Тел: **+7 (707) 900 92 67**
Почта: **saken.yan@yandex.com**

7 ЛЕКЦИЯ

САНДЫҚ ӘДІСТЕР

§7. Қарапайым диференциялды теңдеулер үшін коши есебі (ҚДТ).

Жалпы түсінік.

Ғылыми және инженерлік есептерді шешу барысында көп жағдайда қарастырып отырған динамикалық жүйеге математикалық сипаттама жасау қажеттілігі туындайды. Осыған орай, кейбір жағдайларда ізделініп отырған шамалар арасында бір мәнді байланыс орнату әрдайым мүмкін бола бермейді, дегенмен бұл шамалардың туындылары арасында байланыс орнатып дифференциалдық теңдеулерді алуға болады.

Осы дифференциалдық теңдеулердің шешімдерін интеграл түрінде жазудың көптеген аналитикалық әдістері бар. Алайда, практикалық есептерді шешу барысында бұл интегралдар көп жағдайда элементар функциялар арқылы сипатталмайды. Бұл жағдайда дифференциалдық теңдеулерді шешу үшін сандық әдістерін қолдану қажет.

ҚДТ есептерін негізгі екі топқа бөлуге болады:

Бастапқы шарттар арқылы шешілетін есептер (Коши есебі).

Шектік шарттар арқылы шешілетін есептер.

Бірінші ретті ҚДТ қарастырайық:

$$\frac{dy}{dx} = f(x, y) \tag{7.1}$$

және мынадай бастапқы шарт берілсін:

$$y(x_0) = y_0 \tag{7.2}$$

Мұндағы $f(x, y) -$ әлдебір, жалпы жағдайда екі айнымалыдан тәуелді сызықтық емес функциясы.

Жоғарыдағы (7.1) және (7.2) арқылы жазылған есеп, бастапқы есеп немесе *Коши есебі* деп аталады. (7.1) теңдеуінің қарапайымдылығына қарамастан, оларды аналитикалық түрде шешу тек кейбір арнайы теңдеулердің түрлері үшін орындауға болады. Сол себепті сандық әдістерді қолдану керек.

Коши есебін шешудің сандық әдістері.

ҚДТ шешу үшін біз қадам арқылы орындалатын әдістерді қолданатын боламыз. Бұл әдістердің негізгі идеясы мынада:

$x_i \in [x_0, c]$, $(0 \leq i \leq N)$ нүктелер тізбегі қарастырылады. Бірдей немесе айнымалы қадамдар арқылы бөлінген $h_i = x_{i+1} - x_i$.

Әрбір x_i нүктесінде $y(x_i)$ мәні (шешім) алдыңғы мәндер (шешімдер) арқылы жуықтап табылады.

Алдыңғы k мәндерінен келесі y_{i+1} мәнін аңықтайтын әдіс қадам арқылы шешу әдісі деп аталады. Осындай бір сатылы әдіске Эйлер әдісі мысал бола алады.

ЭЙЛЕР ӘДІСІ

Коши есебін қарастырайық:

$$\frac{dy}{dx} = f(x, y), \;\; y(x_0) = y_0$$

Бұл теңдеуді сандық әдіс арқылы шешу үшін қадамда $h = \frac{c-x_0}{N}$ етіп таңдап аламыз. Сонда түйіндерді мына түрде $x_i = x_0 + i \cdot h = 0, 1, 2, \ldots, N$ жазуға болады. Сонда біздің мақсатымыз $[x_0, c]$ кестеде көрсетілгендей

x	x_0	x_1	$\ldots$	$x_N = c$
y	y_0	y_1	$\ldots$	$y_N = y(c)$

x_i нүктелерінде $y = y(x)$ функциясының y_i жуық мәндерін есептеу болып табылады.

Осы мақсатпен $y = y(x)$ функциясын x_0 −деген нүкте аймағында тейлор қатарына жіктейік.

$$y(x) = y(x_0) + y'(x_0)(x - x_0) + \frac{1}{2}y''(x_0)(x - x_0)^2 + \cdots \quad (7.3)$$

h -тың кіші мәндері үшін $(x_0 + h)$ нүктесінде (7.3) қатарының екі мүшесімен шектесек болады. Сонда

$$y(x_0 + h) = y(x_0) + y'(x_0)h + O(h^2) \quad (7.4)$$

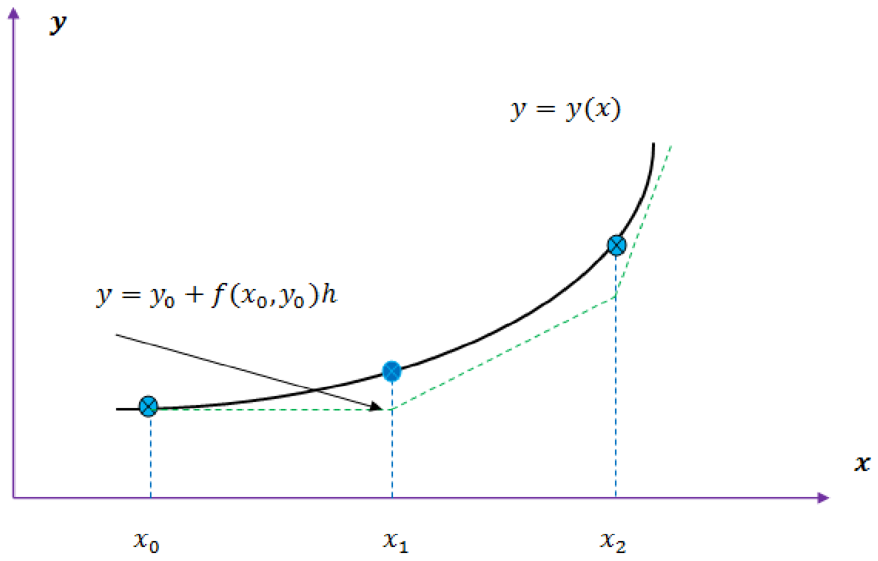
$O(h^2)$ − қалдық функциясы h^2 дәлдігімен сәйкес келеді.

Енді (7.4) ішіндегі $y'(x_0)$ туындысын (7.1) дегі функциямен алмастырайық.

$$y(x_0 + h) = y_0 + f(x_0, y_0)h \quad (7.5)$$

Енді $x_1 = x_0 + h$ нүктесіндегі жуықталған шешімді қайтадан келесі түйінде ізделінетін функцияның бастапқы шарты ретінде қарастыруға болады және (7.5) формуланы пайдаланып, функцияның $x_2 = x_1 + h$ нүктесіндегі мәнін табуға болады.

Нәтижесінде Коши есебін шешудің қарапайым алгоритмін алынамыз. бұл Эйлер әдісі немесе жаңама әдісі деп аталады. Соңғы атау әдістің геометриялық интерпретациясымен байланысты (суретті қараңыз).



Сонда іздеп отырған $y(x)$ функциясы сыңық сызықтармен ауыстырылады, бұлсызықтар $x_0, x_1, x_2, \ldots$ түйіндеріндегі осы функцияға жанамалардың сегменттері болып табылады.

Сонымен, Эйлер әдісін келесі түрде жазуға болады:

$$y_{i+1} = y_i + f(x_i, y_i)h. \quad i = 0, 1, 2 \ldots N-1 \tag{7.6}$$

АЙҚЫН ЕМЕС ЭЙЛЕР ӘДІСІ

ҚДЕ үшін айқын емес эйлер әдісінің түрі мынадай

$$y_{i+1} = y_i + f(x_{i+1}, y_{i+1})h, \; i = 0, 1, 2 \ldots N-1 \tag{7.7}$$

Бұл әдіс эйлер әдісі секілді, h-қа қатысты бірінші дәлдікпен аңықтайды.

Мысал.
Берілген Коши есебін қарастырайық

$$\begin{cases} \dfrac{dy}{dx} = yx^2, & x \in [0, 1] \\ y(0) = 1 \end{cases}$$

мынадай әдістер арқылы:

1. Эйлердің айқын әдісі.

2. Эйлердің айқын емес.

6 түйіннен тұратын тор үшін.

Шешім.
Қадамды аңықтайық:

$$h = \frac{b-a}{N-1} = \frac{1}{5} = 0.2$$

1. Эйлердің айқын әдісі:

$$y_{i+1} = y_i + f(x_i, y_i)h \quad \rightarrow \quad y_{i+1} = y_i + hx_i^2 y_i = y_i(1 + hx_i^2)$$

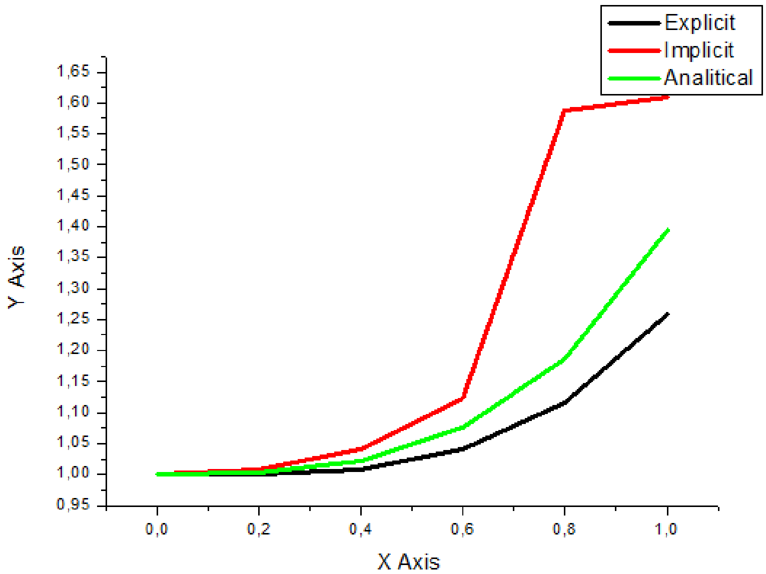
2. Эйлердің айқын емес әдісі:

$$y_{i+1} = y_i + f(x_{i+1}, y_{i+1}) \cdot h \quad \rightarrow \quad y_{i+1} = y_i + h \cdot x_{i+1}^2 \cdot y_{i+1}$$

осыдан

$$y_{i+1} = \quad y_i/(1 - hx_i^2)$$

i	0	1	2	3	4	5
x_i	0	0.2	0.4	0.6	0.8	1.0
Эйлердің айқын әдісі	1	1	1.008	1.04	1.115	1.26
Эйлердің айқын емес әдісі	1	1.008	1.041	1.122	1.587	1.61
Дәл шешім $y(x) = exp\left(\frac{x^3}{3}\right)$	1	1.003	1.022	1.075	1.186	1.396



c# тіліндегі бағдарлама.

```
class Program
{
        public static float function(float x1, float y1, float h)
        {
           float y2 = y1*(1 + h * x1 * x1);  return y2;
        }

        public static void Main(string[] args)
        {
           float a = 0.0f;
           float b = 1.0f;  int n = 20;

           float h = (b-a)/(n-1);

           float[] x = new float[n];  x[0] = a;
           float[] y = new float[n];  y[0] = 1.0f;

           for ( int i = 0; i < n-1; i++ )
           {
             y[i + 1] = function(x[i],y[i],h);
             x[i + 1] = x[i] + h;

             Console.WriteLine("x[{0}]={1} y[{0}]={2}",i+1, x[i+1], y[i+1]);
           }

        }//Main

}//Program
```

Задачи лабораторного урока №7:

№7.1

7-дәрістің соңында көрсетілген кодты алып, мына теңдеуді шешетін етіп жасаңыз:

$$y' - 2xy = 3x^2 - 2x^4\ , \quad y(0) = 1$$

№7.2

Әдіс InPut( ).

Осы әдіс ішінде:

Программаны қолданушы бастапқы шарттарды, соңғы нүктені, қадам санын өзі еңгізе алатындай код жазыңыз. Бұл код қауыпсыз еңгізуді жүзеге асыру керек. Осы бастапқы шарттарды еңгізетін кодты бөлек InPut( ) әдісі ішінде жазыңыз.

InPut( )–әдісі параметр қабылдамайды және егер қолданушы дұрыс шарттарды еңгізсе true дұрыс емес еңгізсе false қайтаратын болуы тиіс.

Сонымен қатар әдісі InPut( ) (глобалды n элементке арналған x және y массивті) инициализация жасауы тиіс.

Қолданушы дұрыс шарттарды еңгізгенге дейін Main әдісі ішінде InPut( ) әдісін қайта қайта шақыруы тиіс.

№7.3

Метод NextPoint

ҚДТ шешетін алгоритмді NextPoint деген бөлек әдіс ішіне шығарыңыз.

Бұл әдіс үш параметрді: алдынғы нүкте және сол нүктедегі функция мәні, қадам h қабылдап, келесі нүктедегі функцияның мәнің есептеп қайтаруы тиіс.

№7.4

Метод Computation( )

Computation( ) деген әдіс параметр қабылдамайды және ештене қайтармайды. Бұл әдіс берілген аралықтағы нүктелер үшін функция мәнің есептеп табуы тиіс яғни КДТ шешу керек x және y массивтерін шешімдермен толтырып шығады. Және бұл әдіс қате туындаған жағдайда "Ошибка при вычислений" деген мәліметті консолға шығарып, әдістен шығып кетуі керек.

№7.5

Метод Show( )

Show( ) әдісі параметр қабылдамайды және нәтиже қайтармайды. Бұл әдіс x және y массивтерінің мәндерін консолға шығару тиіс.

№7.6

Метод Write( )

Show( ) әдісі параметр қабылдамайды және нәтиже қайтармайды. Бұл әдіс программаны қолданушыдан шешімді файлға жазу керек пе деп сұрауы тиіс. Егер қолданушы N деген символды еңгізсе әдіс тоқтайды. Y деген символды еңгізсе шешімді файлға екі баған түрінде (x y) жұмыс үстеліндегі Lab7.txt. деген файлдың ішіне жазады. Және бұл әдіс қате туындаған жағдайда " Ошибка при записи в файл!" деген мәліметті консолға шығарып, әдістен шығып кетуі керек.

Таким образом, мы должны получить программу метод Main которого выглядит след образом:

Монымен біздің Main әдісіміз мынадай болуы тиіс.

```
public static void Main(string[] args)
{
   Console.WriteLine("Программа запущена:");

   while ( !InPut( ) ) { /*еще раз!*/ }

   Computation();
   Show();
   Write();

   Console.WriteLine("Программа завершена!");
   Console.ReadLine();
}//Main
```