Правительство Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования

«Санкт-Петербургский государственный университет»

Факультет прикладной математики — процессов управления

Метод прогонки

Студента группы 547

Суратова В. А.

Проверил:

д.ф.-м.н., профессор
Перегудин С. И.

Санкт-Петербург

2015

Задача (17 вариант)

Решить краевую задачу методом прогонки.

$$y'' + \left(1 - \frac{x^2}{4}\right) * y' + \left(2 + 0.1 * cos(x)\right) * y = \frac{1}{exp(x) + 1}$$

$$0.7 * y(0) - 0.5 * y'(0) = 0$$

$$0.53 * y(2) + 0.19 * y'(2) = -0.4$$

Метод прогонки

Пусть на отрезке $[a, b]$ требуется найти решение дифференциального уравнения:

$$y'' + p(x) * y' + q(x) * y = f(x)$$

Удовлетворяющее следующим краевым условиям:

$$c_1 y(a) + c_2 y'(a) = c$$

$$d_1 y(b) + d_2 y'(b) = d$$
$$|c_1| + |c_2| \neq 0$$
$$|d_1| + |d_2| \neq 0$$

Используем равномерную сетку:

$$h = \frac{b-a}{n}, x_0 = a, x_n = b$$
$$x_i = a + ih = x_0 + ih, i = 0,1, \dots, n$$

Пусть:

$$p(x_i) = p_i$$

$$q(x_i) = q_i$$

$$f(x_i) = f_i$$

$$y(x_i) = y_i$$

$$y'(x_i) = y_i'$$

$$y''(x_i) = y_i''$$

Аппроксимируем первую и вторую производные в каждом внутреннем узле центральными разностными производными:

$$y'(x_i) \approx \frac{y_{i+1} - y_{i-1}}{2h}$$
$$y''(x_i) \approx \frac{y_{i+1} + 2y_i - y_{i-1}}{h^2}$$

И на концах отрезка – односторонними производными:

$$y_0' \approx \frac{y_1 - y_0}{h}, y_n' \approx \frac{y_n - y_{n-1}}{h}$$

Используя эти формулы получаем разностную аппроксимацию исходной задачи:

$$\begin{cases} \dfrac{y_{i+1} + 2y_i - y_{i-1}}{h^2} + p_i \dfrac{y_{i+1} - y_{i-1}}{2h} + q_i y_i = f_i, i = 1,2, \dots, n-1 \\ c_1 y_0 + c_2 \dfrac{y_1 - y_0}{h} = c, d_1 y_n + d_2 \dfrac{y_n - y_{n-1}}{h} = d \end{cases}$$

Чтобы найти приближенные значения $y_0, y_1, \dots, y_n$ искомого решения, необходимо решить систему $n + 1$ линейных уравнений с $n + 1$ неизвестными. Матрица этой системы трехдиагональная, поэтому применим для нее метод прогонки.

Перепишем систему следующим образом:

$$\begin{cases} \beta_0 y_0 + \gamma_0 y_1 = \phi_0 \\ \alpha_i y_{i-1} + \beta_i y_i + \gamma_i y_{i+1} = \phi_i, \qquad i = 1,2, \dots, n-1 \\ \alpha_n y_{n-1} + \beta_n y_n = \phi_n \end{cases}$$

Где:

$$\begin{aligned} &\beta_0 = c_1 h - c_2 \\ &\gamma_0 = c_2 \\ &\phi_0 = hc \\ &\phi_i = f_i h^2 \\ &\alpha_i = 1 - \frac{1}{2} p_i h \\ &\beta_i = q_i h^2 - 2 \\ &\gamma_i = 1 + \frac{1}{2} p_i h \\ &i = 1,2, \dots, n-1 \\ &\alpha_n = -d_2 \\ &\beta_n = h d_1 + d_2 \\ &\phi_n = hd \end{aligned}$$

Будем искать решение новой системы в виде:

$$y_i = u_i + v_i y_{i+1}$$

Тогда для u_i и v_i получаем следующие рекуррентные формулы:

$$v_i = -\frac{\gamma_i}{\beta_i - \alpha_i v_{i-1}}$$

$$u_i = \frac{\phi_i - \alpha_i u_{i-1}}{\beta_i + \alpha_i v_i}$$
$$i = 0,1, \dots, n$$

Чтобы сделать систему однородной положим:

$$\alpha_0 = 0, \gamma_n = 0$$

Прямой ход прогонки состоит в последовательном вычислении коэффициентов u_i и v_i, исходя из значений:

$$v_0 = -\frac{\gamma_0}{\beta_0}$$

$$u_i = \frac{\phi_0}{\beta_0}$$

При обратном ходе прогонки по формуле:

$$y_i = u_i + v_i y_{i+1}$$

Последовательно определяются величины $y_n, y_{n-1}, \dots, y_0$. Так как $\gamma_n = 0$, то $v_n = 0$ и $y_n = 0$.

Таким образом, метод прогонки позволяет найти точное решение первой системы, а, значит, и найти решение краевой задачи.

Реализация в среде MATLAB

Файл `p_function.m`

```
function y = p_function( x )
y = 1 - (x^2) / 4;
end
```

Файл `q_function.m`

```
function y = q_function( x )
y = 2 + 0.1 * cos(x);
end
```

Файл `f_function.m`

```
function y = f_function( x )
y = 1 / (exp(x) + 1);
end
```

Файл tma.m

```
clear all;
a = 0;
b = 2;
c_1 = 0.7;
c_2 = -0.5;
c = 0;
d_1 = 0.53;
d_2 = 0.19;
d = -0.4;
N = 50;

% calculate step
h = (b - a)/N;
%create arrays
u        = zeros(N+1, 1);
v        = zeros(N+1, 1);
alpha    = zeros(N+1, 1);
beta     = zeros(N+1, 1);
gamma    = zeros(N+1, 1);
phi      = zeros(N+1, 1);
x        = zeros(N+1, 1);
y        = zeros(N+1, 1);
%initialize
alpha(1)     = 0;
beta(1)      = c_1 * h - c_2;
gamma(1)     = c_2;
phi(1)       = h * c;
x(1)         = a;
alpha(N+1)   = - d_2;
beta(N+1)    = h * d_1 + d_2;
gamma(N+1)   = 0;
phi(N+1)     = h * d;
x(N+1)       = b;
u(1)         = phi(1) / beta (1);
v(1)         = - gamma(1) / beta(1);

%direct
```

```
for i = 2 : N
    x(i)         = x(i - 1) + h;
    alpha(i)     = 1 - 0.5 * p_function( x(i) ) * h;
    beta(i)      = q_function( x(i) ) * h^2 - 2;
    gamma(i)     = 1 + 0.5 * p_function( x(i) ) * h;
    phi(i)       = f_function( x(i) ) * h^2;

    v(i)= - gamma(i) / ( beta(i) - alpha(i) * v(i-1) );
    u(i)= ( phi(i) - alpha(i) * u(i-1) ) / ( beta(i) + alpha(i) * v(i) );
end

v(N+1)= - gamma(N+1) / ( beta(N+1) - alpha(N+1) * v(N) );
u(N+1)= ( phi(N+1) - alpha(N+1)*u(N))/( beta(N+1) + alpha(N+1)*v(N+1) );

%reverse
y(N+1) = u(N+1);
for i = N:-1:1
    y(i) = u(i) + v(i) * y(i+1);
end
plot(x, y, 'b');
hold on;
plot(x, y, 'bo');
```