计算方法丛书

有限元结构分析并行计算

周树荃 梁维泰 邓绍忠 著

科学出版社

内 容 简 介

有限元结构分析在大型工程计算中至今仍居重要地位．本书系统地论述了有限元方程组形成和求解的各个步骤的并行计算格式和并行程序设计技巧，着重介绍了有限元分析的并行计算、大型稀疏有限元方程组直接解法的并行处理、大型稀疏线性方程组预处理共轭梯度法的并行处理、矩阵向量积的并行计算，还概括了近年来有关研究的主要成果，是一部具有较高理论水平和实用价值的著作．

本书可供计算数学工作者、工程技术人员以及高等院校有关专业的师生阅读参考．

计算方法丛书

有限元结构分析并行计算

周树荃 梁维泰 邓绍忠 著

责任编辑 林 鹏 夏墨英

科学出版社 出版

北京东黄城根北街16号

邮政编码：100717

科地亚印刷厂印刷

新华书店北京发行所发行 各地新华书店经售

*

1994年4月第 一 版 开本：850×1168 1/32

1999年5月第三次印刷 印张：10 1/8

印数：4 501—6 500 字数：265 000

ISBN 7-03-003750-2/O・665

定价：18.00元

（如有印装质量问题，我社负责调换〈新欣〉）

前　言

1972年第一台并行机问世以来，并行算法有了实践的机会。大体上说，70年代并行算法的研究课题比较集中于相关问题以及线代数计算的并行化；随着并行机的发展，70年代后期有限元结构分析的并行计算受到了重视。1984年国产巨型机YH-1投入运转，为国内研究并行算法创造了良好的条件。近年来，在航空科学基金等预研项目的资助下，我们开展了这一领域的研究工作，取得了一些成果，现整理成书供从事并行算法研究及使用并行机作科学与工程计算的有关科学与工程技术人员参考。

因并行算法的研究和应用与并行机的体系结构密切相关，所以本书专辟一章扼要介绍了国内外一些具有代表性的阵列处理机、向量处理机及多处理机系统的体系结构。为使读者对并行机与并行算法的发展、有限元结构分析并行计算的发展以及现代科学技术对高性能计算机的需求有一个概括的了解，作者在绪论中对这些问题逐一作了介绍。

在阅读本书的其余章节之前，最好先学习同步并行算法的一些基础知识（主要是一些相关问题的并行计算与线代数的并行计算），这些内容在参考文献[56—59]中都可找到。

参加本书编写的有周树荃、梁维泰、邓绍忠与叶明等人。

作者感谢南京航空学院领导和数理力学系一些同事的鼓励和支持，感谢武汉大学康立山教授和孙乐林副教授对书稿认真仔细的审阅。本书有关数值试验是在西南计算中心的YH-1机上进行的，曾得到该所王嘉谟、黄清南、马寅国等同志的帮助和支持，作者谨此致谢。

《计算方法丛书》编委会

副主编　石钟慈　李岳生

编　委　王仁宏　王汝权　孙继广　李德元

李庆扬　吴文达　林　群　周毓麟

席少霖　徐利治　郭本瑜　袁兆鼎

黄鸿慈　蒋尔雄　雷晋干　滕振寰

目　　录

试读结束：需要全本请在线购买：　www.ertongbook.com

第一章 绪 论

本章阐述现代科学技术对高性能计算机的需求，计算机与算法的分类，数值并行算法发展的几个阶段，有限元结构分析并行计算发展现状，使读者对并行计算机及并行算法的发展、有限元结构分析并行计算的发展有一个概括的了解.

§1.1 现代科学技术对高性能计算机的需求

现代科学技术对高性能计算机的需求主要来自数值计算和非数值处理(或符号处理)两大方面. 数值计算包括科学计算、系统模拟与工程设计等应用领域，统称为科学/工程计算. 非数值处理包括非数值信息处理(例如，全国性乃至世界性的情报检索系统，政府部门的调查统计、军事情报搜索与分析系统和指挥系统、银行保险业务系统等)，知识处理与智能信息处理[例如，定理证明、专家系统、模式识别、智能机器人、CAD 与 CAM、智能机辅助教学 (ICAI)、博奕]等领域. 参阅文献[1—3].

下面用几个具体事例说明这一点.

(1) 数值天气预报

用电子计算机来预报全球气候，需要求解一组球面坐标系下的一般环流模式方程. 通常，用垂直高度、纬度和经度将大气划分为三维网格，时间作为第四维，用一个指定的时间增量将方程组离散. 在一个半球上给定间距为 270 英里(1 英里=1.609 公里)的网格，并给定一个适当的时间增量，24 小时的预报需要 1 000 亿次操作，在每秒可执行一亿次操作的计算机(如 CRAY-1)上，需要花费大约 100 分钟. 这种网格可以用于纽约和华盛顿的预报，但不能用于费城的预报，费城大约位于华盛顿和纽约的中间. 要得到费城的更精确的预报，在所有四维上还必须将网格大小减半，其计算量

将增加 16 倍. 对于每秒 100 MFLOPS 的计算机 (CRAY-1), 完成 24 小时天气预报需要 24 小时计算. 即使是这种新的网格，也不能有效地作长期预报. 如果我们希望得到更精确的长期预报，必须发展更高性能的计算机.

(2) 计算空气动力学与飞行器设计

今天用于空气动力学设计的两个基本手段是风洞和计算机. 其中，风洞是飞行器设计中进行气流模拟的主要手段，目前计算机只能起辅助作用. 但是风洞实验一方面要受模型规模、风速、密度、温度等干扰，同时还要受许多其他因素的限制. 然而数值流动模拟却没有这些限制，它仅仅受计算机的计算速度和存储空间限制. 另一方面，建造和使用风洞的设施是十分昂贵的. 据介绍，用一个现代化的风洞进行飞机翼型测试，每年要耗资 1.5 亿美元. 如果能用计算机取代风洞的功能，则不但可以节省大量的费用，而且计算机可以实现许多风洞无法实现的空气动力学模拟，使空气动力学特性设计更为科学化. 计算空气动力学的基本任务就是用高速计算机求解流体力学方程组，以实现空气动力学数值模拟. 在实际的空气动力学设计中，计算机只有在 10 到 15 分钟内完成模拟，它才能当作一种工程手段，否则只能作为程序的研究. 比如，要求解的问题是三维雷诺平均 N-S 方程，采用 10^6 个网格点，使用最新的算法，计算机也必须具有持续每秒十亿次以上的浮点运算速度，才能在 10 分钟内完成模拟. 目前市场上的超级计算机的峰值速度大都在每秒 10 亿次以下，实际运行中远远达不到这个峰值速度. 一般仅能达到峰值速度的 10% 到 40%， 而要达到持续每秒 10 亿次以上的浮点运算速度，计算机的峰值速度必须达到每秒 80 亿次浮点运算，这个速度大约相当于 CRAY-2 速度的四倍. 达到这个目标后，计算机可以替代部分风洞实验，但还不能完全代替风洞实验.

目前由美国国家科学基金资助的伊利诺伊大学研究机构已使用超级计算机来研究风剪切，对遇到微爆炸的"飞行"模拟飞机作空气动力学计算. 他们希望通过计算进一步研究微爆炸的危险且

现.

用今天的超级计算机设计整架飞机不可能，如要对整架飞机进行设计，还需要更强大的计算机，其中包括复杂的三维流模拟和相应的化学反应模拟的发动机设计，这需要更大的计算机能力和容量，甚至至少要比今天的超级计算机高出两个数量级.

航天飞机的设计，同样需要高性能的超级计算机.

(3) 人造卫星图象处理

分析卫星发射回来的地球资源数据，在农业、生态学、林业、地质以及土地使用规划方面有着广泛的应用. 然而，卫星发回的图象数据通常都是如此之浩繁，以致于其中简单的计算也要花费大量 CPU 时间. 美国国家航空和航天管理局 (NASA) 已经安装了由 Goodyear 宇航公司制造的大规模并行处理机 (MPP)，它是由 13684 台处理机构成的一个并行系统，使用它来执行卫星图象处理.

(4) 核物理与核工程领域的研究与应用

在先进的武器设计中，特别是核武器的设计、核武器效应的模拟等需要进行大型计算模拟. 这种大型计算模拟可以真正取代实验方法. 类似地，未来核电站的设计、情报的收集、为自动绘制地图而用的图象数据处理等运算，都需要比今天的计算机大几个数量级，才能满足军事上与工程技术上的要求.

(5) 石油资源勘探

石油资源的地震勘探法，要求在沿海 1 公里测量线上，每次人造地震要采集 200万—500 万个数据，再对如此大量的数据通过褶积运算处理，描绘出地层构造，为钻井位置提供准确情报. 目前，我国石油部某单位配备有若干型号的并行超级计算机，昼夜工作，仍不能满足需求.

(6) 电子结构模拟

需要计算成千个变量的积分-微分方程组，成百万个六重积分，并需要求解高阶(达 10^6 阶)矩阵的特征值问题.

(7) 人体心脏三维模拟

普林斯顿大学的数学、医学和计算机科学家一起成功地进行了人体心脏三维模拟，用 64×64×64 网格点求解粘性不可压缩流体的 N-S 方程．每模拟一次心脏收缩，就需用 CRAY-2 机 8 个小时的 CPU 时间．

(8) 人工智能

大多数通用计算机都有一个相对的输入/输出 (I/O) 相互作用面．但是，如果计算机在最高层次上通过说话、图和自然语言，使人类与计算机进行交互作用，这就需要对声音、图和自然语言的实时输入进行处理．这种处理的数据量是相当大的，决非今天的计算机能够解决的．日本从 1982 年开始的“第五代计算机系统研究计划” (Fifth Generation Computing System Project)，其目的之一就是建立一种新的计算机，其能力每秒可执行一亿次到十亿次的逻辑判断．由于一个逻辑判断要执行 100 到 1000 条机器指令，因此所设计的新一代计算机必须每秒完成一百亿到一千亿条指令．实际上也可以说，“五代机”是智能讯息处理系统或简称“智能计算机”．

综上所述，单处理器的冯诺伊曼机的性能已远远不能满足上述各方面的要求．这是因为它一方面要受到顺序执行的限制；另一方面，VLSI 器件本身的开关速度也有一个极限，何况电信号的传播速度也要受到光速的限制．因此，单单从提高 VLSI 器件本身的速度来提高冯诺伊曼机的速度已经接近一个远低于人们要求的速度极限．因此，开发超级计算机已是社会的需要．通过向量机或多处理机实现并行处理是开发超级计算机的重要途径．

美、日两国政府已认识到要在科学技术上保持世界领先地位，就必须在并行处理技术方面占有领先地位．为此，1989 年 3 月美国国防部提出的一份旨在保持国际上技术领先地位的报告中把“并行处理”列入 22 项重大项目的第 3 项；日本政府则列入第二位，介于“软件工程”与“人工智能”之间，这是因为并行处理是人工智能的基础，或者说，人工智能只有基于“极度并行”基础上才能实现．国防科工委以及我国的一些著名科学家对并行机与并行算法

的发展极为关注，目前我国巨型机有了一定的发展、而并行算法的研究有了更大的进步，在国防工业的个别部门与石油勘探方面已获得重要应用，气象部门也给予极大重视，但与美、日等国家相比，还有很大的差距，极需迎头赶上。

§1.2 计算机与算法的分类

众所周知，在计算机上计算一个题目，首先需要将计算问题编制成程序(通称源程序)，然后经过编译得到由机器指令组成的目标程序。该目标程序可以分成两个子序列：

(1) 指令序列——由各条指令中操作符构成的序列。亦称指令流 (Instruction Stream)。

(2) 数码序列——由各条指令中操作数构成的序列。亦称数据流 (Data Stream)。

例 计算

$$\frac{(a+b-c)*d}{e} \Longrightarrow f \tag{1.1}$$

的计算程序是

$$a+b \Longrightarrow u,\ u-c \Longrightarrow u,\ u*d \Longrightarrow u,\ u/e \Longrightarrow f.$$

其中，操作符构成的序列——指令流为

$$+,-,\times,/,\Rightarrow. \tag{1.2}$$

操作数构成的序列——数据流为

$$a,b,u;\ u,c,u;\ u,d,u;\ u,e,f. \tag{1.3}$$

一般地，假定指令序列为

$$\theta_1,\ \theta_2,\ \cdots,\ \theta_i,\cdots, \tag{1.4}$$

其中 θ_i 是通常意义下的操作符。

1.2.1 单指令流-单数据流

如果指令流中的每个操作符 θ_i 仅对相应的一对操作数进行操作(称双目操作)，或对一个操作数进行操作(称单目操作)并产生一个计算结果，则称这种加工方式为单指令流-单数据流加工

试读结束：需要全本请在线购买： www.ertongbook.com

方式，也称为横向加工方式，简记为 SISD (Single Instruction Stream Single Data Stream).

按单指令流-单数据流方式进行计算的计算机，称为 SISD 型计算机，或称为串行机. 现今国内普遍使用的中、小型数字计算机基本上都属于这一类型.

1.2.2 单指令流-多数据流

如果指令流中的操作符对相应的一对数组(或单个数组)进行操作，并产生一组计算结果，则称这种加工方式为单指令流-多数据流加工方式，也称纵向加工方式，简记为 SIMD (Single Instruction Stream Multiple Data Stream).

按单指令流-多数据流加工方式进行计算的计算机称为SIMD型计算机. SIMD 型计算机又包括阵列机与流水线机两种类型. 阵列式计算机：如 ILLIAC-IV; 流水线计算机：如 CDC STAR-100, CYBER 205, CRAY-1 及国产机 YH-1 (银河-1)等都属于这一类. 它们要求一类"同步并行算法"支持. 这类算法基于进行向量与矩阵运算时各分量的高度同步并行性，故这类计算机也称为向量机.

1.2.3 多指令流-多数据流

如果指令序列有 T 个，$T \geqslant 2$，

$$\theta_{1i},\ \theta_{2i},\ \cdots\ \theta_{ii},\ \cdots, \tag{1.5}$$
$$i = 1,2,\cdots,T,\quad i = 1,2,\cdots,$$

每个指令序列中的操作符对相应的单目或双目（或单数组，双数组)执行操作，并分别产生一个(或一组)计算结果，则这种加工方式称为多指令流-多数据流加工方式. 简记为 MIMD (Multiple Instruction Stream Multiple Data Stream).

按这种方式进行计算的计算机，称为 MIMD 型计算机. 多处理机与多计算机就属于这种类型. 例如，武汉大学的 WUPP-80 并行处理系统；卡内基-梅隆大学的 Cmmp 与 C^*_m 系统；加州理

工学院的 hypercube 等多处理机系统以及多向量机 (M-SIMD): HEP, CRAY X-MP 系列, CRAY-2, S-820/80 等等. 这类机器的并行运算过程通常都具有高度的"自治性", 即异步地并行运算,它要求一类"异步并行算法"支持.

1.2.4 并行机

SIMD 型计算机与MIMD 型计算机统称并行机.关于 MIMD 型计算机按其所含处理机的台数又分为浅度并行 (10—100 台), 深度并行(100—1 000台)与极度并行(10 000台以上);从存储的方式上又有共享存储与分布式存储之分; 从机器的连接与通讯方式等方面又分成许多种. 因此,机器分类越细, 算法分类也会越细, 分类过细反而不易抓住主要矛盾.

实用上由于应用问题千差万别,计算机结构五花八门,并行算法尚未完善,故现在并行机上实际运行的是"串、并"混合算法, 而 M-SIMD 计算机应用的是同步与异步混合算法.

§ 1.3 并行算法发展的几个阶段

数值并行算法的发展粗略地可划分为三个阶段[4].

1.3.1 并行算法的预研期(1972 年以前)

1972 年以前是并行机前期, 同时也是并行算法的预研期. 这期间由于大型科学计算的需要,特别是解偏微分方程的需要,人们纷纷设计新型结构的计算机,一是阵列式结构, 一是流水线结构, 它们都是基于解差分方程组或线代数方程组的, 这时期并行算法的特点是建立在理想化的并行模型上,假定

(1) 所有处理机是相同的;

(2) 有任意大的存储器,且能同时为所有处理机存取;

(3) +−×÷运算都用一个时间单位;

(4) 其它操作(如存取、同步、通讯等)都不计时间.

有时还假定处理机台数是无限的(要多少台有多少台).

建立在这种公理化基础上的算法复杂性分析是这个时期评定一个算法好坏的基础. 所以，这阶段研制的算法是一种理想化的同步并行算法. 这类算法虽然没有对计算机加上"单指令流"的限制，但实际上绝大部分算法都是以当时正在设计的 SIMD 型计算机为背景的.

这一时期，采用下面的公式来度量算法的"加速" (Speedup)，

$$S_p = T_1/T_p \quad (\geqslant 1). \tag{1.6}$$

与效率 (Efficiency)，

$$E_p = S_p/p(\leqslant 1). \tag{1.7}$$

其中 p 为处理机台数，T_1 为串行算法计算某一个问题所需要的时间单位，T_p 为并行算法计算同一问题时所需的时间单位.

这里要注意的是，执行串行算法的单处理机与执行并行算法的 p 台并行处理机都是同样的处理机 (假设 1)，而其他操作又不计时间(假设 4). 上述公式当时能为人们所接受是因为在这种假设下，T_1 就是用一台处理机执行的时间，T_p 就是用 p 台处理机执行的时间(理想的).

如计算两个矩阵相乘

$$\begin{bmatrix} a_{11} & a_{12}\cdots a_{1n} \\ a_{21} & a_{22}\cdots a_{2n} \\ \vdots & \vdots \\ a_{n1} & a_{n2}\cdots a_{nn} \end{bmatrix} \begin{bmatrix} b_{11} & b_{12}\cdots b_{1n} \\ b_{21} & b_{22}\cdots b_{2n} \\ \vdots & \vdots \\ b_{n1} & b_{n2}\cdots b_{nn} \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12}\cdots c_{1n} \\ c_{21} & c_{22}\cdots c_{2n} \\ \vdots & \vdots \\ c_{n1} & c_{n2}\cdots c_{nn} \end{bmatrix}, \tag{1.8}$$

其中

$$c_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj}. \tag{1.9}$$

如果有 n^3 台处理机，其步骤如下:

第一步，每台处理机从内存中各取出相应的元素 a_{ik}, b_{kj};

第二步，每台处理机作乘法 $a_{ik}\ b_{kj}$;

第三步，用一部分处理机对它们求和;

第四步，将 c_{ij} 的结果写入内存.

由假设，第一、四步不花时间，第二步花一个时间单位，第三步

花 $\log_2 n$ 个时间单位，故

$$T_n^3 = 1 + \log_2 n.$$

若用串行算法，则第二步要作 n^3 次乘法，第三步要作 $(n-1)n^2$ 个加法，故

$$T_1 = 2n^3 - n^2, \tag{1.10}$$

$$S_n^3 = (2n-1)n^2/(1+\log_2 n), \tag{1.11}$$

$$E_n^3 = (2n-1)/n(1+\log_2 n). \tag{1.12}$$

如果限制处理机台数，比如 $p = n$，则 $S_n < S_n^3$，而 $E_n > E_n^3$. 由此看来，S_p 和 E_p 有时是两个互相矛盾的目标：若要得到最大加速的算法，可能要以降低算法的效率为代价.

在这一时期，递归计算问题是 SIMD 并行算法设计的一个难点，所以许多文献集中在这类问题的研究上.

值得提出的是 1969 年 D. Chazan 与 W. L. Miranker 的关于“混乱松弛法”(Chaotic relaxation)的文章[5]为 MIMD 计算机算法作了开创性工作.

W. L. Miranker[6] 对这一阶段的文献曾作了评述.

1.3.2 同步并行算法实践期、异步并行算法预研期 (1972—1981)

1972 年，阵列式计算机 ILLIAC-IV 在美国国家航空和航天管理局的 Ames 研究中心投入运行，这在并行机的发展史上是一个划时代的重要里程碑，自此进入了并行计算机初期. 同年，得克萨斯仪表公司的第一台先进科学计算机 TI-ASC 在欧洲运转，1974 年 CDC 控制数据公司的第一台流水线计算机（也称为向量计算机）CDC STAR-100 在美国 Lawrence Livermore 国家实验室 (LLNL) 投入使用. 1976 年克雷(CRAY)研究公司的第一台向量机 CRAY-1 在美国 LOS ALAMOS 国家实验室(LANL)运转，与此同时，STAR-100 改进为 CYBER 203, CYBER 205; CRAY-1 改进为 CRAY-1S, 且批量生产. 并行算法的研究进入了实践阶段. 首先是并行算法预研期(并行机前期)的并行算法得

到了检验而被重新估价，新的算法大都带有具体计算机的烙印．尽管都是向量机，但它们要求被计算的向量长度、向量数据的存储方式等都不相同．同是 SIMD 机，它们对并行算法也有不同的选择与评估标准．

经过实践的检验，人们发现，理想并行算法的公理系统与实际大为脱节，关于计算复杂性的研究也有问题．例如，前面谈到的矩阵乘法，其中第一步的取数和第四步的存数实际是不可忽略的．在第一步，数组 [**A**]，[**B**] 的每个元素必须同时为 n 台不同的处理机所索取．对于单进出口存储器（a Single-port memory）并行机，其处理机的每个地址同一时间仅允许一台处理机取数．这样一来，第一步"取数"就需要 n 个单位时间，故取数需要 n^3 个时间单位．

如果将 T_1 改为单台处理机上执行最佳算法解题的时间，T_p 表示在 p 台处理机的并行计算机上解同一问题执行并行算法的时间，则并行处理的"加速"为

$$S_p = \frac{\text{在单机上用最快的串行算法执行时间}}{\text{在 } p \text{ 台处理机上用并行算法执行时间}}, \qquad (1.13)$$

这一定义当时为大多数人所接受．

但"最快的串行算法"并没有定义．例如，解 n 阶线性代数方程的"最快串行算法"是什么？至少目前尚无定论，只有解一些简单的问题才有公认的答案，如解一般三对角方程组的追赶法等．

这阶段许多用户投入到并行算法的研究中去，如 NASA，LLNL 与 LANL 中从事大型科学计算的物理学家、空气动力学家、结构分析学家、计算数学家都转入并行算法的研究．在这阶段里许多同步并行算法形成了相应的软件．NASA，LLNL 和 LANL 在 CRAY-1 上的许多并行应用软件与标准程序正在 SIMD 机上为广大用户所引用．

综上所述，这时期并行算法的研究特点是：

（1）主要是研究同步并行算法，特别是向量式 SIMD 计算机算法；

(2) 并行算法的计算复杂性分析与性能评价有了新的内容;

(3) 并行算法与具体机器的相关性增加了;

(4) 同步并行算法正逐步软件化;

(5) 从事并行算法的研究人员增多,成分复杂了.

这时期由于美国卡内基-梅隆大学的 MIMD 计算机系统 Cmmp 问世,异步并行算法第一次得到了实践的机会,并行算法出现了新方向.

1982 年美国 LLNL 的 G. Rodrigue 主编的"并行计算"一书[7],其中收集了 13 篇论文,计算机体系结构与应用的覆盖面较广. 作者们从自己的专业领域出发总结了他们在各自使用的并行计算机上进行并行计算的经验,包括了CDC STAR-100,CRAY-1, ASC, ILLIAC-IV, CYBER203, C* 等各种 SIMD 与MIMD 计算机. 从这本书里,读者可以对这一时期的并行计算特点得到一个较全面的印象.

1.3.3 同步并行算法成熟期,异步并行算法实践期 (1982 年以后).

1982 年以来是并行机全盛期.

1. 巨型并行机出现

1982 年以 CRAY-1S 为处理机的多机系统 CRAY X-MP 出现(有 2 处理机与 4 处理机两种型号),并于 1984 年在 LLNL 等实验室投入使用.

1983 年 Denelcor 公司的 HEP 批量生产,并在 LANL, Argonne 国家实验室和导弹研究室运转,这是第一台商品化的 MIMD 计算机,也是第一台进行过最系统研究的 MIMD 计算机. 有关它的系统结构、性能、程序设计与语言和算法与应用,在 J. S. Kowalik 主编的一书[8]中作了较详细的介绍.

1985 年更为先进的 CRAY-2 问世,它是由 4 台改进型的 CRAY-1S 紧密耦合成的 MIMD 计算机. 1986 年即有两台分别安装在 LLNL 与 NASA 的 Ames 研究中心. 1987 年又有一台

试读结束: 需要全本请在线购买: www.ertongbook.com