

Helmholtz 方程差分解的迭代方法

刘 金 达

(国家气象局国家气象中心)

提 要

本文提出几种 Helmholtz 方程差分解的迭代方法, 用 Banach 空间线性算子的观点, 提出了算子迭代方法. 最常用的逐步超松弛迭代法是我们提出的算子迭代法的一种. 本文并提出了可变超松弛系数的思想, 解决了因可变 Helmholtz 系数难以确定最佳超松弛系数的问题. 本文还对目前常用的几种迭代方法进行了比较.

关键词: Banach 空间; 线性算子; 算子迭代法; 可变超松弛系数.

一、引 言

Helmholtz 方程是大气科学中常遇到的方程. 如著名的 ω 方程就是 Helmholtz 型方程. 在数值天气预报早期的地转模式中, 大量的计算就是求解 Helmholtz 型方程. 当数值天气预报发展到采用原始方程模式时, 为了加长时间步长, 气象学家们又提出了半隐式格式, 而半隐式格式中又要求解 Helmholtz 型方程. 近年来, 数值天气预报中又出现了很有希望的半隐式半拉格朗日方法, 这方法中又要求解 Helmholtz 型方程. 另外, 在云雾物理学中的积云数值模拟中也需要求解 Helmholtz 型方程. 在其他很多领域中也要求解 Helmholtz 型方程.

60 年代后期和 70 年代国际上陆续提出了各种直接法, 但直接法有一个最大缺陷就是不能求解可变 Helmholtz 系数的方程. 而且对非矩形求解区域, 直接法就非常复杂, 从而也影响求解速度. 所以研究迭代解法还是有重大意义的. 另外, 我们仅讨论二维问题, 三维问题可以类推. 并且只讨论第一边值问题.

二、算子迭代法

我们要求解的问题是

$$\begin{cases} \nabla^2 u - hu = f, & x, y \in \Omega \\ u|_{\Omega_B} = g, & x, y \in \Omega_B \end{cases} \quad (1)$$

$$\overline{\Omega} = \Omega + \Omega_B,$$

Ω 是求解问题的空间区域, Ω_B 为区域的边界. f 是定义在 Ω 上的已知函数, g 是定义在 Ω_B 上的已知函数, Helmholtz 系数 $h \geq 0$.

1988 年 10 月 31 日收到, 1989 年 2 月 19 日收到修改稿.

令 $u = u^{(1)} + u^{(2)}$, $u^{(1)}$ 和 $u^{(2)}$ 分别为下列两个问题的解:

$$(I) \quad \begin{cases} \nabla^2 u^{(1)} - hu^{(1)} = f, & x, y \in \Omega \\ u^{(1)}|_{\Omega_B} = 0, & x, y \in \Omega_B \end{cases}$$

$$(II) \quad \begin{cases} \nabla^2 u^{(2)} - hu^{(2)} = 0, & x, y \in \Omega \\ u^{(2)}|_{\Omega_B} = g, & x, y \in \Omega_B \end{cases}$$

用中央差分法对问题 (I) 和 (II) 进行离散, 可得

$$(I') \quad \begin{cases} Hu^{(1)} = f, & x', y' \in \Omega' \\ u^{(1)}|_{\Omega'_B} = 0, & x', y' \in \Omega'_B \end{cases} \quad (3)$$

$$(4)$$

$$(II') \quad \begin{cases} Hu^{(2)} = 0, & x', y' \in \Omega' \\ u^{(2)}|_{\Omega'_B} = g, & x', y' \in \Omega'_B \end{cases} \quad (5)$$

$$(6)$$

$$\overline{\Omega'} = \Omega' + \Omega'_B,$$

其中“'”表示离散化以后在网格点上的值, Ω' 和 Ω'_B 表示网格点区域和边界. 问题 (I') 和 (II') 中 u, f, g 虽未加“'”, 但也应该理解

为网格点上的函数. 算子 $H \equiv \sum_1^4 -\alpha$, 求和号 $\sum_1^4$ 表示对图 1 中 1, 2, 3, 4 点求和, 以下类推, 其中 $\alpha = 4 + h$.

定义在网格点区域 $\overline{\Omega'}$ 上并且满足 $u|_{\Omega'_B} = 0$ 的所有函数集合构成一个 Banach 空间 L (即 B 型空间 L), 此 B 型空间 L 取范数为

$$\|u\| = \max_{x', y' \in \overline{\Omega'}} |u|, \quad (7)$$

		15	10	14	
	16	6	2	5	13
	11	3	0	1	9
	17	7	4	8	20
		18	12	19	

图 1 网格点示意

显然 H 是由 B 型空间 L 映射到自身的有界线性算子. 我们在 B 型空间 L 上定义线性算子 G , G 满足

$$\|E - GH\| = q < 1, \quad (8)$$

其中 E 为空间 L 中的恒等算子. 令算子 $Q = E - GH$, 则 (3) 式变为

$$u = Qu + Gf. \quad (9)$$

对于 B 型空间 L 中任意两元素 $u^{(1)}$ 和 $u^{(0)}$, 由 (8) 式可得

$$\|Qu^{(1)} - Qu^{(0)}\| \leq \|Q\| \cdot \|u^{(1)} - u^{(0)}\| = q \|u^{(1)} - u^{(0)}\|. \quad (10)$$

由(9)和(10)式,从空间 L 中任意一元素 $u^{(0)}$ 出发,根据 Banach 定理,由迭代公式

$$u^{(v+1)} = Qu^{(v)} + Gf, \quad (11)$$

可求得

$$u^* = \lim_{v \rightarrow \infty} u^{(v)}, \quad (12)$$

u^* 为定义在网格点区域 $\bar{\Omega}'$ 上问题 (I') 的准确解.

在 B 型空间 L 上定义几个线性算子 G :

$$(i) \quad G_0 = -\frac{1}{\alpha_0}, \quad (13)$$

相应于 G_0 的 Q 算子是

$$Q_0 = \frac{1}{\alpha_0} \sum_1^4, \quad (14)$$

这就是最常用的逐步松弛迭代法.

$$(ii) \quad G_1 = \frac{1}{4 - \alpha_0^2} \left(\sum_1^4 + \alpha_0 \right), \quad (15)$$

相应于 G_1 的 Q 算子是

$$Q_1 = \frac{1}{\alpha_0^2 - 4} \left(2 \sum_5^8 + \sum_9^{12} - \sum_1^4 + \alpha_0 \sum_1^4 \right). \quad (16)$$

当 Helmholtz 系数 h 为常数时,可简化为

$$G_1' = \frac{1}{4 - \alpha^2} \left(\sum_1^4 + \alpha \right), \quad (17)$$

和

$$Q_1' = \frac{1}{\alpha^2 - 4} \left(2 \sum_5^8 + \sum_9^{12} \right). \quad (18)$$

(iii) 当 Helmholtz 系数 h 为常数时,定义

$$G_2 = \frac{1}{8 - \alpha^2} \left[\left(\alpha - \frac{4}{\alpha} \right) + \sum_1^4 + \frac{2}{\alpha} \sum_5^8 \right], \quad (19)$$

相应于 G_2 的 Q 算子是

$$Q_2 = \frac{1}{\alpha^2 - 8} \left(\sum_9^{12} + \frac{2}{\alpha} \sum_{13}^{20} \right). \quad (20)$$

在迭代公式(11)中采用算子 G_1 和 Q_1 , G_2 和 Q_2 时,我们只能求解问题 (I'), 因为这两对算子是在 $u|_{\Omega_B} = 0$ 的条件下求得的. 又由于右端函数 f 在边界上无定义,从(15)——(20)式可以看出,迭代公式(11)只能用于边界内第二圈的里面. 紧靠边界内的第一圈, (11)式不能用. 因此我们用迭代公式(11)求解问题 (I') 时, 边界内第一圈上的

u 值还得用其他办法求. 问题 (II') 和 (I') 的方程有相同的算子 H , 而且问题 (II') 是齐次方程. 所以可以将边界内第二圈里面的 $u^{(1)}$ 值看作是要求解的 u 的某次迭代值, 也就是将边界内第二圈里面的 $u^{(1)}$ 看作为 0. 它一定能满足齐次方程. 在边界值为 g 的条件下, 将非第一圈上的 u 值都移到方程右端, 在边界内第一圈上求解方程 (1), 这是一个二阶线性常微分方程的周期解的问题.

为了保证迭代公式 (11) 收敛, 必须且只须 $\|Q\| < 1$. 因此, 必须且只须 $\alpha > 4$ (即 $h > 0$). 上面三个算子 Q_0, Q_1, Q_2 对于 $h > 0$ 是能保证迭代公式 (11) 收敛, 而且当 $h \gg 0$, 则收敛越快. 当 $\|Q\| \leq 1$ 时, 即 $\alpha \geq 4$ 时 ($h \geq 0$), 迭代公式 (11) 是弱收敛, 特别是当 $h = 0$ 时 (即求解 Poisson 方程时), 收敛速度较慢, 因为这时 $\|Q\| = 1$. 对于算子 G_0 和 Q_0 , 为了加速收敛速度, Frankel (1950), Young (1954) 对于 $h = 0$ (即 Poisson 方程) 的情况, 求得了超松弛系数的计算公式

$$\omega = 2 \left\{ 1 + \sin \left[\arccos \frac{1}{2} \left(\cos \frac{\pi}{M} + \cos \frac{\pi}{N} \right) \right] \right\}, \quad (21)$$

其中 M, N 分别是 x 和 y 方向的网格点数. Arnason (1956) 和 Miyakoda (1960) 等对于 $h \neq 0$ 的情况, 求得超松弛系数的计算公式为

$$\omega = \left(\sqrt{2} + \frac{h}{2\sqrt{2}} \right)^2 - \left(\sqrt{2} + \frac{h}{2\sqrt{2}} \right) \left[h \left(1 + \frac{h}{8} \right) + \left(\frac{\pi}{M} \right)^2 + \left(\frac{\pi}{N} \right)^2 \right]^{\frac{1}{2}}. \quad (22)$$

引入超松弛系数对某些定解问题确实加快了收敛速度, 但对某些问题适得其反. 因为这时的 (13) 和 (14) 式变为

$$G_{0\omega} = -\frac{\omega}{\alpha}, \quad (23)$$

和

$$Q_{0\omega} = (1 - \omega) + \frac{\omega}{\alpha} \sum_1^4, \quad (24)$$

并不能证明对所有的问题有

$$\|Q_{0\omega}\| < \|Q_0\|. \quad (25)$$

另外, Arnason 和 Miyakoda 推导出的公式 (22) 是对 h 为常数的情况得到的超松弛系数计算公式. 但在实际的应用中, 经常遇到 Helmholtz 系数是 x, y 的函数的情形. 对于这种情形, 我们提出用相应的变超松弛系数的迭代法求解, 得到加速收敛的效果. 我们用两种变超松弛系数做了试验. 第一种变超松弛系数是

$$\omega(x, y) = 1 + \frac{1}{P * h(x, y) + q}. \quad (26)$$

第二种变超松弛系数就是将公式 (22) 中的常数 h , 改为 $h(x, y)$.

三、环隐式迭代法

隐式迭代法常用的是交替方向隐式迭代法. 但笔者认为在多数情况下行隐式迭代法

比交替方向隐式迭代法好. 迭代收敛的次数相同, 但行隐式迭代法省时省内存. 这两种隐式迭代法都是一个方向上的边界条件在起直接作用. 另一个方向的边界条件不能起直接作用. 为了改进这一不足, 我们提出下面的解法. 沿着边界将边界内网格点连接成环形, 如图 2 所示.

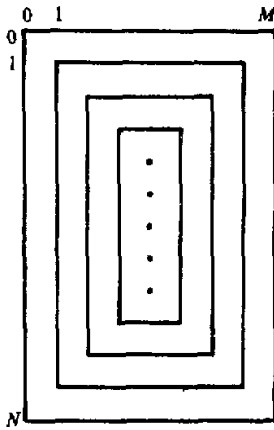


图 2 边界内环形网格点

除了拐弯点之外, 迭代方程为

$$\begin{aligned} u_{i-1,j}^{(v+1)} - (4 + h_{i,j}) u_{i,j}^{(v+1)} + u_{i+1,j}^{(v+1)} \\ = f_{i,j} - u_{i,j-1}^{(*)} - u_{i,j+1}^{(*)}, \end{aligned} \quad (27)$$

和

$$\begin{aligned} u_{i,j-1}^{(v+1)} - (4 + h_{i,j}) u_{i,j}^{(v+1)} + u_{i,j+1}^{(v+1)} \\ = f_{i,j} - u_{i-1,j}^{(*)} - u_{i+1,j}^{(*)}, \end{aligned} \quad (28)$$

上两式中(*)号可能是 $(v+1)$ 或 (v) . 我们将环上的格点值作未知量, 而其余的项都归并到右端项, 它们所取的迭代次数随不同的位置而不同, 但总保持取最新的值.

已知右端项, 在环上求解. 这时相当于求解二阶线性常微分方程周期条件的数值解. 图 2 中 x 方向的网格点数为奇数, 中间的网格点不能连成一个环形, 但可以追赶法求解. 如果网格点为偶数, 就不存在连不成环形的问题.

四、各种解法的比较

评价一种求解方法的优劣, 主要决定于两个因素: (1) 运算量多少, (2) 解的精度高低. 运算量少精度高的方法是比较好的方法. 但是从我们研究试验的结果来看, 对于解 Helmholtz 方程第一边值问题, 很难说某一种方法绝对好, 而是对不同问题, 有不同的方法为最佳.

为了对各种方法进行比较, 我们求解下列六个定解问题, 求解区域是正方形(如图 3), 并将边长分为 32 等分, 也就是有 33×33 个网格点. 网格距为 d :

$$(a) \quad \begin{cases} \nabla^2 u_{i,j} - h_{i,j} * u_{i,j} = A \left[1 - \frac{1}{4} h_{i,j} (x_i^2 + y_j^2) \right], \\ u_{x=0} = \frac{A}{4} y_j^2, \quad u_{x=a} = \frac{A}{4} (a^2 + y_j^2), \\ u_{y=0} = \frac{A}{4} x_i^2, \quad u_{y=a} = \frac{A}{4} (x_i^2 + a^2), \end{cases}$$

相应的解析解为

$$u = A(x^2 + y^2) / 4.$$

$$(b) \quad \begin{cases} \nabla^2 u_{i,j} - h_{i,j} * u_{i,j} = [2(\cos Bd + \cos cd - 2) - h_{i,j}] \\ \quad * A \sin(Bx_i + \beta) \sin(cy_j + \beta), \end{cases}$$

$$\begin{cases} u_{x=0} = A \sin \beta \sin (cy_j + \beta), & u_{x=a} = A \sin (Ba + \beta) \sin (cy_j + \beta), \\ u_{y=0} = A \sin (Bx_i + \beta) \sin \beta, & u_{y=a} = A \sin (Bx_i + \beta) \sin (ca + \beta), \end{cases}$$

相应的解析解为 $u = A \sin (Bx + \beta) \sin (cy + \beta)$.

$$(c) \quad \begin{cases} \nabla^2 u_{i,j} - h_{i,j} * u_{i,j} = -h_{i,j} * A(x_i + y_j), \\ u_{x=0} = A y_j, & u_{x=a} = A(a + y_j), \\ u_{y=0} = A x_i, & u_{y=a} = A(x_i + a), \end{cases}$$

相应的解析解为 $u = A(x + y)$.

$$(d) \quad \begin{cases} \nabla^2 u_{i,j} - h_{i,j} * u_{i,j} = -A h_{i,j}, \\ u_{x=0} = A, & u_{x=a} = A, \\ u_{y=0} = A, & u_{y=a} = A, \end{cases}$$

相应的解析解为 $u = A$ (常数).

$$(e) \quad \begin{cases} \nabla^2 u_{i,j} - h_{i,j} * u_{i,j} = -h_{i,j} * A * x_i * y_j, \\ u_{x=0} = 0, & u_{x=a} = A * a * y_j, \\ u_{y=0} = 0, & u_{y=a} = A * x_i * a, \end{cases}$$

相应的解析解为 $u = A x y$.

$$(f) \quad \begin{cases} \nabla^2 u_{i,j} - h_{i,j} * u_{i,j} = A \left[(x_i + y_j) - h_{i,j} * \frac{1}{6} * (x_i^3 + y_j^3) \right], \\ u_{x=0} = \frac{A}{6} y_j^3, & u_{x=a} = \frac{A}{6} (a^3 + y_j^3), \\ u_{y=0} = \frac{A}{6} x_i^3, & u_{y=a} = \frac{A}{6} (x_i^3 + a^3), \end{cases}$$

相应的解析解为 $u = A(x^3 + y^3)/6$.

因为以上六个定解问题有相应的解析解, 所以便于比较数值解的精度.

将我们提出的算子迭代法、变超松弛系数方法、环隐式迭代法与直接法(正弦展开法(SST)、双向递归减元法(DCR)、递归减元和正弦展开相结合法(FACR))、行隐式迭代法和交替方向隐式迭代法进行了比较, 由于篇幅所限, 不能一一列出, 下面仅叙述比较试验的结果.

当 Helmholtz 系数 $h=0$ 时, 直接法求解速度快, 解的精度高. 但是当 h 较大时, 迭代法的求解速度就可能

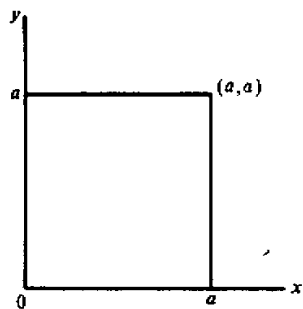


图3 求解区域

超过直接法. 例如当 $h=50$ 时, 求解速度最快的方法是算子迭代法中采用 Q_1' 算子的方法. 表1列出了14种方法求解定解问题(a)的结果.

表1 14种方法求解定解问题(a)的计算时间和精度 ($h=50$)

方 法	时间 (s)	与解析解的最大误差	迭代次数	迭代收敛的最大误差
OI ₁	0.037	0.7748604E-06	3	0.3576279E-05
RII	0.040	0.1072884E-05	4	0.6616116E-05
OI ₂	0.043	0.8344650E-06	3	0.4768372E-06
SOR ₁	0.044	0.1549721E-05	4	0.4634151E-04
ADI	0.046	0.1072884E-05	4	0.6377697E-05
SORV	0.054	0.7152557E-06	4	0.2730612E-04
FACR	0.054	0.1072884E-05		
DCR	0.060	0.2441406E-03		
LII	0.072	0.1072884E-05	4	0.5483627E-05
OIV	0.079	0.7748604E-06	3	0.3576279E-05
SORVV	0.099	0.3755093E-05	4	0.8126099E-04
SST	0.102	0.9059906E-05		
CHEBY	0.529	0.3892183E-04	46	0.8698249E-04
SORO	0.584	0.3826618E-04	55	0.8389470E-04

表1中, OI₁ 是算子迭代法中 Q_1' 迭代法, RII 是行隐式迭代法, OI₂ 是算子迭代法中 Q_2' 迭代法, SOR₁ 是超松弛系数为1的迭代法, 也就是 Gauss-Seidel 迭代法, ADI 是交替方向隐式迭代法, SORV 是变超松弛系数(用(2b)式计算)迭代法, FACR 是递归减元和正弦展开相结合的方法, DCR 是双向递归减元法, LII 是环隐式迭代法, OIV 是算子迭代法中 Q_1' 迭代法, SORVV 是采用(22)式计算的变超松弛系数迭代法, SST 是正弦展开法, CHEBY 是 Chebyshev 迭代法, SORO 是采用(21)式计算的超松弛系数(常数)迭代法.

从表1可以看出算子迭代法 Q_1' 和 Q_2' 不仅速度快、精度也高. 直接法中的双向递归减元法精度最差.

在多数的实际应用中, 例如在大气科学中 Helmholtz 系数为常数的情况是不多见的. 大多数的情况 Helmholtz 系数是 x, y 的函数, 也就是可变 Helmholtz 系数的情况. 这时直接法就不能使用, 只能用迭代法. 用迭代法求解定解问题的速度将随着不同的问题而不同, 不像直接法求解任何问题都有固定的运算量, 因而速度也固定. 但是通过我们的试验结果, 在以上六个定解问题变 Helmholtz 系数的情况下, SORV 方法和 SORVV 方法求解速度快, 精度也高一点. 例如求解定解问题(b), 在 Helmholtz 系数 $h(x, y) = A(x^2 - y^2)/4$ 的情况下, 其试验结果列于表2.

表2中的方法名称与表1同. 从表2可以看出求解变 Helmholtz 系数的问题, SORV 和 SORVV 方法, 即用变超松弛系数方法的求解速度是比较快的, 而且精度也较高. 虽然我们只给了一个定解问题求解的结果, 但多数情况都是这两种方法求解速度快, 精

度高。

但是在某些情况下，隐式迭代法的求解速度也相当快。例如求解定解问题 (f) ，在 Helmholtz 系数 $h = A \sin(Bx + \beta) \sin(cy + \beta)$ 的情况下，其求解速度就较快。计算结果列于表 3。

表 2 9 种可求解变 Helmholtz 系数的迭代法求解定解问题
(b) 的计算时间和精度 ($h = A(x^2 - y^2)/4$)

方 法	时间 (s)	与解析解的最大误差	迭代次数	迭代收敛的最大误差
SORV	0.299	0.1352429E-03	24	0.5316442E-04
SORVV	0.376	0.1087189E-03	27	0.7085013E-04
RII	0.486	0.6766319E-03	53	0.9471178E-04
ADI	0.513	0.7148981E-03	54	0.9697676E-04
CHEBY	0.557	0.7200241E-04	48	0.9915559E-04
SORO	0.793	0.5221367E-04	72	0.6610241E-04
LII	0.796	0.6651282E-03	52	0.9322166E-04
SOR _I	0.961	0.1396000E-02	93	0.9627586E-04
OIV	0.961	0.4554391E-03	42	0.8809566E-04

表 3 9 种可求解变 Helmholtz 系数的迭代法求解定解问题 (f)
的计算时间和精度 ($h = A \sin(Bx + \beta) \sin(cy + \beta)$)

方 法	时间 (s)	与解析解的最大误差	迭代次数	迭代收敛的最大误差
RII	0.287	0.3154874E-03	31	0.8499622E-04
ADI	0.296	0.3180504E-03	31	0.8577108E-04
LII	0.433	0.3070235E-03	28	0.8893013E-04
SORVV	0.470	0.3802776E-04	34	0.8470885E-04
SORV	0.488	0.3886223E-04	38	0.8548651E-04
SOR _I	0.535	0.7025599E-03	52	0.9522180E-04
CHEBY	0.543	0.9280443E-04	46	0.7048412E-04
OIV	0.633	0.1978278E-03	27	0.8666515E-04
SORO	0.777	0.4501315E-04	72	0.4378449E-04

表 3 中的方法名称与表 1 同。从表 3 可以看出行隐式迭代法是比较快的，而且多数情形中，行隐式迭代法都比交替方向隐式迭代法快。另外，本文提出环隐式迭代法的迭代收敛次数比行隐式迭代法和交替方向隐式迭代法都要少，只是由于求解二阶常微分方程周期问题的方法运算量大了一点，所以在计算时间上慢于前两者。如果能找到较快的求解二阶常微分方程周期问题的方法，那末可能环隐式迭代法是较快的方法。

五、结 束 语

通过以上的比较试验，可以得出如下的结论：

(1) 当 $h=0$ 时, 也就是解 Poisson 方程时, 直接法的求解速度快, 精度高. 这时迭代法是逊色的. 当 h 虽为常数, 但较大时, 迭代法的求解速度会快于直接法, 这时双向递归减元法的精度较差. 这可能是由于我们在富士通的 M-360R 计算机上做试验的缘故, 该机器字长是 32 位. 直接法以递归减元和正弦展开相结合的方法为最快.

(2) 当 Helmholtz 系数可变时, 直接法就不能使用, 这时只能用迭代法求解. 而且对于多数的定解问题, 本文提出的变超松弛系数方法其求解速度最快. 迭代法另一个优点就是编制程序较直接法简单. 所以研究迭代法还是很有意义的.

参 考 文 献

- [1] 刘金达, 1982, 离散 Helmholtz 型方程数值解法的比较研究, 中期数值天气预报文集, 气象出版社, 86—91.
- [2] 刘金达, 1981, 快速傅里叶变换和快速正弦变换, 气象科学技术集刊 (1), 气象出版社, 102—109.
- [3] Miyakoda, K., 1960, Test of convergence speed of iterative methods for solving 2 and 3 dimensional elliptic type differential equations, *Journal of the Meteorological Society of Japan*, 38, No. 2, 107—124.
- [4] Hockney, R.W., 1965, A fast direct solution of Poisson's equation using Fourier analysis, *Journal of the Association for Computing Machinery*, 12, 95—113.
- [5] Temperton, C., 1977, On the FACR(L) algorithm for the discrete Poisson's equation, ECMWF Research Department Internal Report, (14).
- [6] 林鹏程, 江本钰, 1985, Poisson 方程的快速直接解法, 高等学校计算数学学报, 7, No. 2, 187—192.
- [7] И. Б. 康脱罗维奇, 1955, 泛函分析与应用数学, 数学进展, 第 1 卷, 第 4 期, 638—747.