Московский Авиационный Институт

(Национальный исследовательский университет)

Лабораторная работа №5

По курсу «Численные методы»

Студент:	Ивченко А.В.
Группа:	М8О-408Б-20
Преподаватель:	Пивоваров Д. Е.

Москва, 2023

Задание:

Используя явную и неявную конечно-разностные схемы, а также схему Кранка - Николсона, решить начально-краевую задачу для дифференциального уравнения параболического типа. Осуществить реализацию трех вариантов аппроксимации граничных условий, содержащих производные: двухточечная аппроксимация с первым порядком, трехточечная аппроксимация со вторым порядком, двухточечная аппроксимация со вторым порядком. В различные моменты времени вычислить погрешность численного решения путем сравнения результатов с приведенным в задании аналитическим решением U(x,t). Исследовать зависимость погрешности от сеточных параметров τ , h

Вариант:

9.

$$\frac{\partial u}{\partial t} = a\frac{\partial^2 u}{\partial x^2} + b\frac{\partial u}{\partial x},\ a > 0,\ b > 0.$$

$$u_x(0,t) - u(0,t) = -\exp(-at)(\cos(bt) + \sin(bt)),$$

$$u_x(\pi,t) - u(\pi,t) = \exp(-at)(\cos(bt) + \sin(bt)),$$

$$u(x,0) = \cos x,$$

Аналитическое решение: $U(x,t) = \exp(-at)\cos(x + bt)$.

Теория:

Нанесём на пространственно − временную область $0 \le x \le,\ 0 \le t \le T$ конечно − разностную сетку $\omega_{h\tau} = \{x_j = jh,\ j = 0...N;\ t^k = k\tau,\ k = 0...K\}$ с пространственным временным шагом $h = \frac{l}{N}$ и шагом по времени $\tau = \frac{T}{K}$

Введем два временных слоя: нижний $t^k = k\tau$, на котором распределение искомой функции $u(x_j, t^k)$ известно (при $k = 0$ распределение определяется начальным условием) и верхний временной слой $t^{k+1} = (k + 1)\tau$, на котором распределение искомой функции подлежит определению

Сеточная функция – однозначное отображение целых аргументов j, k в значения функции $u_j^k = u(x_j, t^k)$

Для определения u_j^{k+1} в задаче заменим дифференциальные операторы отношением конечных разностей, получим:

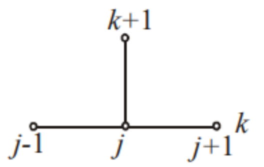
$$\frac{\partial^2 u}{\partial x^2} = \frac{u_{j+1}^k - 2u_j^k + u_{j-1}^k}{h^2} + O(h^2)$$

$$\frac{\partial u}{\partial x} = \frac{u_{j+1}^k - u_{j-1}^k}{2h} + O(h)$$

$$\frac{\partial u}{\partial t} = \frac{u_j^{k+1} - u_j^k}{\tau} + O(\tau)$$

Подставляя полученные выше выражения в задачу, получим **явную** схему:

$$\frac{u_j^{k+1}-u_j^k}{\tau} = a\frac{u_{j+1}^k-2u_j^k+u_{j-1}^k}{h^2} + b\frac{u_{j+1}^k-u_{j-1}^k}{2h}$$



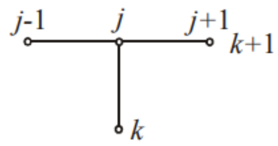
шаблон явной схемы

Если дифференциальный оператор по пространственной переменной аппроксимировать отношением конечных разностей на верхнем временном слое:

$$\frac{\partial^2 u}{\partial x^2} = \frac{u_{j+1}^{k+1}-2u_j^{k+1}+u_{j-1}^{k+1}}{h^2} + O(h^2)$$

, то получим **неявную** схему:

$$\frac{u_j^{k+1}-u_j^k}{\tau} = a\frac{u_{j+1}^{k+1}-2u_j^{k+1}+u_{j-1}^{k+1}}{h^2} + b\frac{u_{j+1}^{k+1}-u_{j-1}^{k+1}}{2h}$$



шаблон неявной схемы

Теперь сеточную функцию u_j^k на верхнем временном слое можно получить из решения СЛАУ

Наконец, запишем явно-неявную схему с весами:

$$\frac{u_j^{k+1}-u_j^k}{\tau} = \theta\left(a\frac{u_{j+1}^{k+1}-2u_j^{k+1}+u_{j-1}^{k+1}}{h^2} + b\frac{u_{j+1}^{k+1}-u_{j-1}^{k+1}}{2h}\right) +$$

$$(1 - \theta)\left(a\frac{u_{j+1}^k-2u_j^k+u_{j-1}^k}{h^2} + b\frac{u_{j+1}^k-u_{j-1}^k}{2h}\right)$$

При $\theta = \frac{1}{2}$ получим схему Кранка-Николсона

Граничные условия:

$$\frac{\partial u(0,t)}{\partial x} - u(0,t) =- e^{-at}(\cos cos\,(bt) + \sin sin\,(bt))$$

Двухточечная аппроксимация с первым порядком:

$$\frac{u_1^{k+1}-u_0^{k+1}}{h} - u_0^{k+1} =- e^{-at}(\cos cos\,(bt) + \sin sin\,(bt))$$

$$u_1^{k+1} = u_0^{k+1}(1 + h) + h * e^{-at}(\cos cos\,(bt) + \sin sin\,(bt))$$

Двухточечная аппроксимация со вторым порядком:

$$u_1^{k+1} = u_0^{k+1} + h\frac{\partial u}{\partial x} + \frac{h^2}{2}\frac{\partial^2 u}{\partial x^2}$$

Из дифференциального уравнения имеем:

$$\frac{\partial^2 u}{\partial x^2} = \frac{1}{a}\frac{\partial u}{\partial t} - \frac{b}{a}\frac{\partial u}{\partial x}$$

=>

$$\frac{\partial u}{\partial x} = \frac{1}{h}\left(u_1^{k+1} - u_0^{k+1} - \frac{h^2}{2}\left(\frac{1}{a}\frac{\partial u}{\partial t} - \frac{b}{a}\frac{\partial u}{\partial x}\right)\right)$$

$$\frac{\partial u}{\partial x} = \frac{1}{\left(1-\frac{bh}{2a}\right)}\left(\frac{u_1^{k+1} - u_0^{k+1}}{h} - \frac{h}{2a}\frac{\partial u}{\partial t}\right)$$

Подставим в граничное условие:

$$\frac{2a}{2a-bh}\left(\frac{u_1^{k+1} - u_0^{k+1}}{h} - \frac{h}{2a}\frac{u_0^{k+1}-u_0^{k}}{\tau}\right) - u_0^{k+1} =- e^{-at}(\cos cos\,(bt) \ + \ \sin sin\,(bt)\,)$$

Трёхточечная аппроксимация:

$$\frac{\partial u}{\partial x} = \frac{-3u_0^{k+1}+4u_1^{k+1}-u_2^{k+1}}{2h}$$

$$\frac{-3u_0^{k+1}+4u_1^{k+1}-u_2^{k+1}}{2h} - u_0^{k+1} =- e^{-at}(\cos cos\,(bt) \ + \ \sin sin\,(bt)\,)$$

Код программы:

```
def yavnaya(n): #явная схема
    u = [0]*n
    for i in range(n):
        u[i] = [0]*n


    for i in range(n):
        u[0][i] = np.cos(x[i])
        #print(u[0][i])

    for j in range(1,n-1):  #внутренние узлы
        for k in range(n-1):
            u[k+1][j] = u[k][j] + sigma1*u[k][j+1] - 2*sigma1*u[k][j] + sigma1*u[k][j-1]
+ sigma2*(u[k][j+1]-u[k][j-1])

    for k in range(0,n-1): #граничные условия
        u[k+1][0] = -(1/h)*u[k+1][1]/(-1-(1/h))+ phi0(t[k+1])/(-1-(1/h))

        #u[k+1][0] = (h*phi0(t[k+1])-u[k+1][1])/(-1-h)
        #u[i][n-1] = h*(-phi0(t[i]))+u[i-1][n-1]*(1+h)
        u[k+1][n-1] = (h*-(phi0(t[k+1]))-u[k+1][n-2])/(-1-h)

    return u


def neyavnaya(n): #неявная схема
    un = [0]*n
    for i in range(n):
        un[i] = [0]*n
    for i in range(n):
        un[0][i] = np.cos(x[i])


    for k in range(n-1):
        A = [[0 for j in range(n+1)] for i in range(n)]
        A[0][0] = 2*a/h+h/tau-(2*a-b*h)
```

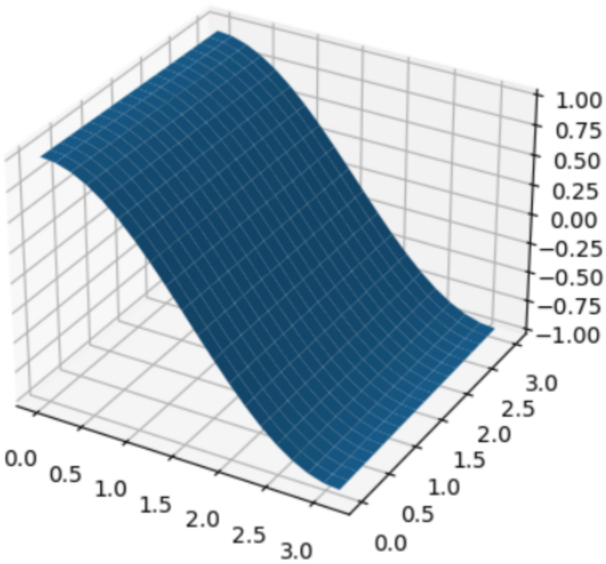
```
        A[0][1] = (-2*a)/h
        A[0][n] = h/tau*un[k][0]+phi0(t[k+1])*(2*a-h*b)
        for j in range(1,n-1):
            A[j][j-1] = -a/h/h+b/(2*h)
            A[j][j] = 1/tau + 2*a/h/h
            A[j][j+1] = -a/h/h-b/(2*h)
            A[j][n] = un[k][j]/tau
        A[n-1][n-2] = -2*a/h
        A[n-1][n-1] = 2*a/h+h/tau+(2*a+b*h)
        A[n-1][n] = h/tau*un[k][n-1]+phi0(t[k+1])*(2*a+h*b)

        res = gauss(A)

        for j in range(n):
            un[k+1][j] = res[j]
    return un

def neyavnoYavnaya(n):
    ukn = [0]*n
    for i in range(n):
        ukn[i] = [0]*n
    for i in range(n):
        ukn[0][i] = np.cos(x[i])

    for k in range(n-1):
        A = [[0 for j in range(n+1)] for i in range(n)]
        A[0][0] = -1-h
        A[0][1] = 1
        A[0][n] = h*phi0(t[k+1])
        for j in range(1,n-1):
            A[j][j-1] = (-a/h/h+b/(2*h))*teta
            A[j][j] = 1/tau+2*a*teta/(h*h)
            A[j][j+1] = (-a/h/h-b/(2*h))*teta
            A[j][n] = ukn[k][j]/tau +
(1-teta)*a/h/h*(ukn[k][j-1]-2*ukn[k][j]+ukn[k][j+1])+b/2/h*(1-teta)*(ukn[k][j+1]-ukn[k][j
-1])
        A[n-1][n-2] = -1-h
        A[n-1][n-1] = 1
        A[n-1][n] = -phi0(t[k+1])*h

        res = gauss(A)

        for j in range(n):
            ukn[k+1][j] = res[j]
    return ukn
```

Результаты:

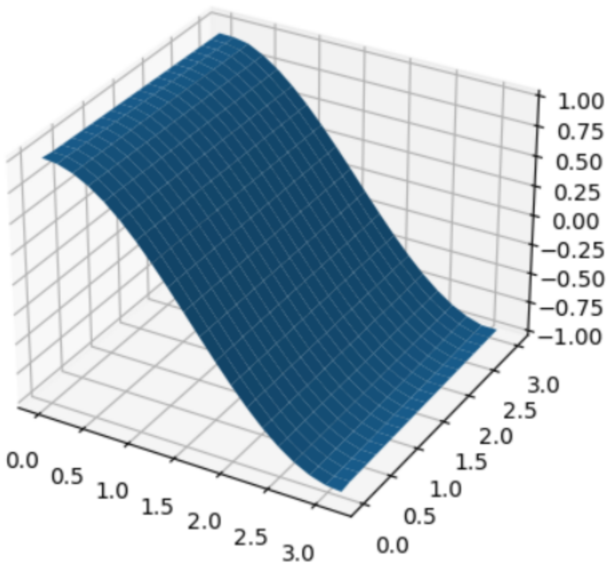
Аналитическое решение

In [4]:

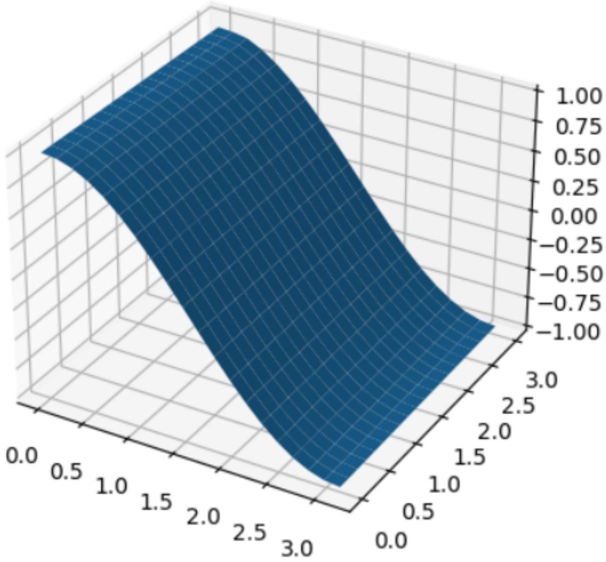
```
U = isxF(x,t)

fig = plt.figure()
ax = plt.axes(projection ='3d')
ax.plot_surface(x_plt,t_plt,np.array(U))
plt.show()
```



Явная схема, двухточечная аппроксимация с первым порядком:



Явная схема, трехточечная аппроксимация со вторым порядком:



Неявная схема, двухточечная апроксимация со вторым порядком:

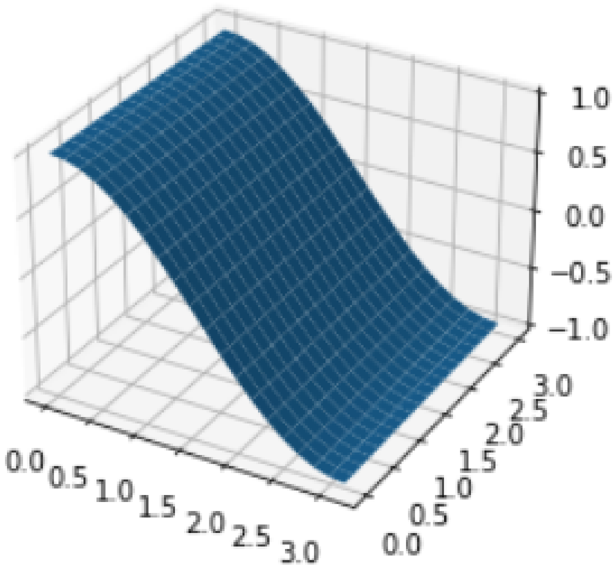
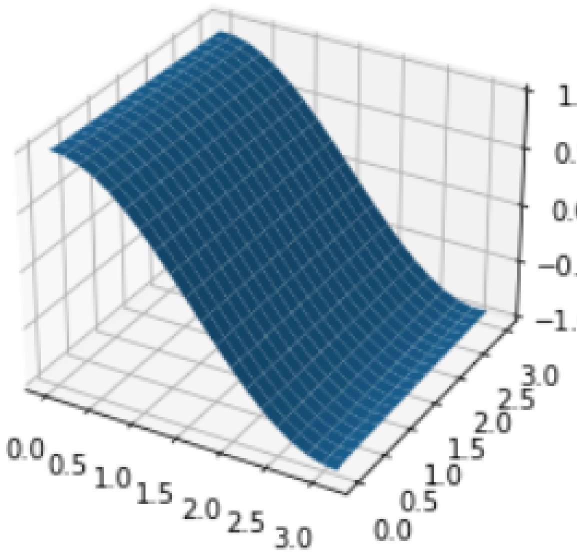


Схема Кранка-Николсона, двухточечная аппроксимация с первым порядком:



Вывод:

Мной были выполнены три различные схемы, позволяющие решить начально-краевую задачу для дифференциального уравнения параболического характера. Каждая из данных схем дала высококачественное решение, близкое к точному.