Әл-Фараби атындағы Қазақ ұлттық университеті

Механика-математика факультеті

Екінші ретті қарапайым дифференциалдық теңдеулерінің шекаралық есептерін шешудің сандық әдістері.

Темирбеков Нурлан Муханович ф-м.ғ.д., профессор

Жоспар

1. Екі нүктелі шекаралық есеп.
2. Ату әдісі.
3. Есепті шығару үлгісі.

Мақсаты

Шеттік дифференциалдық теңдеулер есебін сандық шешімін табу және есептеу бағдарламасын жүзеге асыру.

Екі нүктелі шекаралық есеп

$$y'' = f(x, y, y'), \qquad y(a) = \alpha, \qquad y(b) = \beta.$$

дифференциалдық теңдеуін шешу.

Бастапқы шарттардың жоқтығын жеңудің бір жолы – жетіспейтін бастапқы мәнді болжау. Алынған шешімнің екінші жағындағы шекаралық шарттарды қанағаттандыру. Сәйкессіздікті тексеру арқылы біз қайтадан интегралдау алдында бастапқы шарттарға қандай өзгерістер енгіземіз.

Екі нүктелі шекаралық есептерді шешудің тағы бір құралы болып дифференциалдық теңдеулер біркелкі орналасқан тор нүктелерінде ақырлы айырымдармен жуықталатын ақырлы айырымдар әдісі болып табылады.

Ату әдісі

$$y'' = f(x, y, y'), \qquad y(a) = \alpha, \qquad y(b) = \beta \qquad (1)$$

Енді (1) есебін бастапқы мән есебіне айналдыру

$$y'' = f(x, y, y'), \qquad y(a) = \alpha, \qquad y'(a) = u \qquad (2)$$

Егер шешім белгіленген $y(b) = \beta$ шекаралық шартымен сәйкес келсе, біз есепті шештік; әйтпесе u реттеп, әрекетті қайталауымыз керек.

Егер біз u анықтаудың түбір табу есебі ретінде қарастырамыз

$$y(b) = \theta(u)$$

Демек, u

$$r(u) = \theta(u) - \beta = 0 \qquad (3)$$

мәнінің түбірі, мұндағы $r(u)$ – шекаралық қалдық.

Ату әдісі

(3) теңдеуді 4-модульде қарастырылған түбірлерді табу әдістерінің бірімен шешуге болады. Риддер алгоритмін таңдаймыз. Сызықты емес шекаралық есептерді шешуде қолданатын процедурамыз:

- ✓ (3) теңдеуінің u түбірі үшін u_1 және u_2 бастапқы мәндер аралығын таңдап алу керек.
- ✓ (3) теңдеуді шешу үшін Риддер әдісін қолданыңыз. Әрбір итерация дифференциалдық теңдеуді бастапқы мән есебі ретінде шешу арқылы $\theta(u)$ мәнін бағалауды қажет ететінін ескеріңіз.
- ✓ u мәнін анықтап, дифференциалдық теңдеулерді тағы бір рет шешіп, нәтижелерін жазып алыңыз.

Есепті шығару үлгісі

МЫСАЛ 1

$$y'' + 3yy' = 0 \qquad y(0) = 0 \qquad y(2) = 1$$

шектік есепті шешіңіз.

Шешуі. Эквивалентті бірінші ретті теңдеулер

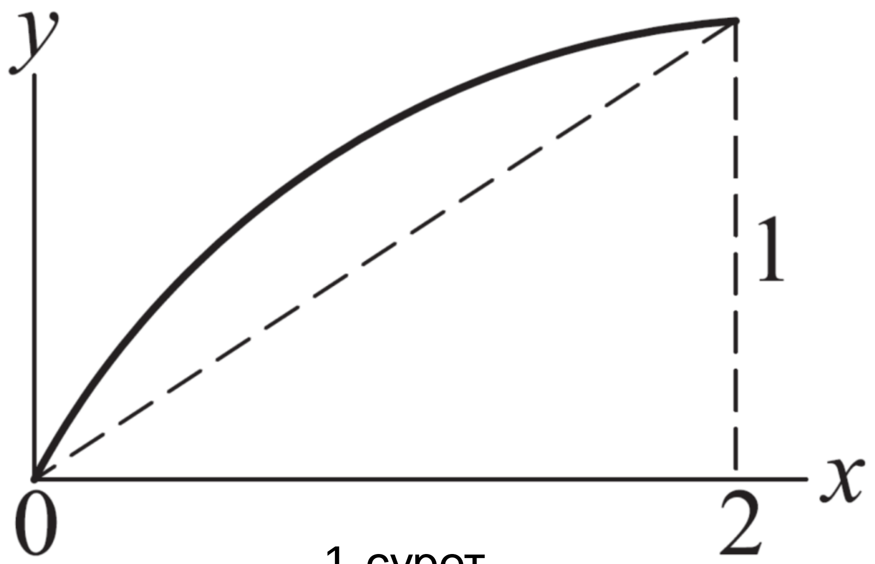
$$y' = \begin{bmatrix} y_0' \\ y_1' \end{bmatrix} = \begin{bmatrix} y_1 \\ -3y_0y_1 \end{bmatrix}$$

шекаралық шарттарымен $y_0(0) = 0 \;\; y_0(2) = 1$ болады.

Енді $y'(0)$ сынақ мәндерін анықтау есебін талдаймыз.

Біз $0 \;\le x \le\; 2$ интервалында y тегіс (тербелмейді) деген негізді болжам жасаудан бастаймыз. Әрі қарай y 0-ден 1-ге дейін ұлғаюы керек екенін ескереміз, бұл үшін $y' > 0$ қажет.

Бұдан y те, y' те оң болса, дифференциалдық теңдеуді қанағаттандыру үшін y'' теріс болуы керек деген қорытындыға келеміз.



1-сурет

Есепті шешу бағдарламасы

Есепті сипаттау үшін пайдаланушы қамтамасыз ететін үш функция қажет.

- ✓ Дифференциалдық теңдеулерді анықтайтын F(x,y) функциясы.
- ✓ Бастапқы шарттарын анықтау үшін initCond(u) функциясы.
- ✓ Риддер әдісін қалдық шекаралық шартпен r(u) функциясы қажет.

Есепті шығару үлгісі

```
## example1
import numpy as np
from run_kut4 import *
from ridder import *
from printSoln import *
def initCond(u): return np.array([0.0, u])
def r(u):
    X,Y = integrate(F,xStart,initCond(u),xStop,h)
    y = Y[len(Y) - 1]
    r = y[0] - 1.0
    return r
def F(x,y):
    F = np.zeros(2)
    F[0] = y[1]
    F[1] = -3.0*y[0]*y[1]
    return F

xStart = 0.0
xStop = 2.0
u1 = 1.0
u2 = 2.0
h = 0.1
freq = 2
u = ridder(r,u1,u2)
X,Y = integrate(F,xStart,initCond(u),xStop,h)
printSoln(X,Y,freq)
```

Есепті шығару үлгісі

Шешімі:

x	y[0]	y[1]
0.0000e+00	0.0000e+00	1.5145e+00
2.0000e-01	2.9404e-01	1.3848e+00
4.0000e-01	5.4170e-01	1.0743e+00
6.0000e-01	7.2187e-01	7.3287e-01
8.0000e-01	8.3944e-01	4.5752e-01
1.0000e+00	9.1082e-01	2.7013e-01
1.2000e+00	9.5227e-01	1.5429e-01
1.4000e+00	9.7572e-01	8.6471e-02
1.6000e+00	9.8880e-01	4.7948e-02
1.8000e+00	9.9602e-01	2.6430e-02
2.0000e+00	1.0000e+00	1.4522e-02

$y'(0) = 1.5145.$

Қорытынды

1. Екі нүктелі шекаралық есептің сандық шешімі.

2. Ату әдісі.

3. Есепті шығарудың сандық үлгісі.

Курстың нәтижесі

- ✓ Python тілі бойынша негізгі құрылымымен, тілдің негізін құрайтын операторлармен танысу.
- ✓ Сызықты алгебралық теңдеулер жүйесін сандық шешудің тура және итерациялық әдістері.
- ✓ Нүктелері бойынша функцияны қалпына келтірудің интерполяция әдістері.
- ✓ Сызықты емес теңдеулерді және жүйелерді шешудің сандық әдістері.
- ✓ Дифференциалдау және интегралдау амалдарын сандық түрде шешу.
- ✓ Дифференциалдық теңдеулердің бастапқы, шектік есептерінің сандық шешімін табу.

Пайдаланылған әдебиеттер тізімі

1. Jaan Kiusalaas. Numerical methods in engineering with Python. Cambridge University Press. ISBN 978-1-107-03385
2. Вабищевич П.Н. Численные методы: Вычислительный практикум. — М.: Книжный дом «ЛИБРОКОМ», 2010. — 320 с.
3. Киреев В. И., Пантелеев А. В. Численные методы в примерах и задачах: Учебное пособие. — СПб.: Издательство «Лань», 2015. — 448 с.