全国计算数学学会并行算法专业委员会

全国第四届并行算法学术会议论文集

航空工业出版社

全国计算数学学会并行算法专业委员会

全国第四届
并行算法学术会议论文集

航空工业出版社

1993

(京)新登字 161 号

全国第四届并行算法学术会议论文集

全国计算数学学会
并行算法专业委员会 编

航空工业出版社出版发行
(北京市安定门外小关东里 14 号)
—邮政编码:100029—
南京航空航天大学印刷厂印刷

1993 年 11 月第 1 版　　1993 年 11 月第 1 次印刷
开本:787×1092　1/16　　印张:24.375
印数:1—500　　字数:608 千字

ISBN 7-80046-694-9/Z·112

定价:30.00 元

《全国第四届并行算法学术会议论文集》

编 辑 委 员 会

序

科学技术的发展，特别是在近代高科技领域，对计算机的要求愈来愈高，而现在计算机的性能还远远不能满足这种要求。70年代以来，计算技术进入了向量化与并行化发展阶段，传统的计算方法也随之发生变化，逐步向量化与并行化，这使得计算功能在科学与工程计算中成量级地提高。有理由认为，大规模的并行计算将是今后计算技术发展的主要途径。

我国的并行计算无论在硬件、软件以及算法方面均起步较晚，与西方发达国家相比，存在着很大的差距，但近十几年来，向量算法与并行算法研究逐步开展起来了，成立了计算数学学会并行算法专业委员会，并先后召开过三次全国并行算法会议。我国在并行算法的基础理论和应用研究方面均取得了可喜的进展。随着国产的或从国外进口的各种型号向量与并行计算机投入使用，并行计算已在若干重要部门如航空航天、气象、石油勘探等领域得到了广泛的应用，这次在南京航空航天大学举行的第四届全国并行算法会议就是一个明显的例证。会议已收到学术论文90多篇，内容涉及到计算数学的各个分支，面很广，水平也很高。与此同时，在全国各高等院校、科研院所与生产部门都已成长了一批以中青年为主的并行计算科学工作者，可以期望这支队伍将会在今后的大规模科学计算中发挥愈来愈大的作用。这次全国并行算法会议的召开以及会议论文集的出版，无疑将会推动我国并行计算事业迅速地向前发展。预祝第四届全国并行算法会议成功召开，并祝愿所有在并行计算领域中辛勤劳动的科技工作者，今后取得更大的成绩，为我国科学与工程计算事业攀登上新的高峰，作出历史性的贡献。

石钟慈

1993.8.29

前　言

并行算法研究与应用从60年代后期到现在只经过廿多年时间，涓涓细水已汇成滔滔洪流，并行计算已获得了举世瞩目的发展。巨型机、小型超级计算机以及各种并行计算机系统不断推出，现已进入普及应用阶段。并行算法研究也取得了长足进步，研究领域和应用范围迅速扩大，受到各界学者越来越大的重视和关注。为适应这种发展形势，国内从1987年召开我国第一届并行算法学术交流会以来到现在已连续召开了四届。本届学术交流会就其论文数量、质量、涉及的领域与应用范围比前几届有了很大发展。这说明我国并行算法发展与应用一直跟踪着国际先进水平。

本届学术交流会是由中国数学学会计算数学学会和国防科工委计算机专业组共同主办的全国性学术会议。会议将于1993年12月上旬在南京召开，全国各个领域从事计算数学、计算机应用和计算机软件的知名专家、学者将参加这次会议，论文集的出版对我国并行算法学科的发展将产生很大影响。

本届会议论文经前后两次审稿，在初审基础上，由计算数学学会并行算法专业委员会全体委员集中终审评定，确定论文集的入选论文，其中特邀大会报告9篇，专题论文90篇。本论文集涉及范围很广，比较全面地反映了我国近二年来并行算法研究与应用所取得的最新成果。

本届论文集的出版，得到了中国数学学会计算数学学会和国防科工委的热情关怀与支持，得到了国防科技大学、科学院计算中心、南京航空航天大学、石油天然气总公司物探局研究院、北京应用物理与计算数学研究所、航天工业总公司204所、华中理工大学、中国工程物理研究院计算机应用研究所、航空工业出版社等单位的大力赞助与支持，在此表示衷心感谢。

专业委员会委托南京航空航天大学十系负责筹办会议和论文集的出版，十系和南航御苑宾馆的许多同志为本届会议召开和论文集出版做了大量工作、尤其是侯树勤同志、邓绍忠同志以及南航印刷厂的许多同志为论文集的编辑、校对、排印、付出了辛勤劳动，编委会对此一并表示衷心感谢。

《全国第四届并行算法学术交流会论文集》
编辑委员会
1993.10

目　　录

(一)

(二)

(三)

(四)

(五)

并行算法及其研究前沿

陈 国 良

（中国科学技术大学计算机系　合肥　230027）

本文重点介绍并行算法理论、研究方法和设计技术及其研究的新领域。

一、并行计算与并行计算模型

1. 基本定义

具有多个处理器可协同并行求解给定问题的计算机叫做并行计算机；在并行机上求解问题的过程和方法称为并行计算或并行算法；使用多项式数目的处理器可于多对数时间内求解的一类问题称为 NC 类问题；能对 NC 类问题并行求解的算法叫做有效并行算法；研究有效并行算法最常使用的计算机是并行随机存取计算机，即 PRAM。

2. 并行计算模型

① 专用计算模型　包括比较器网络模型和 VLSI 阵列模型。

② 通用计算模型　不同互连结构（一维阵列、网孔、树、超立方、立方环、洗牌交换、蝶形网等）的 SIMD-IN 模型；共享存储的 SIMD-SM 模型，包括不允许同时读写的 SIMD-EREW 模型、允许同时读但不允许同时写的 SIMD-CREW 模型（即通常意义下的 PRAM 模型）和允许同时读写的 SIMD-CRCW 模型，而 CRCW 模型又可细分为优先写的 P-CRCW、任意写的 A-CRCW 和写相同数的 C-CRCW；共享存储的（紧耦合）MIMD-SM 模型和异步通信链接的（松散耦合）MIMD-AC 模型。

③ 其他计算模型　包括判定树模型、有向无环图 dag 模型和随机的 PRAM（即 RPRAM）模型等。

二、并行算法的研究层次

1. 并行算法理论

①不同计算模型的能力、限制和等价关系；②计算问题的可并行性；③NC 理论；④下界技术。

2. 并行算法的设计和分析

重点研究有效并行算法的设计和分析，其中包括①并行计算模型；②并行算法的设计；③并行算法的正确性证明；④并行算法的分析等。

3. 并行算法的实现

①算法到体系结构的映射；②并行程序设计语言和编译；③算法的并行运行环境和性能

评估等。

三、并行算法理论

1. 不同 SIMD-SM 模型的能力和相互关系

令 T_M 和 S_M 表示计算模型 M 上算法的运行时间和所需的空间，p 为处理器数目，则

① $$T_{EREW} \geqslant T_{CREW} \geqslant T_{CRCW}$$

② $$\begin{cases} T_{EREW} = O(T_{CREW}\log_2 p) = O(T_{CRCW}\log_2 p) \\ S_{EREW} = O(S_{CREW}\, p) = O(S_{CRCW}\, p) \end{cases}$$

或者，

$$\begin{cases} T_{EREW} = O(T_{CREW}\log_2^2 p) = O(T_{CRCW}\log_2^2 p) \\ S_{EREW} = O(S_{CREW} + p) = O(S_{CRCW} + p) \end{cases}$$

③ P-CRCW＞A-CRCW＞C-CRCW（“＞”表示“功能强于”）

④ 假定处理器数目均为 p，则 P-CRCW 上的一个写指令步，在 C-CRCW 上须用 $O(\log_2 p/\log_2\log_2 p)$ 步，而在 A-CRCW 上须用 $O(\log_2\log_2 p)$ 步。

2. 并行算法理论

① 可并行化问题　令 P 表示在顺序的多项式时间内可求解的问题类；NC 表示用多项式数目的处理器可于多对数时间内求解的问题类。NC⊆P，但 P⊆NC 可能是不成立的。P 与 NC 的边界就是 P-完全问题，即 P 中最难的问题，它是不能用多项式数目的处理器很快并行求解的（如最大流问题、线性规划问题等）。

② 并行复杂度理论　使用并行时间、顺序空间和电路深度三个测度可以形成并行复杂度完美的理论基础。空间和并行时间复杂度之间的对应关系是并行计算理论的推广。定义 NC^K 是由一簇尺寸为 n 的多项式、深度为 $O(\log_2^K n)$ 的均衡电路 $\{C_n\}$ 可计算的一组函数；定义 $EREW^K$、$CREW^K$ 和 $CRCW^K$ 是使用多项式数目的处理器、运行时间为 $O(\log_2^K n)$ 的相应共享存储模型上可计算的一类函数，则

$$NC^K \subseteq EREW^K \subseteq CREW^K \subseteq CRCW^K \subseteq NC^{K+1}$$

$$NC = \bigcup_{K \geqslant 1} NC^K$$

可见，NC 类在不同的并行计算模型上保持不变（正像 P 类问题不依赖于顺序计算模型一样）。因此 NC 类问题的复杂度是至关重要的。

3. 下界技术

求问题的下界常使用“对手论”和“归约技术”。前者是将算法的设计过程视为相互“对抗”的过程；后者的含意是：令 A 和 B 是任意两个计算问题，如果已知 A 的下界为 $L(A)$，而 B 在某种特殊情况下可以变为 A，则 B 的下界亦为 $L(A)$。

四、并行算法的分类和研究内容

1. 并行算法的分类

并行算法可分为同步算法（通常运行在 SIMD 模型上），异步算法（通常运行在 MIMD-

SM 模型上)和分布式算法(通常运行在 MIMD-AC 模型上);也可分为数值计算(主要基于代数关系运算)并行算法和非数值计算(主要基于比较关系运算)并行算法。

2. 并行算法的研究内容

根据计算机学科中有关算法研究的经典内容,并行算法的研究内容大致包括:

① 数值计算问题　系根据数值分析中的数学原理,密切结合体系结构和计算模型,重点讨论诸如矩阵运算、线性方程组求解、非线性方程求根、偏微分方程数值解、FFT、卷积、滤波等计算问题的并行算法的设计和分析,而不像数值分析那样着重讨论计算稳定性、精度和收敛速度等。

② 非数值计算问题　包括选择、归并、排序、搜索、组合运算、图论问题、图像分析、计算几何、组合遍历、串匹配、优化判定、表达式求值、语言识别、语法分析、数据选路、VLSI 计算理论与并行计算等问题的并行算法的设计和分析。

五、并行算法的基本设计技术

1. 平衡树法　沿树自叶向根往返遍历,往往可导出很多问题的有效算法。如求最大(小)者、计算前缀和等。

2. 倍增技术(又叫路径折叠技术、倒塌技术、指针跳越等)　特别适合于处理链表数据结构,每当递归调用时所要处理的数据间的距离逐次加倍,最终可达所要求的数据计算。如计算表序问题、求森林的根、找图的连通分量等。

3. 分治策略　将问题分解成一些特性相同的子问题,然后施行递归调用,最终将各子问题的解归并成原问题的解。分治法的难点在于归并。如双调归并网络、FFT 计算、求凸壳等。

4. 划分原理　将原问题分解成一些尺寸近于相等彼此独立的子问题,然后施行并行求解。划分的难点在于如何划分问题使得子问题的解很容易被组合成原问题的解。如归并排序、(m,n)-选择问题等。

5. 流水线技术　在 VLSI 算法中此技术作用最突出。它将一个计算任务 t 分成一系列子任务 $t_1,t_2,\cdots,t_m$,使得一旦 t_i 完成,后继的子任务就立即开始且以同样的速率进行计算。

6. 加速级联法　一个最优但相对不快的算法与一个甚快但可能非最优的算法相继连接起来,从而产生一个快速而最优的加速级联算法。如欲求 n 个数的最大者,可先在对数深度的树上筛选出一些最大数的候选者,然后再在双对数深度的树上继续进行选择,这样就可导致出在 SIMD-CRCW 模型上求 n 个数最大者的最优算法。

7. 破对称技术　打破某些问题的对称性可导出其有效算法,它常使用在图论问题和随机算法的设计中。如可使用打破有向环图的对称性对其施行快速顶点着色。

六、并行算法的度量、表达、证明和分析

1. 度量标准

① 算法所需的处理器数目 $p(n)$,并行运行时间 $t(n)$,成本 $c(n)=p(n)t(n)$。如果 $c(n)$ 等于最坏情况下的串行算法的运算步数,则称 $c(n)$ 为最优的。

② 度量算法并行运行时间 $t_p(n)$ 对串行运行时间 $t_s(n)$ 的改进程度可使用加速比 $S_p(n)=t_s(n)/t_p(n)$，度量处理器的利用率可使用效率 $E_p(n)=S_p(n)/p(n)$。

③ 算法也可使用总运算量 $W(n)$ 和时间 $T(n)$ 表示之。当分配给 p 台处理器执行时，其运行时间 $t(n)$ 满足 Brent 定理，即 $t(n)=O(W(n)/p+T(n))$。

④ 对于 VLSI 计算模型而言，度量算法使用 AT^2，其中 A 为芯片面积，T 为算法时间。

2. 算法表达

① 形式描述　可使用类-Algol 等语言，引入并行语句 parallel do，for all 等；

② 非形式描述　叙述要规范、精炼。

3. 算法证明

① 解析证明法；② 归纳证明法；③ 构造证明法。

4. 算法分析

① 最坏输入组合情况下的算法性态；② 输入加权平均情况下的算法期望性态。

七、并行算法待进一步研究的课题

1. MIMD-SM 模型上的异步并行算法

①进程的划分与生成；②任务分配与负载平衡；③进程调度（确定性与非确定性调度）；④通信与同步；局存和共享存储器间要交换数据就引起通信，通信可通过共享变量使用通信原语表达，通信常导致同步，同步可使用临界压与信号灯操作来实现；⑤数据的竞争与死锁；⑥Amdahl 效应（加速是问题规模的函数）；⑦Amdahl 定律（加速受算法中串行成分的限制）；⑧算法分析较复杂。

2. MIMD-AC 模型上的分布式算法

①MIMD-AC 模型可抽象为无向图 G(V,E)来研究；②每个处理器只具有局部知识，不知全局情况；③由于无共享存储器，所以要用传递消息进行通信；④假定消息传递可在有限但非确定时间内完成，且同一链路上到达的消息要按一定规则进行服务；⑤由于消息传递时间的非确性致使算法时间分析十分困难，简化的情况是假定通信链路是完美的且相邻处理器之间的通信可在 $O(1)$ 时间内完成；⑥分布式算法的度量标准是时间（通常计算时间远小于通信时间）、空间和通信量（消息交换数目）。

八、并行算法研究的新领域

1. 巨量并行算法(massively parallel algorithm)

① 定义　巨量并行计算机 MPC 是一种可扩展的（即系统的性能近似与处理器数目成正比）并行/分布式处理系统，它拥有先进的 RISC 芯片、成千上万的处理单元、峰值速度可达 100GFLOPS 以上、性能/价格比高和具有多种用途(multi-purpose)。巨量并行算法 MPA 是指运行在 MPC 上的一种可扩展的（即算法的加速近似与处理器数目成正比）主要是开发数据级并行度的并行算法。

② 巨量并行计算模型　常规的并行计算模型必须加以推广，以适应开拓高度并行性的需要：

· ASIMD(自治的单指令流多数据流)模型　是 SIMD 的改进，仍保持 SIMD 的可编程性，但加进了一些灵活性，包括执行自治、连接自治、寻址自治和 I/O 自治等；

· SPMD(单程序流多数据流)模型　在不同的处理机节点上运行相同的程序但访问不同的数据；

· MPMD(多程序流多数据流)模型　系统由若干个 cluster 组成，每个 cluster 内运行 SPMD 模式；不同的 cluster 运行不同的程序。

③ MPA 的设计　设计的目标是可扩展(scalable)。目前的状况是设计 MPA 还是一门艺术，尚谈不上设计技术。不要总试图并行化现有串行算法，要从问题本身描述出发，从头开始设计新算法。避免并行算法随问题规模层层移植。试图先设计小规模并行算法，然后移植到中规模的并行算法，再移植到大规模并行算法的效果是不好的。设计算法时受 Amdahl 定律的约束，即认为程序中的串行成分所占的比例是固定的、与问题的规模无关，这对巨量并行计算未必正确。最近的实践表明，当问题的规模增大时程序中的串行成分的比例大大下降，从而可获得极大的加速。

④ 并行语言、编译和编程环境　并行语言(如并行 Fortran，并行 C，OCCAM 等)达到标准化还为时过早。通过并行编译技术离实用尚有一段距离。目前并行编程环境是采用并行库程序的方法(如 MIMDizer，Express，Linda 等)，基本不改变原串行程序的风格，用户只要在程序中调用并行库函数。

2. 基于神经网模型的并行算法

① 神经计算原理　整体-判定；合作和竞争；学习和自组织；动力学演变；巨量并行、分布存储、模拟处理。

② 神经计算复杂度理论　评价神经计算的能力，即研究神经网络能否解决常规计算机无法解决的问题？神经网络在什么意义下比常规的计算机更有效？目前全面回答这些问题是困难的，已有的成果是：不能用神经网络有效的解决 NP-难度问题，除非 $NP=CO-NP$。即使用神经网络来求 NP-难度问题的近似解也是不可能的，除非 $NP=CO-NP$。神经网络在平均意义上讲很可能比常规计算机更有效，至少大量的实验证明了这一点。对于某些特定的神经网络和某些问题，用神经网络有可能找到有效解。但学习是个难问题。

③ 神经网络的学习和训练算法　学习理论；学习算法的设计；学习算法的并行实现；神经元的划分与并行度的开拓；学习算法的度量——学习速度 Epochs 和训练期间连接权的更新速度 CUPS。

④ 组合优化问题的神经网络算法

· Hopfield 方法　模型与 Ising 玻璃相似；能量函数描述系统状态；高能到低能的动力学演变类似于约束满足问题搜索最佳解。

· 弹性网法　基于几何观点；问题求解过程就是不断修改映射过程；算法就是同时求解一组差分方程的过程。

· 自组织映射法　拓扑保序映射；通过自学习，修改映射直至满足问题的解要求。

· 模拟退火法　将组合优化问题与统计力学中的热平衡问题相类比；通过模拟退火过程，可找到全局或近似最优解。

· 均场退火法　对模拟退火的重大改进；只需在某个关键温度附近施行退火就可代替整个缓慢的退火过程，从而节省了时间。

Schwarz 算法的收敛估计及解有限元方程的单元分裂法

吕 涛　　　　石济民　林振宝　彭志健

(中国科学院成都计算所)　　　　(香港理工学院数学系)

本文分两部分。其一，叙述关于 Schwarz 算法收敛速度的新结果；其二，提出解有限元方程的单元分裂法。此方法并行度特别高，程序简单，能大量节约计算与存储。特别适合解大型问题。

一、Schwarz 算法收敛速度估计新结果

Schwarz 算法收敛速度一直是人们关注的热点，对于规则区域 Laplace 方程已在[7]中给出，本文讨论一般椭圆型方程的 Dirichlet 问题

$$\begin{cases} \mathrm{L}u = a_{ij}D_{ij}u + b_jD_ju + du = f & (\Omega) \\ u = g & (\partial\Omega) \end{cases} \tag{1}$$

其中 Ω 是 R^n 的有界开域，a_{ij}、b_j、d 是 Ω 上连续函数，$D_j=\partial/\partial x_j$，$D_{ij}=D_iD_j$，并约定重复指标从 1 到 n 求和。为了构造 Schwarz 交替法，首先分解：$\Omega=\Omega_1\cup\Omega_2$，$\Omega_1$ 和 Ω_2 相互重叠，令 $\Gamma_1=\partial\Omega_1\cap\Omega_2$，$\Gamma_2=\partial\Omega_2\cap\Omega_1$，$I_1$ 及 I_2 分别表示奇数与偶数集合。Schwarz 交替法描述为：初始 $u^0\in C(\Omega)$ 选好后，逐次在子域上解问题

$$\begin{cases} \mathrm{L}u^k = f \quad (\Omega_j),\text{若 } k\in I_j, \quad k\geq 1, \quad j=1 \text{ 或 } 2 \\ u = g \quad (\partial\Omega\cap\partial\Omega_j), \quad u^k = u^{k-1}, \quad (\Gamma_j) \end{cases} \tag{2}$$

有关(2)的收敛速度有

引理 1(P. L. Lions 见[4]) 设$[a_{ij}(x)]$，$\forall x\in\Omega$ 是非负矩阵且存在常数 d_0. 使 $d(x)\leq d_0<0$，(Ω)及 $\delta=\mathrm{dist}(\Gamma_1,\Gamma_2)>0$，则若(1)有唯一古典解存在，必有与 d_0 及算子系数相关的正数 $\mu>0$，使误差估计有

$$\max_{x\in\Omega}|u(x)-u^k(x)|\leq \exp(-\mu\delta^2)\max_{x\in\Omega}|u(x)-u^{k-1}(x)| \tag{3}$$

引理 1 的缺陷是条件 $d(x)\leq d_0<0$ 太严格，乃至 Poisson 方程也不能纳入此框架内，另方面对矩形域上的 Poisson 方程，康立山和 Evans 曾借助于 Fourier 分析得到类似(3)结论。资料[6]中我们在统一框架：$d(x)\leq 0$ 下得到了比(3)更好的结果

定理 1 设方程(1)的系数有性质：

$$d(x)\leq 0, \quad \forall x\in\Omega \tag{4}$$

矩阵$[a_{ij}(x)]$一致非负，并且至少有一个对角元，不妨设为$a_{11}(x)$，和相关的正常数a_1使

$$\min_{x\in\Omega} a_{11}(x) \geq a_1 > 0 \tag{5}$$

及拟边界点的x_1坐标，有常数$\delta>0$存在使满足

$$x_1 - y_1 \geq \delta > 0, \quad \forall\, x \in \Gamma_1, \quad y \in \Gamma_2 \tag{6}$$

则必定存在与a_{11}及系数相关的常数$\mu>0$，使误差收缩因子为$e^{-\mu\delta}$。

显然，若$\mathrm{dist}(\Gamma_1,\Gamma_2)=\delta$，则(6)一定被满足。定理1由于仅要求较弱的条件(4)所以具有普遍性，并且得到收缩因子为$\exp(-\mu\delta)$，大为改善了引理1中收缩因子为$\exp(-\mu\delta^2)$的结果。

无论引理1或定理1皆表明敛速取决于拟边界矩离$\mathrm{dist}(\Gamma_1,\Gamma_2)$而不是重叠区域的测度。若$\mathrm{dist}(\Gamma_1,\Gamma_2)=0$，则有可能得不到算法收敛性。更详细讨论可见[6]。

对于散度型椭圆型方程

$$\begin{cases} \mathrm{L}u = -D_i(a_{ij}D_ju) + b_iD_iu + du = f & (\Omega) \\ u = g & (\partial\Omega) \end{cases} \tag{7}$$

假定$a_{ij}\xi_i\xi_j\geq 0, \forall\, x\in\Omega, \xi\in R^n$，有关Schwarz敛速讨论很少见于报导，我们证明了以下结果

定理2 若方程(7)有弱解$u\in H^1(\Omega)\cap L^\infty(\Omega)$存在，并且$d(x)\geq d_0>0, \forall\, x\in\Omega$，而且$\mathrm{dist}(\Gamma_1,\Gamma_2)=\delta>0$，又设$a_{ij}\in C^1(\Omega)$，则必定存在仅与$d_0$及系数相关正常数$\mu>0$，使Schwarz交替法的误差收缩因子为$\exp(-\mu\delta^2)$。

还可以证明满足极大值原理的椭圆型偏微分方程组有类似的结论，这里不再叙述。

二、Schwarz算法的加速收敛

由定理1及定理2看出Schwarz算法误差收缩因子为$e^{-\mu\delta}$，其中$\delta=\mathrm{dist}(\Gamma_1,\Gamma_2)$是固定的，加速的技巧是增大$\mu$，既然$\mu$随$d_0$的减小而增大，利用椭圆型算子保序性质我们可以构造加速收敛方法如下

算法1 （加速Schwarz算法）

步1. 选择初始下解$u^0(x)\leq u(x)$（或上解$u^0(x)\geq u(x)$）及负常数$d_1<0$，置$k:=1$。

步2. 若$k\in I_j, j=1,2$，求u^k适合

$$\begin{cases} \mathrm{L}u^k + d_1u^k = d_1u^{k-1} + f & (\Omega_j) \\ u^k = g, (\partial\Omega\cap\Omega_j), u^k = u^{k-1}, (\Gamma_j) \end{cases} \tag{8}$$

步3. 置$k:=k+1$转步2。

利用椭圆型方程极值原理可以证明误差$E^k=u-u^k$是单调下降的，并满足

$$0 \leq E^k \leq e^{-\mu\delta}E^{k-1} \leq \cdots \leqslant e^{-k\mu\delta}E^0 \tag{9}$$

这里μ随d_1减小而增大。在没有合适的上(下)解供选择时，取$u^0\equiv 0$是合理的选择。上面算法1也能用于解非线性椭圆型方程，举简单的半线性方程为例

$$\begin{cases} -\Delta u + f(x,u) = 0 & (\Omega) \\ u = g & (\partial\Omega) \end{cases} \tag{10}$$