

第二章 有限差分方法基础

- § 2.1 有限差分方法概述
- § 2.2 导数的数值逼近方法
- § 2.3 差分格式的性质
- § 2.4 发展方程的稳定性分析

§ 2.1 有限差分方法概述

以一维非定常热传导方程为例，介绍有限差分方法的概念、简单构造方法和求解过程。

2.1.1 基本方程和定解问题

$$\frac{\partial u}{\partial t} = \gamma \frac{\partial^2 u}{\partial x^2} \quad (\gamma > 0) \quad (2.1.1)$$

求解域: $(x, t) \in [0, 1] \times [0, \infty]$

$$\left. \begin{array}{l} \text{初始条件: } u(x, 0) = f(x) \\ \text{边界条件: } u(0, t) = a(t), u(1, t) = b(t) \end{array} \right\} \quad (2.1.2)$$

方程(2.1.1)和初边条件(2.1.2)构成了一个适定的定解问题。

有限差分方法: 对于一个偏微分方程，如果把方程中的所有偏导数近似地用代数差商(Algebraic Difference Quotient)代替，则可以用一组代数方程近似地替代这个偏微分方程，进而得到数值解，这种方法称为**有限差分方法(Finite Difference Method)**。

2.1.2 求解域及偏导数的离散化

为了用有限差分方法求解式 (2.1.1), 需要把其中的偏导数表示为代数形式, 为此, 首先要把自变量从连续的分布变为离散形式。这个过程称为**求解域的离散化**。

1. 空间求解域的离散化

把空间求解域分为 M 段 (均匀剖分)



网格点: $x_0, x_1, x_2, \dots, x_{M-1}, x_M$

显然, $x_k = k\Delta x$ 其中, $\Delta x = 1/M$, 为空间步长。

2. 时间变量的离散化

把感兴趣的时间段($t=T$ 之前)分为 N 段 (均匀剖分), 则时间方向的求解域可以划分为 $N+1$ 个离散时刻: $t_0, t_1, t_2, \dots, t_{N-1}, t_N$

$$t_n = n\Delta t \quad (\Delta t = T/N, \text{ 时间步长})$$

求解域被划分为一系列离散的时空网格点

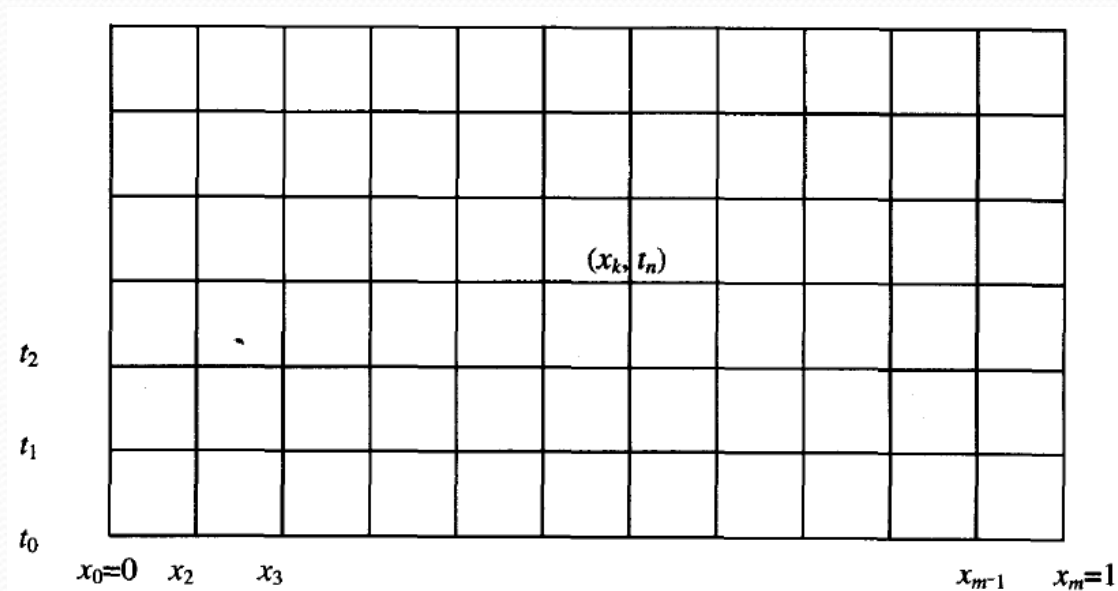


图2.1 求解域的离散化

3. 解的离散表示

目标：求出所有网格点上物理量 u 的近似解。

$$u(x_k, t_n) = u(k\Delta x, n\Delta t) \quad (k = 0, 1, \dots, M; \quad n = 0, 1, \dots, N)$$

后文中，把 $u(x_k, t_n)$ 记为 u_k^n 。

4. 导数的数值逼近

把方程中的偏导数项近似表示为代数形式。

在网格点 (x_k, t_n) , 方程(2.1.1)可以表示为

$$u_t(k\Delta x, n\Delta t) = \gamma u_{xx}(k\Delta x, n\Delta t) \quad (2.1.3)$$

或

$$(u_t)_k^n = \gamma (u_{xx})_k^n \quad (2.1.4)$$

按定义（利用Taylor展开式），偏导数 u_t 可以写成下面几种等价形式：

$$u_t(x, t) = \lim_{\Delta t \rightarrow 0} \frac{u(x, t + \Delta t) - u(x, t)}{\Delta t}$$

$$u_t(x, t) = \lim_{\Delta t \rightarrow 0} \frac{u(x, t) - u(x, t - \Delta t)}{\Delta t}$$

$$u_t(x, t) = \lim_{\Delta t \rightarrow 0} \frac{u(x, t + \Delta t) - u(x, t - \Delta t)}{2\Delta t}$$

其中, $\lim_{\Delta t \rightarrow 0}$ 后面的项称为差商 (difference quotient)。

当 Δt 足够小时, 可以用差商来近似导数。

即:

$$u_t(x, t) \approx \frac{u(x, t + \Delta t) - u(x, t)}{\Delta t}$$

$$u_t(x, t) \approx \frac{u(x, t) - u(x, t - \Delta t)}{\Delta t}$$

$$u_t(x, t) \approx \frac{u(x, t + \Delta t) - u(x, t - \Delta t)}{2\Delta t}$$

在网格点 (x_k, t_n) , 有

$$(u_t)_k^n \text{ 的向前差商: } (u_t)_k^n \approx \frac{u_k^{n+1} - u_k^n}{\Delta t} \quad (2.1.5)$$

$$(u_t)_k^n \text{ 的向后差商: } (u_t)_k^n \approx \frac{u_k^n - u_k^{n-1}}{\Delta t} \quad (2.1.6)$$

$$(u_t)_k^n \text{ 的中心差商: } (u_t)_k^n \approx \frac{u_k^{n+1} - u_k^{n-1}}{2\Delta t} \quad (2.1.7)$$

u 沿时间方向的向前差分(forward difference): $\Delta_t u_k^n = u_k^{n+1} - u_k^n$

向后差分(backward difference): $\nabla_t u_k^n = u_k^n - u_k^{n-1}$

中心差分(central difference): $\bar{\delta}_t u_k^n = u_k^{n+1} - u_k^{n-1}$

其中, Δ_t , ∇_t , $\bar{\delta}_t$ 分别称为时间方向前差、后差和中心差分算子。

空间方向的一阶偏导数 $\frac{\partial u}{\partial x}$ 可以近似为

$$(u_x)_k^n \text{ 的向前差商: } (u_x)_k^n \approx \frac{u_{k+1}^n - u_k^n}{\Delta x}$$

$$(u_x)_k^n \text{ 的向后差商: } (u_x)_k^n \approx \frac{u_k^n - u_{k-1}^n}{\Delta x}$$

$$(u_x)_k^n \text{ 的中心差商: } (u_x)_k^n \approx \frac{u_{k+1}^n - u_{k-1}^n}{2\Delta x}$$

空间方向的向前差分、向后差分和中心差分记为

$$\Delta_x u_k^n = u_{k+1}^n - u_k^n$$

$$\nabla_x u_k^n = u_k^n - u_{k-1}^n$$

$$\bar{\delta}_x u_k^n = u_{k+1}^n - u_{k-1}^n$$

其中, Δ_x , ∇_x , $\bar{\delta}_x$ 分别称为空间方向前差、后差和中心差分算子。

后文中, 将略去差分算子的下标, 简记为 Δ , ∇ , $\bar{\delta}$ 。

$$u_t(k\Delta x, n\Delta t) = \gamma u_{xx}(k\Delta x, n\Delta t) \quad (2.1.3)$$

(2.1.3)中的二阶偏导数应该如何近似呢？

根据数学分析中的知识，我们知道

$$\frac{\partial^2 u}{\partial x^2}(x, t) = \lim_{\Delta x \rightarrow 0} \frac{u(x + \Delta x, t) - 2u(x, t) + u(x - \Delta x, t)}{\Delta x^2}$$

所以，二阶导数可以近似为

$$\left(\frac{\partial^2 u}{\partial x^2} \right)_k^n = \frac{u_{k+1}^n - 2u_k^n + u_{k-1}^n}{\Delta x^2}$$

$u_{k+1}^n - 2u_k^n + u_{k-1}^n$ 称为二阶中心差分。

容易证明： $u_{k+1}^n - 2u_k^n + u_{k-1}^n = \Delta(\nabla u_k^n) = \nabla(\Delta u_k^n)$

2.1.3 差分格式

同一偏导数可以有不同的近似方法，不同的导数近似方法导致方程的不同的有限差分近似。

1. FTCS (Forward difference in Time, Central difference in Space) 格式

时间方向用前差近似，空间二阶导数用中心差分近似。

$$\frac{u_k^{n+1} - u_k^n}{\Delta t} = \gamma \frac{u_{k+1}^n - 2u_k^n + u_{k-1}^n}{\Delta x^2} \quad (2.1.9)$$

对初始条件和边界条件的离散化

$$u_k^0 = f(x_k) \quad (k = 0, 1, \dots, M) \quad (2.1.10)$$

$$u_0^n = a(t_n) \quad (n = 0, 1, \dots) \quad (2.1.11)$$

$$u_M^n = b(t_n) \quad (n = 0, 1, \dots) \quad (2.1.12)$$

式 (2.1.9) ~ (2.1.12) 称为方程 (2.1.1) 的一个有限差分方程或有限差分格式 (finite difference scheme)。

2. BTCS (Backward difference in Time, Central difference in Space) 格式

时间方向用后差近似，空间二阶导数用中心差分近似。

$$\frac{u_k^n - u_k^{n-1}}{\Delta t} = \gamma \frac{u_{k+1}^n - 2u_k^n + u_{k-1}^n}{\Delta x^2} \quad (2.1.13)$$

$$\begin{cases} u_k^0 = f(x_k) & (k = 0, 1, \dots, M) \\ u_0^n = a(t_n) & (n = 0, 1, \dots) \\ u_M^n = b(t_n) & (n = 0, 1, \dots) \end{cases}$$

在研究数值方法时，通常把 t_n 时刻的物理量视为已知量，而把 t_{n+1} 时刻的物理量作为待求的未知量。

因此，式 (2.1.13) 可以改写成

$$\frac{u_k^{n+1} - u_k^n}{\Delta t} = \gamma \frac{u_{k+1}^{n+1} - 2u_k^{n+1} + u_{k-1}^{n+1}}{\Delta x^2} \quad (2.1.14)$$

2.1.4 差分方程的求解

1. FTCS 格式

$$\frac{u_k^{n+1} - u_k^n}{\Delta t} = \gamma \frac{u_{k+1}^n - 2u_k^n + u_{k-1}^n}{\Delta x^2} \quad (2.1.9)$$

可以改写为 $u_k^{n+1} = \sigma u_{k+1}^n + (1 - 2\sigma)u_k^n + \sigma u_{k-1}^n$ (2.1.15)

其中, $\sigma = \frac{\gamma \Delta t}{\Delta x^2}$

可见, 在FTCS格式中, 某一点的数值解只依赖于前一时间步的三个点, 如图2.2所示。

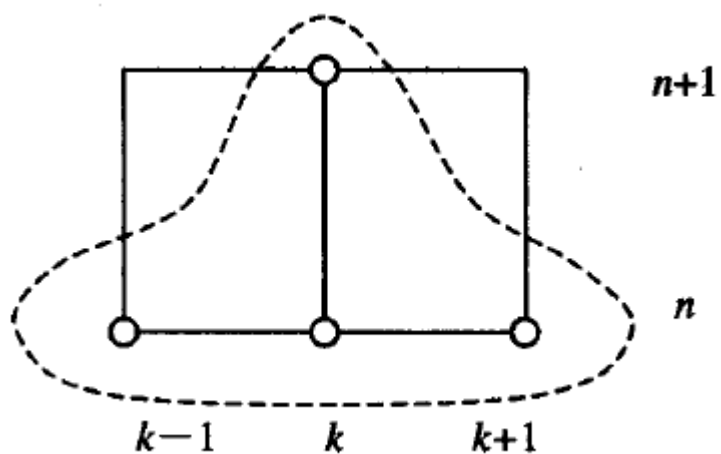
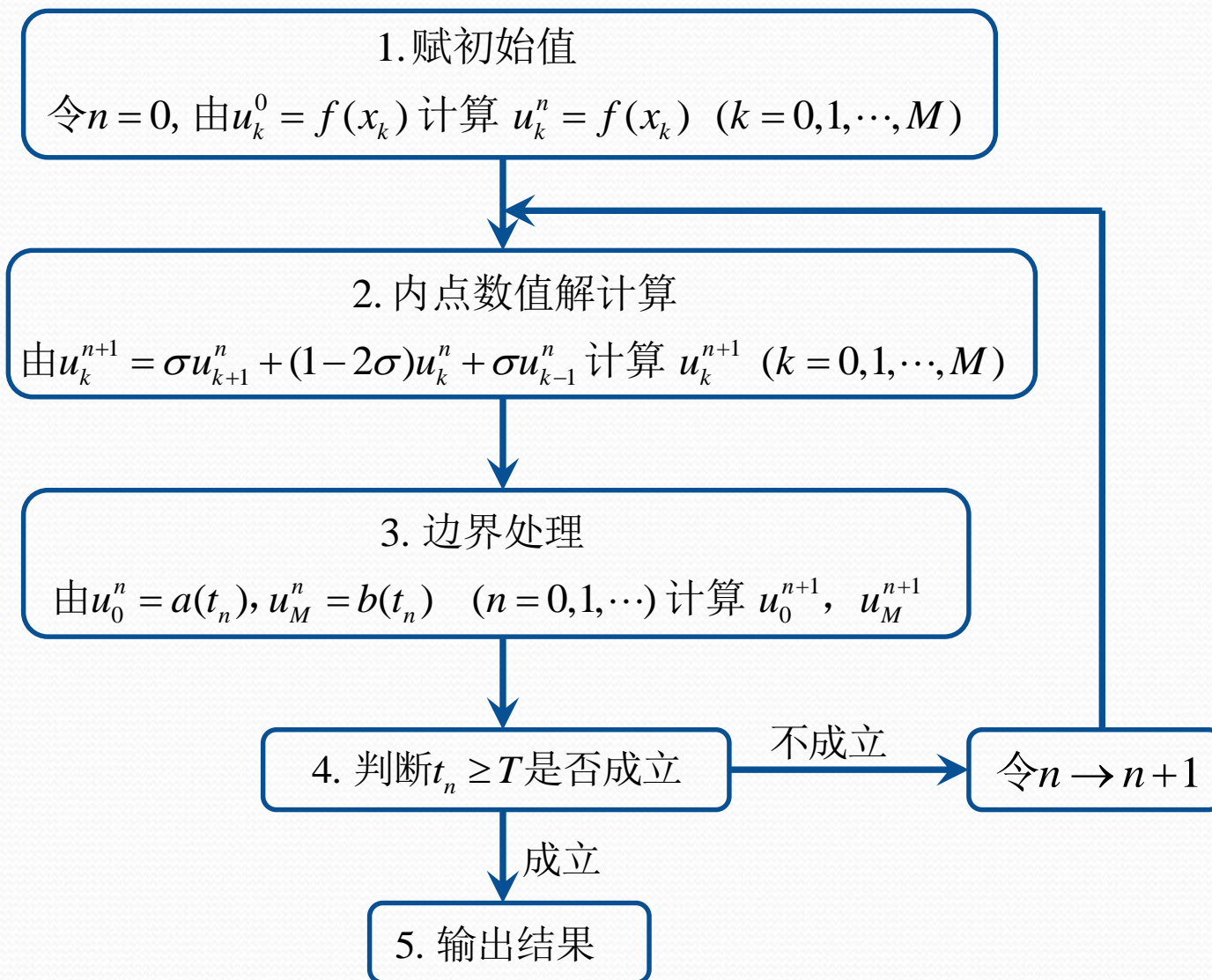


图2.2: FTCS格式的模板点

求解 u_k^{n+1} 所涉及的典型网格点称为格式的模板(stencil)点。

FTCS格式可以通过简单的递推关系(2.1.15)由某一时间步 n 的值 u_k^n 计算出下一个时间步 $n+1$ 的值 u_k^{n+1} , 称为显示格式(explicit scheme)。

FTCS格式的求解过程



2. BTCS 格式

$$\frac{u_k^{n+1} - u_k^n}{\Delta t} = \gamma \frac{u_{k+1}^{n+1} - 2u_k^{n+1} + u_{k-1}^{n+1}}{\Delta x^2} \quad (2.1.14)$$

可以改写为
$$\sigma u_{k+1}^{n+1} - (1 - 2\sigma)u_k^{n+1} + \sigma u_{k-1}^{n+1} = -u_k^n \quad (2.1.16)$$

跟FTCS格式不同，BTCS格式中同时涉及到 $n+1$ 时刻的多个未知量，不能递推求解，称为**隐式格式** (implicit scheme)。

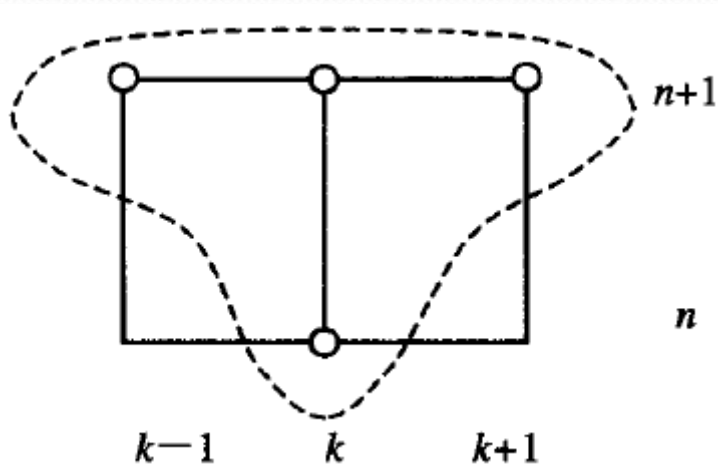


图2.3: BTCS格式的模板点

BTCS格式的求解过程

1. 赋初始值 (令 $n=0$, 计算 u_k^n)

2. 构造求解 u_k^{n+1} 的线性方程组

3. 求解线性方程组

4. 判断 $t_n \geq T$ 是否成立

不成立

令 $n \rightarrow n+1$

成立

5. 输出结果

2. 构造求解 $n+1$ 时刻数值解 u_k^{n+1} 的线性方程组

$$\sigma u_{k-1}^{n+1} - (1-2\sigma)u_k^{n+1} + \sigma u_{k+1}^{n+1} = -u_k^n \quad (2.1.16)$$

列出 $k=0,1,\dots,M$ 各点差分格式的具体形式

$$k=0: \quad u_0^{n+1} = a(t_{n+1}) \quad (\text{边界条件})$$

$$k=1: \quad \sigma u_0^{n+1} - (1-2\sigma)u_1^{n+1} + \sigma u_2^{n+1} = -u_1^n$$

$$k=2: \quad \sigma u_1^{n+1} - (1-2\sigma)u_2^{n+1} + \sigma u_3^{n+1} = -u_2^n$$

$$\vdots$$

$$k=M-1: \quad \sigma u_{M-2}^{n+1} - (1-2\sigma)u_{M-1}^{n+1} + \sigma u_M^{n+1} = -u_{M-1}^n$$

$$k=M: \quad u_M^{n+1} = b(t_{n+1}) \quad (\text{边界条件})$$

写成方程组的形式

$$\begin{pmatrix} 1 & & & & & \\ \sigma & -(1+2\sigma) & \sigma & & & 0 \\ & \sigma & -(1+2\sigma) & \sigma & & \\ & & \ddots & \ddots & \ddots & \\ & 0 & & \sigma & -(1+2\sigma) & \sigma \\ & & & & & 1 \end{pmatrix} \begin{pmatrix} u_0^{n+1} \\ u_1^{n+1} \\ u_2^{n+1} \\ \vdots \\ u_{M-1}^{n+1} \\ u_M^{n+1} \end{pmatrix} = \begin{pmatrix} a(t_{n+1}) \\ -u_1^n \\ -u_2^n \\ \vdots \\ -u_{M-1}^n \\ b(t_{n+1}) \end{pmatrix} \quad (2.1.17)$$

通过求解这个线性方程组，可以得到 $n+1$ 时刻求解域上各个网格点的数值解。

3. 求解线性方程组

系数矩阵

$$\begin{pmatrix} 1 & & & & & & \\ \sigma & -(1+2\sigma) & \sigma & & & & 0 \\ & \sigma & -(1+2\sigma) & \sigma & & & \\ & & \ddots & \ddots & \ddots & & \\ & 0 & & \sigma & -(1+2\sigma) & \sigma & \\ & & & & & & 1 \end{pmatrix}$$

只有主对角线和相邻的两条次对角线上有非零元素，这种形式的方程组称为**三对角线方程组**。

三对角线方程组可以通过**追赶法**直接求解。

追赶法：

- I. 利用一个边界条件将三对角线方程组化为只有主对角线和相邻的一条次对角线上有非零元素的方程组；
- II. 利用另一边界条件逐点求解。

追赶法：是一种高效算法，计算量与未知量个数 $M+1$ 近似呈线性关系。