Санкт-Петербургский политехнический университет

Высшая школа теоретической механики, ФизМех

Направление подготовки

«01.03.03 Механика и математическое моделирование»

Отчет по индивидуальной работе №**03**

**тема "Метод конечных разностей. Уравнение Лапласа"**

**дисциплина "Вычислительная механика"**

Выполнил студент гр. 90301 **М. А.Бенюх**

Преподаватель: Е.Ю. Витохин

Санкт-Петербург

**2021**

Оглавление.

Формулировка задания:……………..................................................................1.   
Постановки задачи ........................................................................................2.  
Метод решения .............................................................................................3.  
Явная схема интегрирования........................................................................4.  
Неявная схема интегрирования....................................................................5. Численный анализ решения задач................................................................6. Заключение……....……………………......................................................................7.  
Код…………………....……………………......................................................................8.

1. **Формулировка задания:**Методом конечных разностей, используя итерационную схему интегрирования, решить уравнение Лапласа.
2. **Постановка задачи:**

Объект моделирования: Среда с однородными граничными условиями.

при заданных начальных условиях . Решение выполнить при для .

**Метод решения:**Разложим в окрестности точки в ряд:

Разложим в окрестности точки в ряд:

Введем сетки для времени и для пространства . Тогда:

Конечно-разностное уравнение примет вид:

Итерационный метод:

*Остановка*

1. **Решение**

Оптимальная константа метода: w= 1.8 1.9

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| w | 0,1 | 0,2 | 0,3 | 0,4 | 0,5 | 0,6 | 0,7 | 0,8 | 0,9 | 1,0 | 1,1 | 1,2 | 1,3 | 1,4 | 1,5 | 1,6 | 1,7 | 1,8 | 1,9 |
| k | 10 | 11 | 12 | 12 | 12 | 12 | 12 | 12 | 11 | 11 | 11 | 11 | 11 | 10 | 10 | 10 | 10 | 9 | 9 |

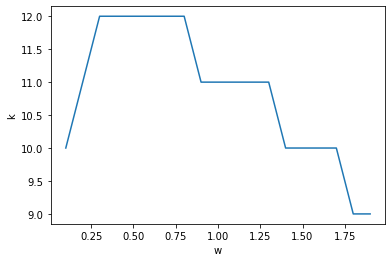


Рис. 1  
зависимость числа итераций от w

Далее будем рассматривать разрезы и поверхность для w=1.9

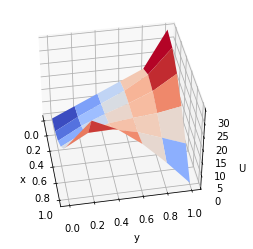


Рис. 2  
Поверхность решения U (x, y) при

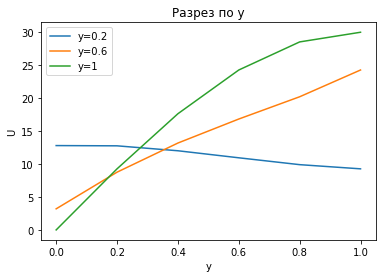
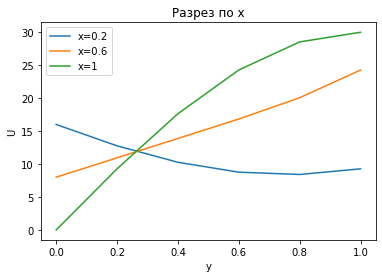
Таблица значений матрицы решений

U =

y

x

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 0 | 0.8 | 3.2 | 7.2 | 12.8 | 20 |
| 4 | 5.69663 | 8.1342 | 11.8401 | 17.805 | 28.5317 |
| 8 | 9.87965 | 11.7877 | 14.2356 | 18.048 | 24.2705 |
| 12 | 14.0219 | 14.9265 | 15.2692 | 15.8909 | 17.6336 |
| 16 | 19.2959 | 18.6298 | 16.0363 | 12.6155 | 9.27051 |
| 30 | 28.5317 | 24.2705 | 17.6336 | 9.27051 | 1.83697e-15 |



x

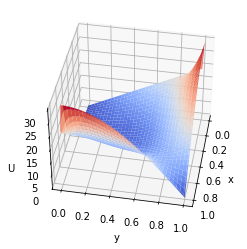
Рис. 4  
Разрезы в разные моменты y

Рис. 3  
Разрезы в разные моменты x

1. **Проверка решения**

Уменьшим шаг разбиения с 0,2 до 0,01 , тогда оптимальная w=0,1

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| w | 0,1 | 0,2 | 0,3 | 0,4 | 0,5 | 0,6 | 0,7 | 0,8 | 0,9 | 1,0 | 1,1 | 1,2 | 1,3 | 1,4 | 1,5 | 1,6 | 1,7 | 1,8 | 1,9 |
| k | 43 | 79 | 110 | 137 | 161 | 182 | 201 | 218 | 233 | 246 | 259 | 270 | 280 | 290 | 300 | 310 | 320 | 331 | 345 |



w=1.9

1. **Заключение**

Было получено решение уравнения Лапласа при помощи итерационного метода. Исходя из поставленных условий и h=0.2 оптимальной константой метода будет w=1,9. Проверка сходимости метода при измельчении шага прошла успешно.

1. **Код**

*x,h,y,eps=1,0.2,1,0.01*

*func\_psi\_0 = lambda y: 20\*y*

*func\_psi\_l = lambda y: [30\*math.cos((math.pi\*i)/2) for i in y]*

*func\_fi\_0 = lambda x: 20\*(x\*\*2)*

*func\_fi\_l = lambda x: [30\*math.cos((math.pi\*i)/2) for i in x] Test=Struna(x, t, h, dt, func\_x\_0, d\_func\_x\_0 ,func2\_x\_0 , func\_0\_t , func\_1\_t)*

*Test=Laplas(x, y, h, func\_fi\_0, func\_fi\_l ,func\_psi\_0 , func\_psi\_l,eps) res2=Test.implicit\_schema()*

*\_return=Test.iterative\_procedures(m)*

*res1=\_return[0]*

class Laplas():

def \_\_init\_\_(self,x, y, h, func\_fi\_0, func\_fi\_l ,func\_psi\_0 , func\_psi\_l,eps):

self.eps=eps

self.x= x

self.x\_0=0

self.y=y

self.y\_0=0

self.h=h

self.func\_fi\_0=func\_fi\_0

self.func\_fi\_l=func\_fi\_l

self.func\_psi\_0=func\_psi\_0

self.func\_psi\_l=func\_psi\_l

self.X=np.arange(self.x\_0,self.x+self.h, self.h)

self.Y=np.arange(self.y\_0,self.y+self.h, self.h)

self.len\_X=len(self.X)

self.len\_Y=len(self.Y)

print("init")

def StartFillMatrix(self):

T=np.zeros((self.len\_X, self.len\_Y))

T[:,0]=self.func\_psi\_0(self.Y)

T[:,-1]=self.func\_psi\_l(self.Y)

T[-1,:]=self.func\_fi\_l(self.X)

T[0,:]=self.func\_fi\_0(self.X)

return T

def iterative\_procedures(self,w):

T=self.StartFillMatrix()

T\_new=np.zeros((self.len\_X, self.len\_Y))

T2=self.StartFillMatrix()

k=0

while (True) :

k=k+1

delta=[]

for i in range(1,self.len\_X-1):

for j in range(1,self.len\_Y-1):

T[i,j]=(1/4)\*(T[i-1,j]+T[i+1,j]+T[i,j-1]+T[i,j+1])

for i in range(1,self.len\_X-1):

for j in range(1,self.len\_Y-1):

T2[i,j]=(1/4)\*(T[i-1,j]+T[i+1,j]+T[i,j-1]+T[i,j+1])

for i in range(0,self.len\_X):

for j in range(0,self.len\_Y):

T\_new[i,j]=T[i,j]+w\*(T2[i,j]-T[i,j])

for i in range(0,self.len\_X):

for j in range(0,self.len\_Y):

delta.append(abs(T\_new[i,j]-T[i,j]))

if(max(delta)<self.eps):

print("break")

break

else:

for i in range(0,self.len\_X):

for j in range(0,self.len\_Y):

T[i,j]=T\_new[i,j]

return [T\_new, k]