向量化理论

范植华 著

科学出版社

向量化理论

范植华 著

科学出版社
1990

内 容 简 介

串行运算向量化与巨型机-向量机硬件相伴而生，它已经成为计算机科学中的崭新课题．本书以作者多年来的工作为背景，详尽地阐明了作者提出的以数组元素为结点的向量化理论，系统地描述了据此理论研制而成的向量化下标追踪法，许多内容尚属首次公开发表．作为国内外有关这一领域的第一部专著，本书也用适量篇幅介绍了坐标方法、超平面方法和相关分析方法所代表的以下标变量为结点的向量化理论，旨在使读者了解迄今为止向量化领域的全貌．

本书尽管论及的是计算机科学中的前沿分支，但内容深入浅出、层次清晰，可供计算机科学与软件工程专业的科技人员和大专院校有关专业的师生阅读、参考．

向 量 化 理 论

范植华 著

责任编辑 那莉莉

科学出版社出版

北京东黄城根北街16号

邮政编码：100707

中国科学院印刷厂印刷

新华书店北京发行所发行 各地新华书店经售

*

1990年7月第 一 版 开本：787×1092 1/16

1990年7月第一次印刷 印张：18 1/2 插页：3

印数：0001—1050 字数：431 000

ISBN 7-03-001655-6/TP·125

定价：20.80 元

序　言

在计算技术的发展史上，向量巨型机已经历了近20年的灿烂历程．因为其运算速度比其它指标大体相同的标量机高出几个数量级，所以向量巨型机在国民经济与国防建设的几乎所有尖端部门都获得了广泛的应用，成为反映一个国家综合性工业与科技发展水平的主要标志之一．银河亿次级系列计算机正是我国的向量巨型机．

为了充分地发挥向量计算机硬件的并发优势，必须有相应的软件支撑，用以自动地识别与改写程序中的向量方式并行成分．于是，一个崭新的软件课题——串行运算向量化应运而生，构成了计算机科学中一个带有相当的理论深度且具有现实的应用背景的前沿分支．

自 L. Lamport 教授于1973年发表坐标方法以来，直接或间接论及向量化的文献为数不少．但是，作者迄今在国内外尚未看到一本专门、系统地阐明这一领域的内容、意义与方法的著作，本书是一个尝试．

在多年从事计算机软件的理论研究和工程实践的基础上，作者于1982年4月在中美计算机软件工程学术讨论会上，首次公开了一套全新的、以数组元素为结点的向量化理论以及该理论指导下的下标追踪法．与早先的以下标变量为结点的坐标方法相比，它不仅大幅度地增强了对赋值语句的识别能力，而且也圆满地解决了控制结构的向量化问题，形成了一个从简洁的公理系统出发，内涵丰富的数学演绎体系．

本书共10章．第一章介绍硬件前提——向量巨型机．第二章介绍软件前提——向量高级语言．第三章描述基本思想、基本理论和基本方法，主要提出构造时序层次的公理系统——繁衍规则，由此出发证明年长顺序定理和判别准则．第四章描述工程化所遭遇的难关及其突破，核心是四个层次片断定理．第五章运用现代集合论工具证明强化定理，这是克服程序中不确定成分的有力武器．第六章运用强化定理解决包含逻辑 IF 语句、单臂块 IF 语句和双臂块 IF 语句的循环的向量化问题．第七章具体探寻由算术 IF 语句和无条件 GOTO 语句所构成的三叉控制转移，向结构程序设计的连接型、选择型和重复型基本模式的程序变换，进而借助于第六章的成果解决它们的向量化问题．由计算 GOTO 语句和无条件 GOTO 语句所构成的多叉控制转移亦可照此办理．第八章和第九章分别研究时序层次结构上的两种极端以及各自特殊的处理办法．第十章是最后一章，它把前面的成果全面地推广于多重循环．

在向量化研究的全过程中，作者得到中国科学院学部委员、国防科技大学慈云桂教授，中国科学院软件研究所唐稚松研究员和许孔时所长，北京大学吴允曾教授的热情鼓励与支持；中国科学院数学研究所陆汝钤研究员仔细地审阅了本书，提出了宝贵意见．在此，作者一并向他们致以衷心的谢意．作者还感谢国防大学外语教研室陆佑珊副教授，她承担了本书大量的资料整理工作．

作者不避浅陋，把研究心得汇集成书奉献给读者．不当之处，敬请指正．伴随向量巨

型机日益实用化与小型微型化，作为一门新兴学科，向量化的研究方兴未艾，本书若能成为一块"引玉之砖"，作者甚幸。

范植华

于北京系统工程研究所

1988年11月20日

目　　录

第一章 引 论

在详细地讨论向量化理论之前，应当对若干相关的内容加以叙述．它们包括：巨型计算机的兴起，巨型计算机与向量计算机之间的联系，向量计算机具备的一种新机制——向量机制，向量机制在程序设计语言中的体现，向计算机科学理论提出的新课题，向量化理论的基本内容及其用途等．

本章也是全书的缩影，整个写作计划(后续章节的安排)将穿插于其中加以说明．

1.1 巨型计算机的兴起

随着人类的社会活动、物质生产、商品流通、科学技术与文化教育的不断发展，有用的数据信息及其加工处理的工作量随时间指数地增长，已进入天文数字的量级．

例如，在航空与航天领域内，飞行器的设计涉及空气动力学的问题，其计算异常复杂．过去对这些问题总是忽略一些次要因素，进行近似计算，实际数据靠风洞试验获取．原先设计飞机的风洞试验约需数百小时，而设计航天飞机的风洞试验却需数万小时，研制周期与研制费用均达到无法承受的地步．更主要的是伴随飞行高度和航速的增加，风洞试验的局限性日益突出，气流的压力、密度、速度和温度，气流的均匀性，洞壁干扰以及气动弹性形变的影响，都给风洞试验造成困难．借助于先进计算机的“计算空气动力学”，不仅可较精确地求解空气动力学微分方程组，大大减轻风洞试验的负担，而且可比风洞更真实地模拟气流场，提高设计质量，缩短设计时间，降低研制费用．

又如在核物理学的研究领域内，仅模拟亚原子粒子的运动，往往就要进行多达 10^{12} 次的计算．核武器或核反应堆的设计需要求解偏微分方程组．过去囿于计算机的性能，在模拟其物理过程时做了过多的近似，把三维模型简化成一维模型，网格分得很粗，致使设计精度严重受损，必须通过真实的核爆炸加以修正(这是核大国频繁进行核试验的症结所在)．欲提高设计水平，须采用能更精确、更全面地描述物理过程的二维或三维模型．加之网格分得更细，时间步长更小，从而导致数据量和运算量的大幅度增加．

又如在数学的研究领域内，人们迄今知晓的最大素数有 65050 位十进制数码，这足以印满两页报纸．为了发现这个素数，人们需要进行数万亿次运算，即便在当今世界上最大的一台计算机(每秒钟可作 4 亿次运算)上进行计算，仍要耗费 3 个多小时．

又如计算机用于医疗诊断的轴向层析扫描设备，根据 X 射线从人体各点折回信号的衰减，经计算机处理成大量剖面图，以产生可旋转的立体图象，供医生在显示屏幕上观察体内各部位，进行诊断．若想取得 1 毫米鉴别率的三维图象，其计算机的运算速度应达到每秒 20 亿条指令．

又如计算机用于人口普查．我国现有 10 亿人口，倘若描述一个人的社会关系、个人经历、生理特征等等方面共需 100 个数据项，全国则需 1000 亿个数据项，仅计算各种统计量的运算次数就已大得惊人了．

再如计算机用于密钥密码系统．如果选择 56 位密钥长度，以 6 微秒试验一个密钥的速度进行密钥试验，那么尚需耗时 6800 年．

其他诸如气象预报、地球物理、石油工程、海洋探测、计算化学、遥感技术等等领域，对于高性能的计算机亦提出类似的要求．

让我们考察一个通俗易懂的例题．

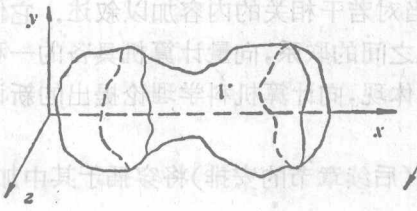

图 1.1

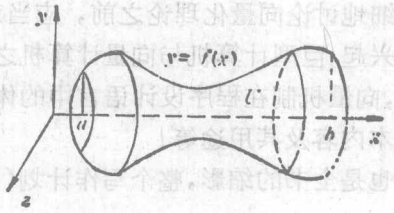

图 1.2

空间中不规则形体 V 的质量 M 原本是由三重积分

$$M=\iiint_V \rho(x,y,z)\mathrm{d}v$$

确定的，其中 $\rho(x,y,z)$ 为体密度函数．所谓“简化成一维模型”乃指：

1. 不规则形体 V 简化成旋转体 U．假设旋转轴为 x 轴，旋转曲线方程为

$$y=f(x) \qquad a\leqslant x\leqslant b$$

2. 沿旋转轴方向把密度简化成轴向分布，即体密度函数 $\rho(x,y,z)$ 简化为线密度函数 $\rho(x)$．于是，求质公式简化成简单积分：

$$M=\int_a^b \rho(x)\cdot \pi f^2(x)\mathrm{d}x$$

从图 1.1 和图 1.2 可知，这样的简化一般说来已失去定量分析的意义，仅能起到定性分析的作用．所以，实际数据要靠试验获取．倘若步长加密 10 倍，再从一维恢复到三维，则这个简单问题的数据量与运算量就将增大 10^3，即 10 万倍之多！

无数事实充分地表明，现代科学技术的数学化趋势日益加强，数学方法已经广泛地运用于各门各类学科的研究．精确地进行定量分析以促使人们思维的精确化，业已成为现代思维方式的主要特征．作为这种定量性思维必不可少的计算工具，高性能的计算机便应运而生．

从 1951 年起，计算机的速度平均每两年翻一番．70 年代以来，基于半导体和集成工艺的飞速发展，高性能的计算机如雨后春笋般地涌现出来．其中，有一类机器在提高运算速度方面不断取得突破，使得单位时间(秒)内的平均运算次数跟数据信息及其加工处理工作量相适应，如天文数字般地增长，正从千万次级、亿次级向十亿、百亿、千亿的量级迈进．这一类机器通常被人们称作“巨型计算机”[1]（简称巨型机）．除去对于存储容量和数据吞吐能力有相应的要求之外，巨型机的运算速度一般认为应该在每秒 2 千万次以上．据不完全统计，至 1984 年上半年止，世界上共有 60 多台巨型机在运转．

巨型机业已经历设计、试制和实用化批量生产的发展阶段，世界各国现又竭力推进巨型机的微型化．例如，美国正在研制能驾驶汽车的超级电脑，体积仅等于一个普通家用电冰箱，运算速度每秒 1 亿次，比同类电脑快两个数量级．巨型机告别庞然大物的时代即将

来临.

综上所述，巨型机已经成为计算机领域中一个生命力旺盛的新方向. 巨型机的兴起既是现代科学技术发展的必然要求，又是现代科学技术进步的必然结果.

巨型机是我国四个现代化建设迫切需要的关键设备，它已经成为我国战略武器研制、航天航空飞行器设计、国民经济的预测和决策、能源开发、天气预报、图象处理、情报分析，以及各种科学研究的强大的计算工具.

1.2 巨型计算机与向量计算机

巨型机是各类计算机中性能最好，规模最大，功能最强的电子计算机，是解决科学计算、工程设计和数据处理等特大课题的重要设备. 表 1.1 列出了迄今美国和日本研制的巨型机系统概况.

加快主频是巨型机提高速度的途径之一. 例如，CRAY-1 计算机主时钟周期缩短为 12.5 纳秒（ns），即主频高达 80 兆赫兹[2,3]，这是一般的计算机所望尘莫及的. 比方说，迄今最先进的微型、小型计算机，主频仅 2.5 兆赫兹左右. 但是，巨型机获得超高速的主要手段是大幅度增强并行处理的能力.

巨型机惊人的并行处理能力，基本上来自下列行之有效的体系结构[1-3]：

1. 流水线技术. 将许多指令所重复的时序过程分解为若干共有的子过程，譬如说取指令、译码、取数、执行等，交由若干专用的功能块去重叠执行.

2. 多功能部件. 配置许多专用的算术运算和逻辑运算功能部件，在操作数不冲突的前提下，不同操作的指令可交由这些功能部件同时执行.

3. 阵列结构. 一台计算机由同构型的多个处理单元组成，处理单元之间存在着有限的连接. 它们在一个控制部件的统一指挥下，同时执行同一种操作.

4. 多处理机系统. 每台处理机有各自的指令流和数据流. 它们可以同时执行各自独立的程序，也可以组合起来共同完成一个大课题的处理任务.

在这些体系结构中，还可配合以诸如指令缓冲站，超高速缓冲存储器，多模块大容量交叉访问存储器和素数模存储器结构，以增强存储器操作的并行度，尤其可以配备向量机制，以增强处理器操作和处理器操作步骤的并行度，构成向量计算机（简称向量机），这是巨型机目前的主要形式[4]. 例如，美国研制的 ILLIAC-IV，STAR-100，CYBER-203 和 CYBER-205，TI-ASC，CRAY-1，CRAY-1S，CRAY X-MP 和 CRAY-2S 机，我国研制的 757 型机（图 1.3，图 1.4）[5] 和银河机（图 1.5），以及日本研制的 FACOM VP-100 和 FACOM VP-200 机[6]都是向量机. 反之，向量机制具有至少高出一、二个数量级的运算并行度，能够明显缩短机器周期. 例如，FACOM VP-100 的机器周期，即 64 位长浮点运算的加法、乘法处理时间仅为 7.5 纳秒，FACOM VP-200 的运算能力还要增强一倍. 因此，向量机同时也是人们一般认为的巨型机.

下一节将从使用的角度说明向量机制的并行功能.

图 1.3—图 1.5 附本书末.

表 1.1　美国和日本研制的巨型计算机系统概况

型　号	研制单位	研制时间	运算速度（万次/秒）	主存容量（万字）	使用单位或用途
ILLIAC-IV	伊利诺伊大学、鲍勒斯公司	1966—1973	15000	13	艾姆斯研究中心
STAR-100	控制数据公司	1965—1974	5000(64位浮点加) 10000 (32 位 浮 点 加)	50—100	劳伦兹辐射实验室研制核武器，兰利研究中心用于航空、航天机设计和环境模拟
TI-ASC	得克萨斯仪器公司	1966—1972	5000	13—100	GSI 奥斯汀公司等
CRAY-1	克雷研究公司	1972—1976	8000(浮点结果)	25—400	劳斯阿拉莫斯实验室研制核武器用，国家气象研究中心，欧洲中期天气预报中心
BSP	鲍勒斯公司	—1978	5000(浮点结果)	50—800	
STARAN	古德依尔宇航公司	—1972	30000 (32 位 浮 点 加)		罗马航空发展中心用于空中交通管制，陆军地形描绘实验室研究自动绘图、图象处理、立体摄影测量、信息检索等，NASA 约翰逊宇宙中心
PEPE	系统发展公司、鲍勒斯公司	1971—1976		1	弹道导弹防御先进技术中心
CRAY-1S	克雷研究公司	—1978	8000(浮点结果)	50—400	
CYBER-203	控制数据公司	1975—1979	5000 (64 位浮点结果) 10000(32位浮点结果)	50—200	NASA，凯特兰空军基地，舰队海洋学中心
CYBER-205	控制数据公司	—1981	40000(64位浮点结果) 80000(32位浮点结果)		特别适于核武器研制，核电站安全运行，飞机及飞行器设计，气象预报，石油勘探等
CRAY X-MP	克雷研究公司	—1978		200—400	
CRAY-2S	克雷研究公司	—1985	约10万(浮点运算)	400—3200	
S-1	斯坦福大学、加州大学、劳伦斯利弗莫实验室	1976—1984	约10万(浮点运算)		美国海军
HEP	邓纳肯公司	—1981	1000—16000 (指令)		美军弹道实验研究室
NASF	鲍勒斯公司、控制数据公司	1977—1986	约10万以上 (浮点运算)	3000 以上	
MPP	古德依尔宇航公司	—1982	60万(8位整数加) 1.6 万 (32 位浮 点加)		用作卫星图象处理
FACOM VP-100	日本富士通公司	—1983	2.5 万(浮点运算)	800—3200	
FACOM VP-200	日本富士通公司	—1983	5 万(浮点运算)	800—3200	
S810/10	日本日立公司	—1983	3.15万(浮点运算)		
S810/20	日本日立公司	—1983	6.3 万(浮点运算)		

注：未标国名者均系美国.

1.3 向量计算机的向量机制

向量机装配有向量处理部件，以日本富士通公司研制的超高速巨型机 FACOM-VP 100/200 为例，它们包括：向量加法运算器、向量逻辑运算器、向量乘法运算器、向量除法运算器、屏蔽寄存器和加载存储器，以及用作保护向量运算数据的向量寄存器(图 1.6)[7]. 凭借这些硬件资源，向量机向用户提供向量运算和向量链接等向量机制.

向量是由同类型数据组成的非空有序集合. 向量运算以向量作为运算对象，向量长度可达数百之多，即数百个以向量元素作为对象的标量运算并发地执行. 例如，从程序的角度来看，CRAY-1 拥有 41 条向量运算指令，配置 8 个向量寄存器，每个向量寄存器由 64 个 64 位的单元 (即向量元素) 组成[1-3]；FACOM-VP 的向量寄存器相当于 CRAY-1 的 16 倍，拥有 82 条向量运算指令[7].

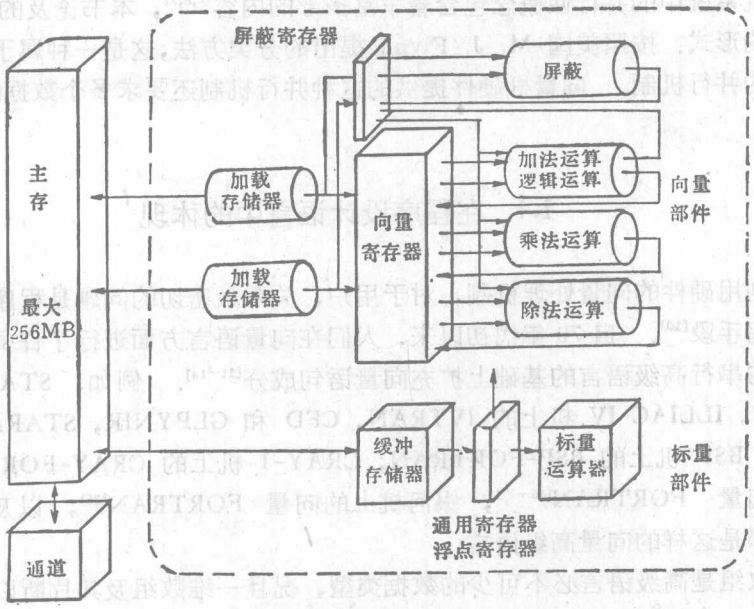

图 1.6 FACOM VP-100/200 处理部件框图

在向量运算的前提下，向量机往往还采用向量链接技术[1,2]. 当前一条指令的结果是下一条指令的操作数(这在程序中是常见的现象)时，向量寄存器可以在同一时钟周期内既接受一个功能部件送来的运算结果，又可把这一结果作为下一次运算的源操作数送交另一个功能部件，从而把两个或两个以上的功能部件链接起来运算，以解决操作数相关的矛盾.

例如，我们欲求出 1 行 5 列的矩阵 X 与 5 行 125 列的矩阵 Y 的矩阵乘积，结果赋予 1 行 125 列的矩阵

$$Z_{1\times 125} = X_{1\times 5} Y_{5\times 125}$$

在标量 FORTRAN 中，用到二重循环：

```
DO 1 I = 1, 125
Z(I) = 0
```

```
      DO 1 J = 1, 5
    1 Z(I) = Z(I) + X(J) * Y(J, I)
```

不计对循环控制变量的修改和判别的开销，尚需执行 125 + 125 * 5 = 750 个语句，而在向量 FORTRAN 中，仅用到 6 个语句构成的简单序列：

```
Z(1:125) = 0
Z(1:125) = Z(1:125) + X(1) * Y(1, 1:125)
Z(1:125) = Z(1:125) + X(2) * Y(2, 1:125)
Z(1:125) = Z(1:125) + X(3) * Y(3, 1:125)
Z(1:125) = Z(1:125) + X(4) * Y(4, 1:125)
Z(1:125) = Z(1:125) + X(5) * Y(5, 1:125)
```

加之借助于多个向量寄存器作链接执行，进一步利用时间上的重叠性，更加快了运算速度．后续章节将说明这些语句的含义．

计算机系统中的并行性概念包含着丰富多彩的内容[2,8,9]，本书论及的并行性专指其向量处理的形式．按照美国 M. J. Flynn 提出的分类方法，这是一种属于单指令流多数据流类型的并行机制． 向量机硬件提供的这种并行机制还要求多个数据的存储地址等间隔．

1.4 在程序设计语言中的体现

为了利用硬件的向量处理机制，对于用户，关系最密切的问题是程序设计语言提供怎样的描写手段[10]． 自 70 年代初以来，人们在向量语言方面进行了许多研究，其主流是在已有的串行高级语言的基础上扩充向量语句成分[11-14]． 例如，STAR-100 机上的 LRLTRAN，ILLIAC-IV 机上的 IVTRAN，CFD 和 GLPYNIR，STARAN-IV 机上的 STARAN，BSP 机上的 BSP-FORTRAN，CRAY-1 机上的 CRAY-FORTRAN[15]，757 型机上的向量 FORTRAN[16-25]，银河机上的向量 FORTRAN[26]，以及向量 ALGOL 60[27-28]，都是这样的向量高级语言．

因为数组是高级语言必不可少的数据类型，况且一维数组及其片断以及多维数组在某一维上的片断都是向量，所以扩充多以数组作为运算对象．比如，CRAY-FORTRAN 的数组处理使得 FORTRAN 语言[29]对标量定义的运算大多可用于数组：数组与数组、数组与标量之间可以施行加、减、乘、除四则运算和逻辑运算，数组可以作为赋值的对象，对数组元素及其运算进行条件选择，数组作为通用函数的入口参数，作为过程参数，等等．

在上一节的例题里，语句

```
Z(1:125) = 0
```

是一个向量赋值语句．其中，1:125 是三元挑选符 1:125:1 的缩写，表示从 1 开始，每次增加 1，直至 125 为止，以此方式为参加运算的数组元素挑选下标值．该语句把 0 并行地赋予 Z(1), Z(2), ···, Z(125)．所谓并行，本书乃指同时、无顺序地一次完成．

类似地，语句

```
Z(1:125) = Z(1:125) + X(1) * Y(1, 1:125)
```

是一个向量运算语句．该语句把

```
Z(1) + X(1) * Y(1, 1)
Z(2) + X(1) * Y(1, 2)
   :
Z(125) + X(1) * Y(1, 125)
```

125 个乘加运算并行地计算出来，再并行地赋予 Z(1), Z(2), ···, Z(125)。

又如数组赋值语句

```
Y = 1
```

是数组赋值语句

```
Y(1:5:1, 1:125:1) = 1
```

的缩写，表示分别按照三元挑选符 1:5:1 和 1:125:1，各自独立地挑选第一维的下标值和第二维的下标值，使得 1 被并行地赋予 5*125 个数组元素。

上述各个向量/数组处理都是单指令流的，它们各自处理的多个运算数据在存储中等间隔，这一点是由编译系统数组编译和存储分配规则保证的[30]。譬如说二维数组按列分配：在列方向上连续配置，间距为 1；在行方向上虽不连续，但间距恒为列长。

第二章将详细地描述这种语言。

1.5 向计算机科学理论提出的新课题

常言道，一心不可二用。这个童叟皆知的道理生动地说明人们传统的思维习惯是串行的，即便思维可并行的事件，这些事件还是串行地、一件一件地思维出来的。譬如说“星期天去公园散步，顺便赏花”。“散步”与“赏花”是可并行的事件，但是人们在安排这两件事时，仍是串行思维的。人类对于神经系统的研究揭示，人体仅仅在下意识一级具有并发机能。然而，向量机与向量高级语言却要求我们用并行的语句描述串行思维出来的算法，下面的例题表明简单的替换会导致错误。

设有 FORTRAN 中的一个整型数组 A(0:101)，其初值如下：

```
A(0) = 0
A(1) = 1
   :
A(101) = 101
```

于是，串行程序

```
    DO 2 I = 1, 100
  2 A(I) = A(I + 1)
```

把 A(2) 至 A(101) 逐个赋予 A(1) 至 A(100)。相应的并行程序应该是

```
  3 A(1:100) = A(2:101)
```

又有串行程序

```
    DO 4 I = 1, 100
  4 A(I) = A(I - 1)
```

把 A(0) 至 A(99) 逐个赋予 A(1) 至 A(100)。相应的并行程序应该是

```
  5 A(1:100) = A(0:99)
```

这四段程序各自的运行结果如表 1.2 所示。

表 1.2　四段程序运行结果比较

内容 \ 下标值 / 语句	0	1	2	3	4	5	6	7	…	95	96	97	98	99	100	101
初值	0	1	2	3	4	5	6	7	…	95	96	97	98	99	100	101
DO 2	0	2	3	4	5	6	7	8	…	96	97	98	99	100	101	101
3	0	2	3	4	5	6	7	8	…	96	97	98	99	100	101	101
DO 4	0	0	0	0	0	0	0	0	…	0	0	0	0	0	0	101
5	0	0	1	2	3	4	5	6	…	94	95	96	97	98	99	101

从表 1.2 可知，貌似相同的一对串行程序在向量化方面的性态是截然不同的：DO 2 循环与向量语句 3 的结果一样，DO 4 循环与向量语句 5 的结果却迥然而异！

又如二重循环

```
      DO 6 I = 1, 100
      DO 6 J = 1, 100
    6 B(I, J) = B(I, J - 1) * *2 - 4 * AC(I, J)
```

和

```
      DO 7 J = 1, 100
      DO 7 I = 1, 100
    7 B(I, J) = B(I, J - 1) * *2 - 4 * AC(I, J)
```

是等效的。但是，对于前者的内层循环，运算处处递推，它的并行算法计算树如图 1.7 所示，深度为 99；对于后者的内层循环，运算各自独立，它的并行算法计算树如图 1.8 所示，深度为 1。显然，二者所含的内在并行性有天壤之别。

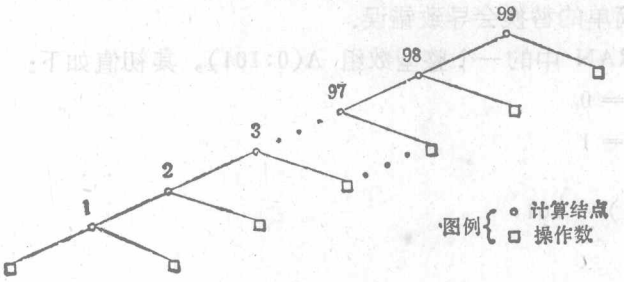

图 1.7　DO 6 内层循环的并行算法计算树

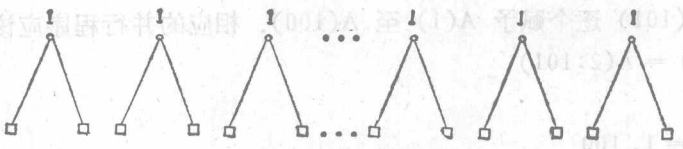

图 1.8　DO 7 内层循环的并行算法计算树

综上所述，伴随向量机与向量高级语言的问世，为了从数学上提供包含尽量多的内在并行性的计算方法，在计算数学领域，并行算法获得了蓬勃发展[31]；为了自动识别隐匿在串行程序里的内在并行性，在计算机科学领域，向量运算的自动识别（简称向量识别，或向

量化)成为一个崭新的课题[32].

并行算法与向量识别是相辅相成的[32,33]. 一般说来，一道程序倘若在数学上采用并行算法越好，那么，经过向量识别产生的向量运算也越多. 二者之间的关系在地质学里可以找到形象的类比. 煤层是在石炭二叠纪生成的，今天的地质勘探则以地质学理论为武器去寻找煤层. 并行算法大体上相当于石炭二叠纪地球环境，向量识别大体上相当于地质勘探. 一个再好的地质学理论，也不能帮我们在根本没有煤层的星球上勘探出煤层；同样地，一个再好的向量识别理论，也无法为我们从根本没有并行潜力的程序里发掘出向量运算.

1.6 向量化理论的基本内容

如果把单指令流单数据流的串行机比喻为只有一个座位，仅能供一个人吃饭的单间餐室，则可把向量机比作有成百上千个座位，可供成百上千个人同时进膳的大饭店. 用餐的单位是“人顿”，即“一人吃一顿”. 当我们接到一张包伙单，譬如说预定 100 人顿的用餐量时，我们并不能肯定饭店的 100 个座位能够同时启用，我们首先得分析这 100 人顿用餐量的分配计划. 仅考虑两种极端的情形，倘若是 100 人吃同一顿，则可安排 100 个座位让他们同时吃；倘若是同一人吃一百顿，那么饭店空位再多也必须为他一顿一顿地开饭. 连叫他同时吃两顿都不可能，更谈不上同时吃 100 顿了. 原因何在？不言而喻，这是因为前面那 100 人顿分给 100 人吃，互不相干；后面那 100 人顿由同一人吃，各餐之间存在着严格的约束关系：仅当早餐入肚活动几小时再吃的那餐饭才是午餐；仅当午餐入肚活动几小时再吃的那餐饭才是晚餐.

向量机的向量机制具有类似的性质. 上一节的 DO 7 内层循环相当于 100 人吃同一餐；DO 6 内层循环相当于同一人吃 100 餐. 前者是可向量化的，后者是不可向量化的.

因此，向量化理论是在深入剖析源程序错综复杂的数据依赖关系的基础上，自动识别其中具有向量方式并行性的成分，并且通过把这些成分自动改写成为向量形式而得到向量化的目标程序. 显然，这是一种旨在提高运行功效的程序变换[34]，源程序与目标程序的等价性是必要前提. 如果把向量识别狭义地理解为识别工作，那么向量化的含义要广泛些，它囊括了识别与改写的全部内容.

根据程序语言理论[35]，源程序是源语言中的句子，目标程序是目标语言中的句子. 所以，源语言和目标语言的表达能力从总体上决定了源程序和目标程序中数据依赖关系的类型与复杂程度，也就决定了向量化理论的识别与改写能力. 迄今为止，所有的向量化算法均以 FORTRAN 语言作为背景，本书亦不例外. 我们所讨论的向量化理论以 FORTRAN 77 作为源语言[29]，以扩充有类似于 CRAY 数组处理成分[15]的 VFORTRAN 作为目标语言(图 1.9). 推导出来的向量化算法大体上适合于各种算法语言. VFORTRAN 的细节将在第二章里介绍.

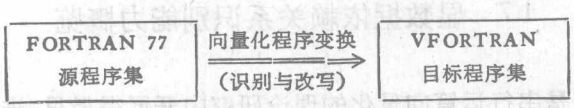

图 1.9 向量化实现的程序变换

纵观向量化研究的历史，其内容可以归纳为下述三方面：

1. 显数据依赖关系，

2. 隐数据依赖关系，

3. 识别指导性指令.

如 1.2 节所述，向量机硬件提供的向量机制具有两个特点：

1. 单指令流多数据流，

2. 多数据在存储中等间隔.

程序设计的经验告诉我们，这是 DO 循环中的语句最容易满足的要求. DO 循环中的每一个语句，给出一条指令(或一个指令序列). 这条指令(或这个指令序列)经由循环往复作用于多个数据之上，便构成单指令流多数据流. 只要步长不变，在编译系统的配合之下，这样的多数据在存储中正是等间隔的. 加之循环集中了程序主要的计算量，使得向量化研究无不瞩目于 DO 循环的向量化. 当然，借助于其他语句（譬如说 GOTO）也能构成循环，不过根据结构程序设计原理[36-38]，它们基本上可以通过源语言自身的程序变换转化为 DO 循环的形式.

由循环体内的语句引起的，作用域限于循环体的一类数据依赖关系，称作显数据依赖关系. 70 年代以来，已经有不少识别显数据依赖关系的方法. 下标追踪法[39]建立在以数组元素作为分析结点的基础上，解决了结构程序设计所有的三种基本模式[36](连接型、选择型和重复型)的向量化问题，构成了本书的核心：第三章介绍它的基本思想、基本理论和基本方法[40,41]，第四章介绍它在工程化中的若干技术[40,42]，第五章引入强化定理[40,43]，这是下标追踪法克服不确定因素的得力工具，第六章描述各种 IF 语句的向量化算法[40,44]，第七章探讨三叉控制转移的程序变换[45]，第八章研究离散层次[46]，提供可原形向量化[47]、可准原形向量化[48]与可反原形向量化算法，第九章引入冗余项与冗余循环的概念，证明了同态定理[49]，第十章是最后一章，讨论向多重循环的拓广[40,50]. 其他诸如坐标方法[51]、超平面方法[52]、相关分析方法[53]等，均以下标变量作为分析结点，是对编译技术通常以语句作为分析单位的思想所作的自然发展，至今均局限于赋值语句循环的范畴内，我们将在 1.7 节里对它们的识别能力作一概述.

由循环体外的语句引起的、作用域超越循环体范围的一类数据依赖关系，称作隐数据依赖关系[54]. 为了确保源程序与目标程序的等价性，向量化理论必须考虑隐数据依赖关系.

现阶段的向量化理论，均限于“静态”分析，即“阅读”而非“运行”源程序来加以识别与改写处理. 静态处理如何利用动态信息，这是程序理论十分关心的一个课题. 识别指导性指令[55]正是用户提供补充信息，主要是静态识别所缺乏的动态信息的手段.

由于隐数据依赖关系和识别指导性指令具有相对的独立性，本书限于篇幅，未收入此二方面的内容，对此感兴趣的读者可参阅文献 [54] 和 [55].

1.7 显数据依赖关系识别能力概览

显数据依赖关系是串行运算向量化的理论研究中开展得最早，进行得最活跃，也最富于成果的领域. 在整体向量化[40,41]，即把 DO 循环作为整体考虑其可向量化性质的范畴