·物理电子学·

五点差分格式求解泊松方程并行算法的研究

廖 臣，祝大军，刘盛纲

(电子科技大学物理电子学院 成都 610054)

【摘要】以二维静电场泊松方程数值求解的串行算法(雅可比迭代、超松弛迭代)为基础，提出了五点差分格式超松弛迭代(SOR)求解二维静电场泊松方程的并行算法，通过与雅可比迭代(Jacobi)并行算法的时间复杂度、加速比和空间复杂度进行对比，得出超松弛迭代的并行算法具有更低的时间复杂度、空间复杂度和更高的加速比与效率。通过实验验证，CHIPIC软件的泊松模块宜采用超松弛迭代并行算法。

关 键 词 雅可比迭代; 并行算法; 泊松; 超松弛迭代

中图分类号 O411.3 文献标识码 A

Parallel Algorithm Research on Solving Poisson Equations Based on Five Point Difference Format

LIAO Chen，ZHU Da-jun，LIU Sheng-gang

(School of Physical Electronics, University of Electronic Science and Technology of China Chengdu 610054)

Abstract In this paper, the efficiency of Jacobi iterative parallel algorithm for solving 2D Poisson equation is analyzed, and then the design of successive over relaxation (SOR) iterative parallel algorithm is presented. The result shows that SOR iterative parallel algorithm should be adopted in developing CHIPIC Poisson module by comparing the time complexity, speedup, and space complexity of the two algorithms in theory. At last, the result is verified by numerical experiment.

Key words Jacobi; parallel algorithm; Poisson; SOR

CHIPIC[1]是我国自行开发的电磁粒子模拟[2]软件，其模拟计算通常花费大量的时间，因此有必要开发其并行版本。作为这一工作的前期实践，本文对其静电场计算模块即泊松模块的并行计算进行了研究。

1 二维静电场泊松方程的串行算法[3]

为简单明了地说明算法的设计思想，本文采用一个最简单求解二维场域内电位的例子。如图1所示，一个长直接地金属矩形槽，其侧壁与底面电位均为0，顶盖电位为100。则求解场域内电位φ的方程为泊松方程(退化为拉普拉斯方程)：

$$\frac{\partial^2\varphi}{\partial x^2}+\frac{\partial^2\varphi}{\partial y^2}=0 \qquad \text{边界条件为：}$$

$$\begin{cases}\varphi=100 & \text{边界上}BC\text{上}\\ \varphi=0 & \text{边界}OA\text{、}OC\text{、}AB\text{上}\end{cases} \tag{1}$$

如图1离散化场域，采用五点差分格式，场域内得到其差分格式：

$$\left(\frac{\partial^2\varphi}{\partial x^2}\right)_{i,j}=\frac{\varphi_{i,j+1}-2\varphi_{i,j}+\varphi_{i,j-1}}{h^2}$$

$$\left(\frac{\partial^2\varphi}{\partial y^2}\right)_{i,j}=\frac{\varphi_{i+1,j}-2\varphi_{i,j}+\varphi_{i-1,j}}{h^2}$$

代入式(1)即得到方程组：

$$\begin{cases}\varphi_{i+1,j}+\varphi_{i-1,j}+\varphi_{i,j+1}+\varphi_{i,j-1}-4\varphi_{i,j}=0\\ \varphi_{i,j}=100 \qquad \text{边界}BC\text{上}\\ \varphi_{i,j}=0 \qquad \text{边界}OA\text{、}OC\text{、}AB\text{上}\end{cases} \tag{2}$$

求解差分方程组(2)即可得场域内各点电位值，可以采用直接法或迭代法[4]进行求解，当网格很小、网格数目较大时，采用迭代法为宜。常见的迭代法有雅可比迭代法和超松弛迭代法(超松弛因子$\omega=1$退化为高斯–赛德尔迭代法)。Jacobi迭代法：$\varphi_{i,j}^{n+1}=$

收稿日期：2006 – 02 – 15；修回日期：2006 – 06 – 20
基金项目：国家863高技术计划(2004AA832101)
作者简介：廖 臣(1982 –)，男，博士生，主要从事电磁粒子模拟软件的开发和并行算法方面的研究.

$(\varphi^n_{i+1,j}+\varphi^n_{i-1,j}+\varphi^n_{i,j+1}+\varphi^n_{i,j-1})/4$；SOR迭代：$\varphi^{n+1}_{i,j}=\varphi^n_{i,j}+\omega(\varphi^n_{i+1,j}+\varphi^n_{i,j+1}+\varphi^{n+1}_{i-1,j}+\varphi^{n+1}_{i,j-1}-4\varphi^n_{i,j})/4$。一般只要超松弛因子选择合适，就可大大加快收敛速度，使迭代次数近似与网格边长h成反比、有阶的改善。因而本文在泊松模块的串行算法中采用SOR迭代。

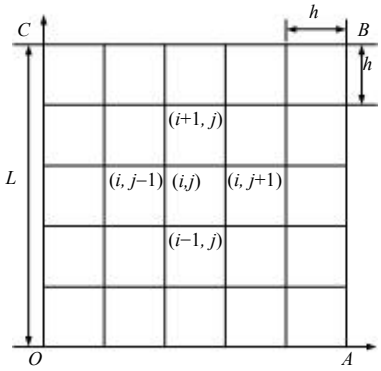

图1 离散化场域

设T_S为计算的总时间，t_i为一次迭代所用的时间，iter为迭代次数，n为问题规模(等于L/h)，无论是Jacobi迭代还是SOR迭代(只是迭代次数iter的值不同)，都满足关系式$T_S=\mathrm{iter}(n-2)^2t_i$。$t_i$不变，要缩短$T_S$，只有减少迭代次数(采用SOR)或减少网格数(采用并行分块)。其算法的空间复杂度[5]为$\Theta(n^2)$，可以通过分块减少其空间复杂度。

2 二维静电场泊松方程的并行算法

当今国外的电磁粒子模拟软件[6-7]并行版本大都是基于消息传递机制[8](MPI)的。MPI被当前所有高性能并行机所支持，程序设计方便，并具有良好的扩展性，非常适合于机群系统。

通用的求解泊松方程的并行模块大都采用Jacobi并行迭代算法[8-9]，因为Jacobi迭代算法中各个更新操作是完全并行的，可以采用分块策略。一般有一维分块和二维分块两种，二维分块的优越性主要在于更好的扩展性。由于在通常的实际问题中网格数非常巨大，而PC机比较有限，目前的通用电磁粒子模拟软件中采用一维分块。SOR迭代算法每次计算第n+1次的$U[i][j]$时，都需要n+1次的$U[i-1][j]$和$U[i][j-1]$，初略看不适合分块并行迭代。目前关于SOR的并行迭代算法主要有着色法[10]，但着色法可扩展性差，不适合通用的电磁粒子模拟软件，因此有必要开发出基于SOR分块并行迭代算法。

2.1 Jacobi并行迭代及其性能分析

如图2所示，每个进程只需要处理n^2/p个元素，各个分块内部点迭代需要的相邻元素都在同一区域内，而边界上的点迭代时需要相邻块的相关元素，因此可以在边沿上引入幻影点来接收存储相邻进程传来的数据。并且区域内部点迭代时，不需要改变边界点的值，可以采用非阻塞通信即在内部点迭代的同时进行边界数据的传输。具体的算法可参阅文献[8-9]。设t_t为传输(包括发送和接收)一次边界上的值所需的时间，则Jacobi并行迭代计算的时间：

$$T_J=\mathrm{iter}_J\max(t_i(n^2/p-2n),t_t)+\mathrm{iter}_J 2nt_i+\sigma$$

式中 t_i为一次迭代所用的时间；iter_J为Jacobi算法的迭代次数。由于通常情况下$t_i(n^2/p-2n)>t_t$，因此上式可写为$T_J=\mathrm{iter}_J t_i n^2/p+\sigma$。其中，$\sigma$为一修正因子，代表传输过程中启动传输和关闭传输的时间以及额外的时间开销，其实际值与具体的机群系统及网络环境有关，其加速比[5]为：

$$\psi(n,p)=\mathrm{iter}_{SOR}(n-2)^2t_i/(\mathrm{iter}_J t_i n^2/p+\sigma)\approx p\,\mathrm{iter}_{SOR}/\mathrm{iter}_J$$

可见Jacobi迭代的迭代次数大于SOR迭代次数是影响并行加速比的主要原因。由于Jacobi迭代采用两套数组存储电位值，其算法空间复杂度为$\Theta(2n^2/p)$。

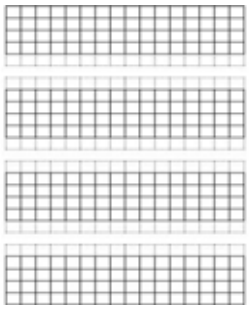

图2 分块策略图

2.2 SOR并行迭代及其性能分析

同样采用图2所示的行分块策略，则每一块要开始进行迭代，必须等待相邻的下一块计算完毕并接收到相邻的下一块上边界的值，似乎不适合并行。但是由于迭代算法的迭代次数非常巨大，不只一次迭代，可以采用如图3所示的时序方法。

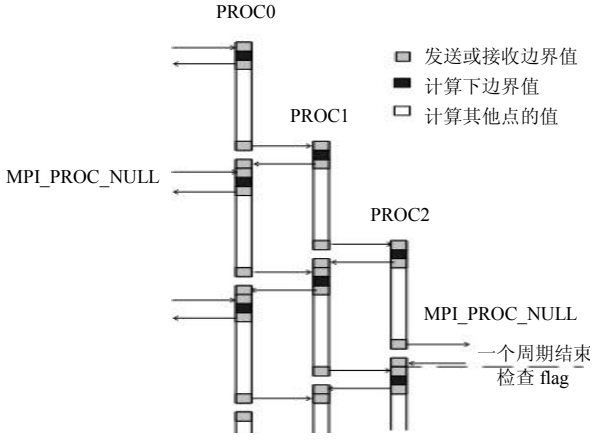

图3 SOR迭代时序图

设p为进程数；myrank进程为当前块对应的进程号；mytop进程为相邻的上一块对应的进程号；mydown进程为相邻的下一块对应的进程号；flag为一逻辑变量，判断迭代是否达到收敛系数要求。为了程序设计的方便，可以取最下块的mydown进程和最上块的mytop进程为MPI_PROC_NULL。其算法的文字描述如下：

(1) 最下面的块(此时为myrank进程)将flag设置为1，开始迭代。迭代过程中，如果某点电位值迭代前和迭代后之差大于W(收敛系数)，将flag置0，迭代完成后将上边界的电位值和flag发送给mytop进程，然后阻塞等待接收mytop进程下边界的值。

(2) 最下面的块的mytop进程接收到该进程的上边界值和flag后，首先计算下边界的电位值，然后将下边界的值传输给myrank进程，再迭代其他点的电位值。迭代过程中，如果某点电位值迭代前和迭代后之差大于W，将flag置0，迭代完成后和myrank进程操作相同。

(3) myrank进程接收mytop进程的下边界后开始第二次迭代。其他块对应的进程都类似处理，直到迭代完成。

实现该算法的关键是如何判断迭代完成，如果是串行算法，最后一个进程完成一次迭代，如果flag仍为1，则结束循环。但是在并行迭代过程中，最后一个进程在完成该次迭代的时候(假设此时迭代次数为iter)，前面的进程已经开始新的迭代，执行最快的进程在执行第iter+p−1次迭代，因此可以利用MPI2.0的单边通信方式由最后一个进程向其他进程写入一个最后迭代的总次数(iter+p−1)。当进程迭代iter+p−1次后就自动退出，其伪代码描述如下：

MPI_Init(&argc, &argv)；非阻塞发送下边界给下mydown进程；

MPI_Comm_size(MPI_COMM_WORLD, &*nproc*)；迭代计算其余行的数据；

初始化mydown和mytop进程//非阻塞发送flag和上边界给mytop进程；

创建窗口//创建单边通信的内存空间

非阻塞接收mytop进程上边界；

初始化电位值//确保接收和发送完成;

//迭代计算//单边写入done的值

done=9 999 999；//初始化迭代的总次数

if(mytop = MPI_PROC_NULL && flag1= 0) {iter=0; //迭代次数初始化为0

if(flag = 1) {while(iter<done) flag1=1;

{//单边写入done的值(为iter+$n$proc−1);

iter++;}//end if

阻塞接收mydown进程上边界值和flag；

}//end if

迭代计算下边界；}//end while

设t_s为发送或接收一次上边界或下边界的时间；t_i为一次迭代所用的时间，则SOR并行计算的时间为：

$$T_{SOR}=[(n^2/p+n)t_i+4t_s][\text{iter}_{SOR}+2(p-1)]+\sigma$$

其加速比为：

$$\psi(n,p)=\frac{\text{iter}_{SOR}(n-2)^2t_i}{[(n^2/p+n)t_i+4t_s][\text{iter}_{SOR}+2(p-1)]+\sigma}\approx p\frac{1}{1+4pt_s/(t_in^2)}$$

因此只要$4pt_s/(t_in^2)<(\text{iter}_J-\text{iter}_{SOR})/\text{iter}_{SOR}$就可以得到比Jacobi迭代更好的加速比。而通常情况下$n^2>>p$，所以SOR并行迭代较Jacobi并行迭代有更好的加速效果。另外其算法空间复杂度为$\Theta(n^2/p)$，较Jacobi迭代节省一半的存储空间。

3　实验结果

实验在由HUB连接的四台计算机(P4 2.4 GHz，内存512 MB)组成的机群系统下完成，并行平台为MPICH2。采用不同的迭代方法和网格大小，所得结果如表1所示(收敛系数为0.000 1)。

表1　算法计算时间与迭代次数的比较

迭代方式	网格数 256×256		512×512		1 024×1 024	
	T/s	iter	*T*/s	iter	*T*/s	iter
SOR 串行	3.009	615	24.762	1 236	187.114	2 315
SOR (p=2)	1.904	616	14.463	1 237	103.641	2 316
SOR (p=4)	1.278	618	8.845	1 239	55.236	2 318
Jacobi (p=2)	153.238	45 254	1668.665	108 850		
Jacobi (p=4)	100.071	45 254	946.135	108 850		

由表1的结果可以看出，Jacobi算法的迭代次数远大于SOR算法的迭代次数，因此其并行算法的效率往往低于SOR串行算法的效率，而SOR并行迭代能取得较好的效果。由表1的结果可以进一步分析网格大小对SOR并行算法加速比和效率的影响，如表2所示。

(下转第127页)

[3] 张　莉, 周伟达, 焦李成. 尺度核函数支撑矢量机[J]. 电子学报, 2002, 30(4): 527-529.
[4] 杜　平, 张燕昆, 刘重庆. 基于不变矩的人脸识别方法的研究[J]. 计算机仿真, 2002, 19(3): 78-81.
[5] 祝海龙, 屈梁生, 张海军. 基于小波变换和支持向量机的人脸检测系统[J]. 西安交通大学学报, 2002, 36(9): 947-950.
[6] VAPNIK V N. The nature of statistical lLearning theory[M]. Beijing: Tsinghua University Press, 2000.
[7] ZHANG L, ZHOU W D, JIAO L C. Scaling Kernel function support vector machines[J]. Acta Electronica Sinica, 2002, 4(4): 527-529.
[8] YAN H, ZHANG X G, MA Y Q, et al. The parameter estimation of RBF Kernel function based on veriogram[J]. Acta Automatica Sinica, 2002, 28(3): 450-455.
[9] OSUNA E, FREUND R, GIROSI F. Training support vector machines: an application to face detetion[C]//Comput Vis Pattern Recogn. [S.l.]: IEEE, 1997: 130-136.
[10] GENOV R. Massively parallel mixed-signal VLSI Kernel machines[D]. Baltimore: Johns Hopkins Univ, 2002.
[11] LASKOV P. Feasible direction decomposition algorithms for tTraining support vector machine[J]. Machine Learning, 2002, 46(1): 315-349.
[12] LIN C F, WANG S D. Fuzzy support vector machines[J]. IEEE Transaction on Neural Networks 2002, 48(1): 85-105.
[13] VAPNIK V, CHAPELL O. Bounds on error expectation for support vector machine[M]. Advance in Large Margin Classifiers: MIT Press, 1999.

编　辑　漆　蓉

(上接第83页)

由表2可知，网格数越大，SOR并行迭代算法可以得到更好的加速比和效率，因为计算时间占总的消耗时间比例越大。

该实验是在HUB连接的LAN下完成的，并且整个局域网与广域网是连通的，如果采用更加高速的网络连接，可以得到更好的加速比和效率。

表2　SOR并行算法的加速比和效率与网格大小的关系

网格数	*P*=2		*P*=4	
	加速比	效率	加速比	效率
256×256	1.58	0.79	2.35	0.59
512×512	1.71	0.86	2.80	0.70
1 024×1 024	1.81	0.90	3.39	0.85

4　结 束 语

在二维CHIPIC软件中，无论实际所需模拟的区域是否规则，都会采用矩形的模拟区域，每完成一次迭代都要进行相应的边界处理。因此，对于复杂的图形结构，只是边界条件的处理和设置不同，可以得到与上述规则结构近似的结果。从而CHIPIC软件的泊松模块宜采用SOR的并行迭代算法实现。

参 考 文 献

[1] 狄　隽, 祝大军, 刘盛纲. CHIPIC软件电磁场计算方法[J]. 电子科技大学学报, 2005, 34(4): 484-488.
[2] BIRDSALL K G, LANGDON A B. Plasma physics via computer simulation[M]. New York: McGraw-Hill, Inc., 1985.
[3] 王秉中. 计算电磁学[M]. 北京: 科学出版社, 2002.
[4] 钟尔杰, 黄廷祝. 数值分析[M]. 北京: 高等教育出版社, 2004.
[5] 孙世新, 卢光辉, 张　艳, 等. 并行算法及其应用[M]. 北京: 机械工业出版社, 2005.
[6] GOPLEN B, LUDEKING L, SMITHE D, et al. User-configurable MAGIC for electromagnetic PIC calculations[J]. Comput. Phys Commun, 1995, 87: 54-86.
[7] 刘大刚, 祝大军, 刘盛纲. 爆炸式发射二极管的粒子模拟研究[J]. 电子科技大学学报, 2005, 34(4): 481-484.
[8] 莫则尧, 袁国兴. 消息传递并行编程环境[M]. 北京: 科学出版社, 2001.
[9] QUINN M J. Parallel programming in C with MPI and openMP[M]. New York: McGraw-Hill, Inc, 2004.
[10] 陈国良. 并行计算[M]. 北京: 高等教育出版社, 1999.

编　辑　漆　蓉