Московский Авиационный Институт (Национальный исследовательский университет)

Лабораторная работа №5

По курсу «Численные методы»

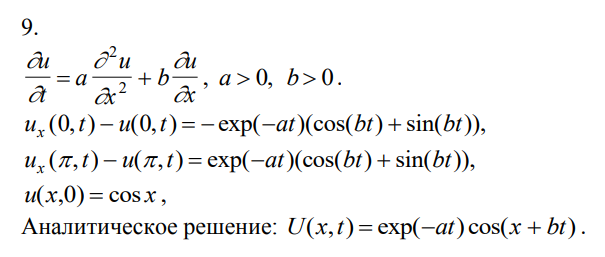
|  |  |
| --- | --- |
| Студент: | Сайгакова А.А. |
| Группа: | М8О-409Б-19 |
| Преподаватель: | Пивоваров Д. Е. |

Москва, 2022

**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:**

****

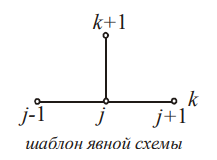
**Теория:**

Введем два временных слоя: нижний , на котором распределение искомой функции известно (при распределение определяется начальным условием) и верхний временной слой , на котором распределение искомой функции подлежит определению

Сеточная функция – однозначное отображение целых аргументов в значения функции

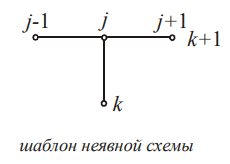
Для определения в задаче заменим дифференциальные операторы отношением конечных разностей, получим:

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

**

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

, то получим **неявную** схему:

**

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

Наконец, запишем явно-неявную схему с весами:

При получим схему Кранка-Николсона

**Граничные условия:**

Двухточечная апроксимация с первым порядком:

Двухточечная апроксимация со вторым порядком:

Из дифференциального уравнения имеем:

=>

Подставим в граничное условие:

Трёхточечная апроксимация:

**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

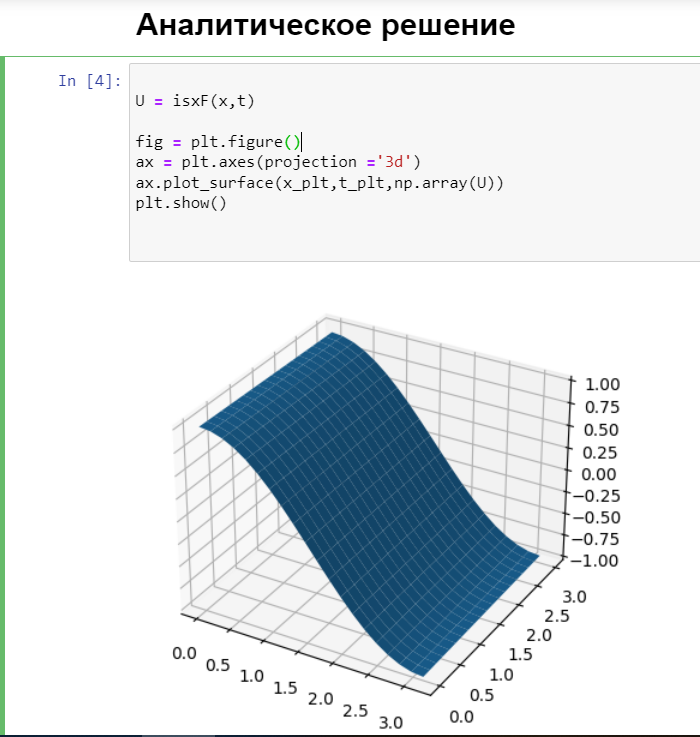
res **=** gauss(A)

**for** j **in** range(n):

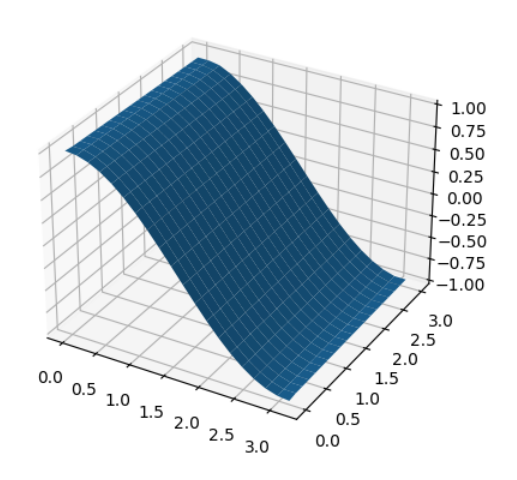
ukn[k**+**1][j] **=** res[j]

**return** ukn

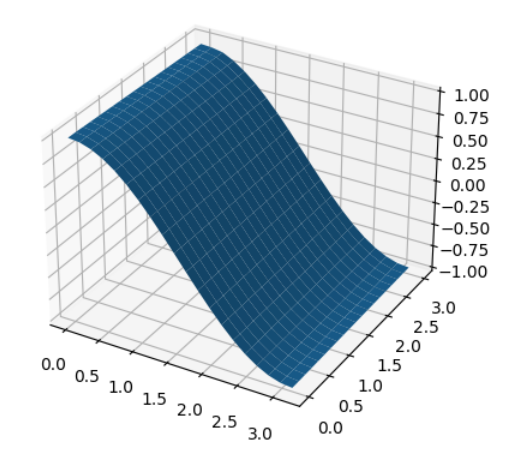
**Результаты:**

****

Явная схема, двухточечная апроксимация с первым порядком:



Явная схема, трехточечная апроксимация со вторым порядком:



Неявная схема, двухточечная апроксимация со вторым порядком:

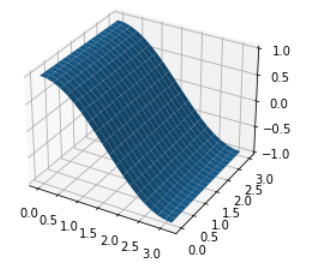
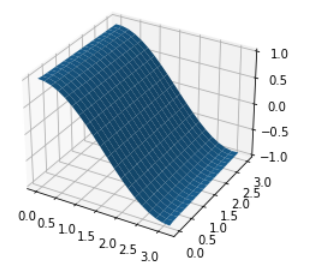


Схема Кранка-Николсона, двухточечная апроксимация с первым порядком:



**Вывод:**

Я реализовала три схемы решения начально-краевой задачи для дифференциального уравнения параболического типа, каждая из схем дала хорошее, близкое к точному решение.