

Решение краевых задач

Метод суперпозиции

Задача № 1

Вязкоупругая жидкость – это жидкость, обладающая памятью. При исследовании течения такой жидкости вдоль бесконечной горизонтальной плоскости приходим к следующей граничной задаче:

$$\nu * \frac{d^2 F}{d\eta^2} - (\lambda_0 \beta^2 + \beta) F = 0, \quad F(0) = U_0, \quad F(\infty) = 0.$$

Рассмотрим простейший случай: параметры в уравнении $\nu, \lambda, \beta = \text{const}$. В новых переменных $y = F / U_0, x = \eta \sqrt{(\lambda_0 \beta^2 + \beta) / \nu}$ задача имеет вид:

$$\frac{d^2 y}{dx^2} - y = 0, y(0) = 1, y(\infty) = 0.$$

В данном случае задача может быть решена аналитически:

$$F = U_0 \exp(-\sqrt{(\lambda_0 \beta^2 + \beta) / \nu} * \eta).$$

Используем аналитическое решение для сравнения с численным решением, полученным методом суперпозиции.

Решением задачи $y(x)$ является суперпозиция функций $y_1(x)$ и $y_2(x)$, т.е. $y(x) = y_1(x) + \mu y_2(x)$.

Для функций $y_1(x)$ и $y_2(x)$ имеем две задачи Коши:

$$\begin{aligned} \frac{d^2 y_1}{dx^2} - y_1 &= 0, y_1(0) = 1, y_1'(0) = 0, \\ \frac{d^2 y_2}{dx^2} - y_2 &= 0, y_2(0) = 0, y_2'(0) = 1. \end{aligned}$$

Константа μ определяется из соотношения $\mu = -y_1(\infty) / y_2(\infty)$.

Решение (предварительное)

% Метод суперпозиции

xstart = 0; xfinis = 8;

n=200;

h=(xfinis-xstart)/(n-1); %Вычисление шага

% ystart = 0.0; yfinis = 0;

y1(1)= 1; V1(1)=0;

y2(1)= 0; V2(1)=1;

% Формирование системы равноотстоящих узлов xi

for i=1:n

x(i)=xstart+(i-1)*h;

end

%Вычисление значение функции в узловых точках методом Эйлера

for i=2:n

y1(i)=y1(i-1)+ h* V1(i-1);

V1(i)=V1(i-1)+ h* y1(i-1);

y2(i)=y2(i-1)+ h* V2(i-1);

```

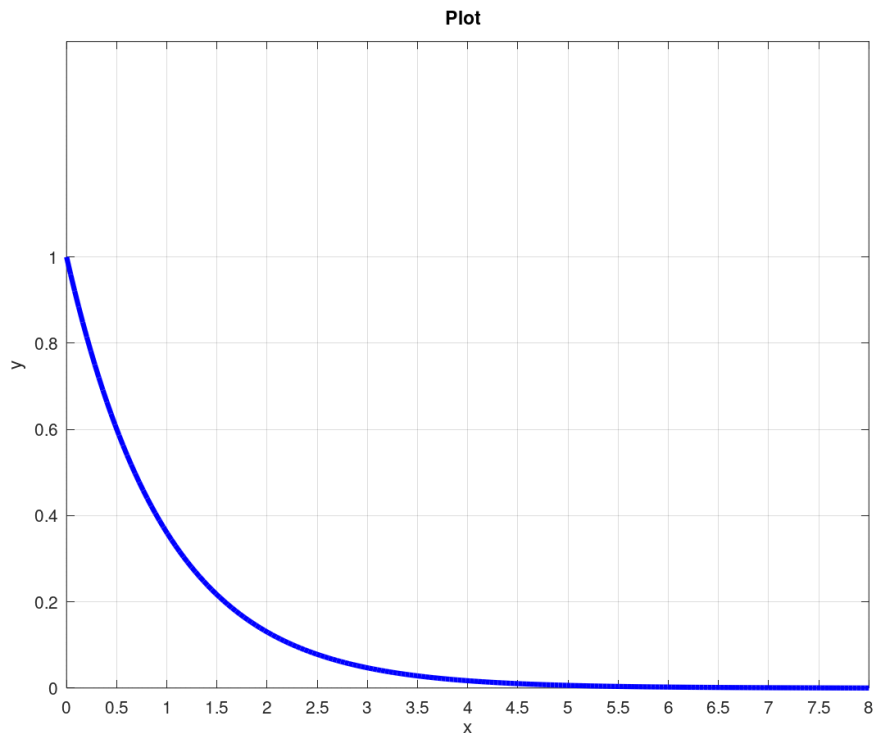
V2(i)=V2(i-1)+ h* y2(i-1);
end
M = -y1(n)/y2(n);
printf("  M= %d  \n ", M ) % Для контроля
nev = 0;
for i=1:n
y(i) = y1(i)+ M*y2(i);
ya(i) = exp(-x(i));
nev(i) = abs((y(i)-ya(i)));
end
for i = 1:5:n
    % Печать таблицы результатов
    printf(" i = %i  \t y1 = %f \t y2= %f \t y = %e\t nev= %e \n",
        i, y1(i),y2(i),y(i), nev(i));
endfor

```

```

okno1=figure();
cla;
pol=plot(x,y);
set(pol,'LineWidth',3,'Color','b')
set(gca,'xlim',[0,x(n)]);
set(gca,'ylim',[0,1.5]);
set(gca,'xtick',[0:0.5:x(n)]);
set(gca,'ytick',[0:0.2:1]);
grid on; xlabel('x'); ylabel('y');
title('Plot ');

```



Задача № 2

Деформация сдвига ψ трехслойной балки, равномерно нагруженной по всей длине, описывается уравнением

$$\frac{d^3\psi}{dx^3} - K^2 \frac{d\psi}{dx} + \alpha = 0, \quad (2.1)$$

где K^2 и α – физические параметры, зависящие от упругих свойств слоев.

Рассчитать деформацию для защемленной балки

$$d\psi(0)/dx = d\psi(1)/dx = 0 \quad (2.2)$$

при $\alpha = 1, K = 5$.

Из условий симметрии $\psi\left(\frac{1}{2}\right) = 0$.

Решение

При решении краевой задачи для линейного дифференциального уравнения используем метод суперпозиции, то есть, будем искать решение в виде (вводим три вспомогательные функции):

$$\psi(x) = \psi_1(x) + \lambda\psi_2(x) + \mu\psi_3(x). \quad (2.3)$$

Вводим три функции, так как в начальной точке $x = 0$ в соответствии с (2.2) задано только одно условие; для решения задачи Коши для уравнения третьего порядка необходимо три условия.

Подставляя данное представление (2.3) - $\psi(x)$ в граничное условие $\frac{d\psi(0)}{dx} = 0$, приходим к условиям в точке $x=0$ для функций $\psi'_1, \psi'_2, \psi'_3$. В соответствии с идеологией метода, положим, что выполняются условия $\psi(0) = \lambda, \psi''(0) = \mu$ (недостающие условия в точке $x = 0$ заменяются неопределенными константами), где значения λ и μ пока не определены.

Исходное уравнение представляется в виде трех уравнений относительно неизвестных функций ψ_1, ψ_2, ψ_3 с граничными условиями в точке $x = 0$. Приходим к трем задачам Коши относительно вспомогательных функций. Определив значения вспомогательных функций в каждой точке области определения, можно рассчитать значения искомой функции, если, конечно, предварительно рассчитать неопределенные коэффициенты λ и μ .

Задачи Коши для функций ψ_1, ψ_2, ψ_3 :

$$\begin{aligned} \frac{d^3\psi_1}{dx^3} - K^2 \frac{d\psi_1}{dx} + \alpha &= 0, & \psi_1(0) &= 0, \\ & & \psi'_1(0) &= 0, \\ & & \psi''_1(0) &= 0, \end{aligned}$$

$$\begin{aligned} \frac{d^3\psi_2}{dx^3} - K^2 \frac{d\psi_2}{dx} &= 0, & \psi_2(0) &= 1, \\ & & \psi'_2(0) &= 0, \\ & & \psi''_2(0) &= 0, \end{aligned}$$

$$\begin{aligned} \frac{d^3\psi_3}{dx^3} - K^2 \frac{d\psi_3}{dx} &= 0, & \psi_3(0) &= 0, \\ & & \psi'_3(0) &= 0, \\ & & \psi''_3(0) &= 1. \end{aligned}$$

Для определения констант λ и μ воспользуемся двумя граничными условиями $d\psi(1)/dx = 0$, $\psi\left(\frac{1}{2}\right) = 0$ которые не были ранее использованы.

Для определения констант λ и μ имеем систему

$$\begin{cases} \psi_1(1) + \lambda\psi_2(1) + \mu\psi_3(1) = 0, \\ \psi_1(0,5) + \lambda\psi_2(0,5) + \mu\psi_3(0,5) = 0. \end{cases}$$

откуда

$$\lambda = \frac{\begin{vmatrix} -\psi_3'(1) & \psi_3'(1) \\ -\psi_1(0,5) & \psi_3(0,5) \end{vmatrix}}{\Delta}, \quad \mu = \frac{\begin{vmatrix} \psi_2'(1) & -\psi_1'(1) \\ \psi_2(0,5) & -\psi_1(0,5) \end{vmatrix}}{\Delta},$$

$$\Delta = \psi_2(1)\psi_3(0,5) - \psi_2(0,5)\psi_3(1).$$

Решение (Питон, Шакиров)

```
x = []
xstart = 0
xfinish = 1
n = 100
h = (xfinish - xstart) / n
a = 1
k = 5

y = [0 for _ in range(n+1)]
Va = 0
Vb = 0
yc = 0
y1 = [0]
y2 = [1]
y3 = [0]
V2 = [0]
V1 = [0]
V3 = [0]
W1 = [0]
W2 = [0]
W3 = [1]

x = [0 for _ in range(n+1)]
i=0
while i < (n) :
    x[i+1] = x[i]+h
    print (x[i])
    i+=1

for i in range(n+1):
    y1.append(y1[-1] + h * V1[-1])
    V1.append(V1[-1] + h * W1[-1])
    W1.append(W1[-1] + h * (k * k * V1[-1] - a))

    y2.append(y2[-1] + h * V2[-1])
    V2.append(V2[-1] + h * W2[-1])
    W2.append(W2[-1] + h * (k * k * V2[-1] - a))
```

```

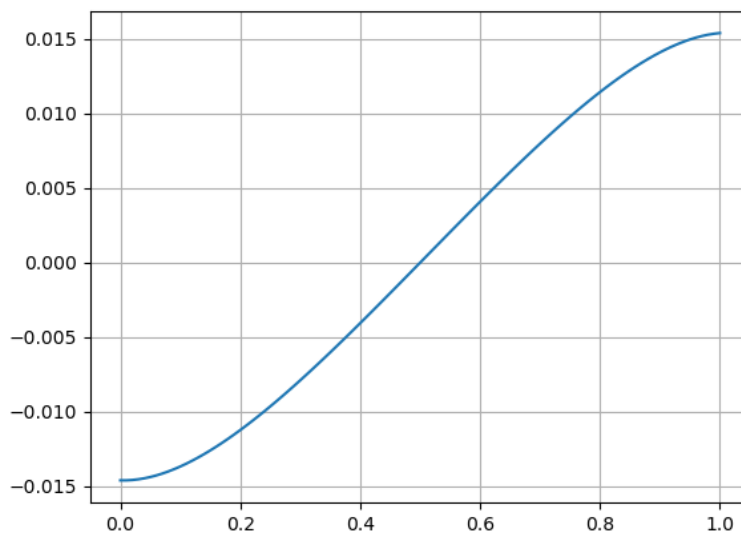
y3.append(y3[-1] + h * V3[-1])
V3.append(V3[-1] + h * W3[-1])
W3.append(W3[-1] + h * (k * k * V3[-1] - a))

det = V2[-1] * y3[n//2] - y2[n//2] * V3[-1]
l = (-V1[-1] * y3[n//2] + y1[n//2] * V3[-1]) / det
m = (-V2[-1] * y1[n//2] + y2[n//2] * V1[-1]) / det
print('lambda=', l, 'm=', m)

for i in range(n + 1):
    y[i] = y1[i] + l * y2[i] + m * y3[i]
    print(y[i])

import matplotlib.pyplot as plt
plt.plot(x,y)
plt.grid(True)
plt.show()

```



Для установки метода рисования графика в Python необходимо установить matplotlib.pyplot. Например,
 pip3 --proxy=<http://99k4:a99@cproxy.udsu.ru:8080> install matplotlib

Решение (Octave)

```

% Метод суперпозиции
% Вводим функции: y1, y2, y3; Производные: V1, V2, V3;
%           Вторые производные: W1, W2, W3
% Граничные условия для вспомогательных функций
% Первое уравнение:
    y1(0) = 0; V1(0) = 0; W1(0) = 0;
% Второе уравнение:

```

```

        y2(0) = 1; V2(0) = 0; W2(0) = 0;
% Третье уравнение:
        y3(0) = 0; V3(0) = 0; W3(0) = 1;
    xstart = 0; xfinis = 1;
    n=200;
    h=(xfinis-xstart)/(n-1); %Вычисление шага

% Формирование системы равноотстоящих узлов xі
for i=1:n
    x(i)=xstart+(i-1)*h;
end
%Вычисление значение функции в узловых точках методом Эйлера
for i=2:n
    y1(i)=y1(i-1)+ h* V1(i-1);
    V1(i)=V1(i-1)+ h* y1(i-1);
    y2(i)=y2(i-1)+ h* V2(i-1);
    V2(i)=V2(i-1)+ h* y2(i-1);
end
M = -y1(n)/y2(n);
printf(" M= %d \n ", M ) % Для контроля
nev = 0;
for i=1:n
    y(i) = y1(i)+ M*y2(i);
    ya(i) = exp(-x(i));
    nev(i) = abs((y(i)-ya(i)));
end
for i = 1:5:n
    % Печать таблицы результатов
    printf(" i = %i \t y1 = %f \t y2= %f \t y = %e\t nev= %e \n",
        i, y1(i),y2(i),y(i), nev(i));
endfor

okno1=figure();
cla;
pol=plot(x,y);
set(pol,'LineWidth',3,'Color','b')
set(gca,'xlim',[0,x(n)]);
set(gca,'ylim',[0,1.5]);
set(gca,'xtick',[0:0.5:x(n)]);
set(gca,'ytick',[0:0.2:1]);
grid on; xlabel('x'); ylabel('y');
title('Plot ');

```

Решение иллюстрируется программой:

```

// https://www.programiz.com/cpp-programming/online-compiler/
#include <iostream>
using namespace std;
int main()
    const int n = 501;

```

```

const int nprint = 25;
double y[n], y1[n], Y1[n], YY1[n], y2[n], Y2[n], YY2[n], y3[n], Y3[n], YY3[n];
double alfa = 1, K = 5, xstart = 0, xfinis = 1, lambda, m, delta;
double h = (xfinis - xstart)/(n-1);
double K2=K*K;
y1[0] = 0; Y1[0] = 0; YY1[0] = 0;
y2[0] = 1; Y2[0] = 0; YY2[0] = 0;
y3[0] = 0; Y3[0] = 0; YY3[0] = 1;
// Расчет вспомогательных сеточных функций
for (int i = 0; i < n; i++) {
    y1[i+1] = y1[i] + h*Y1[i];
    Y1[i+1] = Y1[i] + h*YY1[i];
    YY1[i+1] = YY1[i] + h*(K2*Y1[i]-alfa);

    y2[i+1] = y2[i] + h*Y2[i];
    Y2[i+1] = Y2[i] + h*YY2[i];
    YY2[i+1] = YY2[i] + h*(K2*Y2[i]);

    y3[i+1] = y3[i] + h*Y3[i];
    Y3[i+1] = Y3[i] + h*YY3[i];
    YY3[i+1] = YY3[i] + h*(K2*Y3[i]); }
// Определение введенных констант
int ns = n/2;
delta = Y2[n]*y3[ns] - y2[ns]*Y3[n];
lambda = (-Y1[n]*y3[ns]+y1[ns]*Y3[n])/delta;
m = (-Y2[n]*y1[ns]+y2[ns]*Y1[n])/delta;
cout<<" lambda = "<< lambda<< " m = "<< m <<" delta = "<< delta<<" ns= "
<< ns<< endl;

for (int i = 0; i <= n; i++) {
    y[i] = y1[i] + lambda*y2[i] + m*y3[i];
    if ( i%nprint == 0) { cout<<"y["<<i<<"] = "<< y[i]<< endl; }
//    cout<<"y["<<i<<"] = "<< y[i] <<" y1 = "<< y1[i] << " y2 = "<< y2[i] <<
//        " y3 = "<< y3[i] << endl; }
}
return 0;
}

```

Контрольные точки:

```

lambda = -0.0120987 m = 0.197282 delta = -14.6214 ns= 250
y[0] = -0.0120987
y[25] = -0.0118794
y[50] = -0.0112723
y[75] = -0.0103655
y[100] = -0.00922868
y[125] = -0.00791638
y[150] = -0.00647225
y[175] = -0.00493148
y[200] = -0.00332315
y[225] = -0.00167199
y[250] = 0

```

$y[275] = 0.00167218$
 $y[300] = 0.00332398$
 $y[325] = 0.0049336$
 $y[350] = 0.00647665$
 $y[375] = 0.00792463$
 $y[400] = 0.00924313$
 $y[425] = 0.0103897$
 $y[450] = 0.0113112$
 $y[475] = 0.0119406$
 $y[500] = 0.0121926$

Метод пристрелки

Задача № 3 (Метод Ньютона)

При исследовании распределения температур в течении несжимаемой ньютоновской жидкости между неподвижным внутренним и вращающимся с постоянно скоростью внешним коаксиальными цилиндрами было установлено, что безразмерная температура удовлетворяет уравнению

$$(1/\xi)d(\xi d\theta/d\xi)/d\xi + 4N/\xi^4 = 0$$

и граничным условиям $\theta(x) = 0$, $\theta(1) = 1$.

Найти распределение температуры при $X = 0.9$, $N = 0.4$ и сравнить с точным решением

$$\theta = N + 1 - N/\xi^2 - [N + 1 - N/x^2] \ln \xi / \ln x.$$

Решение

Для решения используем один из вариантов метода пристрелки – метод Ньютона. Система уравнений, определяющая искомую функцию θ , имеет вид:

$$\begin{aligned} \frac{d\theta}{d\xi} &= U, & \theta(x) &= 0, \\ \frac{1}{\xi} \frac{d}{d\xi} (\xi U) &= -\frac{4N}{\xi^4}, & U(x) &= S. \end{aligned}$$

Задачи Коши для вспомогательных функций $V = \frac{\partial \theta}{\partial S}$, $W = \frac{\partial U}{\partial S}$:

$$\frac{dV}{d\xi} = W, \quad V(x) = 0,$$

$$\frac{dW}{d\xi} = -\frac{1}{\xi} W, \quad W(x) = 1.$$

Итерационный процесс для параметра S определяется формулой

$$S^{(K+1)} = S^{(K)} - (\theta(1) - 1)/V(1).$$

Задача № 4

Газ А диффундирует в жидкость, где происходит необратимая реакция первого порядка $A + B \rightarrow AB$. Концентрация газа А может быть найдена как решение дифференциального уравнения

$$-D \frac{d^2 C_A}{dz^2} + K C_A = 0$$

с граничными условиями

$$C_A(0) = C_A^0, \quad dC_A(L)/dZ = 0.$$

Рассчитать распределение концентрации C_A для $M = KL^2/D = 10^{-2}$ и $M = 10^{-3}$.

Решение

В безразмерных переменных $C_A/C_A^0 \rightarrow C, Z/L \rightarrow Z$ дифференциальное уравнение имеет вид:

$$\frac{d^2 C}{dZ^2} - \left(\frac{KL^2}{D} \right) C = 0,$$

граничные условия:

$$C(0) = 1, \quad \frac{dC(1)}{dZ} = 0.$$

Для решения краевой задачи воспользуемся методом пристрелки. В данном случае нелинейное уравнение, определяющее итерационный параметр $S = \frac{\partial C(0)}{\partial Z}$, имеет вид

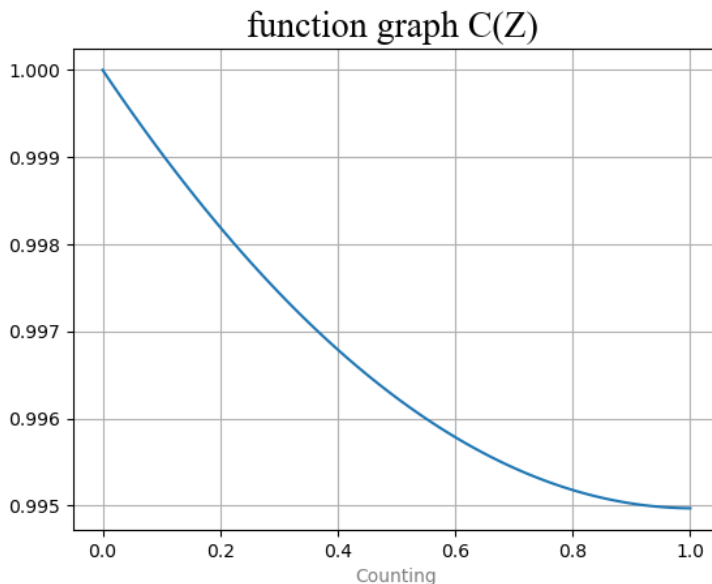
$$\varphi(S) = \frac{\partial C}{\partial Z}(1) - 0 = U(1).$$

Итерационный процесс строится:

$$S^{(n+1)} = S^{(n)} - \frac{U(1, S^{(n)})}{\partial U(1, S^{(n)}) / \partial S}.$$

Результаты расчета при $KL^2/D = 0,01$ приведены в виде графика. Точное решение имеет вид:

$$C = ch[\sqrt{KL^2/D}(1 - Z)] / ch\sqrt{KL^2/D}.$$



Программа (Python) печати графика точного решения

$$C = ch[\sqrt{KL^2/D}(1 - Z)] / ch\sqrt{KL^2/D}.$$

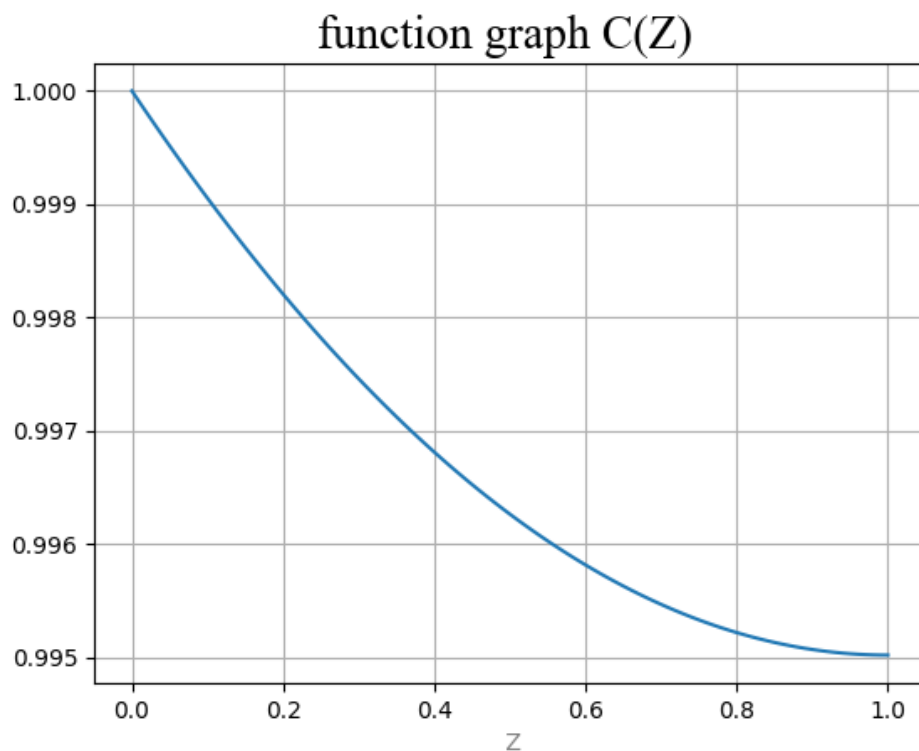
```
import math
import numpy as np
import matplotlib.pyplot as plt
```

```

def funct_ch(Z, M):
    def ch(x):
        return (math.exp(x) + math.exp(-x))*0.5
    s1 = math.sqrt(M)
    return ch(s1*(1-Z))/ch(s1)

n = 100    # Число отрезков по пространству
M = 10 ** -2
Z = np.linspace(0, 1, n+1)
C = np.zeros(n+1)
C[0] = 1
for i in range(n+1):
    C[i] = (funct_ch(Z[i], M))
plt.title('function graph C(Z) ', fontsize=20, fontname='Times New Roman')
plt.xlabel(' Z', color='gray')
plt.plot(Z,C)
plt.grid(True)
plt.show()

```



Задача № 5

Решить краевую задачу

$$\ddot{y} = 2x^2 + 3y^3 + 4xy,$$

$$y(0) = 0, \quad y(1) = 1.$$

Для решения данной краевой задачи используется метод Ньютона.

Вводим параметр S , определив для него начальное значение, например, $S = 0.1$. По смыслу параметр S – это производная искомой функции y в точке $x = 0$, то есть $S = \frac{dy}{dx}(x = 0)$. Система уравнений, определяющая функцию y , имеет вид:

$$\begin{aligned}\frac{dy}{dx} &= U, & y(0) &= 0, \\ \frac{d}{dx}(U) &= 2x^2 + 3y^3 + 4xy, & U(0) &= S.\end{aligned}$$

Дифференцируя два уравнения и условия в точке $x = 0$ по независимой переменной S и, изменяя порядок дифференцирования, приходим к задаче Коши для вспомогательных функций $V = \frac{\partial y}{\partial S}$, $W = \frac{\partial U}{\partial S}$:

$$\begin{aligned}\frac{dV}{dx} &= W, & V(0) &= 0, \\ \frac{dW}{dx} &= (9y^2 + 4x) * V, & W(x) &= 1.\end{aligned}$$

Решая четыре задачи Коши на интервале $x \in [0, 1]$, находим значения неизвестных функций на всем интервале x . В том числе, и в точке $x=1$: $y(S, x=1)$, $U(S, x=1)$, $V(S, x=1)$, $W(S, x=1)$. Все найденные значения зависят от параметра S .

Учитывая граничное условие $y(1) = 1$, запишем функцию $\varphi(S)$ в виде

$$\varphi(S) = y(S, x = 1) - 1 = 0$$

Полученное уравнение относительно параметра S , если известно начальное приближение, можно решить методом Ньютона. Метод Ньютона рассмотрен в задаче №3. В данном случае, имеем:

$$\varphi'(S_i) = \frac{\varphi(S_i, x=1)}{S_i - S_{i+1}}. \text{ Следовательно, } S_{i+1} = S_i - \frac{\varphi(S_i, x=1)}{\varphi'(S_i, x=1)} = \frac{y(S_i, x=1) - 1}{\frac{dy}{dx}(x=1)} = \frac{y(S_i, x=1) - 1}{V(x=1)}$$

Таким образом, итерационный процесс для определения параметра S имеет вид:

$$S_{i+1} = S_i - \frac{\varphi(S_i, x = 1)}{\varphi'(S_i, x = 1)} = \frac{y(1) - 1}{V(1)}$$

Программа для решения на Python (Шакиров А.Д.)

```
import math
import numpy as np
import matplotlib.pyplot as plt
def funct(x, y):
    return 2*x*x + 3*y*y*y + 4*x*y
def prfunct(x, y):
    return 9*y*y + 4*x
n = 100          # Число отрезков по пространству
S = 0.6          # Начальное приближение
h = 1 / n        # Шаг по пространству
eps = 10 ** -6   # Точность вычислений
iter = 0          # Число итераций при решении
x = np.linspace(0, 1, n+1)
condition = False
while not condition and iter < 10:
    iter += 1
    y = np.zeros(n+1)
    y[0] = 0
```

```

V = np.zeros(n+1)
V[0] = 0
W = np.zeros(n+1)
W[0] = 1
U = np.zeros(n+1)
U[0] = S
for i in range(n):
    y[i+1] = y[i] + h * U[i]
    U[i+1] = U[i] + h * funct(x[i], y[i])
    V[i+1] = V[i] + h * W[i]
    W[i+1] = W[i] + h * prfunct(x[i], y[i])*V[i]
Snew = S - (y[n]-1)/V[n]
if abs(y[n] - 1) > eps:
    print("S=",S,"Snew=",Snew,"y=",y[n],"V=",V[n])
    S = Snew
else:
    condition = True
plt.title('function graph y(x) ', fontsize=20, fontname='Times New Roman')
plt.xlabel('Counting', color='gray')
plt.plot(x,y)
plt.grid(True)
plt.show()

```

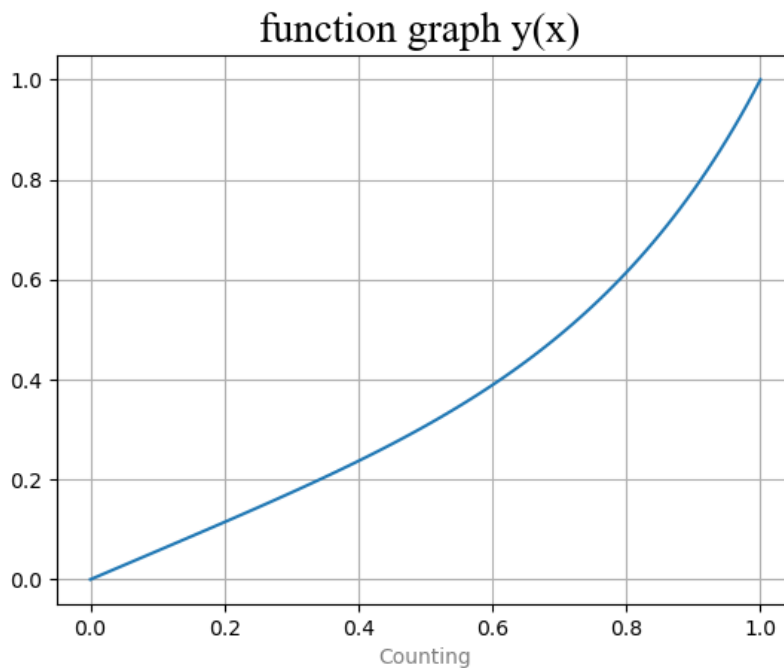
Отладочная печать параметра S и значения функции на конце интервала

S= 0.6 Snew= 0.5713799545982841 y= 1.0471809157051306 V= 1.6485269342829856

S= 0.5713799545982841 Snew= 0.5711311292753489 y= 1.0004032546700365 V= 1.6206335644610987

S= 0.5711311292753489 Snew= 0.5711311112288087 y= 1.0000000292425875 V= 1.6203985405514916

График искомой функции (Шакиров А.Д.)



Задача № 6 (Метод прогонки)

Бесконечная пластина помещена в жидкость с температурой T_∞ . В пластине выделяется тепло с постоянной скоростью q_s . Определить распределение температур в пластине.

$$\frac{\partial^2 T}{\partial x^2} + \frac{q_s}{\lambda} = 0, \quad \frac{\partial T(0)}{\partial x} = 0, \quad -\lambda \frac{\partial T(L)}{\partial x} = h[T(L) - T_\infty].$$

где λ – коэффициент теплопроводности

h – коэффициент конвективной теплопередачи

L – половина толщины пластины

В безразмерных переменных $\frac{x}{L} \rightarrow X$

$\theta = (T - T_\infty)/[q_s L^2 / \lambda]$ задача формулируется

$$\frac{\partial \theta(0)}{\partial x} = 0, \quad \frac{\partial^2 \theta}{\partial x^2} + 1 = 0, \quad -\frac{\partial \theta(1)}{\partial x} = Bi * \theta(1),$$

Bi – критерий Био.

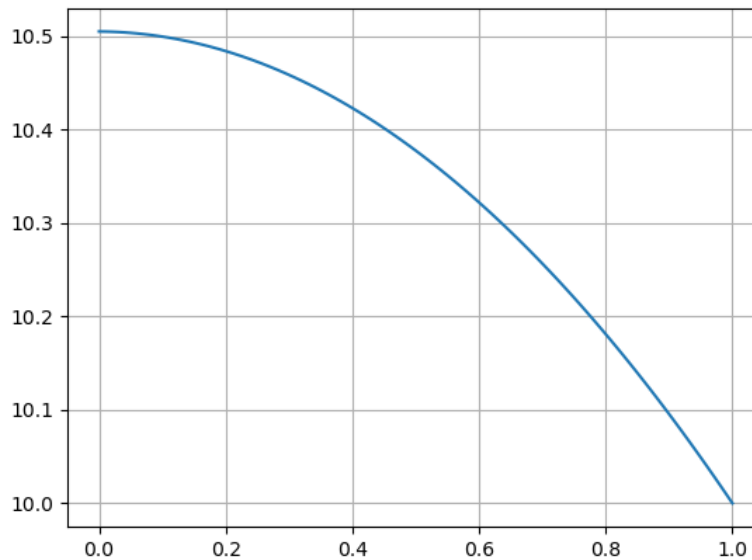
Метод прогонки изложен в книге Самарский А.А. «Численные методы»

Решение (python)

```
import math
import numpy as np
import matplotlib.pyplot as plt
n = 100
Bi = 0.1
h = 1/n
alfa = np.zeros(n+1)
betta = np.zeros(n+1)
Q = np.zeros(n+1)
x = np.linspace(0, 1, n+1)
ksi1 = 1
ksi2 = 1/(1 + h*Bi)
m1 = 0
m2 = 0
alfa[0] = ksi1
betta[0] = m1
for i in range(n):
    alfa[i+1] = 1 / (2 - alfa[i]*1)
    betta[i+1] = (h*h + 1 *betta[i]) / (2 - alfa[i]*1)

Q[n] = ksi2 * betta[n] / (1 - ksi2*alfa[n])

for i in reversed(range(n)):
    Q[i] = alfa[i+1]*Q[i+1] +betta[i+1]
plt.plot(x,Q)
plt.grid(True)
plt.show()
```



```
% Разностная задача  $A_i y_{(i-1)} - C_i y_i + B_i y_{(i+1)} = -F_i$ 
%  $y_0 = \text{ksi}_1 y_1 + \text{nu}_1$ ,  $y_n = \text{ksi}_2 y_{(n-1)} + \text{nu}_2$ 
% Решение
%  $\text{alfa}_{(i+1)} = B_i / (C_i - \text{alfa}_i A_i)$ 
%  $\text{betta}_{(i+1)} = (A_i \text{betta}_i + F_i) / (C_i - \text{alfa}_i A_i)$ 
%  $\text{alfa}_1 = \text{ksi}_1$ ,  $\text{betta}_1 = \text{nu}_1$ 
%  $y_i = \text{alfa}_{(i+1)} y_{(i+1)} + \text{betta}_{(i+1)}$ 
%  $y_n = (\text{nu}_2 + \text{ksi}_2 * \text{betta}_n) / (1 - \text{ksi}_2 * \text{alfa}_n)$ 
% В данном случае  $A_i = 1$ ,  $B_i = 1$ ,  $C_i = 2$ ,  $F_i = h * h$ ;
%  $\text{ksi}_1 = 1$ ,  $\text{nu}_1 = 0$ ,  $\text{ksi}_2 = 1 / (1 + B_i * h)$ ,  $\text{nu}_2 = 0$ .
```

```
% Метод прогонки решения сеточных уравнений 220412
n = 100; % дискретность области по оси x от 0 до 1
h = (1 - 0) / (n - 1);
Bi = 0.1; % критерий Bi
Ai = 1; Bi = 1; Ci = 2; Fi = h * h;
ksi_1 = 1; nu_1 = 0;
ksi_2 = 1 / (1 + Bi * h); nu_2 = 0;
% прямая прогонка
alfa(1) = ksi_1; betta(1) = nu_1;
for i = 1:n
    alfa(i+1) = Bi / (Ci - alfa(i) * Ai);
    betta(i+1) = (Ai * betta(i) + Fi) / (Ci - alfa(i) * Ai);
endfor
% обратная прогонка
Q(n+1) = (nu_2 + ksi_2 * betta(n+1)) / (1 - ksi_2 * alfa(n+1));
for i = n:-1:1
    Q(i) = alfa(i+1) * Q(i+1) + betta(i+1)
endfor
% аналитическое решение
for i = 1:n
```

```

x(i) = (i-1)*h;
Q_a(i) = 0.5 * (1 - x(i)*x(i)) + 1/Bi;
endfor

```

```

% Построение графика функции Q(x) и Q_a(x)+1
cla;
okno1=figure();
R= Rstart:1:Rfinis;      % Создание массива переменой R
dP=expl(R,Qg, M, k)*1000; % Создание массива переменой dP,кПа
pol=plot(R,dP);          % Создание графика
set(pol,'LineWidth',3,'Color','k'); % Установка толщины и цвета линии
set(gca,'xlim',[Rstart,Rfinis]); % Установка осей
set(gca,'ylim',[0,100]);
set(gca,'xtick',[Rstart:10:Rfinis]); % Вывод сетки с заданным диапазоном
set(gca,'ytick',[0:10:1000]);
grid on;                  % Установка сетки
xlabel('R');ylabel('dP,кПа'); % Подписи осей
title('График dP(R)');    % Подпись графика

```

```

okno1=figure();
cla;
plot(t, kr_m, t, li_m);
set(gca,'xlim',[0,40]);
set(gca,'ylim',[0,600]);
set(gca,'xtick',[0:5:40]);
set(gca,'ytick',[0:50:600]);

grid on;
xlabel('t');ylabel('kr_m , li_m');

```

Для решения применим метод прогонки. Задачи Коши для вспомогательных функций:

$$\frac{d\alpha_0}{dx} + \alpha_0^2(x) = 0, \quad \alpha_0(0) = 0,$$

$$\frac{d\alpha_1}{dx} + \alpha_0(x)\alpha_1(x) = -1, \quad \alpha_1(0) = 0.$$

Значение температуры в точке $x = 1$

$$\theta(1) = -\alpha_1(1) / [\alpha_0(1) + Bi].$$

является граничным условием для уравнения

$$d\theta/dx = \alpha_0(x) * \theta + \alpha_1(x),$$

решая которое получаем требуемое распределение температуры.

В распечатке результата для иллюстрации точности метода привести сравнение численного решения с аналитическим, имеющим вид:

$$\theta = 0,5(1 - x^2) + 1/Bi.$$

Результаты иллюстрации, также зависимость точности решения от узлов разностной сетки $N = 20; 50; 100$.

Задача № 7

В модели каталитического конвертера распределение молярной концентрации газа C описывается краевой задачей:

$$\frac{d^2 c}{dz^2} + \frac{1}{2} * \frac{1}{1 - c/2} \left(\frac{dc}{dz} \right)^2 = 0,$$

$$c(z = 0) = c_0, \quad c(z = \delta) = 0.$$

Рассчитать профиль молярной концентрации газа в модели каталитического конвертера, если концентрация газа в основном потоке $C_0 = 0,25$.

Решение

Если ввести новую переменную $x = z/\delta$, то уравнение станет безразмерным и получается следующая граничная задача для уравнения второго порядка

$$\frac{d^2 c}{dx^2} + \frac{1}{2} \left[\frac{1}{1 - c/2} \right] \left(\frac{dc}{dx} \right)^2 = 0,$$

$$c(x = 0) = c_0, \quad c(x = 1) = 0.$$

Для решения воспользуемся методом параллельной пристрелки, разбив интервал $x \in [0; 1]$ на два подынтервала $[0; 0,5]$ и $[0,5; 1]$. Введем разностную сетку ($i = 1, 2, \dots, N$) с шагом $H = 1/(N - 1)$.

В каждом из подынтервалов решаем задачу Коши:

В первом подынтервале $x \in [0; 0,5]$:

$$\frac{dc_1}{dx} = U_1, \quad \frac{dU_1}{dx} = -U_1^2 \frac{1}{2 - c_1},$$

$$c_1(0) = C_0, \quad U_1(0) = S_1. \quad (1)$$

Во втором подынтервале $x \in [0,5; 1]$:

$$\frac{dc_2}{dx} = U_2, \quad \frac{dU_2}{dx} = -U_2^2 \frac{1}{2 - c_2},$$

$$c_2(0,5) = S_2, \quad U_2(0,5) = S_3. \quad (2)$$

где S_1, S_2, S_3 – параметры, подлежащие определению.

В качестве начального приближения для параметров S_1, S_2, S_3 положим:

$$S_1 = -\frac{c_0}{N}, \quad S_2 = \frac{c_0}{2}, \quad S_3 = -\frac{c_0}{N}.$$

Выражения для невязок имеют вид:

$$\begin{aligned} \varphi_1(S_1, S_2) &= c_1(0,5, S_1) - S_2 \rightarrow 0, \\ \varphi_2(S_1, S_3) &= U_1(0,5, S_1) - S_3 \rightarrow 0, \\ \varphi_3(S_2, S_3) &= c_2(1, S_2, S_3) - 0 \rightarrow 0. \end{aligned} \quad (3)$$

Последнее выражение следует из граничного условия при $x=1$.

Разложим выражения для невязок в ряд Тейлора, относительно параметров S_1, S_2, S_3 .

В частности, для первого уравнения невязок (3) имеем:

$$\varphi_1(S_1^{n+1}, S_2^{n+1}) = \varphi_1(S_1^n, S_2^n) + \frac{\partial \varphi_1}{\partial S_1} * (S_1^{n+1} - S_1^n) + \frac{\partial \varphi_1}{\partial S_2} * (S_2^{n+1} - S_2^n) = 0.$$

Аналогично, для $\varphi_2(S_1, S_3)$ и $\varphi_3(S_2, S_3)$. Следующее приближение для искомых параметров: $\bar{S}_1 = S_1^{n+1} - S_1^n$, $\bar{S}_2 = S_2^{n+1} - S_2^n$, $\bar{S}_3 = S_3^{n+1} - S_3^n$ дается матричным уравнением

$$\begin{bmatrix} \bar{S}_1 \\ \bar{S}_2 \\ \bar{S}_3 \end{bmatrix} = A^{-1} \begin{bmatrix} S_2 - c_1(0,5) \\ S_3 - U_1(0,5) \\ -c_2 \end{bmatrix},$$

где, $A = \begin{bmatrix} \frac{\partial c_1(0,5)}{\partial S_1} & -1 & 0 \\ \frac{\partial U_1(0,5)}{\partial S_1} & 0 & -1 \\ 0 & \frac{\partial c_2(1)}{\partial S_2} & \frac{\partial c_2(1)}{\partial S_3} \end{bmatrix}$ (4)

Введем вспомогательные функции

$$\begin{aligned} V_1 &= \frac{\partial c_1}{\partial S_1}, & V_2 &= \frac{\partial c_2}{\partial S_2}, & V_3 &= \frac{\partial c_2}{\partial S_3}, \\ W_1 &= \frac{\partial U_1}{\partial S_1}, & W_2 &= \frac{\partial U_2}{\partial S_2}, & W_3 &= \frac{\partial U_2}{\partial S_3}. \end{aligned}$$

Дифференцируя (1) по S_1 , получим систему уравнений:

$$\frac{dV_1}{dx} = W_1, \quad \frac{dW_1}{dx} = -\frac{U_1^2 V_1}{(2 - c_1)^2} - \frac{1}{2 - c_1} 2U_1 W_1$$

с граничными условиями:

$$V_1(0) = 0, \quad W_1(0) = 1.$$

Дифференцируя (2) по S_2 , приходим к системе:

$$\frac{dV_2}{dx} = W_2, \quad \frac{dW_2}{dx} = -\frac{U_2^2 V_2}{(2 - c_2)^2} - \frac{1}{2 - c_2} 2U_2 W_2$$

с граничными условиями:

$$V_2(0,5) = 1, \quad W_2(0,5) = 0.$$

После дифференцирования (2) по S_3 имеем систему уравнений:

$$\frac{dV_3}{dx} = W_3, \quad \frac{dW_3}{dx} = -\frac{U_2^2 V_3}{(2 - c_2)^2} - \frac{2 * U_2}{2 - c_2} W_3,$$

граничные условия для которой –

$$V_3(0,5) = 0, W_3(0,5) = 1.$$

Уравнения для V_1 и W_1 интегрируются от $x = 0$ до $x = 0,5$, а уравнения для V_2 и W_2 , а также V_3 и W_3 – в интервале от $x = 0,5$ до $x = 1$.

Матрица в новых обозначениях принимает вид:

$$A = \begin{bmatrix} V_1(0,5) & -1 & 0 \\ W_1(0,5) & 0 & -1 \\ 0 & V_2(1) & V_3(1) \end{bmatrix},$$

где все элементы определяются из соответствующих задач Коши для вспомогательных функций.

Обратная матрица A^{-1} :

$$A^{-1} = \frac{1}{\Delta} \begin{bmatrix} V_2(1) & V_3(1) & 1 \\ -W_1(0,5)V_3(1) & V_1(0,5)V_3(1) & V_1(0,5) \\ W_1(0,5)V_2(1) & -V_1(0,5)V_2(1) & V_2(1) \end{bmatrix}$$

Δ – определитель матрицы A

$$\Delta = V_1(0,5) * V_2(1) + W_1(0,5) * V_3(1).$$

Таким образом, итерационный процесс строится в соответствии с уравнением

$$\begin{bmatrix} S_1^{n+1} \\ S_2^{n+1} \\ S_3^{n+1} \end{bmatrix} = \begin{bmatrix} S_1^n \\ S_2^n \\ S_3^n \end{bmatrix} + A^{-1} \begin{bmatrix} S_2^n - c_1(0,5) \\ S_3^n - U_1(0,5) \\ -c_2 \end{bmatrix}$$

Распишем для примера одно из уравнений итерационного процесса (3):

$$\bar{S}_1 = S_1 + \frac{1}{\Delta} * \{V_2(1) * [S_2 - c_1(0,5)] + V_3(1) * [S_3 - U_1(0,5)] + 1 * [-c_1(1)]\}$$

Аналогично строятся уравнения итерационного процесса для определения параметров S_2 и S_3 .

Расширением исходного уравнения являются функции $c_1(x)$, $x \in [0; 0,5]$ и $c_2(x)$, $x \in [0,5; 1]$, являющиеся решением систем (1) и (2) соответственно, где параметры S_1 , S_2 , S_3 определены вышеописанным итерационным процессом.

Программа решения (Питон)

```
import math
import numpy as np
import matplotlib.pyplot as plt
n = 100
dx = 1/(n)
c0 = 0.25
S1 = - c0/n
S2 = c0/2
S3 = - c0/n
rang_A = 3
S = np.array([[S1],[S2],[S3]])
Snew = np.array([[0],[0],[0]])
A = np.zeros((rang_A, rang_A))
B = np.zeros((rang_A, rang_A))
nn = int(n/2)
c1 = np.zeros(n + 1)
U1 = np.zeros(n + 1)
c2 = np.zeros(n + 1)
U2 = np.zeros(n + 1)
V1 = np.zeros(n + 1)
W1 = np.zeros(n + 1)
V2 = np.zeros(n + 1)
W2 = np.zeros(n + 1)
V3 = np.zeros(n + 1)
W3 = np.zeros(n + 1)
kk = 0

while kk < 5:
    kk += 1
    S = Snew
    c1[0] = c0
    U1[0] = S[0]
    c2[0] = S[1]
    U2[0] = S[2]
```

```

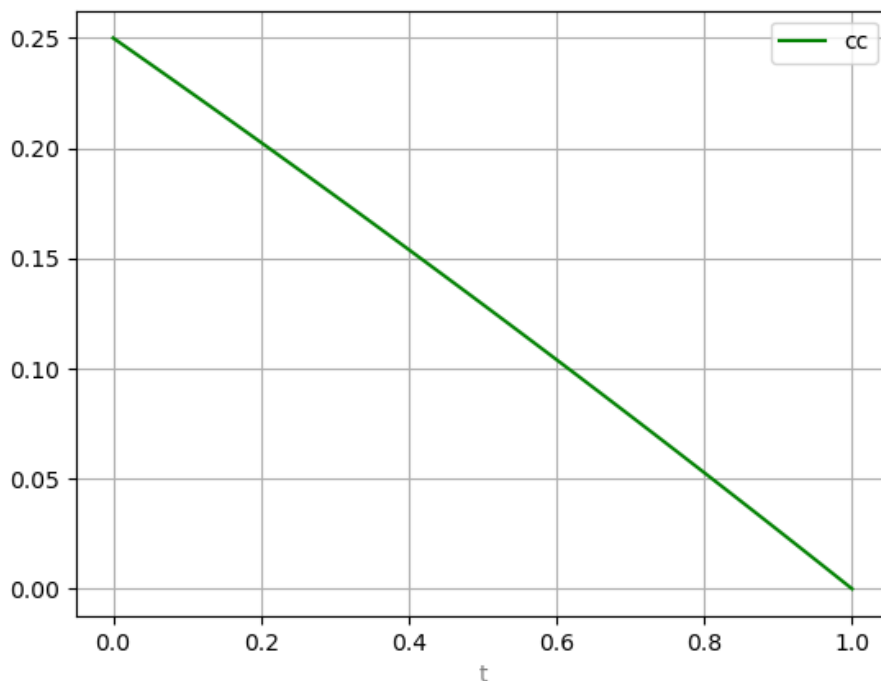
V1[0] = 0
W1[0] = 1
V2[0] = 1
W2[0] = 0
V3[0] = 0
W3[0] = 1
for i in range(nn+1):
    c1[i+1] = c1[i] + dx*U1[i]
    U1[i+1] = U1[i] - dx*(U1[i]*U1[i]/(2 - c1[i]))
    c2[i+1] = c2[i] + dx*U2[i]
    U2[i+1] = U2[i] - dx*(U2[i]*U2[i]/(2 - c2[i]))
    V1[i+1] = V1[i] + dx*W1[i]
    W1[i+1] = W1[i] - dx*(U1[i]*U1[i]*V1[i]/(2 - c1[i])**2 + 2*U1[i]*W1[i]/(2-c1[i]))
    V2[i+1] = V2[i] + dx*W2[i]
    W2[i+1] = W2[i] - dx*(U2[i]*U2[i]*V2[i]/(2 - c2[i])**2 + 2*U2[i]*W2[i]/(2-c2[i]))
    V3[i+1] = V3[i] + dx*W3[i]
    W3[i+1] = W3[i] - dx*(U2[i]*U2[i]*V3[i]/(2 - c2[i])**2 + 2*U2[i]*W3[i]/(2-c2[i]))

A[0,0] = V1[nn+1]
A[0,1] = -1
A[0,2] = 0
A[1,0] = W1[nn+1]
A[1,1] = 0
A[1,2] = -1
A[2,0] = 0
A[2,1] = V2[nn+1]
A[2,2] = V3[nn+1]

#print ("Mampуџа A: \n", A)
B = np.linalg.inv(A)
# print ("Mampуџа B: \n", B)
ss1 = float(S[1] - c1[nn+1])
ss2 = float(S[2] - U1[nn+1])
ss3 = float(-c2[nn+1])
F = np.array([[ss1], [ss2],[ss3]])
Snew = S + B.dot(F)
print (" kk ",kk )
cc = list(c1[:nn+1])+list(c2[:nn+2])
print ("Концентрация: \n",c1[:nn+1], "\n",c2[:nn+2])

x = np.linspace(0, 1, n+3)
plt.xlabel('t', color='gray')
plt.plot(x,cc,color="green", label='cc')
#plt.plot(x[:101],c1,color="red", label='c1')
#plt.plot(x[:101],c2,color="blue", label='c2')
plt.legend()
plt.grid(True)
plt.show()

```



Задача № 8

При исследовании распространения тепла в радиальном течении между параллельными круговыми дисками была получена следующая граничная задача

$$\frac{d^2 f}{d\eta^2} + \eta^2 \frac{df}{d\eta} - 3\alpha\eta f = 0, \quad f(0) = 1, f(\infty) = 0.$$

Решить для $\alpha = 0,5$ и 10 .

ЗАМЕЧАНИЕ

Алгоритм решения можно построить, используя метод суперпозиции. Решение исходной краевой задачи строится как суперпозиция двух функций: $f_1(\eta)$ и $f_2(\eta)$:

$$f(\eta) = f_1(\eta) + \mu f_2(\eta).$$

Для определения $f_1(\eta)$ и $f_2(\eta)$ имеем две задачи Коши:

$$\begin{aligned} \frac{d^2 f_1}{d\eta^2} + \eta^2 \frac{df_1}{d\eta} - 3\alpha\eta f_1 &= 0, & f_1(0) &= 1, & \frac{\partial f_1(0)}{\partial \eta} &= 0, \\ \frac{d^2 f_2}{d\eta^2} + \eta^2 \frac{df_2}{d\eta} - 3\alpha\eta f_2 &= 0, & f_2(0) &= 0, & \frac{\partial f_2(0)}{\partial \eta} &= 1. \end{aligned}$$

Константа μ определяется из второго граничного условия $\mu = -f_1(\infty)/f_2(\infty)$.

ОТВЕТ:

Недостающие граничные условия при $\alpha = 0; 5; 10$:

$f'(0) = -0,77633$; $f'(0) = -1,33778$; $f'(0) = 2,29040$ соответственно.

Задача № 9

При исследовании переноса массы на вращающемся диске в неньютоновской жидкости было найдено, что концентрация диффундирующего компонента является решением граничной задачи

$$\frac{d^2 c}{ds^2} + \frac{1}{9} \left(\frac{7+5n}{2+2n} + \frac{6}{S} \right) \frac{dc}{ds} = 0, \quad c(0) = 0, c(\infty) = c_\infty$$

Решить для $\eta = 0,2; 1,0; 1,5$ и сравнить результат с точным решением

$$\frac{c}{c_{\infty}} = \gamma \left(\frac{1}{3}, \frac{S}{3} * \frac{7+5n}{6+6n} \right) / \Gamma \left(\frac{1}{3} \right),$$

где Γ – гамма-функция, а γ – неполная гамма-функция.

Задача № 10

Рассмотрим электрический контур с последовательно соединенными элементами: индуктивностью $L(=1\Gamma)$, сопротивлением $R(=1000 \text{ Ом})$ и емкостью $C(=6,25*10^{-6}\text{Ф})$. Начальный заряд в контуре равен нулю и в момент времени $t=0$ к нему прикладывается постоянная ЭДС, равная 24В. Требуется определить величину тока в момент времени $t=0$, если известно, что при $t=0,001\text{с}$ величина тока равна 0,031А.

ЗАМЕЧАНИЕ

Ответ дается решением следующей граничной задачи:

$$\frac{d^2 Q}{dt^2} + 1000 \frac{dQ}{dt} + \frac{Q}{6 * 25 * 10^{-6}} = 24,$$

$$Q(0) = 0, i(0,001) = \frac{dQ}{dt}(0,001) = 0,031.$$

ОТВЕТ: $i(0) = dQ/dt(0) = 5\text{А}$.

Задача № 11

Стержень длиной L с постоянным по длине сечением погружен в теплоизолирующий материал. В начальный момент температура стержня $T_0 = 0$, а левый конец скачком приобретает температуру $T_c = 100 \text{ }^{\circ}\text{C}$. Правый конец стержня находится в теплоизолирующем материале. Рассчитать распределение температуры по стержню, соответствующее моменту времени $t = 5$ минут, если длина стержня $L = 0.25 \text{ м}$, коэффициент теплопроводности материала стержня $a = 10^{-5} \text{ м}^2/\text{с}$.

Пояснение.

Распределение температуры $T(x,t)$ описывается уравнением теплопроводности

$$\frac{\partial T}{\partial t} = a * \frac{\partial^2 T}{\partial x^2}$$

с краевыми условиями: $T(x,0) = T_0$, $T(0,t) = T_c$, $\partial T / \partial x (L,t) = 0$.

Разностный аналог дифференциальной задачи расчета распределения температуры для сеточной функции U , которая аппроксимирует непрерывную функцию T , имеет вид

$$\frac{U_i^{k+1} - U_i^k}{\tau} = \frac{U_{i+1}^k - 2 * U_i^k + U_{i-1}^k}{h^2}$$

Аппроксимация краевых условий приводит к выражениям:

$$\begin{aligned} U_i^0 &= T_0 & i &= 0, 1, 2, 3 \dots, n \\ U_0^k &= T_c & k &= 0, 1, 2, 3 \dots, m \\ U_n^k &= U_{n-1}^k & k &= 0, 1, 2, 3 \dots, m \end{aligned}$$

Отметим, что последнее условие отвечает первому порядку аппроксимации; уравнение аппроксимации отвечает второму порядку. Для решения можно воспользоваться явной схемой. Для перехода с временного слоя k на слой $k+1$ имеем выражение

$$U_i^{k+1} = U_i^k + \tau * \frac{U_{i+1}^k - 2 * U_i^k + U_{i-1}^k}{h^2}$$

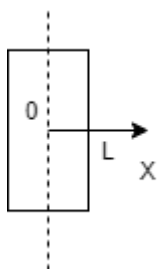
<https://www.programiz.com/cpp-programming/online-compiler/>

<https://www.programiz.com/cpp-programming/online-compiler/>

// Online C++ compiler to run C++ program online

```
#include <iostream>
using namespace std;
int main() {
    const int n=20,
        stepPrint = 20 ; // Шаг печати()
    const double tay = 1.0;
    const double L = 0.25, a = 1.0E-5;
    double T0 = 0, Tc = 100, tfinis = 5*60, h = L/(n-1), A = tay*a/(h*h) ;
    double Tk[n], Tk1[n];
    int kfinis = int ((tfinis/tay)/stepPrint);
    for(int i=0; i<n; i++) Tk[i] = Tk1[i] = T0;
    Tk[0] = Tk1[0] = 100;
    for(int kk=0; kk<kfinis; kk++) {
        for(int k=0; k<stepPrint; k++) {
            for(int i=1; i<n-1; i++) Tk1[i] = Tk[i] + A*(Tk[i+1]-2*Tk[i]+Tk[i-1]);
            Tk1[n-1] = Tk1[n-2];
            for(int i=0; i<n; i++) Tk[i] = Tk1[i] ;
        }
        for(int i=0; i<n; i++) std::cout << "    "<<Tk[i] ;
        std::cout << "    "<< endl;
        std::cout << "    "<< endl;
    }
    return 0;
}
```

Задача № 12



Боковая поверхность латунного (Cu+1.0% Zn) диска толщиной 2 см хромирована, чтобы предотвратить испарение цинка. Примем, что диффузия цинка происходит в направлении, перпендикулярном торцам диска. Площадь торца – 50 см². Коэффициент диффузии цинка при температуре отжига $D = 10^{-11}$ м²/с. Рассчитать количество цинка, испарившегося в течение 16 часов отжига. Отжиг производится в вакууме. Плотность латуни – 8900 кг/м³.

РЕШЕНИЕ

Ось x из центра диска проходит перпендикулярно торцу диска. Граница $x=L$ отделяет латунь от вакуума. На границе происходит испарение.

Концентрационное поле в диске $C(x,t)$ описывается уравнением диффузии

$$\frac{\partial C}{\partial t} = D * \frac{\partial^2 C}{\partial x^2}.$$

Начальные условия $C(x, t = 0) = C_0 = 0.01 * 8900$ кг/м³.

Граничные условия:

$$\frac{\partial C(x=0)}{\partial x} = 0; \quad C(x = L) = 0.$$

Первое граничное условие следует из симметрии концентрационного поля относительно плоскости, проходящей через центр диска параллельно его торцам. Второе условие справедливо для случая отжига в вакуум.

$$\text{Диффузный поток } J = -D \frac{\partial C}{\partial x}$$

Поток массы через торец диска

$$I(t) = - \int_0^t D \frac{\partial C}{\partial x} * S dt, \quad S \text{ площадь торца диска.}$$

В безразмерных переменных $x / L \rightarrow x$, $D * t / L^2 \rightarrow t$ задача расчета концентрационного профиля формулируется

$$\frac{\partial C}{\partial t} = \frac{\partial^2 C}{\partial x^2} \quad C(x, 0) = C_0, \quad C'(0, t) = 0, \quad C(1, t) = 0.$$

Количество цинка, испарившегося в течении времени t через торцы диска:

$$I(t) = -2 * \frac{\tau * L * S}{h} \sum_l^k (U_N^l - U_{N-1}^l), \quad k = \frac{D * t}{L^2 * \tau}$$

где τ , h – безразмерные шаги по времени и пространству;

L, S – полутолщина и площадь торца диска

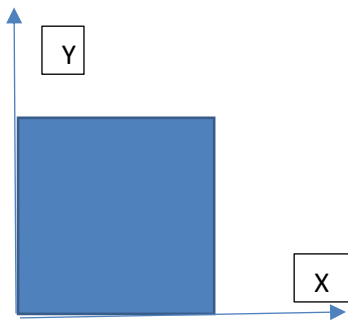
U – значение сеточной функции на l временном слое

Задача № 13

Элемент квадратного сечения (сплав на основе меди) погружается в жидкий теплоноситель из расплавленного металла с температурой $T_g = 600$ °С. Температура элемента перед погружением $T_0 = 100$ °С. Размеры сечения элемента 0.2×0.2 м. В третьем измерении размер элемента на порядок превосходит характерный размер в плоскости сечения. Коэффициент температуропроводности материала элемента $A = 10^{-5}$ м²/с. Рассчитать значение температуры в центре элемента, которое будет достигнуто к моменту времени $t = 10$ минут.

РЕШЕНИЕ

Оси X и Y расположим в плоскости сечения элемента.



Температурное поле в элементе $T(x,y,t)$ описывается уравнением теплопроводности

$$\frac{\partial T}{\partial t} = A * \left[\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \right].$$

Начальные условия $T(x, y, t = 0) = T_0 = 100.0$.

Граничное условие $T(0, y, t) = T(L, y, t) = T(x, 0, t) = T(x, L, t) = T_g = 600$.

Разностный аналог дифференциальной задачи расчета распределения температуры для сеточной функции $U_{i,j}$, которая аппроксимирует непрерывную функцию T , имеет вид

$$\frac{U_{i,j}^{k+1} - U_{i,j}^k}{\tau} = \frac{U_{i+1,j}^k - 2*U_{i,j}^k + U_{i-1,j}^k}{hx^2} + \frac{U_{i,j+1}^k - 2*U_{i,j}^k + U_{i,j-1}^k}{hy^2}$$

Аппроксимация краевых условий приводит к выражениям:

$$U_{i,j}^0 = T_0 \quad i = 0, 1, 2, 3 \dots, nx; \quad j = 0, 1, 2, 3 \dots, ny$$

$$U_g^k = T_g \quad k = 0, 1, 2, 3 \dots, m$$

Для решения можно воспользоваться явной схемой. Для перехода с временного слоя k на слой $k+1$ имеем выражение

$$U_{i,j}^{k+1} = U_{i,j}^k + \tau * A * \left[\frac{U_{i+1,j}^k - 2*U_{i,j}^k + U_{i-1,j}^k}{h^2} + \frac{U_{i,j+1}^k - 2*U_{i,j}^k + U_{i,j-1}^k}{h^2} \right]$$

Решение задачи иллюстрируется программой:

<https://www.programiz.com/cpp-programming/online-compiler/>

```
// Online C++ compiler to run C++ program online
```

```
#include <iostream>
```

```
using namespace std;
```

```
int main() {
```

```
    const int n=20,
```



```

        stepPrint = 50 ; // Шаг печати()
const double tay = 1.0;
const double L = 0.20, a = 1.0E-5;
double T0 = 100, Tg = 600, tfinis = 10*60, h = L/(n-1), A = tay*a/(h*h) ;
int iexit = (int)(n/2.0); int yexit = iexit;
double time = 0;
double Tk[n][n], Tk1[n][n];
int kfinis = int ((tfinis/tay)/stepPrint);
for(int i=0; i<n; i++) // Начальное условие по температуре
    for(int j=0; j<n; j++)
        Tk[i][j] = Tk1[i][j] = T0;
for(int i=0; i<n; i++)
    Tk[i][0] = Tk1[i][0] = Tk[i][n-1] = Tk1[i][n-1]=Tg;
for(int j=0; j<n; j++)
    Tk[0][j] = Tk1[0][j] = Tk[n-1][j] = Tk1[n-1][j]=Tg;

for(int kk=0; kk<kfinis; kk++) {
    for(int k=0; k<stepPrint; k++) {
        for(int i=1; i<n-1; i++)
            for(int j=1; j<n-1; j++)
                Tk1[i][j] = Tk[i][j] + A*(Tk[i+1][j]-2*Tk[i][j]+Tk[i-1][j]+
                Tk[i][j+1]-2*Tk[i][j]+Tk[i][j-1]);
        for(int i=0; i<n; i++)
            for(int j=0; j<n; j++)
                Tk[i][j] = Tk1[i][j] ;
        time+=tay;
    }
    std::cout << " time,c= " << time << " T,град=" << Tk[iexit][yexit] << endl;
}
return 0;
}

```

Результат работы программы отражает динамику изменения температуры в центре элемента T,град в зависимости от времени процесса нагрева time,c

```

time,c = 50 T,град=104.455
time,c= 100 T,град=153.306
time,c= 150 T,град=230.567
time,c= 200 T,град=305.152
time,c= 250 T,град=367.794
time,c= 300 T,град=418.034
time,c= 350 T,град=457.669
time,c= 400 T,град=488.749
time,c= 450 T,град=513.065
time,c= 500 T,град=532.073
time,c= 550 T,град=546.927
time,c= 600 T,град=558.533

```

