起来。目前急需解决的主要问题包括下面几类：

(1) 时空尺度问题，也就是如何模拟大尺度和稠密等离子体的问题。这个问题在PIC模拟中是普遍存在的，但对于不同的物理问题，表现形式有所不同。

对于放电和材料表面处理问题，由于自加热问题对于能谱影响很大，空间网格的德拜长度限制是主要的困难所在。这方面，隐格式和能量守恒格式都可以作为解决问题的变通手段。其中，能量守恒格式设计最为简单，但从我们实测结果来看，这个格式对鞘层行为的描述并不可靠，因此对于鞘层行为主导的问题，其应用可能会出现问題。对于鞘层行为影响不大的问题，能量守恒格式则可以发挥作用（如ECR和霍尔发动机）。隐格式则会导致等离子体的自冷却，而且隐格式的场求解器设计相当困难。隐格式的主要优点是可以使用任意的时间和空间步长，因此对于鞘层占优问题，可以考虑使用隐格式的多尺度模拟来处理，这样在鞘层部分可以使用 $\Delta x < \lambda_D$ ，而在其他部分使用隐格式的宽网格。目前，国外已经有一些隐格式用于放电问题（包括CCP和Hall Thruster）的研究，至于能量守恒格式的应用刚刚开始，还需要时间来验证。国内尚很少这方面的研究工作。

至于这类问题的显格式多维模拟，常常需要用超级计算机才可能完成。主要困难在于并行化，特别是静电模式模拟的并行化。传统的矩阵转置并行Poisson求解或者使用并行库的求解是很难有实际意义的。各种快速求解器的并行化还需要更多的数学进展。

对于强激光和晕区等离子体的相互作用问题，目前的主要问题在于过长的时间尺度。但随着Boost Frame技术的发展，利用Lorentz变换，这问题有可能得到较好的解决。尽管如此，由于这类问题的三维模型和两维模型在物理上有根

本的区别，最终的解决仍然需要三维研究，这通常意味着超级计算机。

最后，一个很特殊的问题是强激光和高密度等离子体（甚至固体）的相互作用，快点火问题也属于这一类。由于此时存在等离子体频率远远大于激光频率的高密度区，根本的解决手段依赖于相对论的隐格式模拟技术。目前，直接隐式方法和矩方法都被推广到了相对论区，但其可靠性还需要验证，而且三维下隐格式的电磁场求解器和粒子运动推进器性能也存在很大的问题。另外，基于电磁波包络近似和高度并行化的隐格式模拟器也需要更多的发展。但无论如何，并行的隐格式模拟是现阶段看来唯一可行的解决方案，其他路线现在看不到走通希望。

(2)复杂几何形状和背景介质性质的问题。这个问题主要在放电等离子体和微波器件问题中较为突出，因为实际工程中的设备形状可能非常复杂（包括探针导致的扰动），常常需要使用非均匀甚至非结构化网格分割，这就带来了大量的问题，如网格分界面上的虚假反射，非结构化网格的场求解，粒子跟踪，粒子的分裂合并等等。尽管这些问题现在都有对应的解决方案，但都容易导致性能的严重退化（例如，粒子跟踪算法会成量级地降低PIC推动代码的性能），而性能下降对于PIC模拟带来的影响是致命的。

一个变通的手段是使用浸没方法，即仍然使用标准的直角坐标网格，然后在求解电场的时候使用浸没有限元方法。由于直角坐标网格的元胞是立方体，可以简单剖分成四面体，因此直接套用有限元方法没有实质性的困难；对于边界不规则区域则使用有限元方法按照边界和元胞的相交形状进行离散。这一手法兼顾了粒子追踪和边界处理，是一种不错的折中手段。

(3)碰撞和电离问题。这个问题就其物理和数学方面来说，已经有了很成熟的解决方法，但是和复杂几何问题一样，其解决是靠降低PIC模拟方法的性能获得的。特别是两体库仑碰撞代码传统上依赖于粒子的排序，这是非常耗费时间的。Tskhakaya的工作表明使用粒子按照网格分配的“Pull Mode”大约可以提升一半左右的代码性能，这是个令人鼓舞的消息，但还需要更多的验证和改进，才能使得碰撞模拟的效率达到可接受的水平。由于涉及到电离的情况下，每个网格的粒子数至少要上百才能充分抑制随机噪声导致的加热，如何削减粒子数同时抑制随机噪声也是未来需要解决的问题。这类问题，在激光与重原子靶作用的研究中尤为重要。

(4)诊断和验证问题。通常情况下，对于PIC模拟过程，能量和动量的守恒性是用来验证模拟可靠性的基本手段。但如同在Particle Rezone和能量守恒格式中看到的那样，单纯的物理量守恒，往往不如相空间的分布函数和轨道形态重要。那么，对于具体物理问题，应该选择什么参数，才能判断模拟使用的算法是否合理？

所有这些问题都需要进一步的研究来解决。目前看来，隐格式，“Pull Mode”和复杂网格问题具有中心的地位。解决这些问题，将使得PIC模拟方法从重要的理论研究工具发展为工程上的核心工具（如同在电子电路CAD领域中的情况）。

同时，即使物理研究方面，PIC方法仍然有其重要的未知领域，即极端条件下的带电粒子运动问题。随着现代激光技术的发展，光场中的电子颤动动量可以高达 $50m_e c$ 甚至更多，这种强度下，宏观电磁规律尚有很多不确定的地方（如辐射阻尼等），而PIC作为研究高能量带电粒子集体运动规律的基本手段，可以为这方面的研究提供重要的依据。此外，PIC方法是经典的，对于广义相对论和量子力学主宰的带电粒子运动，如何建立类似的模拟手段，也是未来可能需要解决的问题。

参考文献

- [1] Birdsall C K, Landon A B. Plasma Physics via Computer Simulation. Longdon: Institute of Physics Publishing, 1991.
- [2] Matsumoto H, Sato T. Computer Simulation of Space Plasmas. tyoko: Terra Scientific Publishing Company, 1985.
- [3] Matsumoto H, Omura Y. Computer Space Plasma Physics: Simulation Techniques and Software. tokyo: Terra Scientific Publishing Company, 1993.
- [4] Usui H, Omura Y. Advanced Methods for Space Simulations. Tokyo: Terra Scientific Publishing Company, 2007.
- [5] Buchner J, Dum C T, Scholer M. Space Plasma Simulation. Lecture Notes in Physics, Berlin: Springer-Verlag, 2003.
- [6] Hockney R W, Eastwood J W. Computer Simulation Using Particles. New York: Adam Hilger, 1988.
- [7] 傅竹风, 胡友秋. 空间等离子体数值模拟. 合肥: 安徽科技出版社, 1995.
- [8] Pukhov A. Strong field interaction of laser radiation. Reports of Progress in Physics, 2003, 66:47.
- [9] Birdsall C K. Particle-in-Cell Charged-Particle Simulations, Plus Monte Carlo Collisions With Neutral Atoms, PIC-MCC. IEEE TRANSACTIONS ON PLASMA SCIENCE, 1991, 19(2):65.
- [10] Goplen B, Ludeking L, Smithe D, et al. ser-configurable MAGIC for electromagnetic PIC calculations. Computer Physics Communications, 1995, 87(1/2):54.
- [11] Zhou J, Liu D G, Liao C, et al. CHIPIC: An Efficient Code for Electromagnetic PIC Modeling and Simulation. IEEE TRANSACTIONS ON PLASMA SCIENCE, 2009, 37(10):2002.
- [12] Wang J G, Zhang D H, Liu C L, et al. UNIPIC code for simulations of high power microwave devices. PHYSICS OF PLASMAS, 2009, 16:033108.
- [13] Qiang J, Ryne R D, Habib S, et al. An Object-Oriented Parallel Particle-in-Cell Code for Beam Dynamics Simulation in Linear Accelerators. Journal of Computational Physics, 2000, 163:434.
- [14] Lieberman M A, Lichtenberg A J. Principles of Plasma Discharges and Materials Processing. second ed., 2005.
- [15] Umstadter D. Review of physics and applications of relativistic plasmas driven by ultra-intense lasers. PHYSICS OF PLASMAS, 2001, 8(5):1774.

-
- [16] Mourou G A, Tajima T, Bulanov S V. Optics in the relativistic regime. *REVIEWS OF MODERN PHYSICS*, 2006, 78:309.
- [17] Matyash K, Schneide R, Taccogna E, et al. Particle in Cell Simulation of Low Temperature Laboratory Plasmas. *Contribute on Plasma Physics*, 2007, 47(8-9):595.
- [18] Sentoku Y, Kemp A J. Numerical methods for particle simulations at extreme densities and temperatures: Weighted particles relativistic collisions and reduced currents. *Journal of Computational Physics*, 2008, 227:6846.
- [19] Gordon D F, Mori W B, Antonsen T M. A Ponderomotive Guiding Center Particle-in-Cell Code for Efficient Modeling of Laser Plasma Interactions. *IEEE TRANSACTIONS ON PLASMA SCIENCE*, 2000, 28(4):1135.
- [20] Messmer P, Bruhwiler D L. Simulating laser pulse propagation and low-frequency wave emission in capillary plasma channel systems with a ponderomotive guiding center model. *PHYSICAL REVIEW SPECIAL TOPICS - ACCELERATORS AND BEAMS*, 2006, 9:031302.
- [21] Lapenta G. The algorithms of the implicit method, 2008. arxiv:0801.4134v1.
- [22] Brackbill J U, Cohen B I. *Multiple Time Scales*. Orlando: Academic Press, 1985.
- [23] Vay J L. Simulation of beams or plasmas crossing at relativistic velocity. *PHYSICS OF PLASMAS*, 2008, 15:056701.
- [24] Verboncoeur J P. Particle simulation of plasmas: review and advances. *Plasma Physics and Controlled Fusion*, 2005, 47:A231.
- [25] Tskhakaya D, Matyash K, Schneider R, et al. The Particle-In-Cell Method. *Contribute in Plasma Physics*, 2007, 47(8-9):563.
- [26] ABE H, SAKAIRI N, ITATANI R. High-Order Spline Interpolations in the Particle Simulation. *JOURNAL OF COMPUTATIONAL PHYSICS*, 1986, 63:247.
- [27] Lewis H R. Energy-Conserving Numerical Approximations for Vlasov Plasmas. *Journal of Computational Physics*, 1970, 6:136.
- [28] Pointon T D. Second-order, exact charge conservation for electromagnetic particle-in-cell simulation in complex geometry. *Computer Physics Communications*, 2008, 179:535. Some processes about energy conseving mode of PIC simulation.
- [29] Langdon A B. Energy Conserving Plasma Simulation Algorithms. *Journal of Computational Physics*, 1973, 12:247.
- [30] Lapenta G. Automatic Adaptive Multi-Dimensional Particle In Cell. *Proceedings of Advanced Methods for Space Simulations*. Tyoko: Terra Scientific Publishing Company, 2007:.
- [31] Soria-Hoyo C, Pontiga F, Castellanos A. Two dimensional numerical simulation of gas discharges: comparison between particle-in-cell and FCT techniques. *JOURNAL OF PHYSICS D: APPLIED PHYSICS*, 2008, 41.

-
- [32] Verboncoeur J P. Symmetric Spline Weighting for Charge and Current Density in Particle Simulation. *Journal of Computational Physics*, 2001, 174:421.
- [33] Vazquez P A, Castellanos A. Weighting of Charge in PIC Codes for Unstructured Meshes in Cylindrical Coordinates: Application to Charged Jets. *Proceedings of 2005 Annual Report Conference on Electrical Insulation and Dielectric Phenomena*, 2005.
- [34] Chanrion O, Neubert T. A PIC-MCC code for simulation of streamer propagation in air. *Journal of Computational Physics*, 2008, 227:7222.
- [35] Takekida H, Nanbu K. Weighting Factor for Particle Modeling of Axisymmetrical Low Temperature Plasmas. *Journal of the Physical Society of Japan*, 2004, 73(3):756.
- [36] Lapenta G. Particle Rezoning for Multidimensional Kinetic Particle-In-Cell Simulations. *Journal of Computational Physics*, 2002, 181:317.
- [37] Yee K S. Numerical Solution of Initial Boundary Value Problems involving Maxwell's Equation in isotropic Media. *IEEE Transactions of Antennas and Propagation*, 1966, 14:302.
- [38] Cai D S, Li Y T, Nishikawa K I, et al. Parallel 3-D Electromagnetic Particle Code Using High Performance FORTRAN: Parallel TRISTAN. In: Büchner J, Dum C T, Scholer M, (eds.). *Proceedings of Space Plasma Simulation*. Berlin: Springer, 2003:.
- [39] I P J M, Verboncoeur J P. Charge conservation in electromagnetic PIC codes; spectral comparison of Boris/DADI and Langdon-Marder methods. *Computer Physics Communications*, 1997, 106:219.
- [40] Villasenor J, Buneman O. Rigorous charge conservation for local electromagnetic field solvers. *Computer Physics Communications*, 1992, 69.
- [41] Eastwood J W. The virtual particle electromagnetic particle-mesh method. *Computer Physics Communications*, 1991, 64:252.
- [42] Esirkepov T Z. Exact charge conservation scheme for Particle-in-Cell simulation with an arbitrary form-factor. *Computer Physics Communications*, 2001, 135:144.
- [43] Umeda T, Omura Y, Tominaga T, et al. A new charge conservation method in electromagnetic particle-in-cell simulations. *Computer Physics Communications*, 2003, 156:73.
- [44] Munz C D, Omnes P, Schneider R, et al. Divergence Correction Techniques for Maxwell Solvers Based on a Hyperbolic Model. *Journal of Computational Physics*, 2000, 161:484.
- [45] Jacobs G B, Hesthaven J S. High-order nodal discontinuous Galerkin particle-in-cell method on unstructured grids. *Journal of Computational Physics*, 2005, 214:96.
- [46] Zheng F H, Chen Z Z, Zhang J Z. Toward the Development of a Three-Dimensional Unconditionally Stable Finite-Difference Time-Domain Method. *IEEE TRANSACTIONS ON MICROWAVE THEORY AND TECHNIQUES*, 2000, 48(9):1550.

-
- [47] Yuan C H, Chen Z Z. A Three-Dimensional Unconditionally Stable ADI - FDTD Method in the Cylindrical Coordinate System. *IEEE TRANSACTIONS ON MICROWAVE THEORY AND TECHNIQUES*, 2002, 50(10):2401.
- [48] Greenwood A D, Cartwright K L, Luginsland J W, et al. On the elimination of numerical Cerenkov radiation in PIC simulations. *Journal of Computational Physics*, 2004, 201:665.
- [49] Shang J S. High-Order Compact-Difference Schemes for Time-Dependent Maxwell Equations. *Journal of Computational Physics*, 1998, 153:312.
- [50] Pukhov A. Three-dimensional electromagnetic relativistic particle-in-cell code VLPL (Virtual Laser Plasma Lab). *Journal of Plasma Physics*, 1999, 61:425.
- [51] Benedetti C, Sgattoni A, Tomassini P. ALaDyn: A HIGH ACCURACY CODE FOR THE LASER-PLASMA INTERACTION. *Proceedings of Proceedings of EPAC08, Genoa, Italy*, 2008. 2794.
- [52] Qiang J, Ryne R D. Parallel 3D Poisson solver for a charged beam in a conducting pipe. *Computer Physics Communications*, 2001, 138:18.
- [53] BUZBEE B L, GOLUB G H, NIELSON C W. ON DIRECT METHODS FOR SOLVING POISSON'S EQUATIONS. *SIAM Journal of Numerical Analysis*, 1970, 7(4):627.
- [54] Banegas A. Fast Poisson Solvers for Problems with Sparsity. *Mathematics of Computation*, 1978, 32(142):441.
- [55] Demmel J W. *Applied Numerical Linear Algebra*. New York: SIAM, 1997.
- [56] Meurant G. *Computer Solution of Large Linear Systems. Studies In Mathematics and Its Applications*, Amsterdam: Elsevier, 1999.
- [57] Wang H Y, Jiang W, Wang Y N. Parallelization and optimization of electrostatic Particle-in-Cell/Monte-Carlo Coupled codes as applied to RF discharges. *Computer Physics Communications*, 2009, 180:1305.
- [58] Lai M C, Lin W W, Wang W C. A fast spectral/difference method without pole conditions for Poisson-type equations in cylindrical and spherical geometries. *IMA Journal of Numerical Analysis*, 2002, 22:537.
- [59] Swarztrauber P N. A DIRECT METHOD FOR THE DISCRETE SOLUTION OF SEPARABLE ELLIPTIC EQUATIONS. *SIAM Journal of Numerical Analysis*, 1974..
- [60] George B L, Fox L A, Gent C, Graves G. The Direct Solution of the Discrete Poisson Equation on Irregular Regions. *SIAM Journal on Numerical Analysis*, 1971, 8(4):722.
- [61] Hewett D W, Larson D J, S D. Solution of Simultaneous Partial Differential Equations Using Dynamic ADI: Solution of the Streamlined Darwin Field Equations. *JOURNAL OF COMPUTATIONAL PHYSICS*, 1992, 101:11.
- [62] Lambert M I A, Rodrigue G H, Hewett D W. A parallel DSDADI method for solution of the steady state diffusion equation. *Parallel Computing*, 1997, 23:2041.

-
- [63] Saad Y. Iterative methods for sparse linear systems. New York: SIAM, 1996.
- [64] Hackbush W. Multigrid Methods and Applications. Berlin: Springer, 1985.
- [65] Trottenberg U, Oosterlee C, Shülle A. Multigrid. New York: Academic Press, 2001.
- [66] Wang H Y, Jiang W, Wang Y N. Implicit and electrostatic Particle-in-cell/Monte Carlo model in two-dimensional and axisymmetric geometry I: analysis of numerical techniques. Plasma Source Science and Technology, 2010, 19:045023.
- [67] SHAPIRA Y. MATRIX-BASED MULTIGRID. Numerical Methods and Algorithms, Berlin: Springer, 2008.
- [68] Takekida H, Nanbu K. Particle modelling of inductively-coupled argon plasmas with wafer biasing. JOURNAL OF PHYSICS D: APPLIED PHYSICS, 2005, 38:3461.
- [69] Verboncoeur J P, Alves M V, Vahedi V, et al. Simultaneous Potential and Circuit Solution for 1D Bounded Plasma Particle Simulation Codes. Journal of Computational Physics, 1993, 104:321.
- [70] Vahedi V, DiPeso G. Simultaneous Potential and Circuit Solution for Two-Dimensional Bounded Plasma Simulation Codes. JOURNAL OF COMPUTATIONAL PHYSICS, 1997, 131:149.
- [71] Huang C, Decyk V K, Ren C, et al. QUICKPIC: A highly efficient particle-in-cell code for modeling wakefield acceleration in plasmas. Journal of Computational Physics, 2006, 217:658.
- [72] Wang H Y, Xie B S. Numerical Simulation of Ultrafast Laser Pulse Propagation in Tenuous Plasmas: Envelope Evolving and Modulation. Communication in Theoretical Physics, 2006, 46:563.
- [73] Hur M S, Wurteleb J S. Two-dimensional simulations of the amplification and focusing of intense laser pulses in the kinetic regime of Raman backward amplification in plasmas. Computer Physics Communications, 2009, 180:651.
- [74] Kawamura E, Lichtenberg A J, Lieberman M A, et al. ADVENTURES IN TWO-DIMENSIONAL PARTICLE-IN-CELL SIMULATIONS OF ELECTRONEGATIVE DISCHARGES, 2009. <http://www.eecs.berkeley.edu/~lieber>.
- [75] Taccogn F, Longo S, Capitelli M, et al. Stationary plasma thruster simulation. Computer Physics Communications, 2004, 164:160.
- [76] Taccogna F, Longo S, Capitelli M, et al. Particle-in-Cell Simulation of Stationary Plasma Thruster. Contribute to Plasma Physics, 2007, 47(8-9):635.
- [77] Cohen B I, Langdon A B, Friedman A. Implicit Time Integration for Plasma Simulation. Journal of Computational Physics, 1982, 46:15.
- [78] Langdon A B, Cohen B I, Friedman A. Direct Implicit Large Time-Step Particle Simulation of Plasma. Journal of Computational Physics, 1983. 107.

-
- [79] Friedman A. A Second-Order Implicit Particle Mover with Adjustable Damping. *Journal of Computational Physics*, 1990, 90:292.
- [80] Hewett D W, Langdon A B. Electromagnetic Direct Implicit Plasma Simulation. *Journal of Computational Physics*, 1987, 72:121.
- [81] Welch D R, Rose D V, Oliver B V, et al. Simulation techniques for heavy ion fusion chamber transport. *Nuclear Instruments and Methods in Physics Research A*, 2001, 464:134.
- [82] Welch D R, Rose D V, Clark R E, et al. Implementation of a non-iterative implicit electromagnetic field solver for dense plasma simulation. *Computer Physics Communications*, 2004, 164:183.
- [83] Drouin M, Gremillet L, Adam J C, et al. Particle-in-Cell modelling of relativistic Laser-Plasma interaction with the adjustable-damping, direct implicit method, 2009. arXiv:0906.2912v1.
- [84] Brackbill J U, Forslund D W. An Implicit Method for Electromagnetic Plasma Simulation in Two Dimensions. *Journal of Computational Physics*, 1982, 46:271.
- [85] Vu H X, Brackbill J U. CELEST1D: an implicit, fully kinetic model for low-frequency, electromagnetic plasma simulation. *Computer Physics Communications*, 1992, 69:253.
- [86] Lapenta G, Brackbill J U, Ricci P. Kinetic approach to microscopic-macroscopic coupling in space and laboratory plasmas. *PHYSICS OF PLASMAS*, 2006, 13:055904.
- [87] Ricci P, Lapenta G, Brackbill J U. A Simplified Implicit Maxwell Solver. *Journal of Computational Physics*, 2002, 183:117.
- [88] Knoll D A, Lapenta G, Brackbill J U. A Multilevel Iterative Field Solver for Implicit, Kinetic, Plasma Simulation. *Journal of Computational Physics*, 1999, 149:377.
- [89] Elman H C, Ernst O G, O'leary D P. A MULTIGRID METHOD ENHANCED BY KRYLOV SUBSPACE ITERATION FOR DISCRETE HELMHOLTZ EQUATIONS. *SIAM Journal of Scientific Computing*, 2001, 23:1291.
- [90] Zhu Y, Cangellaris A C. *Multigrid Finite Element Methods for Electromagnetic Field Modeling*. Hoboken: John Wiley & Sons, Inc., 2006.
- [91] Noguchi K, Tronci C, Zuccaro G, et al. Formulation of the relativistic moment implicit particle-in-cell method. *PHYSICS OF PLASMAS*, 2007, 14:042308.
- [92] Friedman A, Parker S E, Ray S L, et al. Multi-scale Particle-in-Cell Plasma Simulation. *Journal of Computational Physics*, 1991, 96:54.
- [93] Parker S E, Friedman A, Birdsall C K. Bounded Multi-scale Plasma Simulation: Application to Sheath Problems. *Journal of Computational Physics*, 1993, 107:388.
- [94] Vay J L, Colella P, McCorquodale P, et al. Mesh Refinement for Particle-in-cell plasma simulations: Applications to and Benefits for heavy ion fusion. *Laser and Particle Beams*, 2002, 20:569.

-
- [95] Vay J L, Colella P, Kwan J W, et al. Application of adaptive mesh refinement to particle-in-cell simulations of plasmas and beams. *Physics of Plasmas*, 2004, 11(5):2928.
- [96] Fujimoto K, Machida S. Electromagnetic full particle code with adaptive mesh refinement technique: Application to the current sheet evolution. *Journal of Computational Physics*, 2005, 214:550.
- [97] Vay J L, Colella P, Friedman A, et al. Implementations of mesh refinement schemes for Particle-In-Cell plasma simulations. *Computer Physics Communications*, 2004, 164:297.
- [98] Vay J L. An Extended FDTD Scheme for the Wave Equation: Application to Multiscale Electromagnetic Simulation. *Journal of Computational Physics*, 2001, 167:72.
- [99] Vay J L, Adam J C, Héron A. Asymmetric PML for the absorption of waves. Application to mesh refinement in electromagnetic Particle-In-Cell plasma simulations. *Computer Physics Communications*, 2004, 164:171.
- [100] Vahedi V, Surendra M. Monte-Carlo Collision Model for Particle-In-Cell Method: Application to Argon and Oxygen Discharges. *Computational Physics Communication*, 1995, 87:179.
- [101] GEORGIEVA V. Computer Modeling of Low-pressure Fluorocarbon-based Discharges for Etching Purposes[Doctor Thesis]. Antwerpen: University of AntWerpen, 2005.
- [102] Takizuka T, H H A. A binary collision model for plasma simulation with a particle code. *Journal of Computational Physics*, 1977, 25:205.
- [103] Nanbu K, Yonemura S. Weighted Particles in Coulomb Collision Simulations Based on the Theory of a Cumulative Scattering Angle. *JOURNAL OF COMPUTATIONAL PHYSICS*, 1998, 145:639.
- [104] Evans R G. A Binary Collision Model for the PIC Code ' OSIRIS' . *Proceedings of Central Laser Facility Annual Report 2001/2002*, 2002. 71.
- [105] Nanbu K. Theory of cumulative small-angle collisions in plasmas. *Physical Review E*, 1997, 55:4642.
- [106] Nanbu K. Momentum relaxation of a charged particle by small-angle Coulomb collisions. *Physical Review E*, 1997, 56:7314.
- [107] Griebel M. *Numerical Simulation in Molecular Dynamics*. Berlin: Springer, 2007.
- [108] GREENCARD L, ROKHLIN V. A Fast Algorithm for Particle Simulations. *Journal of Computational Physics*, 1987, 73:325.
- [109] Fonseca R A, Silva L O, Tsung F S, et al. OSIRIS: a three-dimensional, fully relativistic particle in cell code for modeling plasma based accelerators. *Proceedings of Lecture Notes in Computer Science*. 2002: 342.
- [110] Nieter C, Cary J R. VORPAL: a versatile plasma simulation code. *Journal of Computational Physics*, 2004, 196:2004.

- [111] 曹小林 张爱清 莫则尧. 基于面向对象的粒子类模拟并行计算研究. 计算机研究与发展, 2007, 44(10):1647.
- [112] Lu Q M, Cai D S. Implementation of parallel plasma particle-in-cell codes on PC cluster. Computer Physics Communications, 2001, 135:93.
- [113] Anderson D V, Shumaker D E. Hybrid Ordered Particle Simulation (HOPS) code for plasma modelling on vector-serial, vector-parallel, and massively parallel computers. Computer Physics Communications, 1995, 87:16.
- [114] Decyk V K, Karmesin S R, deBoer A, et al. Optimization of particle-in-cell codes on reduced instruction set computer processors. Computer in Physics, 1996, 10:290.
- [115] Bowers K J. Accelerating a Particle-in-Cell Simulation Using a Hybrid Counting Sort. Journal of Computational Physics, 2001, 173:2001.
- [116] Tskhakaya D, Schneider R. Optimization of PIC codes by improved memory management. Journal of Computational Physics, 2007, 225:829.
- [117] Cockshott W P. SIMD Programming manual for Linux and Widnows. London: Springer, 2004.
- [118] Quinn M J. Parallel Programming in C with MPI and OpenMP. New York: McGraw-Hill Inc, 2004.
- [119] Sloan J D. High Performance Linux Clusters with OSCAR, Rocks, OpenMosix, and MPI. Sebastopol: O'Reilly Media, Inc, 2004.
- [120] Snir M, Otto S, Otto S, et al. MPI: The Complete Reference. London: The MIT Press, 1996.
- [121] Liu J X, Mamidala A, Vishnu A. Performance Evaluation of InfiniBand with PCI Express. Proceedings of Proceedings of 12th Symposium on High Performance Interconnects, 2004.
- [122] Eastwood J W, Arter W, Brealey N J, et al. Body-fitted electromagnetic PIC software for use on parallel computers. Computer Physics Communications, 1995, 87:155.
- [123] Gjonaj F E T W. PARALLEL PARTICLE-IN-CELL (PIC) CODES. Proceedings of Proceedings of ICAP 2006, Chamonix, 2006.
- [124] Othmer C, Schule J. Dynamic load balancing of plasma particle-in-cell simulations:The taskfarm alternative. Computer Physics Communications, 2002, 147:741.
- [125] Plimpton S J, Seidel D B, Pasik M F, et al. A load-balancing algorithm for a parallel electromagnetic particle-in-cell code. Computer Physics Communications, 2003, 152:227.
- [126] Decyk V K. Skeleton PIC codes for parallel computers. Computer Physics Communications, 1995, 87:87.
- [127] Borges L, Daripay P. A Fast Parallel Algorithm for the Poisson Equation on a Disk. Journal of Computational Physics, 2001, 169:151.
- [128] Decyk V K, Norton C D. UCLA Parallel PIC Framework. Computer Physics Communications, 2004, 164:80.

- [129] Xu J, Ostroumov P N, Nolen J. A Parallel 3D Poisson Solver for Space Charge Simulation in Cylindrical Coordinates. *Computer Physics Communications*, 2007. doi:10.1016/j.cpc.2007.09.008.
- [130] Choi J D, Walker D. Parallel Matrix Transpose Algorithms on Distributed Memory Concurrent Computers. *Parallel Computing*, 1994, 21:599.
- [131] 陈国良. 并行计算-结构算法编程. 北京: 高等教育出版社, 1999.
- [132] Messmer P, Bruhwiler D L. A parallel electrostatic solver for the VORPAL code. *Computer Physics Communications*, 2004, 164:118.
- [133] Mardahl P J, Verboncoeur J P. Progress in Parallelizing XOOPIC. Technical report.
- [134] Heroux M, Bartlett R, Howle V, et al. An Overview of Trilinos. Technical report, Sandia National Laboratories, 2003.
- [135] Smith B. Petsc, 2008. <http://www.mcs.anl.gov/petsc>. ARGONNE NATIONAL LABORATORY.
- [136] Guerrero M S. Parallel Multigrid algorithms for computational fluid dynamics and heat transfer[Doctor Thesis]. Terrassa: Politècnica de Catalunya, 2000.
- [137] Wang H H. A Parallel Method for Tridiagonal Equation. *ACM Transactions in Mathematics and Software*, 1981, 7:170.
- [138] Schumann U, Strietzel M. Parallel Solution of Tridiagonal Systems for the Poisson Equation. *Journal of Scientific Computing*, 1995, 10(2):181.
- [139] 张林波. 关于采用流水线方式进行一簇递推关系式的并行计算. *数值计算与计算机应用*, 1999, (3):184.
- [140] Press W H, Teukolsky S A, Vetterling W T, et al. *Numerical Recipes in C++ (Second Edition)*. London: Cambridge University Press, 2002.
- [141] Nguyen H. *GPU Gems 3*. Addison Wesley Professional, 2007.
- [142] Nvidia C. NVIDIA CUDA 统一计算架构编程指南, 2007. <http://www.nvidia.com>.
- [143] Stantchev G, W D, Gumerov N. Fast parallel Particle-To-Grid interpolation for plasma PIC simulations on the GPU. *Journal of Parallel and Distribute Computing*, 2008, 68:1339.
- [144] Verboncoeur J P, Langdon A B, Gladd N T. An object-oriented electromagnetic PIC code. *Computer Physics Communications*, 1995, 87:199.
- [145] Decyk V K, Gardner H J. *A Factory Pattern in Fortran 95*. Berlin: Springer, 2007.
- [146] Geddes C G R, Bruhwiler D L, Cary J R, et al. Computational studies and optimization of wakefield accelerators. *Journal of Physics: Conference Series*, 2008, 125.
- [147] Huang C, An W, Decyk V K, et al. Recent results and future challenges for large scale particle-in-cell simulations of plasma-based accelerator concepts. *Journal of Physics: Conference Series*, 2009, 180:012005.

- [148] Gatsonis N A, Spirkin A. A three-dimensional electrostatic particle-in-cell methodology on unstructured Delaunay - Voronoi grids. *Journal of Computational Physics*, 2009, 228:3742.
- [149] Wu J S, Hsu K H. Parallel implementation of a three-dimensional electrostatic PIC - MCC method using an unstructured tetrahedral mesh. *Journal of Plasma Physics*, 2006, 72:845.
- [150] Wu J S, Hsu K H, Li F L, et al. Development of a parallelized 3D electrostatic PIC-FEM code and its applications. *Computer Physics Communications*, 2007, 177:98.
- [151] Mora P, Antonsen T M. Kinetic modeling of intense, short laser pulses propagating in tenuous plasmas. *Physics of Plasmas*, 1997, 4:217.
- [152] Feng B, Huang C, Decyk V, et al. Enhancing parallel quasi-static particle-in-cell simulations with a pipelining algorithm. *Journal of Computational Physics*, 2009, doi:10.1016/j.jcp.2009.04.019.

致 谢

本研究是在王友年教授悉心指导下完成的。王友年教授广博的理论知识、敏锐的洞察力、严谨的科研作风和丰富的科研经验是我学习的典范。由衷的感谢王友年教授给予我的热情指导和帮助。

本文中的大部分章节直接或者间接地来自和姜巍同学几年来的合作研究。此外，感谢物理310的同学们给我的帮助。



大连理工大学博士后研究报告使用授权协议

本人同意按照学校要求提交博士后研究报告并提供检索与浏览服务；学校可以采用影印、缩印、数字化或其它复制手段保存和汇编该研究报告。

（保密研究报告在解密后适应此规定）

作者签名： _____ 导师签名： _____

日 期： _____