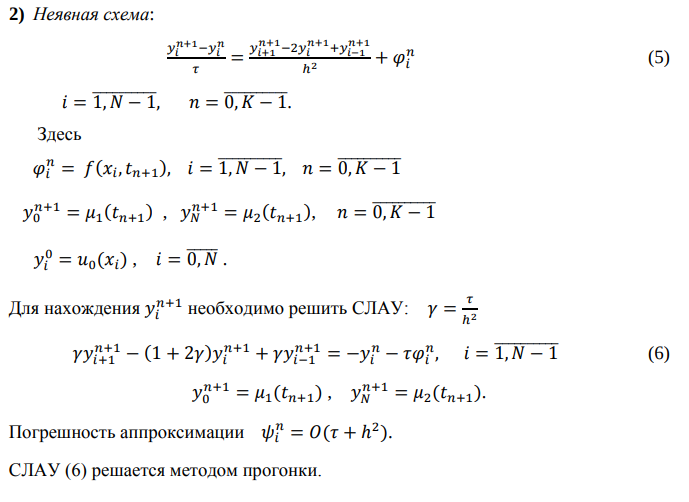
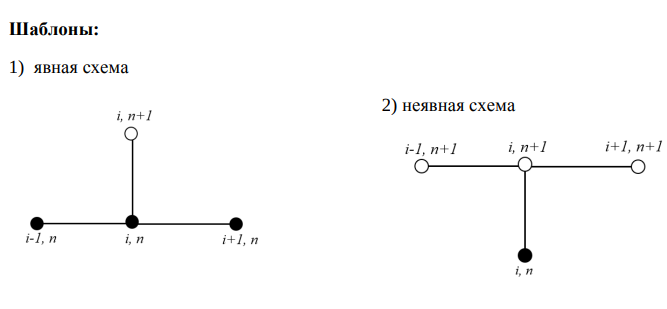
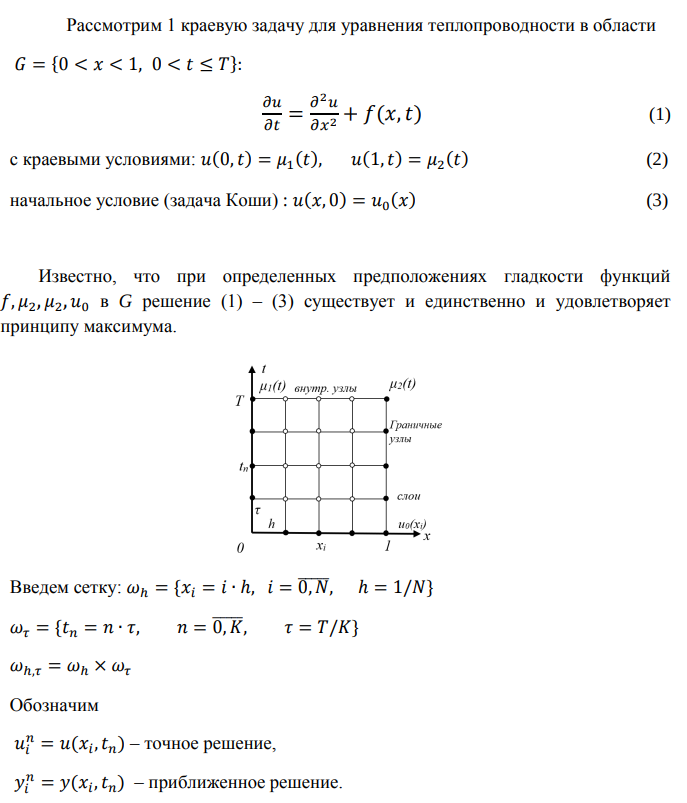
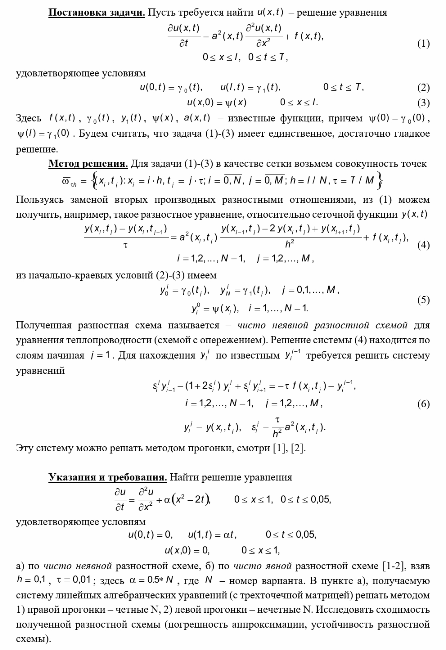
Лабораторная работа 3 УМФ

Вариант 16



**Неявная схема**

def implicit\_scheme(matrix):

tmp = (tau)/(h\*\*2)

A = [0 for i in range(nx)]

B = [0 for i in range(nx)]

C = [0 for i in range(nx)]

# T = [[0 for i in range(nx)] for i in range(nx)]

for i in range(nx):

if(i != nx-1):

B[i] = 2\*tmp + 1

if(i < nx-1):

C[i] = -tmp

else:

C[i] = 0

if(i > 0):

A[i] = -tmp

else:

A[i] = 0

B[0] = 1

B[nx-1] = 1

C,A = A,C

for k in range (1,nt):

d = [matrix[k-1][i] + tau\*f(x[i],t[k-1]) for i in range(nx)]

# Прямой ход метода прогонки

alpha = [0 for \_ in range(nx)]

beta = [0 for \_ in range(nx)]

alpha[0] = -C[0]/B[0]

beta[0] = d[0]/B[0]

for i in range(1, nx-1):

alpha[i] = C[i]/(-B[i] - A[i]\*alpha[i-1])

beta[i] = (d[i] - A[i]\*beta[i-1])/(B[i] + A[i]\*alpha[i-1])

beta[nx-1] = (A[nx-1]\*beta[nx-2] - d[nx-1])/(-B[nx-1] - A[nx-1]\*alpha[nx-2])

# Обратный ход метода прогонки

U = [0 for i in range(nx)]

U[nx-1] = beta[nx-1]

for i in range(nx-2,-1,-1):

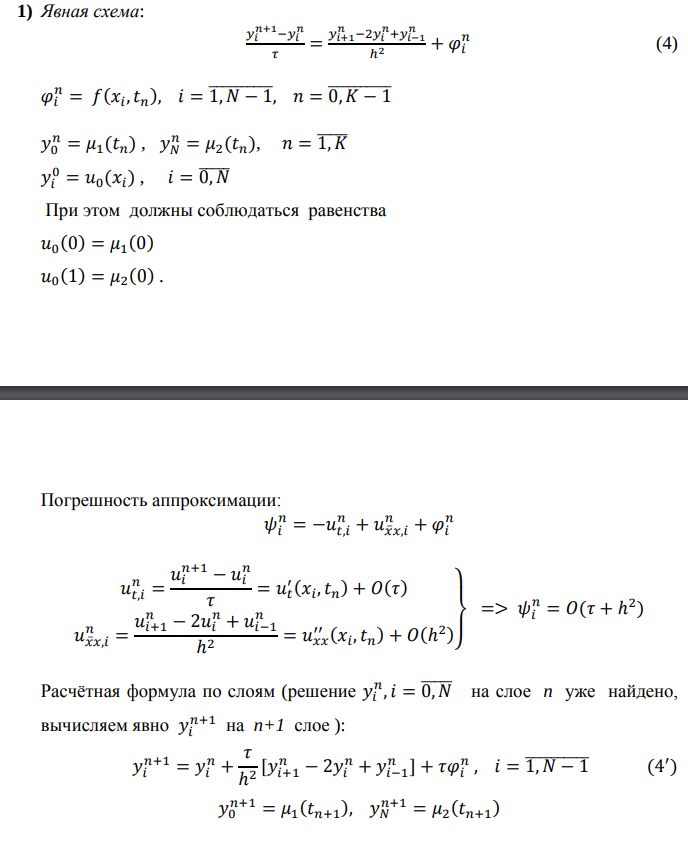
U[i] = alpha[i]\*U[i+1] + beta[i]

for i in range (1,nx-1):

matrix[k][i] = U[i]

print\_matrix(matrix, "Неявная схема")

return matrix



**Явная схема**

def obvious\_scheme(matrix):

for i in range(1,nt):

for j in range (1,nx-1):

matrix[i][j] = matrix[i-1][j] + (tau/h\*\*2)\*(matrix[i-1][j+1] - 2\*matrix[i-1][j] + matrix[i-1][j-1]) + tau\*f(x[j],t[i-1])

print\_matrix(matrix, "Явная схема")

