

Одномерная задача теплопроводности

Ищется температура $u(x, t)$ при $x \in [a; b]$, $t \geq 0$; коэффициент теплопроводности α , плотность источников тепла f . Начально-краевая задача:

$$\begin{cases} u'_t(x, t) = (\alpha(x) u'_x(x, t))'_x + f(x); \\ u|_{x=a} = 0, \quad u|_{x=b} = 0; \\ u|_{t=0} = u_0(x). \end{cases}$$

Сетка: $h = \frac{b-a}{n}$, $x_i = a + i \cdot h$, $i = 0, 1, \dots, n$; $x_0 = a$, $x_n = b$; $t_k = k \cdot \tau$, $k = 0, 1, 2, \dots$. Ищем $u_{ik} \approx u(x_i, t_k)$.

Явная разностная схема:

$$\frac{u_{i,k+1} - u_{i,k}}{\tau} = \frac{\alpha_{i-\frac{1}{2}}}{h^2} u_{i-1,k} - \frac{\alpha_{i-\frac{1}{2}} + \alpha_{i+\frac{1}{2}}}{h^2} u_{i,k} + \frac{\alpha_{i+\frac{1}{2}}}{h^2} u_{i+1,k} + f_i, \quad i = 1, 2, \dots, n-1$$

(подразумевается $\alpha_{i \pm \frac{1}{2}} = \alpha(x_{i \pm \frac{1}{2}}) = \alpha(x_i \pm \frac{h}{2})$ и $f_i = f(x_i)$).

Неявная разностная схема:

$$\frac{u_{i,k+1} - u_{i,k}}{\tau} = \frac{\alpha_{i-\frac{1}{2}}}{h^2} u_{i-1,k+1} - \frac{\alpha_{i-\frac{1}{2}} + \alpha_{i+\frac{1}{2}}}{h^2} u_{i,k+1} + \frac{\alpha_{i+\frac{1}{2}}}{h^2} u_{i+1,k+1} + f_i, \quad i = 1, 2, \dots, n-1$$

В матричной форме, соответственно,

$$\vec{u}_{k+1} = \vec{u}_k + \tau(A_h \vec{u}_k + \vec{f}) \quad \text{— явная схема;}$$

$$(E - \tau A_h) \vec{u}_{k+1} = \vec{u}_k + \tau \vec{f} \quad \text{— неявная схема.}$$

Здесь $\vec{f} = (f_i)_{i=0}^n$, $\vec{u}_k = (u_{i,k})_{i=0}^n$ (имеются в виду числовые столбцы), E — единичная матрица и

$$A_h = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ \frac{\alpha_{\frac{1}{2}}}{h^2} & -\frac{\alpha_{\frac{1}{2}} + \alpha_{\frac{3}{2}}}{h^2} & \frac{\alpha_{\frac{3}{2}}}{h^2} & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & \frac{\alpha_{\frac{3}{2}}}{h^2} & -\frac{\alpha_{\frac{3}{2}} + \alpha_{\frac{5}{2}}}{h^2} & \frac{\alpha_{\frac{5}{2}}}{h^2} & 0 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \dots & \frac{\alpha_{n-\frac{5}{2}}}{h^2} & -\frac{\alpha_{n-\frac{5}{2}} + \alpha_{n-\frac{3}{2}}}{h^2} & \frac{\alpha_{n-\frac{3}{2}}}{h^2} \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 & \frac{\alpha_{n-\frac{3}{2}}}{h^2} & -\frac{\alpha_{n-\frac{3}{2}} + \alpha_{n-\frac{1}{2}}}{h^2} \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 1 \end{pmatrix}$$

— матрица разностного оператора. Её первая и последняя строки отвечают граничным условиям Дирихле; соответственно, первый и последний элементы столбца \vec{f} следует переопределить: $f_0 = 0$, $f_n = 0$.

Неявная схема абсолютно устойчива, т.е. разностное решение сходится к истинному при стремлении к нулю шагов h и τ независимо от соотношения между ними (например, можно задать и по координате, и по времени одинаковое количество узлов). Для устойчивости явной же схемы должно выполняться условие

$$\tau < \frac{h^2}{2 \max_x \alpha(x)}.$$

Это условие довольно обременительно, т.к. резко увеличивает время вычислений по сравнению с неявной схемой (необходимое для которой решение системы линейных уравнений с трёхдиагональной матрицей требует времени того же порядка, что и умножение на эту матрицу).

В работе требуется вычислить решение по явной и по неявной схемам, оценить максимально допустимый для явной схемы шаг τ при данном h и, отталкиваясь от этой оценки, опытным путём подобрать фактическое значение τ , при котором только начинает наблюдаться неустойчивость.

Отчёт должен содержать (помимо листингов *.m-файлов):

- 1) графики $\alpha(x)$, $f(x)$ и $u_0(x)$ (в последнем случае надо раздвинуть график по вертикали, иначе его горизонтальные участки сольются с краями окна);
- 2) трёхмерные графики решения по неявной и явной схемам (на графике явной схемы должна наблюдаться слабая неустойчивость на фоне примерно той же картинки, что и для неявной);
- 3) максимум $\alpha(x)$ (оценивается по графику, т.е. по использованному для его построению массиву) и вычисленная по нему теоретическая оценка максимального значения τ ;
- 4) фактическое значение τ , при котором начинается неустойчивость.

Фактическая граница устойчивости должна оказаться немного выше, чем теоретическая; подбирается она далеко не сразу. Скажем, в приведённом ниже примере она выше в 1.04366 раза; при этом превышение лишь в 1.0436 раз видимой неустойчивости ещё не даёт, превышение же в 1.0437 раз приводит к неустойчивости уже катастрофической. На отчётном графике должны наблюдаться регулярное поведение температуры в начале и заметные осцилляции, характеризующие неустойчивость, в конце.

Рекомендация разбивать отрезок на именно 100 частей обусловлена не столько тем, что при большем n понадобилось бы слишком много времени для явной схемы (при машинном счёте это не существенно), сколько тем, что слишком густая сетка не будет смотреться на чёрно-белой распечатке (по этой же причине выводить на график следует не каждый временной слой, а примерно такое же их количество, сколько есть координатных узлов).

Демонстрационный пример:

$$\alpha(x) = \frac{1}{\operatorname{tg} \frac{x}{6} + 1}, \quad f(x) = \frac{1}{x^2 + 1}, \quad u_0(x) = \begin{cases} 3 & (x < 1); \\ 1 & (1 \leq x < 2); \\ 4 & (x \geq 2). \end{cases} \quad x \in [0; 3], \quad t_{\max} = 9$$

Задание функций:

alpha.m

```
function y=alpha(x)
y=1./(tan(x/6)+1);
```

f.m

```
function y=f(x)
y=1./(x.^2+1);
```

u0.m

```
function y=u0(x)
y=3*(x<1) + 1*(x>=1).*(x<2) + 4*(x>=2);
```

Пояснение. Логические значения TRUE и FALSE представляются в Matlab числами 1 и 0 соответственно. Поэтому использование логических операций в арифметических выражениях позволяет задавать многовариантные функции (применять же для этой цели условный оператор `if` крайне невыгодно). Следует только следить за строгостью неравенств для того, чтобы избежать потерю или дублирование точек.

Создание матрицы разностного оператора:

getah.m

```
function a=getah(x)
% Составляем базовую матрицу разностного оператора
h=x(2)-x(1);
% Побочные диагонали:
d1=alpha(x-h/2)/h^2;    d1(1)=[];
% Главная диагональ:
d0=-[0, d1] - [d1, 0];
% Создаём трёхдиагональную матрицу:
a=diag(d0) + diag(d1, 1) + diag(d1, -1);
% Корректируем первую и последнюю строки под задачу Дирихле:
[n,m]=size(a);
a(1,1)=1;  a(1,2)=0;    a(n,m)=1;  a(n,m-1)=0;
```

Пояснения. Команда `d1(1)=[]` удаляет из числовой строки `d1` первый элемент (отвечавший неиспользуемому узлу $x_{-1/2}$). Выражения `[0,d1]` и `[d1,0]` добавляют к строке нулевой элемент слева и справа. Второй аргумент команды `diag` указывает, с каким смещением относительно главной диагонали должна быть вставлена строка в матрицу.

Стартовый сценарий (задание параметров, вывод графиков функций и создание матрицы оператора):

`wb.m`

```
format short g
a=0, b=3, tmax=9
n=100;      % к-во отрезков по t (для вывода) и по x
h=(b-a)/n;   x=a:h:b;
% Нужно для оценки максимального tau в явной задаче:
amax=max(alpha(x))    % (у нас alpha>0)

plot(x, alpha(x)),    grid on
title(sprintf('\alpha(x): max=%1.5g', amax)),    pause;
print('-dpng', 'gr_alpha')

plot(x, f(x)),    grid on
title('f(x)'),    pause;
print('-dpng', 'gr_f')

% При выводе ступенчатой функции надо немного расширить
% вертикальные границы графика:
umin=min(u0(x));      umax=max(u0(x));
uming=umin-(umax-umin)/20;    umaxg=umax+(umax-umin)/20;

plot(x, u0(x)),    grid on
ylim([uming, umaxg])    % - задание вертикальных границ
title('u_0(x)')
print('-dpng', 'gr_u0')

% Создаём матрицу разностного оператора:
ah=getah(x);
```

Пояснение. При выводе в заголовок графика символ подчёркивания означает переход к нижнему индексу; комбинация `\\alpha` выводит греческую букву α .

Сценарий, реализующий неявную схему:

w_im.m

```
tau=tmax/n
t=0;    tt=t;           % - массив времён для графика

ff=f(x)';
ff(1)=0;    ff(n+1)=0; % - для поддержания граничных условий

u=u0(x)';
u(1)=0;    u(n+1)=0;    % - граничные условия в нач. момент
uu=u;      % - для накопления массива температур

for i=1:n
    t=t+tau;
    u=(eye(n+1) - tau*ah) \ (u+tau*ff);
    uu=[uu, u];
    tt=[tt, t];
end

[xu, tu]=meshgrid(x, tt);
mesh(tt,x, uu)
title(sprintf('Implicit (x: %d; t: %d) tau=%1.5g', n,n,tau))
print('-dpng', 'gr_imp')
```

Пояснения. Апострофы при задании `ff` и `u` в начале сценария означают транспонирование (массив `x` — это строка, а нам нужны столбцы). Функция `eye` возвращает единичную матрицу; символ `\` означает деление на матрицу слева (т.е. умножение на обратную). Внутри цикла накапливаются двумерный массив `uu` температур и одномерный `tt` временных узлов (последний нужен только для оцифровки временной оси на графике).

Сценарий, реализующий явную схему:

w_ex.m

```
taumax=h^2/(2*amax)      % - теор. граница устойчивости
tau=taumax*1.04366;

t=0;   tt=t;              % - массив времён для графика
nt=round(tmax/tau);       % - количество шагов по времени
dt=round(nt/n);           % - через сколько шагов выводить

ff=f(x)';
ff(1)=0;   ff(n+1)=0;     % - для поддержания граничных усл.

u=u0(x)';
u(1)=0;   u(n+1)=0;       % - граничные условия в нач. момент
uu=u;                                           % - для накопления температур

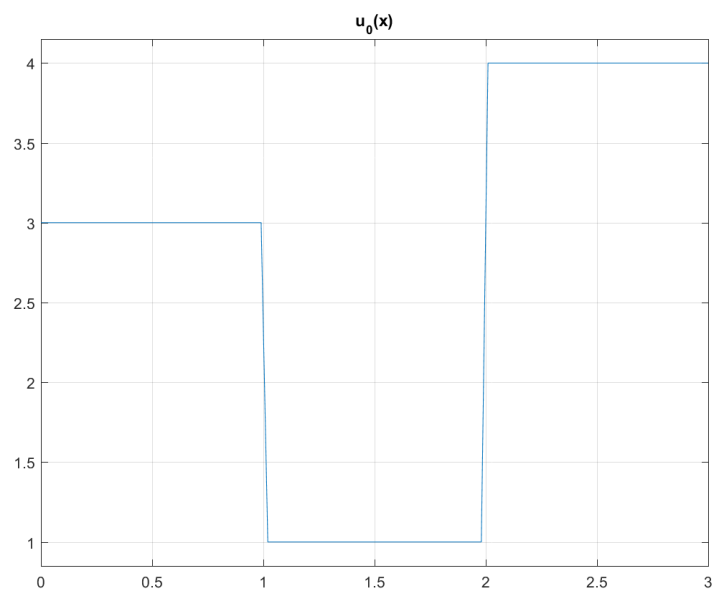
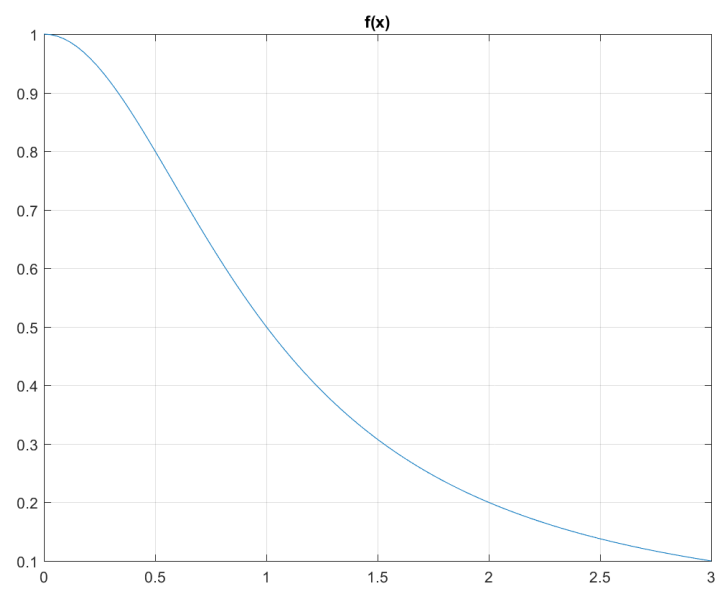
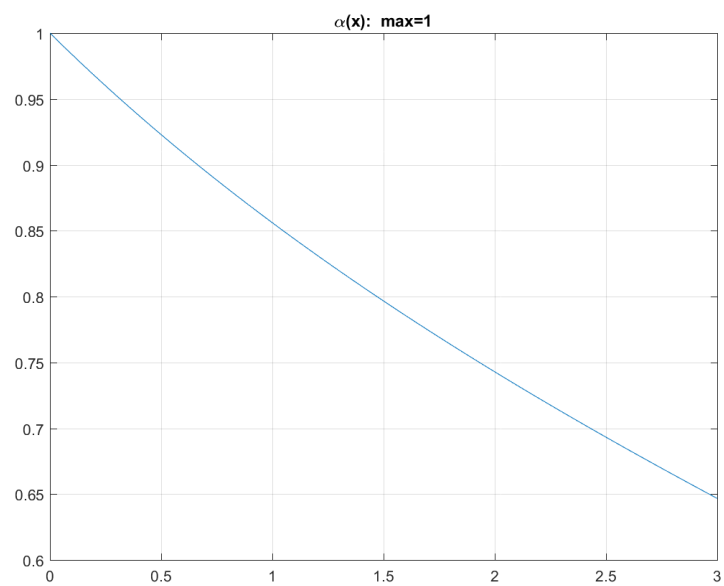
for i=1:nt
    t=t+tau;
    u=u + tau*(ah*u + ff);
    if mod(i,dt)==0
        uu=[uu, u];
        tt=[tt, t];
    end
end

[xu, tu]=meshgrid(x, tt);
mesh(tt,x, uu)
title(sprintf('Explicit (x: %d; t: %d) tau=%1.5g', n,n,tau))
print('-dpng', '_exp')
```

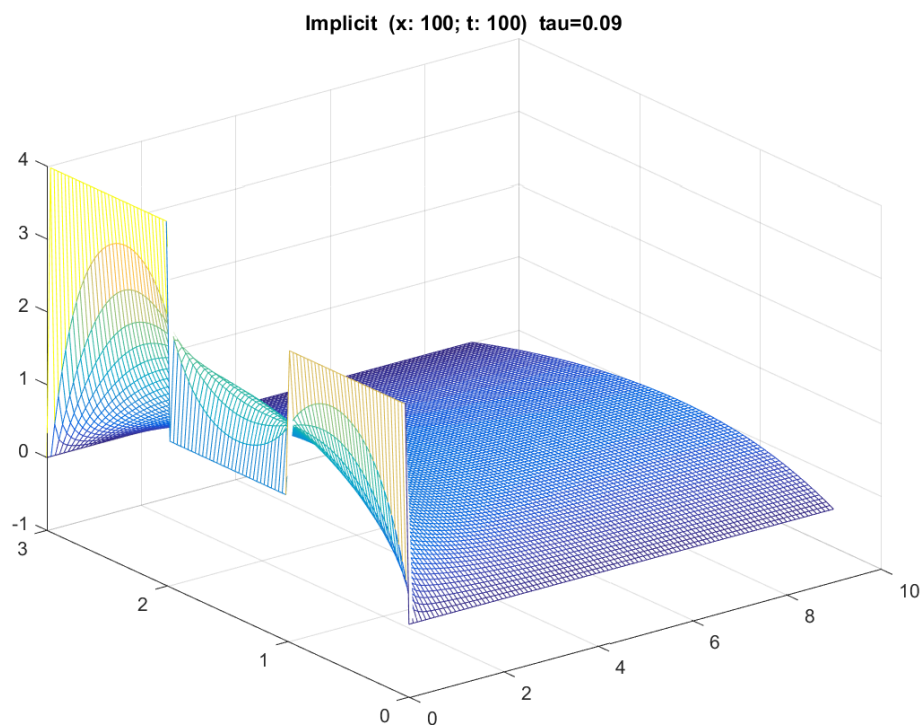
Пояснение. Внутри цикла массив температур, выводимый затем на трёхмерный график, накапливается не для всех временных слоёв (их в явной схеме слишком много), а только через каждые `dt` шагов.

Ниже приведены результаты запуска этих трёх сценариев.

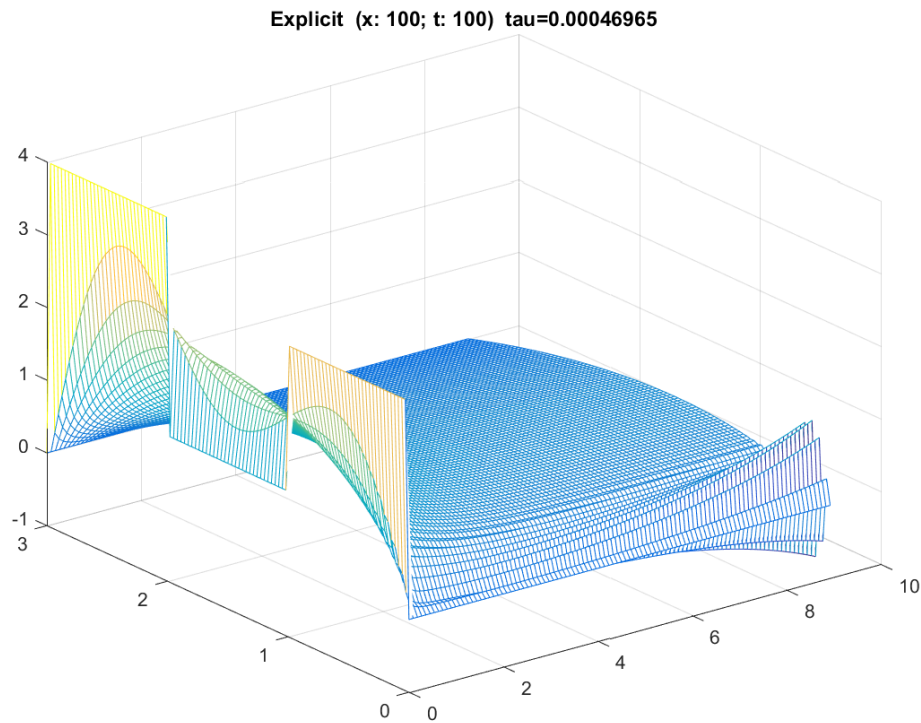
Коэффициенты уравнения теплопроводности и начальное условие:



Результат расчёта по неявной схеме:



Результат расчёта по явной схеме:



Видно, что где-то около $t = 4$ начинает проявляться численная неустойчивость на правом краю графика (в окрестности $x = 0$ — там, где $\alpha(x)$ максимальна).