

文章编号:1000-582X(2002)02-0075-04

抛物型问题的边界元重叠型区域分解法

张太平,祝家麟

(重庆大学 数理学院,重庆 400044)

摘要:边界元法是一种求解偏微分方程数值解的计算方法,用边界元法来求解抛物型方程,如采用与时间有关的基本解,较其它方法可以采用较长的时间步长,从而节省计算时间,且计算结果精度高。区域分解法是把计算区域分解成若干子区域来分别求解,由于它将原问题分解,由大化小,由复杂化简单,并且可以并行计算,优越性是显而易见的。将这两种方法结合起来(边界元重叠型区域分解法)求解抛物型方程,利用区域分解法将求解区域划分为两个小的子区域,然后在子区域上用边界元法并行求解方程。数值算例表明边界元重叠型区域分解法是行之有效的,数值试验显示这种方法的收敛速度依赖于子区域重叠面积。

关键词:抛物型方程;区域分解;边界元法;并行计算

中图分类号:O 241.82

文献标识码:A

由于并行技术的发展和当代并行计算机的出现,并行处理和并行计算成为工程计算中的重要手段,而区域分解法正是并行计算大规模科学问题的最有效方法,因此对区域分解法的研究已成为计算数学最有前途的分支之一,该方法是经典的 Schwarz 交替法为基础。通过分解区域,一方面可以降低所处理问题的规模,另一方面可使人们根据不同子域上问题的特点采用不同的方法,这样实现了工程问题的并行计算并提高了计算速度。区域分解法在椭圆问题中的应用,已有很多人在这方面做了大量工作(见文献[1~3])。而对抛物型问题来说,由于问题的特殊性,对区域分解法在抛物型问题中的应用研究却很少并都是以有限元法为基础,但有限元法必须采用较小的时间步长,而不象边界元法,由于基本解跟时间有关,所以可以采用较长的时间步长。基于此笔者在这方面做了一些工作,以二维热传导问题为例,给出了求解抛物型问题的边界元重叠型区域分解算法为基础的并行计算模式,以前一时间步的近似解作为后一时间步的迭代初值,避免了因初始值选择不当造成的误差。并做了数值试验。

1 问题的描述和算法构造

考虑如下二维典型热传导方程的初边值问题:

$$\begin{cases} u_t - \Delta u = 0 & x \in \Omega, t \in (t_0, T) \\ u(x, t) = \bar{u}(x, t) & x \in \Gamma \\ u(x, t) = u_0(x, t_0) & x \in \Omega, \text{初始条件} \end{cases}$$

其中 Ω 是 R^2 中的单连通域, $\Gamma = \partial\Omega$ 。算法构造如下:

为了简单地说明计算方法本文仅考虑把求解区域分成两个子区域的情形,将 Ω 分解为两个有重叠区域的子域 Ω_1, Ω_2 ,且 $\partial\Omega_1 \cap \Omega = \Gamma_{11}, \partial\Omega_2 \cap \Omega = \Gamma_{22}$,具体情况见图 1。

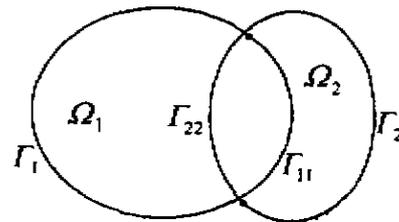


图 1 区域分解情形

计算步骤:

1) 任意给定在人工边界 Γ_{11} 和 Γ_{22} 上的初始值 u^0 和 v^0 。给定循环控制误差 ϵ ,对每个时间步长 $T = n\Delta t$ 循环。

2) 边界元法并行求解方程(1),(2);

$$\begin{cases} u_t^{2n+1} - \Delta u^{2n+1} = 0 & x \in \Omega_1 \\ u^{2n+1} = \bar{u} & x \in \Gamma_1 \\ u^{2n+1} = v^{2n} & x \in \Gamma_{11} \end{cases} \quad (1)$$

• 收稿日期:2001-10-15

作者简介:张太平(1975-),男,四川南充人,重庆大学硕士生。主要从事边界元法及其在工程中的应用研究。

$$\begin{cases} v_i^{2n+1} - \Delta v^{2n+1} = 0 & x \in \Omega_2 \\ v^{2n+1} = \bar{u} & x \in \Gamma_2 \\ v^{2n+1} = u^{2n} & x \in \Gamma_{22} \end{cases} \quad (2)$$

3) 直到 Γ_{11} 上前后两次 v 值的差的绝对值和 Γ_{22} 上前后 u 两次值的差的绝对值都小于某个预先设定的误差或迭代次数超过某指定步数, 则进入下一个时间步循环。

4) 达到所求的时间步, 停止计算。

第2步方程(1), (2) 利用通用的直接边界元方法进行求解。

其边界积分方程为:

$$\begin{aligned} & c(\xi)u(\xi, t_f) + \\ & k \int_0^{t_f} \int_{\Gamma} u(x, t) q^*(\xi, x, t_f, t) d\Gamma(x) dt = \\ & k \int_0^{t_f} \int_{\Gamma} q(x, t) u^*(\xi, x, t_f, t) d\Gamma(x) dt + \\ & \int_{\Omega} u_0(x, t_0) u^*(\xi, x, t_f, t) d\Omega \end{aligned}$$

其中基本解

$$\begin{aligned} u^*(\xi, x, t_f, t) &= \frac{1}{4\pi(t_f - t)} \exp\left(-\frac{r^2}{4\pi(t_f - t)}\right) \\ q^*(\xi, x, t_f, t) &= \frac{-1}{8\pi k^2(t_f - t)^2} \exp\left(\frac{-r^2}{4\pi(t_f - t)}\right) \frac{\partial r}{\partial n} \end{aligned}$$

$c(\xi)$ 是与边界光滑相关的系数。 $c(\xi) = \beta/2\pi\alpha\beta$ 是 ξ 点的平面角或立体角, 对区域外的点 $\beta = 0$, 对区域内的点 $\beta = 2\pi$, 光滑边界上的点 $\beta = \pi$ 。

2 边界积分方程的数值解法

为了数值求解边界积分方程, 将边界 Γ 离散为 N 个常数单元, 时间被划分为 F 个常单元, 区域 Ω 划分为 L 个网格。可以用两种离散方式通过配点法得到两类方程。

格式 1:

$$\begin{aligned} c_i u_i^f + k \sum_{j=1}^N \left(\int_{\Gamma_j} \int_{t_{f-1}}^{t_f} q^* dt d\Gamma \right) u_j^n = \\ k \sum_{j=1}^N \left(\int_{\Gamma_j} \int_{t_{f-1}}^{t_f} u^* dt d\Gamma \right) q_j^n + \sum_{i=1}^L \int_{\Omega_i} u^* u_{f-1} \end{aligned}$$

此格式是把每一个时间步长看作一个新的问题, 所以在每一时间步长结束时, 必须计算足够多内部点上的值, 以便用着下一时间步长的“伪初始值”进行计算。

格式 2:

$$\begin{aligned} c_i u_i^f + k \sum_{j=1}^N \sum_{f=1}^F \left(\int_{\Gamma_j} \int_{t_{f-1}}^{t_f} q^* dt d\Gamma \right) u_j^n = \\ k \sum_{j=1}^N \sum_{f=1}^F \left(\int_{\Gamma_j} \int_{t_{f-1}}^{t_f} u^* dt d\Gamma \right) q_j^n + \sum_{i=1}^L \int_{\Omega_i} u^* u_0 \end{aligned}$$

此格式是时间积分过程始终从 t_0 开始计算, 当时间步推进时, 尽管中间步数增加了, 但不必重新计算内部点上的值, 如果 u_0 满足 Laplace 方程, 区域积分可以转化为等价的边界积分。本文的数值例子就采用这种格式。

格式 2 的矩阵形式

$$\sum_{f=1}^F H_{ff} U_f = \sum_{f=1}^F G_{ff} Q_f + B \bar{U}_0$$

格式 2 矩阵元素的计算(时间和空间都是常单元情形)

$$h_{ffv} = \hat{h}_{ffv} + c_i \delta_{ff} \delta_v$$

其中 δ_{ff}, δ_v 是克罗内克尔(Kronecker)符号

$$\hat{h}_{ffv} = k \int_{\Gamma_j} \int_{t_{f-1}}^{t_f} q^* dt d\Gamma$$

$$g_{ffv} = k \int_{\Gamma_j} \int_{t_{f-1}}^{t_f} u^* dt d\Gamma$$

$$b_d = \int_{\Omega_i} u^* d\Omega$$

$$\hat{h}_{ffv} = k \int_{\Gamma_j} \int_{t_{f-1}}^{t_f} q^* dt d\Gamma =$$

$$k \int_{\Gamma_j} \frac{-1}{2\pi k r} \frac{\partial r}{\partial n} \int_{t_{f-1}}^{t_f} \frac{r^2}{4k(t_f - t)^2} \exp\left(\frac{-r^2}{4k(t_f - t)}\right) dt d\Gamma =$$

$$\frac{-1}{2\pi} \int_{\Gamma_j} \frac{\partial r}{r \partial n} [\exp(-a_{f-1}) - \exp(-a_f)] d\Gamma$$

$$g_{ffv} = k \int_{\Gamma_j} \int_{t_{f-1}}^{t_f} u^* dt d\Gamma =$$

$$k \int_{\Gamma_j} \frac{1}{\pi r^2} \int_{t_{f-1}}^{t_f} \frac{r^2}{4k(t_f - t)} \cdot$$

$$\exp\left(\frac{-r^2}{4k(t_f - t)}\right) dt d\Gamma = k \int_{\Gamma_j} \frac{1}{4\pi k} \int_{a_{f-1}}^{a_f} \frac{e^{-x}}{x} dx d\Gamma =$$

$$\frac{1}{4\pi} \int_{\Gamma_j} [E_1(a_{f-1}) - E_1(a_f)] d\Gamma$$

$$\text{令 } x = \frac{r^2}{4k(t_f - t)}$$

式中 E_1 是指数积分函数, $a_f = \frac{r^2}{4k(t_f - t)}$

1) 当 $f \neq F$ 时, 时间积分没有奇异性, 空间积分有奇异性。

a. 当 $i = j$ 时, 由于 $\frac{\partial r}{\partial n} = 0$, 所以 $h_{ffv} = 0$

$$g_{ffv} = \frac{1}{4\pi} \int_{-l, l}^{l, l} \left[E_1\left(\frac{x^2}{4k(t_f - t_{f-1})}\right) -$$

$$E_1\left(\frac{x^2}{4k(t_f - t_f)}\right) \right] dx = \frac{l_i}{8\pi} \int_{-1}^1 E_1\left[\left(\frac{l_i^2 \eta^2}{16k(t_f - t_{f-1})}\right) - E_1\left(\frac{l_i^2 \eta^2}{16k(t_f - t_f)}\right)\right] d\eta$$

由于 $E_1(z)$ 可展开为级数, 所以此时 g_{ffv} 可以解析求出。

$$E_1(z) = -c - \ln(z) + \sum_{n=1}^{\infty} (-1)^{n-1} \frac{z^n}{nn!}$$

通过封闭形式计算

$$g_{f\bar{u}} = \frac{l_i}{4\pi} \left[\ln(\bar{a}_f) - \ln(\bar{a}_{f-1}) + \sum_{n=1}^{\infty} (-1)^{n-1} \frac{\bar{a}_{f-1}^n - \bar{a}_f^n}{n(2n+1)n!} \right] \quad (3)$$

$$\bar{a}_f = \frac{l_i^2}{16k(t_f - t_j)}$$

b. 当 $i \neq j$ 时

$$h_{f\bar{u}} = \frac{-1}{2\pi} \int_{\Gamma_j} \frac{\partial r}{r \partial n} [\exp(-a_{f-1}) - \exp(-a_f)] d\Gamma =$$

$$\frac{l_j}{4\pi} \int_{-1}^1 \frac{\partial r}{r \partial n} [\exp(-a_{f-1}) - \exp(-a_f)] d\xi \quad (4)$$

$$g_{f\bar{u}} = \frac{1}{4\pi} \int_{\Gamma_j} [E_1(a_{f-1}) - E_1(a_f)] d\Gamma$$

将指数积分函数展开为级数形式 =

$$\frac{1}{4\pi} \int_{\Gamma_j} \left[\ln(a_f) - \ln(a_{f-1}) + \sum_{n=1}^{\infty} (-1)^{n-1} \frac{a_{f-1}^n - a_f^n}{nn!} \right] d\Gamma =$$

$$\frac{l_j}{8\pi} \int_{-1}^1 \left[\ln(a_f) - \ln(a_{f-1}) + \sum_{n=1}^{\infty} (-1)^{n-1} \frac{a_{f-1}^n - a_f^n}{nn!} \right] d\xi \quad (5)$$

利用高斯积分公式进行数值计算。通常采用四点高斯积分公式就能保证所要求的精度。

2) 当 $f = F$, 时间和空间积分都有奇异性, $E_1(a_f) = 0$

a. 当 $i = j$ 时, 由于 $\frac{\partial r}{\partial n} = 0$, 所以 $h_{f\bar{u}} = 0$

$$g_{f\bar{u}} = \frac{1}{4\pi} \int_{-l/2}^{l/2} E_1\left(\frac{x^2}{4\pi\Delta t_f}\right) dx =$$

$$\frac{l_i}{8\pi} \int_{-1}^1 E_1(a\eta^2) d\eta$$

式中

$$a = \frac{l_i^2}{16k\Delta t_f}, \Delta t_f = t_f - t_{f-1} =$$

$$\frac{l_j}{8\pi} \int_{-1}^1 \left[-c - \ln(a\eta^2) + \sum_{n=1}^{\infty} (-1)^{n-1} \frac{(a\eta^2)^n}{nn!} \right] d\eta =$$

$$\frac{l_i}{4\pi} \left[2 - c - \ln(a) + \sum_{n=1}^{\infty} (-1)^{n-1} \frac{a^n}{n(2n+1)n!} \right] \quad (6)$$

计算时根据需要选择所需项数(一般 6 项就足以满足计算精度)。

b. 当 $i \neq j$ 时, 因 $\exp(-a_f) = 0, E_1(a_f) = 0,$

$$h_{f\bar{u}} = \frac{-1}{2\pi} \int_{\Gamma_j} \frac{\partial r}{r \partial n} \exp(-a_{f-1}) d\Gamma =$$

$$\frac{-l_j}{4\pi} \int_{-1}^1 \frac{\partial r}{r \partial n} \exp(-a_{f-1}) d\xi \quad (7)$$

$$g_{f\bar{u}} = \frac{1}{4\pi} \int_{\Gamma_j} E_1(a_{f-1}) d\Gamma =$$

$$\frac{1}{4\pi} \int_{\Gamma_j} \left[-c - \ln(a_{f-1}) + \sum_{n=1}^{\infty} (-1)^{n-1} \frac{a_{f-1}^n}{nn!} \right] d\Gamma =$$

$$\frac{l_j}{8\pi} \int_{-1}^1 \left[-c - \ln(a_{f-1}) + \sum_{n=1}^{\infty} (-1)^{n-1} \frac{a_{f-1}^n}{nn!} \right] d\xi \quad (8)$$

利用高斯积分进行数值计算。

当 u_0 不为 0 时必须进行区域积分

$$b_u = \int_{\Omega_i} u^* d\Omega$$

$$\int_{\Omega_i} \frac{1}{4\pi k(t_f - t)} \exp\left(\frac{-r^2}{4k(t_f - t)}\right) d\Omega \quad (9)$$

同样可利用高斯积分进行区域积分数值计算。

3 数值例子

利用上述算法来求解下面的例子:考虑一个 3 m × 3 m 的方形区域,其初始温度 $u_0 = 30^\circ\text{F}$ 和热扩散系数 $k = 1.25 \text{ m}^2/\text{h}$,对于任何 $t > t_0$ 时,沿 Γ 的狄利克莱边界条件 $u = 0$ 。

考虑区域分解方案 1,如图 2。两子区域重叠部分大小为 1.8 m × 3 m,每个区域边界被剖分为 36 个常单元,区域被划分为 40 个三角形网格。区域分解法计算结果,边界元法解、有限元法解、解析解列于表 1 中,其中有限元结果和解析解来自文献[4],可见尽管采用比较粗糙的离散格式,区域分解的解与直接边界元解和有限元解具有同样量级的精度,而且在所有点上都优于有限元解。为了得到收敛速度与子区域重叠程度的关系,作了区域分解方案 2,如图 3。

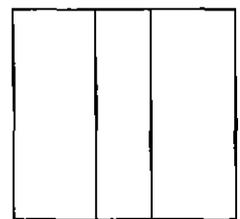
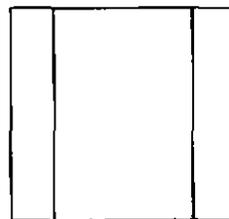


图 2 区域分解方案 1

图 3 区域分解方案 2

表 1 时间步长 $\Delta t = 0.12 \text{ h}$, $t = 1.2 \text{ h}$ 时的温度值

x	y	区域分解法	边界元	有限元 $\Delta t = 0.1 \text{ h}$	解析结果
0.0	0.0	1.916	1.963	2.108	1.812
0.3	0.0	1.834	1.868	2.005	1.723
0.6	0.0	1.571	1.594	1.706	1.466
0.9	0.0	1.154	1.177	1.239	1.065
1.2	0.0	0.620	0.626	0.652	0.560
0.3	0.3	1.745	1.777	1.907	1.639
0.6	0.6	1.275	1.287	1.380	1.186
0.9	0.9	0.681	0.674	0.728	0.626
1.2	1.2	0.168	0.186	0.201	0.173

区域分解方案2的两子区域重叠部分大小 $0.6\text{ m} \times 3\text{ m}$,每个子区域边界被剖分为30个常单元,区域划分采用方案1同样的三角形剖分。计算得到表2。

表2 重叠区域大小与收敛速度的关系

时间步	人工边界节点上的误差(需迭代次数)	
	区域分解方案1	区域分解方案2
1	$9.5367\text{E} - 7(3)$	$9.5367\text{E} - 7(12)$
2	$6.2943\text{E} - 5(2)$	$1.9073\text{E} - 5(8)$
3	$2.4796\text{E} - 5(2)$	$1.7166\text{E} - 5(8)$
4	$1.3351\text{E} - 5(2)$	$1.1444\text{E} - 5(8)$
5	$9.5367\text{E} - 7(3)$	$1.9073\text{E} - 6(10)$
6	$5.7220\text{E} - 6(2)$	$1.9073\text{E} - 6(9)$
7	$3.8147\text{E} - 6(2)$	$4.7684\text{E} - 6(8)$
8	$2.8610\text{E} - 6(2)$	$2.8610\text{E} - 6(8)$
9	$1.9073\text{E} - 6(2)$	$2.8610\text{E} - 6(8)$
10	$1.4305\text{E} - 6(2)$	$1.9073\text{E} - 6(8)$

上述数值结果表明,区域分解法是非常有效的。对于迭代控制误差($\epsilon = 1\text{e} - 5$),用方式1计算时,每个时间步只需迭代2,3次就收敛,而对于方式2每个时间步需要迭代8次以上才收敛。从数值试验,可以看出区域分解法的收敛速度与重叠区域大小有关,重叠区域越

大,收敛速度愈快。

参考文献:

- [1] 余德浩. 无界区域上基于自然边界归化的一种区域分解算法[J]. 计算数学, 1994, 16(4): 448 - 459.
- [2] 余德浩. 无界区域非重叠区域分解算法的离散化及其收敛性[J]. 计算数学, 1996, 18(3): 328 - 336.
- [3] ZHU JIALIN. Domain Decomposition Method with Boundary Elements, in Boundary Element Methods [M]. Tanaka M, DU Q. and Homma, Elsevier science Publishers, 1993, 61 - 66.
- [4] 布瑞比亚 C A, 泰勒斯 J C F, 诺贝尔 L C. 边界单元法的理论和工程应用[M]. 龙述兴, 刘腾喜, 蔡松柏译. 北京: 国防工业出版社, 1988.
- [5] 吕涛, 石济民, 林振宝. 区域分解算法[M]. 北京: 科学出版社, 1992.
- [6] 祝家麟. 椭圆边值问题的边界元分析[M]. 北京: 科学出版社, 1991.
- [7] WROBEL L C, BREBBIA C A. Time - dependent potential problems, in progress in boundary element Methods [A]. (Brebbia C A) Pentech press, London, Halstead Press, NY, 1981. 192 - 212.

Boundary Element Methods with Overlapping Domain Decomposition for Parabolic Problem

ZHANG Tai-ping, ZHU Jia-lin

(Institute of Mathematics and Physics, Chongqing University, Chongqing 400044, China)

Abstract: Boundary element method is a numerical method for solving partial differential equations. There are several formulations of boundary element method (BEM) applied to solve a parabolic differential equation. The approach, which employs time - dependent fundamental solution, allows longer time steps in time integration than other approaches, and this can cut down on time for computer implementation with high precision. Domain decomposition method, which decompose the domain that a given problem is to be solved into subdomains, has the advantages of reducing the large problem into smaller ones and reducing the complex problem into simpler ones, and allows parallel computing. An overlapping domain decomposition method is applied combining a boundary element formulation with time - dependent fundamental solution to solve a diffusion equation. Firstly, by domain decomposition, the problem divided into two problems on subdomains, and then the initial - Boundary problems are solved by boundry element method on each subdomain. Some numerical examples are presented to illustrate feasibility and efficiency of the method. The numerical experiments show that the convergence rate of the method is dependent with the overlapping degree of the subdomains.

Key words: parabolic equation; domain decomposition; boundary element; parallel algorithm

(责任编辑 张 苹)