

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.

# Метод прогонки

Пусть на отрезке требуется найти решение дифференциального уравнения:

Удовлетворяющее следующим краевым условиям:

Используем равномерную сетку:

Пусть:

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

И на концах отрезка – односторонними производными:

Используя эти формулы получаем разностную аппроксимацию исходной задачи:

Чтобы найти приближенныe значения искомого решения, необходимо решить систему линейных уравнений с неизвестными. Матрица этой системы трехдиагональная, поэтому применим для нее метод прогонки.

Перепишем систему следующим образом:

Где:

Будем искать решение новой системы в виде:

Тогда для и получаем следующие рекуррентные формулы:

Чтобы сделать систему однородной положим:

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

При обратном ходе прогонки по формуле:

Последовательно определяются величины . Так как , то и .

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');

