МОД-КП Практика (Неделя 5, 6): Тема 3. Уравнение теплопроводности, функция `pdepe()`

1 Решение уравнения теплопроводности с помощью функции pdepe()

Цель данного практического занятия – решение смешанную краевую задачу для уравнения теплопроводности с помощью функции Матлаб `pdepe()`. Данная функция имеет структуру аналогичную таким функциям в других системах программирования. Поэтому освоение этой функции является важным и полезным.

Конкретно, мы будем решать уравнение теплопроводности в безразмерной форме:

$$\frac{\partial_t T}{\partial t} = \frac{\partial^2 T}{\partial x^2} + q_{\text{вн}}(x,t).$$

с следующими начальными и граничными условиями:

$$T(x,0) = T_0(x), \quad T(0,t) = f_0(t), \quad T(1,t) = f_1(t).$$

где $x = [0,1]$, $q_{\text{вн}}(x,t)$ – плотность внешних источников / стоков тепла. В данном случае, заданы граничные условия 1-го рода. Но в общем случае могут быть также условия 2-го и 3-го родов.

1.1 Псевдокод решения

Для решения уравнения теплопроводности будем использовать функцию `pdepe()`, стандартный вызов которой имеет следующий вид:

```
sol = pdepe(m,pdefun,icfun,bcfun,xmesh,tspan,options)
```

Таким образом, для решения УТ необходимо выполнить следующие шаги

1. Определить 3 функции: `pdefun()`, `icfun()`, `bcfun()`.
2. Задать `xmesh`, `tspan`.
3. Задать опции `options` – параметры функции `pdepe()`.

4. Решить уравнение `sol = pdepe(m,@pdefun,@icfun,@bcfun,xmesh,tspan,options)`.

5. Построить график $u(x,T)$.

Синтаксис `pdepe()` с примерами функций и основного скрипта описан в Приложении 1. Пример основного кода см. в шаблоне `pract04_shab.m`.

1.2 Задачи для Практики и Домашнего задания

1. Решить уравнение теплопроводности:

$u_t = u_{xx} + q, \quad u(x,0) = 4x(1-x).$

1) $q = 0, \quad u(0,t) = 2, \quad u(1,t) = 1.$
2) $q = 0, \quad u_x(0,t) = 2, \quad u_x(1,t) = 2.$
3) $q = 0, \quad u_x(0,t) = 2, \quad u_x(1,t) = 1.$
4) $q = 0, \quad u_x(0,t) = -1, \quad u_x(1,t) = 1.$
5) $q = 0, \quad u_x(0,t) = 1, \quad u_x(1,t) = -1.$
6) $q = 1, \quad u(0,t) = 1, \quad u(1,t) = 1.$

2. Решить уравнение теплопроводности ($u_{th}(x,t)$ – точное решение)

$$u_t = u_{xx} + \frac{2x(1+x+x\,t+(1+t)^2}{4(1+x+x\,t)^{3/2}},$$

$$u(0,t) + 2u_x(0,t) = 2+t, \quad u(1,t) + 2\,u_x(1,t) = \frac{3+2t}{\sqrt{2+t}},$$

$$u(x,0) = \sqrt{1+x}, \quad u_{th}(x,t) = \sqrt{1+x+x\,t}.$$

3. Решить уравнение теплопроводности ($u_{th}(x,t)$ – точное решение)

$$u_t = u_{xx} + \frac{3(3x^2-t-1)}{2(1+t+x^2)^2},$$

$$u(0,t) - u_x(0,t) = \frac{3}{2}\ln(1+t), \quad u_x(1,t) = \frac{3}{2+t},$$

$$u(x,0) = \frac{3}{2}\ln(1+x^2), \quad u_{th}(x,t) = \frac{3}{2}\ln(1+t+x^2).$$

A Приложение. Краткая документация по функции `pdepe()`

A.1 Какие типы УЧП решает функция `pdepe()`?

Функция `pdepe()` решает начально-краевые задачи для систем УЧП для одной пространственной переменной x и времени t.

Функция `pdepe()` использует неформальную классификацию для одномерных уравнений, которые он решает:

Параболические уравнения, например, уравнение теплопроводности $\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}$.

Эллиптические уравнения, например, уравнение Лапласа $\frac{\partial^2 T}{\partial x^2} = 0$.

Функция `pdepe()` также решает некоторые двумерные и трехмерные задачи, которые сводятся к одномерным задачам из-за угловой симметрии.

Набор инструментов PDE Toolbox расширяет эту функциональность на обобщенные задачи в 2D и 3D с граничными условиями Дирихле и Неймана.

A.2 Решение одномерного УЧП

Одномерное УЧП определяет функцию $u(x,t)$, которая зависит от времени t, и одной пространственной переменной x. Функция `pdepe()` решает системы 1D параболических и эллиптических УЧП вида

$$c(x,t,u,\frac{\partial u}{\partial x})\,\frac{\partial u}{\partial t} = x^{-m}\frac{\partial}{\partial x}\left(x^m f(x,t,u,\frac{\partial u}{\partial x})\right) + s(x,t,u,\frac{\partial u}{\partial x}). \quad (1)$$

Уравнение обладает свойствами:

Области независимых переменных: $t = [0, t_{end}]$, $\quad x = [a, b]$.

Параметр m может быть 0, 1 или 2, что соответствует плоской, цилиндрической или сферической симметрии, соответственно. Если $m > 0$, то $a \geq 0$.

Функция $f(x,t,u,\frac{\partial u}{\partial x})$ опрределяет поток, является членом потока, а $s(x,t,u,\frac{\partial u}{\partial x})$ – плотность источников. Член потока должен зависеть от частной производной $\frac{\partial u}{\partial x}$.

Связь частных производных по времени ограничивается умножением на диагональную матрицу $c(x,t,u,\frac{\partial u}{\partial x})$. Диагональные элементы этой матрицы либо равны нулю, либо положительны. Элемент, равный нулю, соответствует эллиптическому уравнению, а любой другой элемент соответствует параболическому уравнению. Должно быть хотя бы одно параболическое уравнение. Элемент c, соответствующий параболическому уравнению, может обращаться в нуль при изолированных значениях x, если они являются точками сетки (точками, в которых оценивается решение).

A.3 Процесс решения

Чтобы решить УЧП с помощью `pdepe()`, надо определить коэффициенты уравнения для c, f и s, начальные условия, поведение решения на границах и сетку точек для оценки решения. Вызов функции

`sol = pdepe(m,pdefun,icfun,bcfun,xmesh,tspan)`

дает решение на указанной сетке. Аргументами функции являются:

- `m` – параметр симметрии уравнения.
- `pdefun` – функция, определяющая решаемые уравнения.
- `icfun` – функция, определяющая начальные условия.
- `bcfun` – функция, определяющая граничные условия.
- `xmesh` – вектор пространственных значений для x.

- tspan – вектор значений времени для t.

Вместе векторы xmesh и tspan образуют двумерную сетку, на которой pdepe() оценивает решение.

A.3.1 Задание уравнений

Нужно выразить УЧП в стандартной форме, ожидаемой функцией pdepe(). Это можно сделать с помощью функции следующего вида, которая возвращает значения коэффициентов c, f и s:

```
function [c,f,s] = pdefun(x,t,u,dudx)
c = 1;
f = dudx;
s = 0;
end
```

В этом случае pdefun() определяет уравнение $\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}$. Если задано несколько уравнений, то c, f и s являются векторами, каждый элемент которых соответствует одному уравнению.

A.3.2 Задание начальных условий

В начальный момент времени $t = 0$ для всех x компоненты решения удовлетворяют начальным условиям вида

$$u(x, 0) = u_0(x).$$

В MATLAB вы можете закодировать начальные условия с помощью функции вида

```
function u0 = icfun(x)
u0 = 0.5;
end
```

В этом случае $u_0 = 1$ определяет начальное условие $u(x, 0) = 1$. Если существует несколько уравнений, то u_0 – это вектор, каждый элемент которого определяет начальное условие одного уравнения.

A.3.3 Задание граничных условий

На границе $x = a$ или $x = b$ для всех t компоненты решения удовлетворяют граничным условиям вида

$$p(x, t, u) + q(x, t)f(x, t, u, \frac{\partial u}{\partial x}) = 0,$$

где $q(x, t)$ – диагональная матрица с элементами, которые либо равны нулю, либо никогда не равны нулю. Обратите внимание, что граничные условия выражены в терминах потока f, а не в виде частной производной $\partial u/\partial x$. Кроме того, из двух коэффициентов $p(x,t,u)$ и $q(x,t)$ только p может зависеть от u.

В MATLAB вы можете закодировать граничные условия с помощью функции вида

```
function [pL,qL,pR,qR] = bcfun(xL,uL,xR,uR,t)
pL = uL;      qL = 0;
pR = uR - 1;  qR = 0;
end
```

Переменные `pL` и `qL` являются коэффициентами для левой границы, в то время как `pR` и `qR` являются коэффициентами для правой границы. В этом случае `bcfun()` определяет граничные условия

$$u(x_L,t) = 0, \quad u(x_R,t) = 1.$$

Если существует несколько уравнений, то выходные данные `pL`, `qL`, `pR` и `qR` являются векторами, каждый элемент которых определяет граничное условие одного уравнения.

A.3.4 Задание опций интегрирования

Опции интегрирования по умолчанию в функции `pdepe()` выбраны для решения распространенных проблем. В некоторых случаях вы можете повысить производительность решателя, переопределив эти значения по умолчанию. Для этого используйте `odeset()` для создания необходимых параметров. Затем передайте эти параметры в функцию `pdepe()` в качестве последнего входного аргумента.

```
options=odeset('RelTol',1e-7,'AbsTol',1e-7);
sol = pdepe(m,pdefun,icfun,bcfun,xmesh,tspan,options)
```

Здесь задаются относительная точность (`RelTol`) и абсолютная точность (`AbsTol`).

A.3.5 Оценка решения

Функция `pdepe()` возвращает решение в виде трехмерного массива функции `sol`, где `sol(i,j,k)` содержит k-ю составляющую решения, оцененную при $t(i)$ и $x(j)$. В общем случае вы можете извлечь k-ю компоненту решения с помощью команды `u = sol(:,:,k)`.

Указанная вами временная сетка используется исключительно для целей вывода и не влияет на внутренние временные шаги, выполняемые решателем. Однако указанная вами пространственная сетка может повлиять на качество и скорость решения.

A.3.6 Основной скрипт

```
% Используем сетку из 20 точек по x  и 30 точек по t.
% Т.к. решение быстро достигает устойчивого состояния, временные точки
% вблизи  t = 0 расположены ближе  друг к другу, чтобы учесть это поведение.
L = 1;
x = linspace(0,L,20);
t = [linspace(0,0.05,20), linspace(0.5,5,10)];

% Решаем уравнение теплопроводности, используя функции для задания уравнения,
% начальных условий и граничных условий
m = 0;
sol = pdepe(m,@heatpde,@heatic,@heatbc,x,t);

% Строим график, используя  функцию imagesc() (можно использовать surf(x,t,u))
colormap hot
imagesc(x,t,sol)
colorbar
xlabel('Координата x')
ylabel('Время t')
title('Уравнение теплопроводности для  x = [0, 1], t = [0,5]')
```

Дополнительная информация

Более подробную информацию по функции `pdepe()` см. в подсказке Матлаба.