非线性微分—代数系统迭代法的并行实现

孙 卫1,2, 樊晓光1, 李 立2, 周友运1

(1. 空军工程大学工程学院，陕西西安 710038；2. 西安交通大学理学院信息与系统科学研究所，陕西西安 710049)

摘 要：讨论了非线性微分—代数系统的并行迭代算法所涉及的理论和具体算例的实现。利用动力学迭代法对微分—代数系统进行剖分迭代，并在曙光3000超级服务器上选用实际算例测试这些并行迭代算法。结果显示：这些迭代方法能够有效地并行实现，具有优良的加速比，这也证实了动力学迭代法在理论上的内在并行性。

关键词：微分—代数系统；动力学迭代法；并行实现

中图分类号：O241.8 **文献标识码**：A **文章编号**：1009-3516(2005)05-0082-03

近年来,对大型瞬态系统的解耦或分割方法的研究已成为科学计算领域中的关键课题之一，这些瞬态系统通常通过状态演化方程来表述。演化方程的具体表现形式可能是常微分方程,微分－代数方程,偏微分方程,积分－微分－代数方程,等等。本文将要研究的就是指标－3的微分－代数系统的并行迭代算法及其在曙光3000并行机器上的实现。

首先考虑非线性指标－3的微分－代数方程初值问题

$$\frac{\mathrm{d}y}{\mathrm{d}t}=f(t,y(t),z(t)),\frac{\mathrm{d}z}{\mathrm{d}t}=k(t,y(t),z(t),u(t)),g(t,y(t))=0 \tag{1}$$

其中函数f,k和g是充分可微的，并假设不等式$\|(g_y(y)f_z(y,z)k_u(y,z,u))^{-1}\|\leqslant M$在某一邻域内成立。进一步，在时间域$[t_0,t_T]$内，初始值$(y_0,z_0,u_0)$一致满足条件:$g=0,g_yf=0,g_{yy}(f,f)+g_yf_yf+g_yf_zk=0$。为了求解指标－3的微分－代数系统(1)，我们考虑动力学迭代或波形松弛(Waveform Relaxation－WR)方法。这种方法在1982年由Lelarasmee首次提出[1]，它能够解耦整个系统,将系统的每个方程剖分成几个部分。微分－代数系统(1)的动力学迭代形式如下:

$$\begin{cases}\dfrac{\mathrm{d}y}{\mathrm{d}t}y^{k+1}(t)=F(t,y^{k+1}(t),y^k(t),z^{k+1}(t),z^k(t)),G(t,y^{k+1}(t),y^k(t))=0\\ \dfrac{\mathrm{d}}{\mathrm{d}t}z^{k+1}(t)=K(t,y^{k+1}(t),y^k(t),z^{k+1}(t),z^k(t),u^{k+1}(t),u^k(t))\end{cases} \tag{2}$$

定义初始值$y^{(0)}(\cdot)\equiv y_0,z^{(0)}(\cdot)\equiv z_0$，和$u^{(0)}(\cdot)\equiv u_0$，第$K+1$次迭代的初始值定义为$y^{(k+1)}(t_0)=y_0$，$z^{(k+1)}(t_0)=z_0,u^{(k+1)}(t_0)=u_0,k=1,2,\cdots$，函数$F,K$和$G$满足条件$F(t,y,y,z,z)=f(t,y,z),K(t,y,y,z,z)=k(t,y,z,u)$和$G(t,y,y)=g(t,y)$。

动力学迭代算法是一种具有内在并行性的算法,比通常由串行方法改造的并行算法更有效;因此如何挖掘这种算法的内在并行性,以及如何在并行机器上有效实现是该类算法要解决的重要课题。

1 并行算法

对于微分－代数系统来说,Jacobi－WR和Gauss－seidel－WR算法是两种常用的动力学迭代算法[2~3]。系统(1)的Jacobi－WR并行算法的实现过程概括如下。

1.1 系统(1)的Jacobi－WR并行算法

1)初始化。①按进程数划分系统,并在每个进程上定义激励函数;②将区间$[t_0,t_T]$剖分为N个子区

收稿日期:2005－01－25
作者简介:孙 卫(1964－),女,辽宁大连人,副教授,博士,主要从事复杂系统的稳定性、系统的仿真与建模等研究.

间,在每个进程中用 $X^0=(x_i^0(t_j))_{n\times N}$,$Z^0=(z_i^0(t_j))_{n\times N}$和 $U^0=(u_i^0(t_j))_{n\times N}$记录初始波形矩阵,即按给定猜测波形赋波形猜测值。

2)块 Jacobi 迭代。①计算:在每个进程上调用 rk4-jacobi 子函数求解迭代方程,各进程的计算同步进行;②通信:MPI 并行程序设计采用同步通信模式,在每个进程都计算完毕后,进程之间相互通信以更新各进程中的初始波形矩阵;③判断停机条件:判断所求取的解是否满足迭代要求或停机条件,否则重复上两步进行下一轮迭代。

3)输出最后迭代更新的初始波形矩阵 $X^0=(x_i^0(t_j))_{n\times N}$,$Z^0=(z_i^0(t_j))_{n\times N}$和 $U^0=(u_i^0(t_j))_{n\times N}$,即得到该微分-代数系统的解。

1.2　系统(1)的 Gauss-seidel-WR 并行算法

其并行算法的实现过程概括如下。

1)初始化:①按进程数划分系统,并在每个进程上定义激励函数;②将时间区间$[t_0,t_T]$被剖分为由 N 个时间节点组成的子区间,在每个进程中用 $X^0=(x_i^0(t_j))_{n\times N}$,$Z^0=(z_i^0(t_j))_{n\times N}$和 $U^0=(u_i^0(t_j))_{n\times N}$记录初始波形矩阵,即按给定猜测波形赋波形猜测值。

2)红黑 Gauss-seidel 迭代:①红点计算:在每个进程上调用 rk4-jacobi 子函数求解红点所在迭代方程,该子函数使用古典四阶 Runge-Kutta 方法,各进程的计算同步进行;②第一次通信:MPI 并行程序设计采用同步通信模式,在每个进程都计算完毕后,进程之间相互通信以更新各进程中红点所在位置的初始波形矩阵;③黑点计算:在每个进程上调用 rk4-jacobi 子函数求解红点所在迭代方程,该子函数使用古典四阶 Runge-Kutta 方法,各进程的计算同步进行;④第二次通信:在每个进程都计算完毕后,进程之间相互通信以更新各进程中黑点所在位置的初始波形矩阵;⑤判断停机条件:判断所求取的解是否满足迭代要求或停机条件,若是,转到下一步,否则重复上面步骤进行下一轮迭代。

3)输出最后迭代更新的初始波形矩阵 $X^0=(x_i^0(t_j))_{n\times N}$,$Z^0=(z_i^0(t_j))_{n\times N}$和 $U^0=(u_i^0(t_j))_{n\times N}$即得到该大非线性瞬态系统的解。

2　算法的具体实现

并行试验的硬件环境由6台 IBM RS/6000 工作站和一台曙光天潮系列-曙光3000超级服务器组成。曙光3000由国家智能计算研究开发中心研制,它基于分布式存储机群系统和消息传递体系结构,节点采用 IBM Power3-II 和 PowerPC RS64-III 微处理器,节点之间通过10/100/1000Mbps高速以太网和高速系统网络互连,是通用的可扩展超级服务器系统。并行机器上计算节点所用操作系统为 AIX4.3.3,支持 C、Fortran、Java 等程序设计语言,以及 ESSL、BLAS、Scalapack、Gauss 等数学和工程库。并行程序设计环境为 PVM 和 MPI[4],我们选用的是编程语言为 C/C++语言以及并行进程通信的 MPI 库。

例: Pendulum problem 问题(指标-3)

$$\begin{cases} p'=u,q'=v,u'=-p\lambda,v'=-q\lambda-1,0=p^2+q^2-1, \\ p(0)=1,q(0)=0.\ i(0)=0,v(0)=0,\lambda(0)=0,t\in[0,2] \end{cases} \tag{3}$$

取时间步长为 $\Delta t=0.01$,并假设初值为零函数。采用的基本技术手段为并行 Jacobi WR 方法和 Gauss-Seidel WR 方法。单个微分方程采用古典四阶 Runge-Kutta 法;单个代数方程采用 GS 迭代法。定义逐次迭代误差为 $E_i^{(k+1)}e=\sqrt{\sum_{j=0}^{199}\|w^{(k+1)}(t_j)-w^{(k+1)}(t_j)\|^2}$,其中 $w^{(l)}(t)(l=k,k+1)$ 表示波形,时间 t_j满足 $t_j=j\Delta t(j=0,1,\cdots,199)$。离散的动力学迭代算法1-15次迭代误差的数值结果分别为:29.5523, 54.8104, 61.8648, 8.00105, 1.22224, 0.15605, 1.26386×10^{-2}, 2.30096×10^{-3}, 2.43815×10^{-4}, 2.91207×10^{-5}, 2.39788×10^{-6}, 1.88226×10^{-7}, 1.21042×10^{-8}, 6.98735×10^{-10}, 3.55364×10^{-11}。迭代的误差曲线见图1。图2给出了此算例的加速比。图2中,在计算节点较少时,加速比基本上呈线性增加的趋势。对于此算例的加速比超过最大计算节点数的问题,是因为我们在计算中通讯和计算重叠,并有效利用了高速缓存;另外,通讯与计算的重叠,高速缓存的充分利用是增大加速比的重要因素[4-5]。

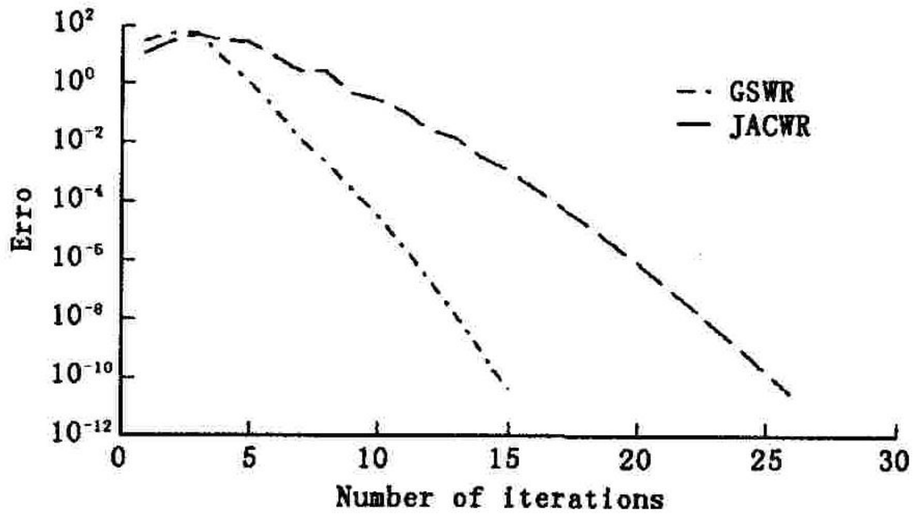

图 1 该算例动力学迭代方法的收敛误差曲线

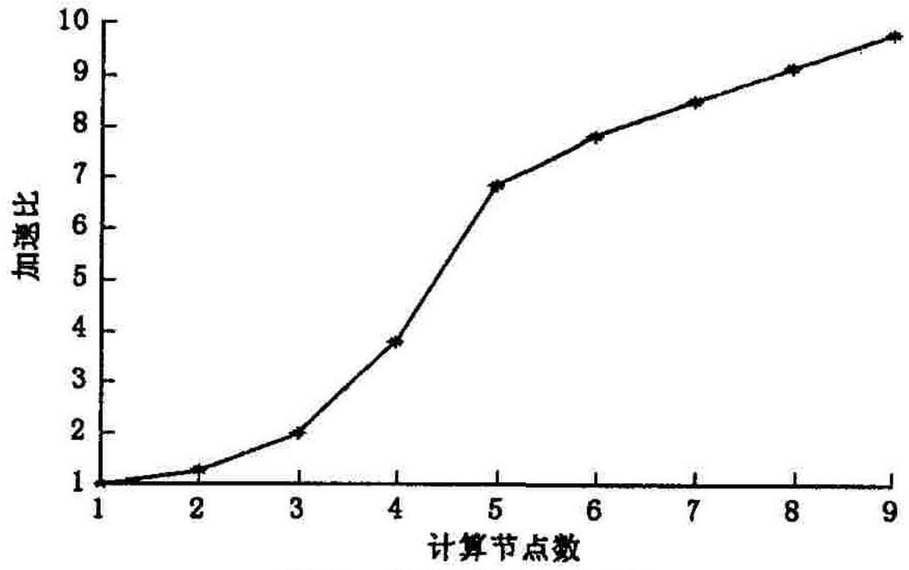

图 2 该算例的加速比

3 结论

本文讨论了非线性微分 - 代数系统的动力学迭代公式和相应的并行迭代算法，完成了该并行算法的两种形式：Jacobi WR 和 Gauss - Seidel - WR 算例的具体实现，并做了误差估计。结果显示：并行的 Jacobi WR 和 Gauss - Seidel WR 迭代算法是具有内在并行性的有效算法，能准确有效地求解系统的数值解，求解非线性微分 - 代数系统时能够节省存储空间。

参考文献：

[1] Lelarasmee E, Ruehli A E, Sangiovanni - Vincentellli A L. The waveform relaxation method for time - domain analysis of large scale integrated circuits[J]. IEEE Trans. CAD IC Syst, 1982, (1): 131 - 145.

[2] Burrage K, Parallel and sequential methods for ordinary differential equations[M]. Oxford: Clarendon Press, 1995.

[3] Hairer E, Wanner G. Solving Ordinary Differential Equations II[M]. Berlin: Springer - Verlag, 1991.

[4] 都志辉. 高性能计算并行编程技术 - MPI 并行程序设计[M]. 北京:清华大学出版社,2001.

[5] 沈志宇. 并行编译方法[M]. 北京: 国防工业出版社, 2000.

(编辑:姚树峰)

The Parallel Implement of Iteration Methods for Non -Linear Differential - Algebraic Systems

SUN Wei1,2, FAN Xiao -guang1, LI Li2, ZHOU You- yun1

(1. The Engineering Institute, Air Force Engineering University, Xi'an, Shaanxi 710038, China; 2. School of Science, Xi'an Jiaotong University, Xi'an, Shaanxi 710049, China)

Abstract:This paper discusses the theoretical models and numerical experiments of parallel iteration methods for solving non -linear differential -algebraic systems. Dynamics iteration methods are used to split the differential - algebraic equations, and the numerical experiments are selected to test the parallel iteration methods on Dawning 3000 Super Server Systems. The result shows that the iterative methods can implement parallel availably and possess a superior accelerated ratio. It is testified that the dynamics iteration methods possess immanent parallel in theory.

Key words:differential- algebraic systems; dynamics iteration methods; parallel implement