Модуль 6. Жай дифференциалдық теңдеулерді сандық шешу.

6.1 Бастапқы есепті шешу әдістері

$$y' = F(x, y)$$

дифференциалдық теңдеуін $y(a) = \alpha$ шарттарымен шешіңіз.

Кіріспе

Бірінші ретті дифференциалдық теңдеудің жалпы түрі:

$$y' = f(x, y) \qquad (6.1a)$$

мұндағы $y' = dy/dx$ және $f(x, y)$ берілген функция. Бұл теңдеудің шешімі белгісіз тұрақтыны (интегралдау тұрақтысы) қамтиды. Бұл тұрақтыны табу үшін шешім қисығында нүктені білу керек; яғни у функциясы қандай да бір х-тің мәнінде көрсетілуі керек, айталық $x = a$. Бұл көмекші шартты былай жазамыз.

$$y(a) = \alpha \qquad (6.1b)$$

n ретті қарапайым дифференциалдық теңдеуді

$$y^{(n)} = f\left(x, y, y', y'', \dots, y^{(n-1)}\right) \qquad (6.2)$$

әрқашанда бірінші ретті n теңдеулерге түрлендіруге болады. Келесі белгілеулерді қолданып,

$$y_0 = y, \qquad y_1 = y', \qquad y_2 = y'' \quad \dots \quad y_{n-1} = y^{(n-1)} \qquad (6.3)$$

эквивалентті бірінші ретті теңдеулер мына түрде жазылады

$$y_0' = y_1, \qquad y_1' = y_2, \qquad y_2' = y_3 \quad \dots \quad y_n' = f(x, y_0, y_1, y_2, \dots, y_{n-1}) \qquad (6.4a)$$

Шешім енді n көмекші шартты білуді талап етеді. Егер бұл шарттар х-тің бірдей мәнінде көрсетілсе, есеп бастапқы мән есебі деп аталады. Сонда бастапқы шарттар деп аталатын көмекші шарттар

$$y_0(a) = \alpha_0, y_1(a) = \alpha_1, y_2(a) = \alpha_2, \dots, y_{n-1}(a) = \alpha_{n-1} \qquad (6.4b)$$

түрінде болады.

Егер y_i әр түрлі х мәндерінде көрсетілсе, есеп шекаралық есеп деп аталады.

Мысалы,

$$y'' = -y \qquad y(0) = 1 \qquad y'(0) = 0$$

бастапқы мән есебі болып табылады, себебі шешімге берілген көмекші шарттардың екеуі де $x = 0$ мәнінде берілген. Керісінше,

$$y'' = -y \qquad y\,(0) = 1 \qquad y(\pi) = 0$$

шекаралық есеп, себебі екі шарт х-тің әртүрлі мәндерінде көрсетілген.

Бұл сабақта біз тек бастапқы мән есептерін қарастырамыз. Сонымен қатар бірінші ретті теңдеулер жиынын қысқаша түрде өңдеуге мүмкіндік беретін векторлық белгілерді кеңінен қолданамыз. Мысалы, (6.4) теңдеуі мына түрде жазылады

$$y' = F(x, y) \qquad y(a) = \alpha \qquad\qquad (6.5a)$$

мұндағы

$$F(x, y) = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ f(x, y) \end{bmatrix} \qquad\qquad (6.5b)$$

Дифференциалдық теңдеулердің сандық шешімі негізінде x-тің дискретті аралықтарында тізімделген x және y-мәндерінің кестесі болып табылады.

Эйлер әдісі

Эйлердің шешу әдісінің тұжырымдамасы қарапайым. Оның негізі x-ке жуық y-тің қысқартылған Тейлор қатары болып табылады:

$$y(x + h) \approx y(x) + y'(x)h \qquad\qquad (6.6)$$

Себебі (6.6)теңдеу x нүктесінде қолжетімді ақпараттан x+h кезінде y-ті болжайды, оны x пен y-ның берілген бастапқы мәндерінен бастап h қадамымен шешімді алға жылжыту үшін пайдалануға болады.

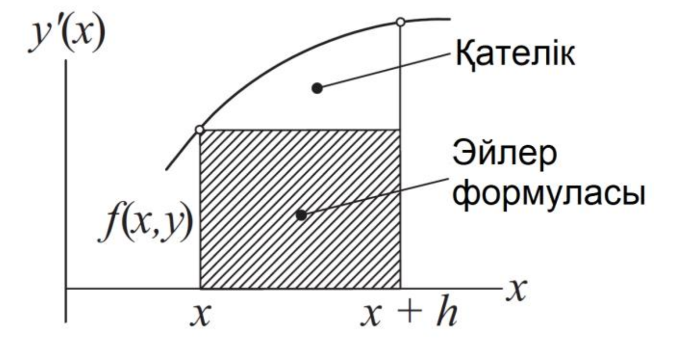
(6.6) теңдеудегі қателік Тейлор қатарының қысқаруынан туындаған теңдеумен берілген:

$$E = \frac{1}{2} y''(\xi)h^2 = O(h^2), \qquad x < \xi < x + h \qquad\qquad (6.7)$$

Жинақталған қателік туралы түсінікті $E_{жин}$ интегралдау кезеңінде әр қадамдағы қателік тұрақты деп есептеу арқылы табуға болады. Содан кейін x_0-ден x_n аралығын қамтитын n интегралдау қадамдары бар деп есептейміз

$$E_{жин} = nE = \frac{x_n - x_0}{h} E = O(h) \qquad\qquad (6.8)$$

Демек, жинақталған қателік қадам сайынғы қателік ретінен бір ретке кем.



6.1-сурет. Эйлер формуласының графикалық көрінісі.

Енді Эйлер теңдеуінің графикалық интерпретациясын қарастырайық. Қарапайымдылық үшін дифференциалдық теңдеу $y' = f(x,y)$ болатындай жалғыз тәуелді айнымалы y бар деп есептейміз. x пен x+h арасындағы y шешімін өзгерісі

$$y(x+h) - y(x) = \int_{x}^{x+h} y' dx = \int_{x}^{x+h} f(x,y) dx$$

болады және ол $y'(x)$ сызбасының астындағы бөліктің ауданы, 6.1-суретте көрсетілген. Эйлер формуласы бұл ауданды көлденең сызылған тіктөртбұрыштың ауданына жақындатады. Тіктөртбұрыш пен қисық арасындағы аумақ қысқарту қателігін білдіреді. Әлбетте, қысқарту қателігі қисық көлбеуіне пропорционал; яғни $(y')' = y''(x)$ пропорционал. Эйлер әдісі есептеудегі тиімсіз болғандықтан тәжірибеде сирек қолданылады. Қысқарту қателігін қолайлы деңгейге дейін төмендету үшін өте аз h қажет, нәтижесінде дөңгелектеу қателігінің ұлғаюымен қатар жүретін көптеген интегралдау қадамдары болады. Әдістің құндылығы негізінен оның қарапайымдылығында, ол орнықтылық сияқты белгілі бір маңызды тақырыптарды талқылауды жеңілдетеді.

Euler модулі

Бұл функция Эйлер интегралдау әдісін жүзеге асырады. Ол бірінші ретті дифференциалдық теңдеулердің кез келген санын өңдей алады. Пайдаланушы

$$F(x,y) = \begin{bmatrix} y_0' \\ y_1' \\ y_2' \\ \vdots \end{bmatrix}$$

массивінің дифференциалдық теңдеулерін анықтайтын $F(x,y)$ функциясын беруі қажет. Функция h аралықтарында x және y мәндерін қамтитын X және Y массивтерін қайтарады.

```
##module euler
''' X,Y = integrate(F,x,y,xStop,h).
{y}' = {F(x,{y})} бастапқы мән есебін
шешуге арналған Эйлер әдісі,
мұндағы {y} = {y[0],y[1],...y[n-1]}.
x,y=бастапқы шарттар
xStop = x-тің ақырлы мәні
h = интегралдаудағы x өсімшесі
F= пайдаланушы ұсынған функция
```

```
F(x,y)={y'[0],y'[1],. ..,y'[n-1]}
массивін қайтарады
'''
import numpy as np
def integrate(F,x,y,xStop,h):
    X = []
    Y = []
    X.append(x)
    Y.append(y)
    while x < xStop:
        h = min(h,xStop - x)
        y = y + h*F(x,y)
        x = x + h
        X.append(x)
        Y.append(y)
    return np.array(X),np.array(Y)
```

printSoln

Біз бұл функцияны сандық интегралдаудан алынған X және Y басып шығару үшін пайдаланамыз. Деректер саны freq параметрімен бақыланады. Мысалы, егер freq = 5 болса, әрбір бесінші интегралдау қадамы көрсетіледі. Егер freq = 0 болса, тек бастапқы және соңғы мәндер көрсетіледі.

```
## module printSoln
'''printSoln(X,Y,freq).
Дифференциалдық теңдеуді шешушіден
қайтарылған X және Y мәндерін
басып шығару жиілігін «freq» пайдаланып
басып шығарады.
freq = n әрбір n-қадамды басып шығарады.
freq = 0 тек бастапқы және соңғы мәндерді
басып шығарады.
'''
def printSoln(X,Y,freq):

    def printHead(n):
```

```
        print("\n x ",end=" ")
        for i in range (n):
            print(" y[",i,"] ",end=" ")
        print()
    def printLine(x,y,n):
        print("{:13.4e}".format(x),end=" ")
        for i in range (n):
            print("{:13.4e}".format(y[i]),end=" ")
        print()

    m = len(Y)
    try: n = len(Y[0])
    except TypeError: n = 1
    if freq == 0: freq = m
    printHead(n)
    for i in range(0,m,freq):
        printLine(X[i],Y[i],n)
    if i != m - 1: printLine(X[m - 1],Y[m - 1],n)
```

МЫСАЛ 6.1

$$y' + 4y = x^2 \qquad\qquad y(0) = 1$$

бастапқы мән есебін $h = 0{,}01$ қадамымен x=0-ден 0,03-ке дейін интегралдаңыз. Сондай-ақ

$$y = \frac{31}{32}e^{-4x} + \frac{1}{4}x^2 - \frac{1}{8}x + \frac{1}{32}$$

аналитикалық шешімін және әрбір қадамдағы жинақталған қысқарту қателігін есептеңіз.

Шешуі. Келесі ыңғайлы белгілеулерді пайдаланып,

$$x_i = ih \qquad y_i = y(x_i)$$

Эйлер формуласын мына түрде жазамыз

$$y_{i+1} = y_i + y_i' h$$

мұндағы

$$y_i' = x_i^2 - 4y_i$$

1-қадам. ($x_0 = 0$ ден $x_1 = 0.01$ дейін):

$$y_0 = 0$$

$$y_0' = x_0^2 - 4y_0 = 0^2 - 4 \cdot 1 = -4$$

$$y_1 = y_0 + y_0'h = 1 + (-4) \cdot 0.01 = 0.96$$

$$(y_1)_{\text{дәл}} = \frac{31}{32}e^{-4\cdot 0.01} + \frac{1}{4}\cdot(0.01)^2 - \frac{1}{8}\cdot 0.01 + \frac{1}{32} = 0.9608$$

$$E_{\text{жин}} = 0.96 - 0.9608 = -0.0008$$

2-қадам. ($x_1 = 0.01$ ден $x_2 = 0.02$ дейін):

$$y_1' = x_1^2 - 4y_1 = 0.01^2 - 4 \cdot 0.96 = -3.840$$

$$y_2 = y_1 + y_1'h = 0.96 + (-3.840) \cdot 0.01 = 0.9216$$

$$(y_2)_{\text{дәл}} = \frac{31}{32}e^{-4\cdot 0.02} + \frac{1}{4}\cdot(0.02)^2 - \frac{1}{8}\cdot 0.02 + \frac{1}{32} = 0.9231$$

$$E_{\text{жин}} = 0.9216 - 0.9231 = -0.0015$$

3-қадам. ($x_2 = 0.02$ ден $x_3 = 0.03$ дейін):

$$y_2' = x_2^2 - 4y_2 = 0.02^2 - 4 \cdot 0.9216 = -3.686$$

$$y_2 = y_2 + y_2'h = 0.9216 + (-3.686) \cdot 0.01 = 0.8847$$

$$(y_3)_{\text{дәл}} = \frac{31}{32}e^{-4\cdot 0.03} + \frac{1}{4}\cdot(0.03)^2 - \frac{1}{8}\cdot 0.03 + \frac{1}{32} = 0.8869$$

$$E_{\text{жин}} = 0.8847 - 0.8869 = -0.0022$$

Қадамдық қателіктің шамасы шамамен 0.008 шамасында тұрақты екенін ескереміз. Осылайша, 10 интегралдау қадамынан кейін жинақталған қате шамамен 0.08 болады, осылайша шешім бір мәнді цифр дәлдігіне дейін төмендейді. 100 қадамнан кейін барлық мәнді сандар жоғалады.

МЫСАЛ 6.2

$$y'' = -0{,}1y' - x \qquad y(0) = 0 \qquad y'(0) = 1$$

бастапқы мән есебін Эйлер әдісімен x=0-ден 2-ге дейін h=0.05 қадаммен интегралдаңыз. Есептелген y мәнін аналитикалық шешіммен

$$y = 100x - 5x^2 + 990(e^{-0.1x} - 1)$$

бірге сызыңыз.

Шешуі. $y_0 = y$ және $y_1 = y'$ белгілеуімен эквивалентті бірінші ретті теңдеулер мен бастапқы шарттар мына түрде болады

$$F(x, y) = \begin{bmatrix} y_0' \\ y_1' \end{bmatrix} = \begin{bmatrix} y_1 \\ -0{,}1y_1 - x \end{bmatrix} \qquad y(0) = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

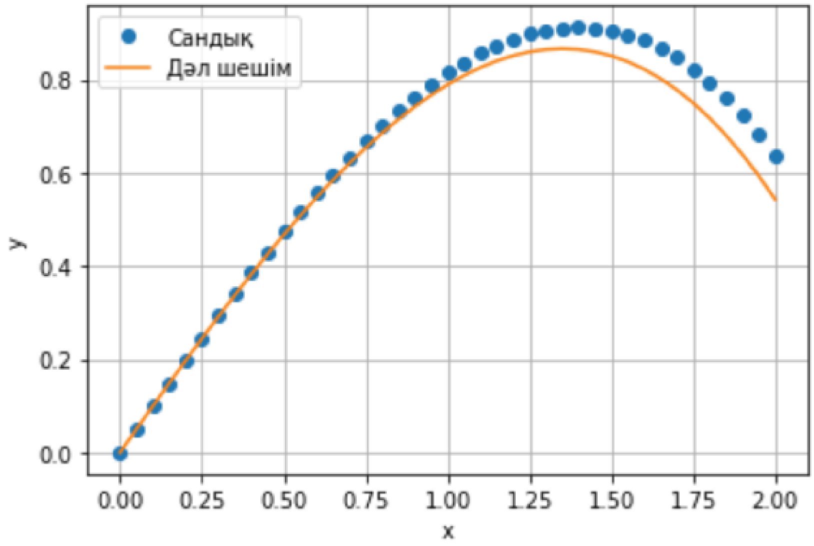
Мұнда `euler` функциясын пайдаланатын бағдарлама:

```
## example6_2
```

```
import numpy as np
from euler import *
import matplotlib.pyplot as plt

def F(x,y):
    F = np.zeros(2)
    F[0] = y[1]
    F[1] = -0.1*y[1] - x
    return F
x = 0.0 # Интегралдаудың басы
xStop = 2.0 # Интегралдаудың соңы
y = np.array([0.0, 1.0]) #{y} Бастапқы мәндері
h = 0.05 # Қадам өлшемі
X,Y = integrate(F,x,y,xStop,h)
yExact = 100.0*X - 5.0*X**2 + 990.0*(np.exp(-0.1*X) - 1.0)
plt.plot(X,Y[:,0],'o',X,yExact,'-')
plt.grid(True)
plt.xlabel('x'); plt.ylabel('y')
plt.legend(('Сандық','Дәл шешім'),loc=0)
plt.show()
```

Алынған график



Графиктің бастапқы бөлігі дерлік түзу сызықтас. Сандық шешімдегі қысқарту қателігі y''-ге пропорционал болғандықтан, екі шешімнің арасындағы сәйкессіздік аз. Графиктің қисықтығы ұлғайған сайын қысқарту қателігі де өседі.

Бағдарламаны h=0.025 мәнімен беру арқылы h өзгерісі қалай әсер ететінін көруге болады. Бұндай жағдайда қысқарту қателігі екі есе азаяды.

Рунге-Кутта әдістері

Эйлер әдісі бірінші ретті дәлдіктегі әдіс ретінде қарастырылады, өйткені оның жинақталған қысқарту қателігі O(h) ретінде әрекет етеді. Оның негізі

$$y(x+h) = y(x) + y'(x)h$$

қысқартылған Тейлор қатары болды. Сандық интегралдаудың дәлдігін қатардың көбірек мүшелерін сақтау арқылы жақсартуға болады. Осылайша, n-ші реттік әдіс қысқартылған Тейлор қатарын

$$y(x+h) = y(x) + y'(x)h + \frac{1}{2!}y''(x)h^2 + \cdots + \frac{1}{n!}y^{(n)}(x)h^n$$

Бірақ енді $y' = F(x, y)$ қайта-қайта дифференциалдау арқылы $y'', y''', \dots y^{(n)}$ өрнектерін шығаруымыз керек және оларды бағалау үшін ішкі есептеулерді жүргізу керек. Бұл қосымша жұмысты Рунге-Кутта әдістерін қолдану арқылы болдырмауға болады, олар сондай-ақ қысқартылған Тейлор қатарларына негізделген, бірақ y(x) жоғары туындыларын есептеуді қажет етпейді.

Екінші ретті дәлдіктегі Рунге-Кутта әдісі

Екінші ретті Рунге-Кутта әдісіне келу үшін

$$y(x+h) = y(x) + c_0 F(x, y)\cdot h + c_1 F\,[x + ph, y + qhF(x, y)]\cdot h \qquad (a)$$

түріндегі интегралдау формуласын қабылдаймыз және (a) теңдеуін Тейлор қатарына сәйкестендіріп, c_0, c_1, p және q параметрлерді табуға тырысамыз,

$$y(x+h) = y(x) + y'(x)h + \frac{1}{2!}y''(x)h^2 = y(x) + F(x, y)h + \frac{1}{2}F'(x, y)h^2 \qquad (b)$$

Атап өтейік,

$$F'(x, y) = \frac{\partial F}{\partial x} + \sum_{i=0}^{n-1}\frac{\partial F}{\partial y_i}y_i' = \frac{\partial F}{\partial x} + \sum_{i=0}^{n-1}\frac{\partial F}{\partial y_i}F_i(x, y)$$

мұндағы n – бірінші ретті теңдеулер саны, (b) теңдеуді келесі түрде жазуға болады

$$y(x+h) = y(x) + F(x, y)h + \frac{1}{2}\left(\frac{\partial F}{\partial x} + \sum_{i=0}^{n-1}\frac{\partial F}{\partial y_i}F_i(x, y)\right)h^2 \qquad (c)$$

(a) теңдеуге қайта оралсақ, біз соңғы мүшені бірнеше айнымалылардағы Тейлор қатарын қолдану арқылы қайта жаза аламыз,

$$F\,[x+ph,y+qhF(x,y)] = F(x,y) + \frac{\partial F}{\partial x}ph + qh\sum_{i=1}^{n-1}\frac{\partial F}{\partial y_i}F_i(x,y)$$

Осыдан (a) теңдеуі

$$y(x+h) = y(x) + (c_0+c_1)F(x,y)\cdot h + c_1\left[\frac{\partial F}{\partial x}ph + qh\sum_{i=1}^{n-1}\frac{\partial F}{\partial y_i}F_i(x,y)\right]\cdot h \qquad (d)$$

(c) және (d) теңдеулерін салыстырып, егер

$$c_0+c_1=1, \qquad c_1p=\frac{1}{2}, \qquad c_1q=\frac{1}{2} \qquad (e)$$

болса, олардың бірдей екенін табамыз, өйткені (e) теңдеулерде төрт белгісіз параметрде үш теңдеуді көрсетілген, біз параметрлердің біріне кез келген мән бере аламыз. Кейбір танымал таңдаулар және алынған формулалармен байланысты атаулар төмендегідей:

$c_0=0$	$c_1=1$	$p=1/2$	$q=1/2$	Модификацияланған Эйлер әдісі
$c_0=1/2$	$c_1=1/2$	$p=1$	$q=1$	Хен әдісі
$c_0=1/3$	$c_1=2/3$	$p=3/4$	$q=3/4$	Ралстон әдісі

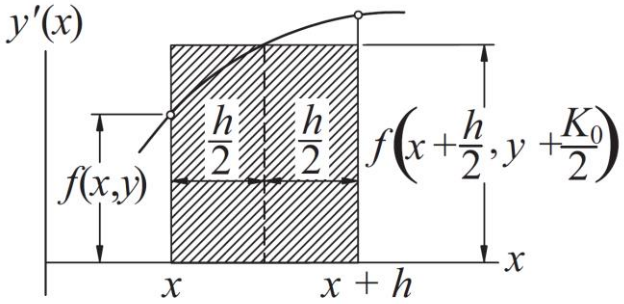
Барлық осы формулалар екінші ретті Рунге-Кутта әдістеріне жатқызылады, оның үстіне формулалардың ешқайсысы басқа формулалардан сандық артықшылыққа ие емес. Модификацияланған Эйлер әдісін таңдап, сәйкес параметрлерді (a) теңдеуіне ауыстырсақ

$$y(x+h) = y(x) + F\left[x+\frac{h}{2}, y+\frac{h}{2}F(x,y)\right]\cdot h \qquad (f)$$

Бұл интегралдау формуласын келесі әрекеттер тізбегі арқылы ыңғайлы түрде жазуға болады:

$$\begin{aligned} &K_0 = h\cdot F(x,y)\\ &K_1 = h\cdot F\left(x+\frac{h}{2}, y+\frac{1}{2}K_0\right) \qquad (6.9)\\ &y(x+h) = y(x) + K_1 \end{aligned}$$

Екінші ретті дәлдіктегі әдістер компьютерлік қолданбаларда танымал емес. Көптеген бағдарламашылар аз есептеу күшімен берілген дәлдікке жететін төртінші ретті дәлдіктегі интегралдау формулаларды қалайды.



7.2-сурет. Модификацияланған Эйлердің формуласының графикалық көрінісі.

7.2-суретте $y' = f(x, y)$ қарапайым дифференциалдық теңдеу үшін модификацияланған Эйлер формуласының графикалық интерпретациясы көрсетілген. (7.9) теңдеулердің біріншісі Эйлер формуласы бойынша бөліктің ортаңғы нүктесіндегі у бағасын береді: $y(x + h/2) = y(x) + f(x, y)\,h/2 = y(x) + K_0/2$. Содан кейін екінші теңдеу бөліктің ауданын көлденең сызылған тіктөртбұрыштың K1 ауданына жақындатады. Мұндағы қателік сызбаның $(y')'' = y'''$ қисықтығына пропорционал.

Төртінші ретті дәлдіктегі Рунге-Кутта әдісі

Төртінші ретті дәлдіктегі Рунге-Кутта әдісі екінші ретті әдіспен бірдей амалдар бойынша Тейлор қатарынан алынған. Шығарылуы өте ұзақ болғандықтан, біз оны өткізіп жібереміз. Интегралдау формуласының соңғы түрі қайтадан параметрлерді таңдауға байланысты; яғни бірегей Рунге-Кутта төртінші ретті формуласы жоқ. Рунге-Кутта әдісі ретінде белгілі ең танымал нұсқа келесі әрекеттер тізбегін қамтиды:

$$
\begin{aligned}
&K_0 = h \cdot F(x, y) \\
&K_1 = h \cdot F\left(x + \frac{h}{2}, y + \frac{K_0}{2}\right) \\
&K_2 = h \cdot F\left(x + \frac{h}{2}, y + \frac{K_1}{2}\right) \\
&K_3 = h \cdot F(x + h, y + K_2) \\
&y(x + h) = y(x) + \frac{1}{6}(K_0 + 2K_1 + 2K_2 + K_3)
\end{aligned}
\qquad (6.10)
$$

Бұл әдістің негізгі кемшілігі – қысқарту қателігін бағалауға мүмкіндік бермейді. Сондықтан, h интегралдау қадамының өлшемін болжау керек немесе оны сынақ және қателіктер арқылы анықтау керек.

run_kut4 модулі

Бұл модульдегі integrate функциясы төрт ретті дәлдіктегі Рунге-Кутта әдісін жүзеге асырады. Пайдаланушы $y' = \mathrm{F}(x, y)$ бірінші ретті дифференциалдық теңдеулерді анықтайтын F(x,y) функциясымен интегралдауды қамтамасыз етуі керек.

```
## module run_kut4
''' X,Y = integrate(F,x,y,xStop,h).
{y}' = {F(x,{y})} бастапқы мән есебін
 шешудің 4-ші ретті Рунге-Кутта әдісі,
мұндағы {y} = {y[0],y[1],...y[n-1]}.
x,y = бастапқы шарттар
xStop = x-тің ақырлы мәні
h = интегралдаудағы x өсімшесі
F = пайдаланушы ұсынған функция
F(x,y) = {y'[0],y'[1],...,y'[n-1]} массивін қайтарады.
'''
import numpy as np
def integrate(F,x,y,xStop,h):
    def run_kut4(F,x,y,h):
        K0 = h*F(x,y)
        K1 = h*F(x + h/2.0, y + K0/2.0)
        K2 = h*F(x + h/2.0, y + K1/2.0)
        K3 = h*F(x + h, y + K2)
        return (K0 + 2.0*K1 + 2.0*K2 + K3)/6.0
X = []
Y = []
X.append(x)
Y.append(y)
while x < xStop:
    h = min(h,xStop - x)
    y = y + run_kut4(F,x,y,h)
    x = x + h
    X.append(x)
    Y.append(y)
return np.array(X),np.array(Y)
```

МЫСАЛ 6.3

$$y'' = -0{,}1y' - x \qquad y(0) = 0 \qquad y'(0) = 1$$

бастапқы мән есебін төртінші ретті Рунге-Кутта әдісімен x=0-ден 2-ге дейін h=0.2 қадаммен интегралдаңыз. Есептелген y мәнін аналитикалық шешіммен

$$y = 100x - 5x^2 + 990(e^{-0.1x} - 1)$$

бірге сызыңыз.

Шешуі. $y_0 = y$ және $y_1 = y'$ белгілеуімен эквивалентті бірінші ретті теңдеулер мен бастапқы шарттар мына түрде болады

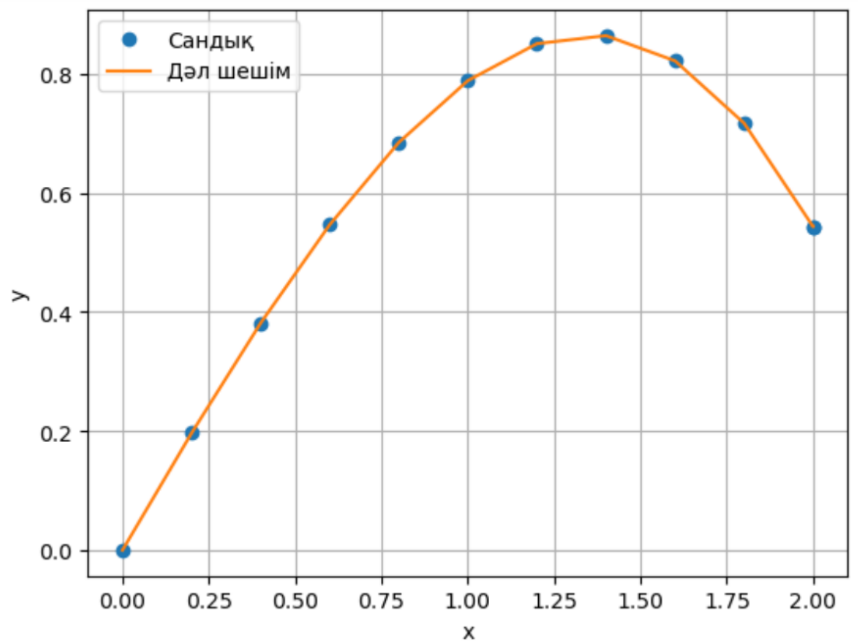
$$F(x,y) = \begin{bmatrix} y_0' \\ y_1' \end{bmatrix} = \begin{bmatrix} y_1 \\ -0{,}1y_1 - x \end{bmatrix} \qquad y(0) = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Екі операторды қоспағанда, келесіде көрсетілген бағдарлама 6.2-мысалда қолданылған бағдарламамен бірдей

```
## example6_3
import numpy as np
from printSoln import *
from run_kut4 import *
import matplotlib.pyplot as plt

def F(x,y):
    F = np.zeros(2)
    F[0] = y[1]
    F[1] = -0.1*y[1] - x
    return F
x = 0.0                          # Start of integration
xStop = 2.0                      # End of integration
y = np.array([0.0, 1.0])         # Initial values of {y}
h = 0.2                          # Step size

X,Y = integrate(F,x,y,xStop,h)
yExact = 100.0*X - 5.0*X**2 + 990.0*(np.exp(-0.1*X) - 1.0)
plt.plot(X,Y[:,0],’o’,X,yExact,’-’)
plt.grid(True)
plt.xlabel(’x’); plt.ylabel(’y’)
plt.legend((’Numerical’,’Exact’),loc=0)
plt.show()
```



Осы графикті 6.2-мысалдағымен салыстыра отырып, біз төртінші ретті Рунге-Кутта әдісінің Эйлер әдісінен айқын артықшылығын көреміз. Бұл есепте екі әдіс шамамен бірдей есептеу көлемін қамтиды. Эйлер әдісі әр қадам үшін F(x,y) бір бағалауды қажет етеді, ал Рунге-Кутта әдісі төрт бағалауды қамтиды. Алайда Эйлер әдісінің қадамдар саны төрт есе көп. Нәтижелердің дәлдігіндегі үлкен айырмашылықты ескере отырып, біз төртінші ретті Рунге-Кутта әдісі жақсы дәлірек есептейді деген қорытындыға келуге болады.

Қолданылған әдебиеттер тізімі

1) Шакенов Қ.Қ. Есептеу математикасы әдістері лекциялар курсы.Алматы,2019. – 193б
2) P. Dechaumphai, N. Wansophark. Numerical Methods in Science and Engineering Theories with MATLAB, Mathematica, Fortran, C and Python Programs. Alpha Science International Ltd. 2022
3) Н. С. Бахвалов, Н. П. Жидков, Г. М. Кобельков Численные методы: Классический университетский учебник. —М.: Издательство «Бином. Лаб. знаний»,2020. — 636 с.
4) Вабищевич П.Н. Численные методы: Вычислительный практикум. — М.: Книжный дом «ЛИБРОКОМ», 2020. — 320 с.

Интернет ресурстар

1) https://docs.python.org/3/
2) http://math-hse.info/f/2018-19/py-polit/instruction_JN.pdf
3) https://jupyter-notebook-beginner-guide.readthedocs.io/en/latest/execute.html
4) https://colab.research.google.com/
5) https://planetcalc.ru/search/?tag=2874