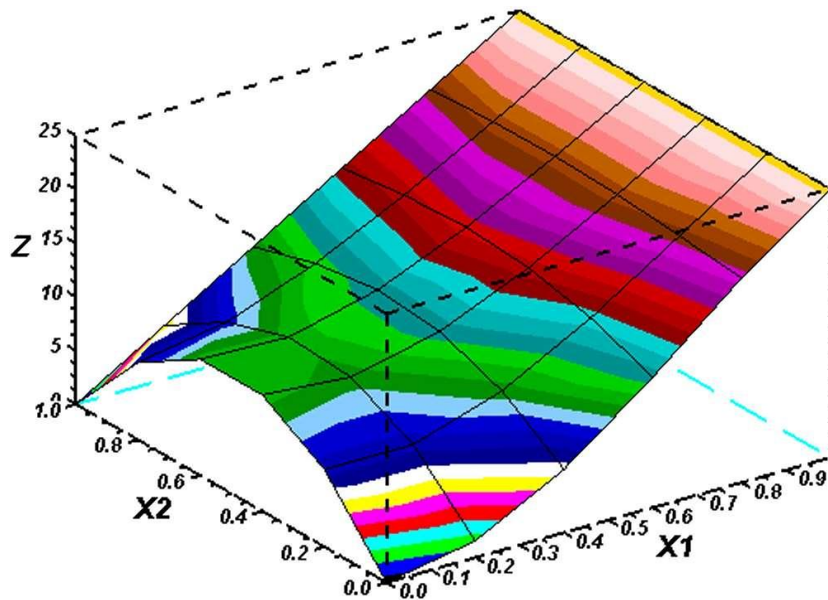


В. Г. Мусіяка

**РОЗВ'ЯЗАННЯ ЗАДАЧ МЕХАНІКИ
ЧИСЛОВИМИ МЕТОДАМИ
У ПАКЕТІ SCILAB**



2012

**Міністерство освіти і науки, молоді та спорту України
Дніпропетровський національний університет
ім. Олеся Гончара**

В. Г. Мусіяка

**РОЗВ'ЯЗАННЯ ЗАДАЧ МЕХАНІКИ
ЧИСЛОВИМИ МЕТОДАМИ
У ПАКЕТІ SCILAB**

Ухвалено вченою радою університету як посібник

**Дніпропетровськ
РВВ ДНУ
2012**

УДК 519.6(075.8)
М91

Рецензенти: д-р фіз.-мат. наук, проф. В.І. Кузьменко
д-р техн. наук, проф. В. Л. Красовський

М91 **Мусіяка, В. Г.** Розв'язання задач механіки числовими методами у пакеті Scilab [Текст]: посібник/ В.Г. Мусіяка — Д.: РВВ ДНУ, 2012. — 64 с.

Стисло викладено деякі числові методи, застосовані в процесі розв'язання математичних моделей, покладених в основу задач механіки.

Наведено, крім того, завдання для лабораторних робіт з числових методів та навчальні Scilab-програми.

Для студентів природничо-інженерних спеціальностей ДНУ, а також охочих ознайомитися з основами числових методів та методикою розв'язання математичних моделей.

Темплан 2012, поз. 29

Навчальне видання

Вольєн Григорович Мусіяка

Розв'язання задач механіки числовими методами у пакеті Scilab

Посібник

Редактор В. О. Насєкан
Технічний редактор Л. П. Замятіна
Коректор А. А. Гриженко

Підписано до друку 03.01.2012. Формат 60 × 84/16. Папір друкарський. Друк плоский.
Ум. друк. арк. 3,72. Ум. фарбовідб. 3,72. Обл.-вид. арк. 3,64. Тираж 150 пр. Зам. №

РВВ ДНУ, вул. Наукова, 13, м. Дніпропетровськ, 49050
Друкарня ДНУ, вул. Наукова, 5, м. Дніпропетровськ, 49050

© Мусіяка В.Г., 2012

Вступ

Одним із способів розв'язання задач механіки, що мають місце в розрахунковій та конструкторській практиці, є аналіз конструкції або явища із застосуванням *математичних методів*.

Більшість модельних задач, які таким чином доводиться розв'язувати, можна сформулювати так:

$$A(u, x) = 0, \quad (1)$$

де u — невідомий розв'язок, а величина x та функція (оператор) A — відомі.

Зауважимо, що тут u і x можуть бути числами, їх масивами або функціями однієї чи багатьох змінних із деякого класу.

Під час розв'язування (1) числовими методами A , u , x замінюють відповідно на B , y , ξ , які є зручніші для обчислень (іноді досить замінити одну з них), унаслідок чого приходимо до задачі

$$B(y, \xi) = 0. \quad (2)$$

Природно, заміна має бути такою, щоб розв'язки (1) і (2) були близькими, тобто $u \approx y$.

Оскільки числові методи практично завжди передбачають користування обчислювальною технікою, то перед будь-яким дослідником стоїть проблема вибору програмного забезпечення і, зазвичай, переважно у вигляді сучасного вільно поширюваного пакета. Саме таким є пакет наукових програм Scilab з потужним відкритим середовищем для інженерних і наукових розрахунків. З 1994 року він поширюється через Інтернет у вигляді відкритого початкового коду. Тепер Scilab підтримує компанія Scilab Consortium, створена в 2003 р. Вона налічує 25 учасників, зокрема Mandriva, INRIA та ENPC (Франція). Scilab має схожу з Matlab мову програмування і має у своєму складі утиліту, що дозволяє конвертувати документи Matlab \rightarrow Scilab.

Scilab дозволяє працювати з елементарними функціями і великим числом спеціальних функцій, має потужні засоби роботи з матрицями, поліномами (у тому числі і символьними), проводити числові розрахунки і розв'язання задач лінійної алгебри, оптимізації і симуляції, могутні статистичні функції, а також засіб для побудови і роботи з графіками. Для обчислень послуговуються бібліотеками Lapack, Linpack, Odepack, Atlas та ін. Крім того, до складу пакета також входить Scicos — інструмент для редагування блокових діаграм і симуляції (аналог simulink в пакеті — Matlab). Є можливість спільної роботи Scilab з програмою LabView. Програма доступна для різних операційних систем, у тому числі й GNU/Linux та MS Windows. Є можливість розширити пакет зовнішніми програмами і модулями, написаними різними мовами програмування, зокрема C, C++, Fortran ...

Вищенаведене й зумовило вибір Scilab, а саме його україномовну версію Scilab-5.2.2, для розв'язання числовими методами задач механіки.

1. Наближення функцій

Наближення функцій полягає в заміні аналітично або таблично заданої функції $y(x)$ на зручнішу для обчислень апроксимувальну функцію $\varphi(x)$, яка б для необхідних нам значень аргументу задовольняла співвідношення $y(x) \approx \varphi(x)$. Цього досягають введенням вектора вільних параметрів $C = (C_1, C_2, \dots, C_n)$ до $\varphi(x)$ [тобто $\varphi(x) \equiv \varphi(x; C)$], який визначають із умови близькості функцій $y(x)$ та $\varphi(x)$. Зазвичай при цьому вважають, що графіком таблично заданої функції є плавна крива.

Узявши за умову близькості $\varphi(x)$ і $y(x)$ їх збіжність в n точках (вузлах інтерполяції), можна одержати **інтерполяційний поліном Ньютона (ІПН)**:

$$y(x) \approx P_n(x) = y(x_0) + \sum_{k=1}^n y(x_0, x_1, \dots, x_k) \prod_{i=0}^{k-1} (x - x_i).$$

$$\text{Тут } y(x_i, x_j) = \frac{y(x_i) - y(x_j)}{x_i - x_j}; \quad y(x_i, x_j, x_k) = \frac{y(x_i, x_j) - y(x_j, x_k)}{x_i - x_k}, \dots$$

Приклад 1.1. Застосовуючи табличні значення $y(0)=0$, $y(1)=1/2$, $y(3)=1$ функції $y(x) = \sin \pi x / 6$, знайдемо її значення у деяких точках відрізка $[0, 3]$ за допомогою ІПН.

Лістинг 1.1. Файл `ipn`:

```
// ipn
// Побудова ІПН на нерівномірній сітці
function [f] = ff(x); f = sin(%pi*x/6) endfunction;
// Задана функція
function [pln] = plnwt(n, xp, xx, yy);
//Функція plnwt обчислення ІПН у точці xp
y=yy(n+1);
for k=1:n
    n1=n-k+1;    y=y*(n1)+y*(xp-xx(n1));
end;
pln=y
endfunction;
// Функція ipn формування значень ІПН у точках xd
function [Pn, y, x] = ipn(n, M, xx, yy, xd);
// n - степінь ІПН, M - число точок у масиві xd
// Знаходження поділених різниць
m=n+1;
for k=1:m
    nt=n-k+1;
    for i=1:nt
        n1=n-i+1; n2=n1+1;
        yy(n2)=(yy(n2)-yy(n1))/(xx(n2)-xx(n1-k+1));
```

```

end
end;
// Обчислення ІПН у точках xd
for k=1:M
    xp=xd(k); p(k)=plnwt(n,xp,xx,yy); x(k)=xp; y(k)=ff(xp)
end;
// Перетворення векторів на матриці-рядки
x=x.'; y=y.'; Pn=p.';
endfunction;
// main program
// Вихідні дані для побудови ІПН
xx=[0,1.0,3]; yy=[0,0.5,1];
// Координати точок для таблиць і графіків
xd=[0,0.5,1,1.5,2.,2.5,3];
// Коментар до фактичних параметрів функції ipn
//[Pn,y,x] = ipn(n,M,xx,yy,xd)
[Pn,y,x] = ipn(2,7,xx,yy,xd)
// Побудова графіків заданої функції та ІПН
plot(x,Pn,'k*',x,y,'k-'); xgrid();
xtitle(' Pn=P(X)      Y=Y(X) ','X','Pn      Y');
legend(' Pn=P(X) ',' Y=Y(X) ', 4, %t);

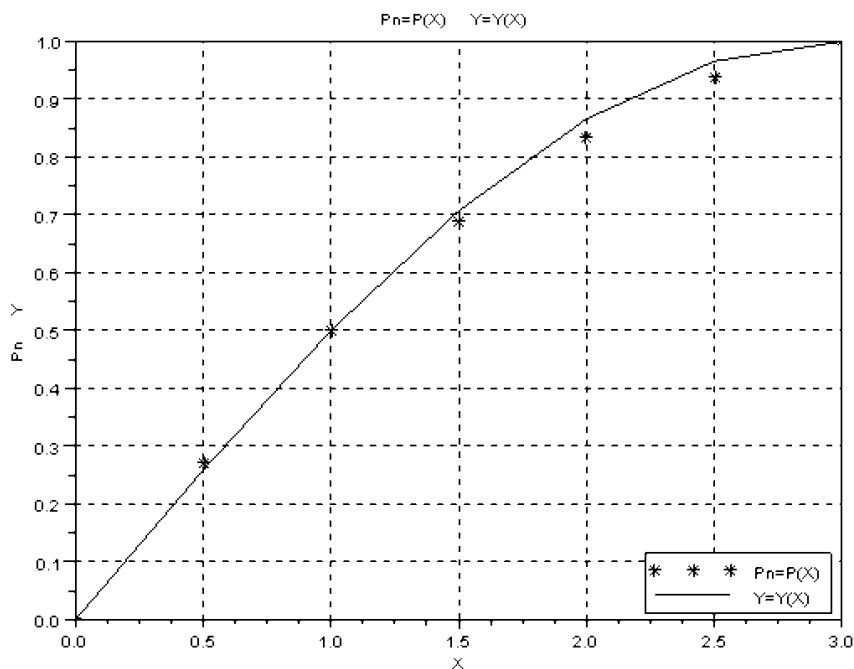
```

Результати виконання *ipn*

у табличному вигляді:

x	= 0.	0.5	1.	1.5	2.	2.5	3.
y	= 0.	0.2588190	0.5	0.7071068	0.8660254	0.9659258	1.
Pn	= 0.	0.2708333	0.5	0.6875	0.8333333	0.9375	1.

графічному вигляді:



В інженерній практиці для побудови графіка функції $y(x_i)$ ($i = \overline{0, N}$), якщо точки розташовані рідко, здебільшого беруть гнучку лінійку (spline), ставлять її на ребро і згинають так, щоб вона одночасно проходила через усі точки. Оскільки наближене рівняння згину пружного бруса має вигляд $y^{IV}(x) = 0$, то можна припустити, що її форма між вузлами є алгебричним поліномом 3-го степеня. Тому, очевидно, інтерполювальну функцію між кожними двома вузлами можна взяти, наприклад, у такому вигляді (**кубічний сплайн**):

$$S^{(i)}(x) = a_i + b_i(x - x_{i-1}) + c_i(x - x_{i-1})^2 + d_i(x - x_{i-1})^3 \\ [x_{i-1} \leq x \leq x_i; i = \overline{1, N}].$$

Невідомі коефіцієнти у цьому виразі a_i , b_i , c_i , d_i знаходять із умов у вузлах інтерполяції, зокрема c_i визначають із такої системи лінійних алгебричних рівнянь (СЛАР):

$$h_{i-1}c_{i-1} + 2(h_{i-1} + h_i)c_i + h_ic_{i+1} = \\ = 3 \left[\frac{y_i - y_{i-1}}{h_i} - \frac{y_{i-1} - y_{i-2}}{h_{i-1}} \right] (i = \overline{2, N}); \quad c_1 = c_{N+1} = 0. \quad (1.1)$$

Для розв'язання системи лінійних алгебричних рівнянь (1.1) (знаходження c_i) варто скористатися алгебричною прогонкою, після чого a_i , b_i та d_i визначити такими виразами, якщо кінці лінійки вільно відпущені (вільні кубічні сплайни – ВКС):

$$a_i = y_{i-1} (i = \overline{1, N}); \quad d_N = -\frac{c_N}{3h_N}, \quad d_i = \frac{c_{i+1} - c_i}{3h_i} (i = \overline{1, N-1}); \\ b_N = \frac{y_N - y_{N-1}}{h_N} - \frac{2h_N c_N}{3}; \quad b_i = \frac{y_i - y_{i-1}}{h_i} - \frac{h_i(c_{i+1} + 2c_i)}{3} (i = \overline{1, N-1}); \\ c_1 = c_{N+1} = 0.$$

Приклад 1.2. Скориставшись табличними значеннями $y(0)=0$, $y(1)=0.5$, $y(3)=1$, $y(2)=\sqrt{3}/2$ функції $y(x) = \sin \pi x / 6$, знайдемо її значення у деяких точках відрізка $[0, 1]$ за допомогою вільних кубічних сплайнів.

Лістинг 1.2. Файл `splncub`:

```
// splncub
// Побудова ВКС на нерівномірній сітці
function [f] = ff(x); f = sin(%pi*x/6) endfunction;
// Задана функція
function [ac] = spln3(n, na, xx, yy);
// Функція spln3 обчислення коефіцієнтів ВКС
c(1) = 0; aa(1) = 0; na1 = na + 1; n1 = n + 1; c(n1) = 0; m = n - 1;
for i = 1:n
    h(i) = xx(i+1) - xx(i)
```

```

end;
for i=2:n
    aa(i-1)=h(i-1); bb(i-1)=-2*(h(i-1)+h(i));
    cc(i-1)=h(i); ayp = yу(i+1); ay0 = yу(i);
    aym = yу(i-1);
    dd(i-1)=((ayp-ay0)/h(i)-(ay0-aym)/h(i-1))*3
end;
// Обчислення коефіцієнтів c ВКС
[z] = algprog(na1,aa,bb,cc,dd);
// Формування масиву ac(n,4) коефіцієнтів ВКС
for i=1:m
    c(i+1)=z(i)
end;
for i=1:n
// ac(i,1) - коефіцієнти a ВКС
    ac(i,1)=yу(i); ayp = yу(i+1); ay0 = yу(i);
// ac(i,2) - коефіцієнти b ВКС
    ac(i,2)=(ayp-ay0)/h(i)-...
                h(i)*(c(i+1)+c(i)*2)/3
// ac(i,3) - коефіцієнти c ВКС
    ac(i,3)=c(i)
// ac(i,4) - коефіцієнти d ВКС
    ac(i,4)=(c(i+1)-c(i))/(3*h(i))
end;
endfunction;
// Розв'язання СЛАР алгебричною прогонкою
function [x] = algprog(n1,a,b,c,d);
n=n1-1; k(1)=0; h(1)=0; c(n)=0;
a(1)=0; k(n1)=0; x(n1)=0;
for i=1:n
    ai =a(i);
    zn =b(i)-ai*k(i);
    k(i+1)=c(i)/zn;
    h(i+1)=(ai*h(i)-d(i))/zn
end;
j=n1;
for i=1:n
    j =j-1; x(j)=k(j+1)*x(j+1)+h(j+1)
end;
endfunction;
function [s,y,xd] = splncub(n,M,N,a,b,xd);
// Функція splncub формування ВКС та у(x) у точках xd
h=(b-a)/(N-1);
// Знаходження заданої функції у вузлах інтерполяції
for i=1:N

```



```

        xx(i)=a+(i-1)*h; yy(i)=ff(xx(i));
    end;
// Обчислення ВКС у вузлах інтерполяції
    na=n-1;
// Обчислення коефіцієнтів ВКС за допомогою spln3
    [ac] = spln3(n,na,xx,yy);
    for j=1:M
        xj=xd(j); x(j)=xj;
// Установлення підвдрізка розташування точки xj
        for i=1:n
            if(xj>=xx(i)&xj<xx(i+1)) then
                k=i;
                break;
            end
        end;
// Обчислення ВКС у точці xj
        ss=ac(k,4); t=xj-xx(k);
        for i=1:3
            ss=ss*t+ac(k,4-i)
        end;
// Формування значень ВКС та у(x) у точках масиву xd
        s(j)=ss;
        y(j)=ff(xj);
    end;
// Перетворення векторів на матриці-рядки
y=y.'; s=s.';
endfunction;
// main program
// Координат точок для таблиць і графіків
xd=[0,0.5,1,1.5,2.,2.5,3];
// Коментар до фактичних параметрів функції splncub
// [s,y,xd] = splncub(n,M,N,a,b,xd);
    [s,y,xd] = splncub(3,7,4,0,3,xd)
// Побудова графіків заданої функції та ВКС
x=xd.';
plot(x,s,'k+',x,y,'k-');
xgrid();
xtitle(' S=S(X)      Y=Y(X) ', 'X', 'S      Y');
legend(' S=S(X) ', ' Y=Y(X) ',4,%t);

```

Результати виконання *splncub*

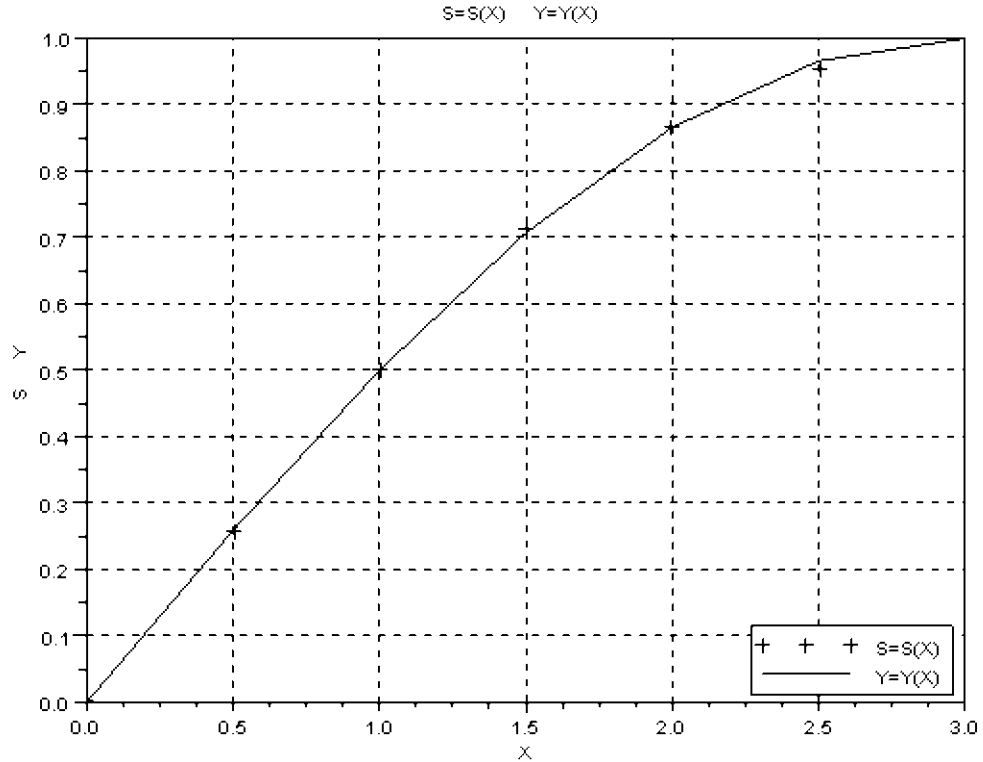
у табличному вигляді:

```

x = 0. 0.5      1.  1.5      2.      2.5      3.
y = 0. 0.2588190 0.5 0.7071068 0.8660254 0.9659258 1.
s = 0. 0.2575962 0.5 0.7104646 0.8660254 0.9528684 1.

```

графічному вигляді:



Оскільки ПН забезпечує зручну апроксимацію лише на невеликому відрізку, тобто на декількох вузлах інтерполяції, то якщо для знаходження ВВП потрібно мати єдину апроксимувальну функцію $\varphi(x;C)$ на значному відрізку або ж коли значення заданої функції $y(x)$ знайдено неточно, вдаються, зокрема, до **дискретного середнього квадратичного наближення (ДСКН)** або **методу найменших квадратів (МНК)**, в якому ВВП C визначають із такої умови:

$$I(C_1, \dots, C_n) \equiv \delta^2 \sum_{i=1}^N \rho_i = \sum_{i=1}^N \rho_i \left[y(x_i) - \sum_{k=1}^n C_k \varphi_k(x_i) \right]^2 = \min. \quad (1.2)$$

Прирівнявши до нуля частинні похідні від $I(C_1, \dots, C_n)$ за всіма складовими C_k , дійдемо такої системи лінійних алгебричних рівнянь для їх визначення:

$$\sum_{k=1}^n a_{mk} C_k = b_m \quad (m = \overline{1, n}), \quad (1.3)$$

де $a_{mk} = (\varphi_m, \varphi_k)$, $b = (y, \varphi_k)$ $[(f, g) = \sum_{i=1}^N \rho_i f(x_i) g(x_i)]$.

За умови лінійної незалежності $\{\varphi_k\}$ визначник (1.3) ненульовий, тобто дискретне середнє квадратичне наближення існує і воно єдине.

Оптимальне число коефіцієнтів апроксимації $n_{\text{опт}}$ визначають, беручи $n=1$, коли, напевне, дискретне середнє квадратичне відхилення (ДСКВ)

$$\delta_n = \sqrt{\sum_{i=1}^N \rho_i [y_i - \varphi(x_i)]^2 / \sum_{i=1}^N \rho_i} \quad (1.4)$$

δ_1 суттєво більша ε , і збільшують число коефіцієнтів n доти, доки не почне виконуватись умова $\delta_n \approx \varepsilon$. Якщо n суттєво менше N , то вираз $\varphi(x)$ вибрано вдало. Якщо ж $n_{\text{опт}} \approx N$, то апроксимувальну функцію $\varphi(x)$ потрібно взяти в іншому вигляді.

Приклад 1.3. Знайдемо коефіцієнти квадратичної апроксимувальної функції $\varphi(x) = C_1 + C_2x + C_3x^2$ для функції $y(x) = \sin \pi x / 6$ за допомогою МНК, застосовуючи вузли $x_i = (i-1)h$ $[i = \overline{1,14}; h = 1/13]$ відрізка $[0, 3]$, та обчислимо її наближені значення у деяких точках та ДСКВ.

Лістинг 1.3. Файл *mnk*:

```
// mnk
// Побудова апроксимувальної функції fi МНК та ДСКВ
function [f] = ff(x); f = sin(%pi*x/6) endfunction;
// Задана функція
function [dskv,C,y,fi,x] = mnk(n,N,M,a,b);
h=(b-a)/(N-1);
m=n+1;
// Обчислення вузлів та значень у(x) для МНК
for i=1:N
    xx(i)=a+(i-1)*h; yу(i)=ff(xx(i));
end;
// Формування проміжних масивів для побудови А і В
n2=n*2; n1=n+1;
for j=1:n2
    s=0;
    for i=1:N
        s=s+xx(i)^j;
    end;
    p(j)=s
end;
for i=1:n1
    j1=1+(i-1)*n1; j2=i*n1;
    k=i-j1-1;
    for j=j1:j2
        if (j==1) then
            d(1)=N
        end;
        if (~j==1) then
            d(j)=p(j+k)
        end;
    end;
end;
end;
```

```

// Формування вектора ПЧ В СЛАР (1.3)
s=0;
for i=1:N
    s=s+yy(i)
end;
B(1)=s;
for i=2:n1
    s=0;
    for j=1:N
        s=s+yy(j)*xx(j)^(i-1)
    end;
    B(i)=s
end;
// Формування матриці А СЛАР (1.3)
for i=1:m
    for j=1:m
        A(i,j)=d(i+m*(j-1));
    end;
end;
// Розв'язання СЛАР (1.3) методом Гаусса
CC=rref([A B]); [NN,MM]=size(CC); C=CC(:,MM);
// Обчислення ДСКВ
dskv=0;
for i=1:N
    xi=xx(i); fi =C(1);
    for j=1:n
        fi=fi+C(j+1)*xi^j;
    end;
    dskv=dskv+(fi-yy(i))^2;
end;
dskv=sqrt(dskv/N);
// Обчислення значень у(х) та АФ fi
h=(b-a)/(M-1);
for i=1:M
    x(i)=a+(i-1)*h;
    xi=x(i);
    ffi=C(1);
    for j=1:n
        ffi=ffi+C(j+1)*xi^j;
    end;
    fi(i)=ffi;
    y(i)=ff(xi);
end;
// Перетворення векторів на матриці-рядки
x=x.'; fi=fi.'; y=y.'; C=C.';

```

```

endfunction;
// main program
// Коментар до фактичних параметрів функції mnk
//[dskv,C,y,fi,x] = mnk(n, N,M,a,b)
[dskv,C,y,fi,x] = mnk(2,14,7,0,3)
// Побудова графіків заданої функції та АФ
plot(x,fi,'k.',x,y,'k-'); xgrid();
xtitle(' FI=FI(X)    Y=Y(X) ','X','FI    Y');
legend(' FI=FI(X) ', ' Y=Y(X) ',4,%t);

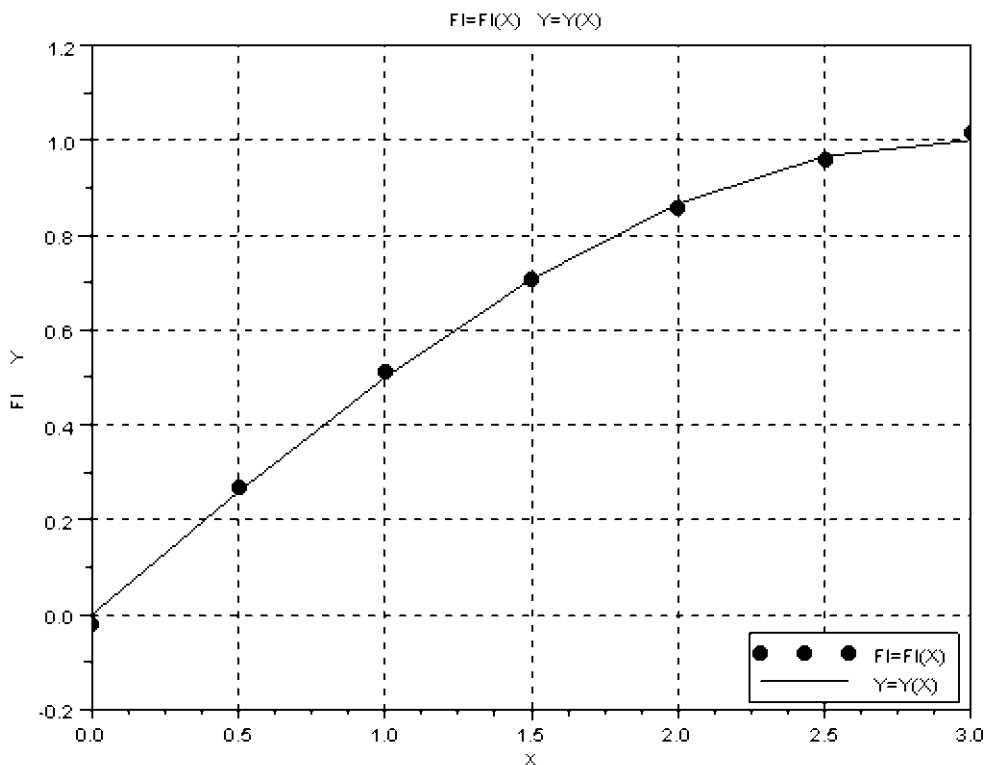
```

Результати виконання *mnk*

у табличному (за збереження чотирьох значущих цифр за десятичною крапкою) вигляді:

x =	0.	0.5	1.	1.5	2.	2.5	3.
fi=	-0.0186	0.2690	0.5105	0.7059	0.8552	0.9584	1.0156
y =	0.	0.2588	0.5	0.7071	0.8660	0.9659	1.
C =	-0.0186	0.6212	- .0922	dskv = 0.0101			

графічному вигляді:



2. Числове інтегрування

У практичних розрахунках, у тому числі й у задачах механіки, нерідко виникає потреба в обчисленні визначених інтегралів:

$$F = \int_a^b \rho(x)f(x)dx \quad (\rho(x) > 0), \quad (2.1)$$

де функція $f(x)$ неперервна на відрізку $[a, b]$, а вагова функція $\rho(x)$ неперервна на інтервалі (a, b) .

За числового інтегрування (ЧІ) найчастіше підінтегральну функцію $f(x)$ замінюють на алгебричний поліном, причому за умови $\rho(x) \equiv 1$.

Узявши на відрізку $[a, b]$ єдиний вузол у квадратурній формулі (КФ) $\bar{x} = (a+b)/2$, тобто апроксимуючи підінтегральну функцію $f(x)$ сталою $f(\bar{x})$, матимемо КФ **середніх (ФС)** $F \approx (b-a)f(\bar{x})$.

Природно, що точність ФС для довільної $f(x)$ можна підвищити, скориставшись детальнішою сіткою x_i . Якщо $x_i - x_{i-1} = h = \text{const}$ (рівномірна сітка), матимемо

$$F_h^C \approx h \sum_{i=1}^N f_{i-1/2}; \quad f_{i-1/2} = f\left(x_i - \frac{h}{2}\right); \quad (2.2)$$

$$R \approx \frac{h^2}{24} \int_a^b f''(x)dx \approx \frac{h^3}{24} \sum_{i=1}^N f''(\bar{x}_i) \approx O(h^2). \quad (2.3)$$

Наведені оцінки R справедливі, якщо неперервна $f''(x)$.

КФ **трапецій (ФТ)** одержують, замінивши функцію $f(x)$ на відрізку $[a, b]$ ПІ першого степеня з вузлами $x = a, x = b$, що відповідає заміні кривої $f(x)$ на січну.

На рівномірній сітці x_i ($i = 0, N$) вона стає такою:

$$F_h^T \approx h \left(\frac{f_0 + f_N}{2} + \sum_{i=1}^{N-1} f_i \right). \quad (2.4)$$

Узагальнена КФ парабол (ФП) для рівномірної сітки має вигляд ($N = 2K$)

$$F_h^{\text{II}} \approx \frac{b-a}{6K} \left(f_0 + f_{2K} + 4 \sum_{i=1}^K f_{2i-1} + 2 \sum_{i=1}^{K-1} f_{2i} \right). \quad (2.5)$$

Отже, ФП дає добру точність за відносно невеликого числа вузлів, якщо $f^{IV}(x)$ не дуже велика.

Для підвищення точності наведених КФ можна застосувати *правило Рунге*, якщо точне значення інтеграла F пов'язане з наближеним F_h , знайденим на сітці з кроком h :

$$F = F_h + h^p C + O(h^{p+2}).$$

Для формул середніх (трапецій) і парабол відповідно $p = 2$ і $p = 4$.

Перша формула Рунге для інтегрування — *оцінка похибки ЧІ*:

$$R \approx h^p C = \frac{F_h - F_{gh}}{g^p - 1} + O(h^{p+2}). \quad (2.6)$$

Тут F_{gh} — наближене значення F , знайдене на сітці з кроком gh .

Уточнене значення інтеграла знаходять за другою формулою Рунге:

$$\tilde{F} \approx F_h + \frac{F_h - F_{gh}}{g^p - 1}. \quad (2.7)$$

Приклад 2.1. Скориставшись значенням підінтегральної функції $\ln(1+x)$ на відрізьку $[0, 1]$ $f(0)=0$, знайдемо за КФ середніх, трапецій та парабол наближені

значення інтеграла $F = \int_0^1 \ln(1+x) dx = 0.3863$.

Лістинг 2.1. Файл *qfpr*:

```
// qfpr
//Підінтегральна функція (ПФ)
function [f] =ff(x);
f = log(1+x)
endfunction;
// Обчислення інтеграла за різними КФ
function [trap,ser,par] = trapserpar(a,b,N);
h=(b-a)/N; // Крок сітки
// Формування значень ПФ для КФ середніх
for i=1:N
    xi=a+(i-0.5)*h; y(i)=ff(xi)
end;
// КФ середніх
ser=0;
for i=1:N
    ser=ser+y(i)
end;
ser=ser*h;
// Формування значень ПФ для КФ трапецій та парабол
N1=N+1;
for i=1:N1
    xi=a+(i-1)*h; y(i)=ff(xi)
end;
// КФ трапецій
trap=(y(1)+y(N1))/2;
for i=2:N
    trap=trap+y(i)
end;
trap=trap*h;
// КФ парабол
K=N/2; s1=y(1)+y(N1); s2=0; s4=0;
for i=2:K
    s2=s2+y(i*2-1)
end;
for i=1:K
    s4=s4+y(i*2)
```

```

end;
par=(s1+s2*2+s4*4)*h/3
endfunction;
// Уточнення КФ за правилом Рунге
function [Ftrap,Fser,Fpar] = qfpr(a,b,N);
[trap1,ser1,par1] = trapserpar(a,b,N);
[trap2,ser2,par2] = trapserpar(a,b,N*2);
Ftrap=trap2+(trap2-trap1)/3; Fser=ser2+(ser2-ser1)/3;
Fpar=par2+(par2-par1)/15
endfunction;
// main program
// N - парне
// Точне значення інтеграла F = 0.3862944
// Коментар фактичних параметрів функції qfpr
//[Ftrap,Fser,Fpar] = qfpr(a,b,N);
[Ftrap,Fser,Fpar] = qfpr(0,1,4)

```

Результати виконання *qfpr*

Fpar = 0.3862942 Fser = 0.3862964 Ftrap = 0.3862920

3. Нелінійні алгебричні рівняння

Досить часто у прикладних дослідженнях і механіці зокрема виникає потреба у розв'язанні нелінійних алгебричних рівнянь або їх систем, причому це може бути як самостійною задачею, так і допоміжною у складнішій задачі.

Розглянемо *рівняння з одним невідомим*

$$f(x) = 0, \quad (3.1)$$

для якого потрібно знайти всі або лише окремі корені, де $f(x)$ — неперервна функція. Якщо $f(x) \equiv 0$, то x називають *коренем рівняння* (3.1) або нулем функції $f(x)$. Здебільшого корені (3.1) можна знаходити лише ітераційними методами. Корені (3.1) вважають *відокремленими*, якщо для кожного з них існує окіл, де немає інших коренів цього рівняння.

Удаючись до *методу дотичних* (МД), здійснюємо знаходження кореня x для двічі неперервно диференційованої функції $f(x)$ на досліджуваному проміжку за наявності деякого наближення $x^{(s)}$ до нього здійснюють за допомогою такого ітераційного процесу:

$$x^{(s+1)} = x^{(s)} - f(x^{(s)}) / f'(x^{(s)}) \quad (s=0,1,\dots). \quad (3.2)$$

Геометрично це заміна графіка $y = f(x)$ на кожній ітерації дотичною.

У разі довільного $x^{(0)}$, якщо всюди $|ff''| < f'^2$, ітерації вважають збіжними.

Ітерації збігаються до x з кінця відрізка, де виконується умова

$$f(x^{(\text{гран})})f''(x^{(\text{гран})}) \geq 0. \quad (3.3)$$

Похибка кожного наступного наближення приблизно дорівнює квадрату похибки попереднього наближення, що забезпечує швидку збіжність до кореня. У разі числового задання рівняння (3.1) похідну у знаменнику (3.2) замінюють на першу поділену різницю, відтак уточнення виконують за формулою (метод січних)

$$x^{(s+1)} = x^{(s)} - \frac{[x^{(s)} - x^{(s-1)}]f(x^{(s)})}{f(x^{(s)}) - f(x^{(s-1)})} \quad (s = 1, 2, \dots). \quad (3.4)$$

Для початку процесу вибирають $x^{(0)}$ і $x^{(1)}$ наближення, тому цей метод називають двокроковим.

Приклад 3.1. Знайдемо методом дотичних наближене значення кореня рівняння $f(x) \equiv x - \operatorname{tg}x = 0$ на проміжку $\left(\pi, \frac{3\pi}{2}\right)$. Його можна одержати переходом до безрозмірних величин в задачі визначення критичної сили $P_{\text{кр}}$ пружного стержня, верхній кінець якого шарнірно рухомий, а нижній — затиснутий. За нульове наближення візьмемо саме $x^{(0)} = 4.4$, бо в цій точці функція та її друга похідна — одного знака.

Лістинг 3.1. Файл *dotych*:

```
// dotych
// Розв'язання (3.1) методом дотичних (3.2)
function [s , x1] = dotych(x, eps , smax)
// x - нульове наближення, eps - точність обчислення
// кореня, smax - допустиме число ітерацій.
// s - число виконаних ітерацій, x1 - розв'язок
// Реалізація (3.2)
deff('сс = ff(x)', 'сс = x - sin(x)/cos(x)')
// Похідна від заданої f f (x)
deff('ср = fp(x)', 'ср = 1.0 - 1.0/cos(x)^2')
// Ітераційний процес (3,2)
for i = 1:smax
x1 = x - ff(x)/fp(x); d = abs(x1-x); x = x1;
if d <= eps then break
end;
end;
// Формування результату виконання dotych
s = i; x1 = x
endfunction;
// main program
// Коментар до фактичних параметрів функції dotych
//[s , x1] = dotych(x0,eps,smax)
[s , x1] = dotych(4.4,0.001,9)
```

Результати виконання *dotych*

$$x_1 = 4.49341 \quad s = 4.$$

Розглянемо *систему нелінійних рівнянь* (СНР) з n невідомими:

$$f_k(x_1, x_2, \dots, x_n) = 0 \quad (k = \overline{1, n}). \quad (3.6)$$

Її можна записати у векторному вигляді:

$$f(x) = 0, \quad (3.7)$$

де $x = (x_1, x_2, \dots, x_n)$ і $f = (f_1, f_2, \dots, f_n)$.

Розв'язання (3.6) — набагато складніша задача, ніж розв'язання одного рівняння. Такі системи, як правило, розв'язують лише ітераційними методами.

Якщо $n = 2$, то за нульове наближення $x^{(0)}$ до розв'язку \underline{x} можна взяти координати точок перетину графіків

$$f_1(x_1, x_2) = 0 \quad \text{і} \quad f_2(x_1, x_2) = 0.$$

До речі, якщо $f(x) = 0$, де $x = x_1 + ix_2$, замість такого рівняння можна розглядати систему нелінійних рівнянь:

$$f_1(x_1, x_2) = 0; \quad f_2(x_1, x_2) = 0, \quad (3.8)$$

де

$$f_1(x_1, x_2) = \operatorname{Re} f(x_1 + ix_2); \quad f_2(x_1, x_2) = \operatorname{Im} f(x_1 + ix_2).$$

Для трьох і більше невідомих задовільних способів знаходження нульових наближень немає, але іноді $x^{(0)}$ можна вибрати, виходячи з фізичних міркувань або аналізу задачі.

Удаючись до *методу Ньютона* (МН) знаходження розв'язку (3.6), якщо відомий будь-який наближений розв'язок $x^{(s)}$ до \underline{x} , здійснюють розв'язанням методом Гаусса такої СЛАР відносно приростів $\Delta x_i^{(s)}$:

$$\sum_{i=1}^n \frac{\partial f_k(x^{(s)})}{\partial x_i} \Delta x_i^{(s)} = -f_k(x^{(s)}) \quad (k = \overline{1, n}; s = 0, 1, \dots). \quad (3.9)$$

Тут $\Delta x^{(s)} = \underline{x} - x^{(s)}$ ($s = 0, 1, \dots$).

Розв'язавши СЛАР методом Гаусса (МГ), знайдемо наступні наближення:

$$x^{(s+1)} = x^{(s)} + \Delta x^{(s)} \quad (s = 0, 1, \dots). \quad (3.10)$$

За вдалого вибору $x^{(0)}$ розрахунки за методом Ньютона збігатимуться, і досить швидко (3 – 5 ітерацій).

На практиці ітерації припиняють за виконання умови

$$\sqrt{\frac{1}{n} \sum_{k=1}^n [x_k^{(s+1)} - x_k^{(s)}]^2} \leq \varepsilon. \quad (3.11)$$

Приклад 3.2. Знайдемо методом Ньютона наближене значення розв'язку СНР

$$x_1^2 + x_2^2 = 4,$$

$$e^{x_1} + x_2 = 1$$

за початкового наближення $x^{(0)}=(1.5, -1.5)$, яке є точкою перетину графіків $f_1(x_1, x_2)=0$ і $f_2(x_1, x_2)=0$.

Лістинг 3.2. Файл *newton*:

```
// newton
// Розв'язання СНР (3.7) методом Ньютона (3.9)
function [s , x1] = newton(x, eps , smax);
// x - нульове наближення, eps - точність визначення
// кореня, smax - допустиме число ітерацій.
// s - число виконаних ітерацій, x1 - розв'язок (3.6)
s=0;
while (s<=smax) then
s=s+1;
// Звернення до sdfd для обчислення складових СНР (3.9)
[ff ,dd] = sdfd(x);
//Модифікація ПЧ СНР (3.9)
ff(1)=-ff(1);
ff(2)=-ff(2);
// Розв'язання СЛАР (3.9) МГ на кожній ітерації
CC=rref([dd ff]); [NN,MM]=size(CC); dx =CC(:,MM);
// Уточнення розв'язку за (3.10)
x(1)=x(1)+dx(1);
x(2)=x(2)+dx(2);
// Перевірка виконання умови (3.11)
if(sqrt((dx(1)^2+dx(2)^2)/2)<=eps) then
break;
end
end;
x1=x
endfunction;
// Функція sdfd формування складових СНР (3.9)
function [ff ,dd] = sdfd(x);
x1=x(1); x2=x(2);
// Обчислення матриці СНР (3.9)
dd (1,1) = x1*2; dd(2,1) = exp(x1);
dd (1,2) = x2*2; dd(2,2) = 1;
// Обчислення ПЧ СНР (3.9)
ff(1)=x1*x1+x2*x2-4; ff(2)=exp(x1)+x2-1;
endfunction;
//main program
// Коментар до фактичних параметрів функції newton
//[s , x1] = newton( x, eps , smax)
[s , x1] = newton([1.5 -1.5], 0.001 , 9)
```

Результати виконання newton

$$x_1(1) = 1.0041687 \quad x_1(2) = -1.7296373 \quad s = 4.$$

4. Задачі лінійної алгебри

Розв'язання багатьох задач математики, механіки й інженерної практики зводиться до розв'язання задач лінійної алгебри. Відомо, що коли визначник системи лінійних алгебричних рівнянь

$$Ax = b \text{ або } \sum_{j=1}^n a_{ij}x_j = b \quad (i = \overline{1,n}) \quad (4.1)$$

дорівнює нулю $D = 0$, система лінійних алгебричних рівнянь або не має дійсних розв'язків, або має їх нескінченно багато. Якщо ж $D \neq 0$, то розв'язок існує й притому єдиний.

Для системи лінійних алгебричних рівнянь загального вигляду і $n < 200$ застосовують прості й універсальні прямі (точні) методи, наприклад *метод Гаусса*, а у випадку слабо заповнених матриць і досить великих n — ітераційні методи.

На першому етапі розв'язання рівнянь методом Гаусса початкову матрицю A перетворюють на верхню трикутну.

На k -му кроці ($k = \overline{1, n-1}$) формули визначення перетворених елементів A та складових b мають такий вигляд (*формули прямого ходу*):

$$\begin{aligned} a_{ij}^{(k)} &= a_{ij}^{(k-1)} - m_i^{(k-1)} a_{kj}^{(k-1)} \quad (i = \overline{k+1, n}; j = \overline{k, n}); \\ b_i^{(k)} &= b_i^{(k-1)} - m_i^{(k-1)} b_k^{(k-1)} \quad (i = \overline{k+1, n}); \\ \left[m_i^{(k-1)} &= \frac{a_{ik}^{(k-1)}}{a_{kk}^{(k-1)}} \quad (i = \overline{k+1, n}; a_{kk}^{(k-1)} \neq 0) \right]. \end{aligned} \quad (4.2)$$

Значення невідомих знаходять за формулами *зворотного ходу*:

$$\begin{aligned} x_n &= b_n^{(n-1)} / a_{nn}^{(n-1)}; \quad x_j = (b_j^{(j-1)} - \sum_{k=j+1}^n a_{jk}^{(j-1)} x_k) / a_{jj}^{(j-1)} \\ &\quad (j = \overline{n-1, 1}; a_{1k}^{(0)} = a_{1k}; b_1^{(0)} = b_1). \end{aligned}$$

Якщо $a_{kk}^{(k-1)} = 0$, то, переставляючи місцями рядки на головній діагоналі, можна розмістити ненульовий елемент (краще, щоб він був максимальним за абсолютною величиною у відповідному стовпчику; тоді цей метод називають *МГ з вилученням головного елемента*) і продовжити процес вилучення невідомих.

Приклад 4.1. Знайдемо за методом Гаусса в пакеті *Scilab* розв'язок такої системи лінійних алгебричних рівнянь:

$$\begin{pmatrix} 1 & 2 & -3 \\ 4 & -1 & 1 \\ 2 & 6 & -1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 4 \\ 7 \end{pmatrix}.$$

Лістинг 4.1. Файл *gauss*:

```
// gauss
A=[1 2 -3; 4 -1 1; 2 6 -1]; b=[ 0; 4; 7];
C=rref([A b]); [NN,MM]=size(C); x=C(:,MM); X=x.'
```

Результати виконання *gauss*

```
X = 1. 1. 1.
```

Примітка. Під час розв'язування прикладних задач за наявності потрібних функцій (підпрограм) у пакеті *Scilab* слід послуговуватися саме ними, оскільки вони побудовані на основі спеціальних швидких алгоритмів.

Метод алгебричної прогонки застосовують для розв'язання системи лінійних алгебричних рівнянь з тридіагональними матрицями

$$a_i x_{i-1} - b_i x_i + c_i x_{i+1} = d_i \quad (a_1 = c_n = 0; i = \overline{1, n}). \quad (4.3)$$

Після вилучення невідомих на нижній діагоналі одержимо формули *зворотного ходу*:

$$x_i = \xi_{i+1} x_{i+1} + \eta_{i+1} \quad (x_{n+1} = 0; i = \overline{n, 1}). \quad (4.4)$$

Коефіцієнти ξ_{i+1} і η_{i+1} визначаємо за формулами *прямого ходу*:

$$\xi_{i+1} = \frac{c_i}{b_i - a_i \xi_i}; \quad \eta_{i+1} = \frac{a_i \eta_i - d_i}{b_i - a_i \xi_i} \quad (\xi_1 = \eta_1 = x_{n+1} = c_n = 0; i = \overline{1, n}). \quad (4.5)$$

Узагалі, якщо хоч для одного рівняння справджується умова $|b_i| \geq |a_i| + |c_i|$, то розв'язок системи лінійних алгебричних рівнянь з тридіагональними матрицями існує, єдиний і стійкий до похибок округлення.

Приклад 4.2. Знайдемо методом алгебричної прогонки розв'язок такої системи лінійних алгебричних рівнянь з тридіагональною матрицею:

$$\begin{pmatrix} 2 & 1 & 0 \\ 1 & -1 & 1 \\ 0 & -1 & -2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 4 \\ 2 \\ -8 \end{pmatrix}.$$

Лістинг 4.2. Файл *algprog*:

```
// algprog
// Розв'язання СЛАР алгебричною прогонкою
function [xx] = algprog(n1, a, b, c, d);
// Зворотний хід
n=n1-1; k(1)=0; h(1)=0; c(n)=0; a(1)=0; k(n1)=0; x(n1)=0;
for i=1:n
```

```

    ai =a(i); zn =b(i)-ai*k(i);
    k(i+1)=c(i)/zn; h(i+1)=(ai*h(i)-d(i))/zn
end;
// Прямий хід
j=n1;
for i=1:n
    j = j-1; x(j)=k(j+1)*x(j+1)+h(j+1)
end;
for i=1:n
    xx(i)= x(i)
end;
// Перетворення вектора на матрицю-рядок
xx=xx.'
endfunction;
//main program
// Коефіцієнти СЛАР (4.3)
a=[ 0 1 -1]; b=[-2 1 2]; c=[ 1 1 0]; d=[ 4 2 -8]; n1=4;
// Виклик функції algprog
[xx] = algprog(n1,a,b,c,d)
Результати виконання algprog
xx =      1.      2.      3.

```

У разі великих n ефективнішими для розв'язування систем лінійних алгебричних рівнянь будуть *ітераційні* методи, зокрема **метод Гаусса—Зейделя (МГЗ)**. За цим методом на головній діагоналі розміщують максимальні за абсолютною величиною елементи *відповідних* стовпчиків вихідної матриці, а потім знаходять невідомі $x_1, x_2, x_3, \dots, x_n$ з першого, другого, третього, ..., n -го рівнянь системи лінійних алгебричних рівнянь (5.1), тобто

$$x_k = (b_k - \sum_{j \neq k} a_{kj} x_j) / a_{kk} \quad (k = \overline{1, n}).$$

Скориставшись цим записом, можна побудувати такий ітераційний процес:

$$x_k^{(s+1)} = \left(b_k - \sum_{j=1}^{k-1} a_{kj} x_j^{(s+1)} - \sum_{j=k+1}^n a_{kj} x_j^{(s)} \right) / a_{kk} \quad (k = \overline{1, n}; s = 0, 1, \dots). \quad (4.6)$$

Взявши за нульові наближення, наприклад, $x_k^{(0)} = 0$ ($k = \overline{1, n}$), наступні знайдемо за

формулами (5.6). Якщо $\left| \frac{a_{kj}}{a_{kk}} \right| < 1$, то процес (5.6) має бути збіжним.

Ітерації завершують після виконання умови $\delta = \sqrt{\frac{1}{n} \sum_{k=1}^n [x_k^{(s+1)} - x_k^{(s)}]^2} \leq \varepsilon$.

Приклад 4.3. Знайдемо за методом Гаусса-Зейделя наближене значення розв'язок системи лінійних алгебричних рівнянь

$$\begin{pmatrix} 4 & -1 & 1 \\ 2 & 6 & -1 \\ 1 & 2 & -3 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 4 \\ 7 \\ 0 \end{pmatrix}$$

за початкового наближення $x^{(0)} = (0, 0, 0)$.

Лістинг 4.3. Файл mgz:

```
// mgz
// Функція розв'язання СЛАР МГЗ за формулою (4.6)
function [x1,s]=mgz(n,a,b,smax,eps)
// n - порядок СЛАР, a - матриця A, b - вектор ПЧ,
// smax - допустиме число ітерацій, eps - точність.
// s - число виконаних ітерацій, x1 - розв'язок.
// Формування початкового наближення розв'язку
for k=1:n
    x0(k)=0.0
end;
// Початок ітераційного процесу (4.6)
s=0;
// Перевірка умови (s <= smax)
while (s <= smax)
    s=s+1
    for k=1:n
        km=k-1; kp=k+1; ss=0;
        if (km == 0) then
            for j=kp:n
                ss=ss+a(k,j)*x0(j)
            end
        end;
// Реалізація (4.6)
        if (km <> 0) then
            for j=1:km
                ss=ss+a(k,j)*x1(j)
            end
            for j=kp:n
                ss=ss+a(k,j)*x0(j)
            end
        end;
        x1(k)=(b(k)-ss)/a(k,k)
    end;
// Середнє квадратичне відхилення dlt розв'язку
    ss=0
for k=1:n
    ss=ss+(x0(k)-x1(k))**2
end;
```

```

dlt=sqrt(ss/n);
for k=1:n
    x0(k)=x1(k)
end;
// Перевірка умови збіжності процесу (4.6) (dlt <= eps)
if (dlt <= eps) then break
end
end
endfunction;
//main program
// Значення складових СЛАР (4.6)
b=[4. 7. 0.]; a=[4. -1. 1.; 2. 6. -1.; 1. 2. -3.];
// Коментар до фактичних параметрів функції mgz
//[x1,s]=mgz(n,a,b,smax,eps)
[x1,s]=mgz(3,a,b,20,0.00001)

```

Результати виконання mgz

s = 7. x(1) = 1. x(2) = 0.9999996 x(3) = 0.9999997

Визначення головних напружень тензора напружень та їх напрямків у лінійній теорії пружності, критичних зусиль — у стійкості лінійних пружних систем, власних частот — у теорії лінійних коливань та ін. приводить до **задачі на власні значення (ВЗ) та власні вектори (ВВ) для квадратної матриці**.

У цій задачі потрібно знайти n дійсних або комплексних чисел (ВЗ) λ і відповідних кожному з них ненульових векторів (ВВ) x , тобто

$$Ax = \lambda x \text{ або } (A - \lambda E)x = 0 \quad (x \neq 0), \quad (4.7)$$

де A — квадратна матриця n -го порядку. Для існування ненульового розв'язку однорідної системи лінійних алгебричних рівнянь (4.7) її визначник має перетворитися на нуль, тобто

$$D(\lambda) \equiv \det(A - \lambda E) = 0. \quad (4.8)$$

Визначник $D(\lambda)$ називають *характеристичним поліномом* (ХП), а алгебричне рівняння (4.8) — *характеристичним рівнянням* (ХР), корені якого слід знаходити методом парабол.

У загальному випадку $D(\lambda)$ є алгебричний поліном n -го степеня відносно λ і характеристичний поліном можна записати у такому вигляді:

$$(-1)^n [\lambda^n - p_1 \lambda^{n-1} + p_2 \lambda^{n-2} + \dots + (-1)^n p_n] = 0. \quad (4.8')$$

Коефіцієнти цього рівняння визначають за виразами

$$p_1 = \sum_{i=1}^n a_{ii}; \quad p_2 = \sum_{i < j} \begin{vmatrix} a_{ii} & a_{ij} \\ a_{ji} & a_{jj} \end{vmatrix}; \quad p_3 = \sum_{i < j < k} \begin{vmatrix} a_{ii} & a_{ij} & a_{ik} \\ a_{ji} & a_{jj} & a_{jk} \\ a_{ki} & a_{kj} & a_{kk} \end{vmatrix}, \dots, p_n = \det(A).$$

Досить просто знаходити всі власні значення, якщо матриця A трикутна або діагональна, оскільки для них характеристичне рівняння (4.8) набуває вигляду

$$D_{\Delta}(\lambda) \equiv \prod_{i=1}^n (a_{ii} - \lambda) = 0; \quad D_{\text{diag}}(\lambda) \equiv \prod_{i=1}^n (d_i - \lambda) = 0. \quad (4.8'')$$

Отже, можна записати, що $\lambda_i = a_{ii}$ або $\lambda_i = d_i$ ($i = \overline{1, n}$).

Оскільки у задачах механіки нерідко становить інтерес лише одне власне значення (наприклад, найбільше головне напруження), то його можна знайти за допомогою такого ітераційного процесу (**степеневий метод**):

$$y^{(s+1)} = Ay^{(s)} \quad (s = 0, 1, \dots), \quad (4.9)$$

де $y^{(0)}$ — довільний вектор.

За досить великого s можна записати $y^{(s)} = \lambda_1^s \xi_1(x^1 + \dots)$, тобто вектор $y^{(s)}$ збігається до власного вектора x^1 за напрямком.

Отже, маємо

$$y_i^{(s+1)} \approx \lambda_1 y_i^{(s)} \quad (i = \overline{1, n}) \quad \text{або} \quad \lambda_1 = \frac{\|y^{(s+1)}\|}{\|y^{(s)}\|}. \quad (4.10)$$

У практичних розрахунках ітерації припиняють за умови $|y_i^{(s+1)} - y_i^{(s)}| \leq \varepsilon$ або

$$\sqrt{\frac{1}{n} \sum_{k=1}^n [y_k^{(s+1)} - y_k^{(s)}]^2} \leq \varepsilon.$$

Запобігають переповненню під час знаходження λ_1 на ЕОМ діленням у на кожній ітерації його складових на норму.

Приклад 4.4. Визначимо степеневим методом головне напруження такого тензора напружень та його напрямки:

$$\begin{pmatrix} 20 & 2 & 6 \\ 2 & 10 & 4 \\ 6 & 4 & 5 \end{pmatrix}$$

за початкового вектора $y^{(0)} = (1, 0, 0)$.

Лістинг 4.4. Файл *stepmet*:

```
// stepmet
// Степеневий метод визначення ВЗ та ВВ на прикладі
// визначення головного напруження тензора напружень
// та його напрямків
// Добуток вектора на матрицю
function [v] = Ax (n, A, x);
for i=1:n
    summa=0;
    for j=1:n
        summa = summa +A(i, j)*x(j)
    end;
    v(i) = summa
end;
endfunction;
```

```

// Обчислення найбільшої за модулем компоненти вектора
// та його нормування
function [lm,x] = norma (n,x);
xmax=abs(x(1));
if(abs(x(2))>xmax) then
xmax=abs(x(2))
end;
if(abs(x(3))>xmax) then
xmax=abs(x(3))
end;
// Нормування компонент вектора
for i=1:n
x(i)=x(i)/xmax
end;
lm=xmax
endfunction;
// Реалізація степеневого методу
function [x,lam,s] = stepmet (n,x,A,eps,smax,lam1);
// n - порядок матриці A, x - початковий вектор,
// A,eps - точність, smax - допустиме число ітерацій,
// lam1 - початкове значення ВЗ. x - ВВ,lam - знайдене
// значення ВЗ,s - виконане число ітерацій
// Початок ітераційного процесу (4.10)
s=0;
// Перевірка умови (s <= smax)
while (s<=smax) then
    s=s+1;
    [v] = Ax (n,A,x);
    for j=1:n
        x(j)=v(j)
    end;
    [lam2,x] = norma (n,x);
    // Перевірка умови завершення ітерацій
    if(abs((lam1-lam2)/lam2)<=eps) then
        break;
    end;
    lam1=lam2;
end;
lam=lam2;
endfunction;
// main program
// Складові довільного вектора x та матриці A
x = [1, 0, 0]; A= [20, 2, 6; 2, 10, 4; 6, 4, 5];
// Коментар до фактичних параметрів функції stepmet
//[x,lam,s] = stepmet (n,x,A, eps,smax,lam1)

```

[x, lam, s] = stepmet (3, x, A, 0.01, 9, 1)

Результати виконання stepmet

s = 4 lam = 22.753126

x = 1. 0.2589132 0.3888574.

Отже, найбільше головне напруження 22.753 знайдено за 4 ітерації і має такі напрямні косинуси: (1, 0.2589132, 0.3888574).

5. Звичайні диференціальні рівняння

Математичними моделями задач механіки безпосередньо або після відокремлення змінних є задачі для звичайних диференціальних рівнянь (ЗДР). Причому це можуть бути звичайні диференціальні рівняння будь-якого порядку або їх системи.

Систему p звичайних диференціальних рівнянь 1-го порядку записують у такому вигляді:

$$u'_k(x) = f_k(x, u_1, u_2, \dots, u_p) \quad (k = \overline{1, p}) \quad \text{або} \quad u'(x) = f(x, u), \quad (5.1)$$

де $u = (u_1, u_2, \dots, u_p)$; $f = (f_1, f_2, \dots, f_p)$.

Із теорії звичайних диференціальних рівнянь відомо, що розв'язок $u(x)$ звичайного диференціального рівняння p -го порядку в загальному випадку залежить від p довільних сталих $C = (C_1, C_2, \dots, C_p)$, тобто $u = u(x; C)$. Для визначення C (виділення єдиного або потрібного нам розв'язку) на розв'язок слід накласти p додаткових умов (ДУ).

Якщо всі додаткові умови, тобто значення всіх функцій $u_k(x)$, задають в одній точці $x = \xi$, то їх називають *початковими (задача Коші - ЗК)* :

$$u_k(\xi) = \eta_k \quad (k = \overline{1, p}) \quad \text{або} \quad u(\xi) = \eta. \quad (5.2)$$

Вони є координатами початкової точки $(\xi, \eta_1, \eta_2, \dots, \eta_p)$ інтегральної кривої $u(x)$ у $p + 1$ -вимірному просторі $(x, u_1, u_2, \dots, u_p)$. Розв'язок у цьому випадку визначають на деякому відрізку $[\xi, X]$ або $[X, \xi]$. Тому точку ξ можна вважати початковою точкою цього відрізка.

Найуніверсальнішими методами розв'язування задач Коші є *числові*, оскільки уможливають визначення наближених значень розв'язку $u(x)$ на довільній сітці значень аргументу x_n ($n = \overline{0, N}$) відрізка $[\xi, X]$, тобто $\xi = x_0 < x_1 < \dots < x_N = X$. При цьому задача Коші має *бути коректно поставленою та добре обумовленою*, тобто такою, щоб незначна зміна початкових умов (ПУ) спричиняла й незначні зміни розв'язку.

Розглянемо застосування *числових методів* на прикладі задачі Коші

$$u'(x) = f(x, u(x)) \quad [\xi \leq x \leq X]; \quad (5.1')$$

$$u(\xi) = \eta. \quad (5.2')$$

Якщо розв'язок $u(x)$ розкласти у ряд Тейлора на інтервалі $x_n < x < x_{n+1}$ й позначити $u(x_n) = u_n$, отримаємо вираз

$$u_{n+1} = u_n + h_n u'_n + h_n^2 \frac{u''_n}{2!} + \dots \quad (h_n = x_{n+1} - x_n). \quad (5.3)$$

Похідні в правій частині можна легко одержати, продиференціювавши вираз (5.1') потрібне число разів:

$$u' = f(x, u); \quad (u')' = \frac{d}{dx}[f(x, u(x))] = f_x + f f_u \quad \left(f_x = \frac{\partial f}{\partial x}; f_u = \frac{\partial f}{\partial u} \right). \quad (5.4)$$

Ефективними для розрахунків є методи Рунге—Кутти (МРК), а найпростішою з них є формула 2-го порядку точності, яку можна одержати, утримавши у формулі (5.3) члени порядку $O(h^2)$. Однопараметрична сім'я двочленних різницевих схем методів Рунге—Кутти ($0 < \alpha \leq 1$) має такий вигляд:

$$y_{n+1} = y_n + h \left[(1 - \alpha) f_n + \alpha f \left(x_n + \frac{h}{2\alpha}, y_n + \frac{h}{2\alpha} f_n \right) \right]. \quad (5.5)$$

Ця схема має непогану точність, її нерідко застосовують у числових розрахунках. Беручи, наприклад, $\alpha = 1$, одержимо

$$y_{n+1} = y_n + h f \left(x_n + \frac{h}{2}, y_n + \frac{h f_n}{2} \right). \quad (5.6)$$

Зрозуміло, що початкові умови в обох випадках такі:

$$y_0 = \eta \quad [n = \overline{0, N-1}; u(x_n) \approx y_n].$$

Посилаючись на метод Рунге—Кутти 2-го порядку точності, писатимемо МРК2, вказуючи значення α . Формули методу Рунге—Кутти легко поширити на розв'язання систем звичайних диференціальних рівнянь.

Розрахункові формули методів Рунге—Кутти 2-го порядку ($\alpha = 1$) для системи звичайних диференціальних рівнянь 2-го порядку мають такий вигляд:

$$\begin{aligned} y_1^{n+1} &= y_1^n + h f_1 \left(x_n + \frac{h}{2}, y_1^n + h \frac{f_1^n}{2}, y_2^n + h \frac{f_2^n}{2} \right); \\ y_2^{n+1} &= y_2^n + h f_2 \left(x_n + \frac{h}{2}, y_1^n + h \frac{f_1^n}{2}, y_2^n + h \frac{f_2^n}{2} \right), \end{aligned} \quad (5.7)$$

де $y_k^0 = \eta_k \quad [n = \overline{0, N-1}; u_k(x_n) \approx y_k^n; f_k^n = f_k(x_n, y_1^n, y_2^n); k = \overline{1, 2}]$.

Основним способом оцінювання точності числового розв'язку (ЧР) задачі Коші для звичайних диференціальних рівнянь є використання *правила Рунге*. Якщо є числові розв'язки на двох сітках $y(x; h)$ і $y(x; gh)$ ($g > 1$), то похибку розв'язку на сітці з меншим кроком можна знайти за такою формулою:

$$\Delta y(x; h) \approx [y(x; h) - y(x; gh)] / [g^p - 1]. \quad (5.8)$$

Замість оцінювання точності знайдену похибку можна додати до числового розв'язку $y(x;h)$, уточнивши його:

$$\tilde{y}(x;h) \approx y(x;h) + \Delta y(x;h), \quad (5.9)$$

але тоді залишиться нез'ясованим питання оцінки точності \tilde{y} .

Отже, застосування правила Рунге дозволяє одержувати прийнятні результати навіть за схемами невисокого порядку точності.

Приклад 5.1. Визначимо за допомогою методу Рунге-Кутти 2-го порядку ($\alpha=1$) числовий розв'язок безрозмірного лінеаризованого рівняння коливань математичного маятника в порожнині $u''(x) + u(x) = 0$, $u(0) = 2$, $u'(0) = 0$, точний розв'язок якого $u(x) = 2\cos x$. Для застосування методу Рунге—Кутти переходимо до системи ЗДР 1-го порядку $u'(x) = u_1(x)$, $u_1'(x) = -u(x)$; $u(0) = 2$, $u_1(0) = 0$.

Лістинг 5.1. Файл `szdrmrk2`:

```
// szdrmrk2
// Числове розв'язання системи ЗДР МРК2 (alpha=1)
function [dmrk2,u,y,x] = szdrk2(y10,y20,N,a,b)
// ПЧ рівнянь (6.1) та точний розв'язок u(x)
deff('ff1 = f1(x,y1,y2)', 'ff1 = -y2')
deff('ff2 = f2(x,y1,y2)', 'ff2 = y1')
deff('uu = f(x)', 'uu = 2*cos(x)')
// Крок сітки
h=(b-a)/(N-1);
hh=h/2;
xx=a;
// Початкові умови
y1=y10;
y2=y20;
// Числовий розв'язок системи ЗДР МРК2 (alpha=1)
for n=1:N
    x(n) =xx; y(n)=y1;
    f11=f1(xx,y1,y2); f21=f2(xx,y1,y2);
    f12=f1(xx+hh,y1+hh*f11,y2+hh*f21);
    f22=f2(xx+hh,y1+hh*f11,y2+hh*f21);
    y1=y1+h*f12; y2=y2+h*f22; xx=xx+h
end ;
// Обчислення ЧР та точного розв'язку
for n=1:N
    xn=x(n); u(n)=f(xn);
end;
// Обчислення ДСКВ ЧР від точного розв'язку
dmrk2=0;
for k=1:N
```

```

dmrk2=dmrk2+(u(k)-y(k))^2;
end;
dmrk2=sqrt(dmrk2/N);
// Перетворення векторів на матриці-рядки
x = x.'; y = y.'; u = u.';
endfunction;
// main program
// Коментар до фактичних параметрів функції szdrk2
//[dmrk2,u,y,x] = szdrk2(y10,y20,N,a,b)
[dmrk2,u,y,x] = szdrk2( 2, 0,5,0,1)
// Побудова графіків числового та точного розв'язків
plot(x,y,'k*','x,u','k-');
xgrid();
xtitle(' Y=Y(X)      U=U(X) ', 'X', 'Y      U');
legend(' Y=Y(X) ', ' U=U(X) ');

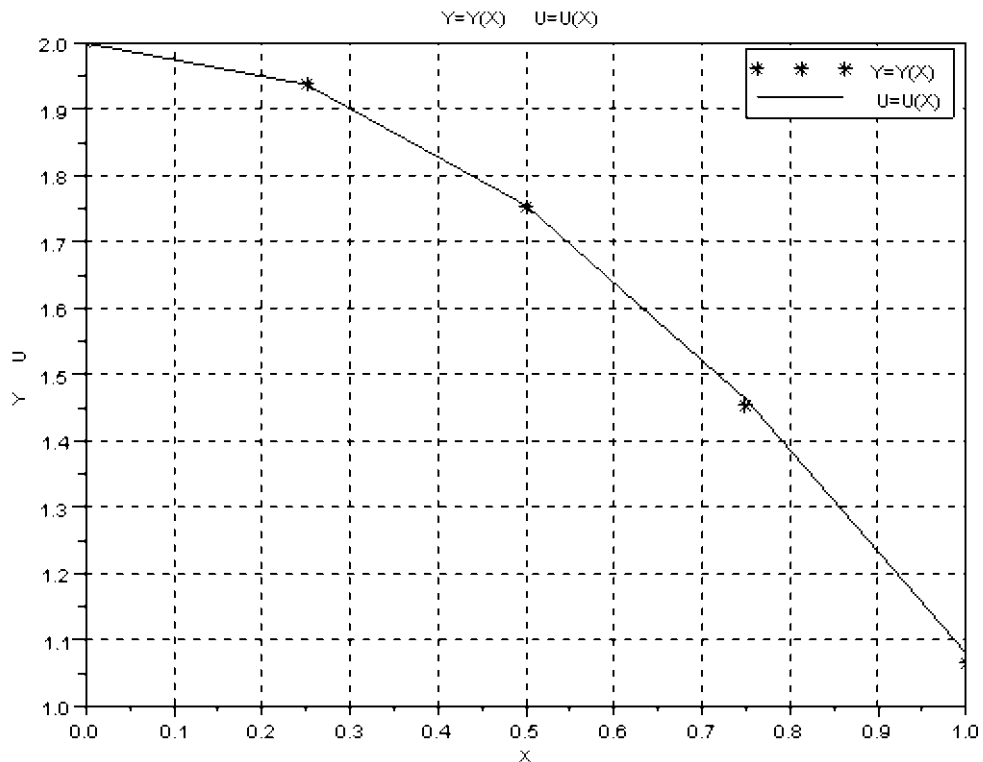
```

Результати виконання *szdrmrk2*

у табличному вигляді:

x =	0.	0.25	0.5	0.75	1.
y =	2.	1.9375	1.7519531	1.4550171	1.0654316
u =	2.	1.9378248	1.7551651	1.4633777	1.0806046
dmrk2 =		0.0078809			

графічному вигляді:



Якщо додаткові умови для звичайного диференціального рівняння задані не в одній точці відрізка інтегрування, то задачу називають **крайовою**. Зрозуміло, що її

можна поставити лише для звичайного диференціального рівняння, у яких $p \geq 2$. У цьому разі додаткові умови називають *крайовими*. Вони можуть бути дуже різноманітними. Наведемо один із поширених типів крайових умов (КУ):

$$\Phi[u(\xi_1, \xi_2, \dots, \xi_k)] = \eta \text{ чи } \varphi_k[u_1(\xi_k), \dots, u_p(\xi_k)] = \eta_k \left\{ \xi_k \in [a, b]; k = \overline{1, p} \right\}. \quad (5.12)$$

Незважаючи на різноманітність крайових умов, крайові задачі (КЗ) для звичайного диференціального рівняння розв'язують в основному одними й тими самими методами.

Розглянемо застосування *різницевого методу* (РМ) на прикладі розв'язування крайової задачі для лінійного звичайного диференціального рівняння 2-го порядку зі змінними коефіцієнтами

$$u''(x) + p(x)u'(x) + q(x)u(x) = f(x) \quad (x \in [a, b]) \quad (5.13)$$

і крайові умови загального вигляду

$$\alpha_1 u(a) + \beta_1 u'(a) = \gamma_1; \quad \alpha_2 u(b) + \beta_2 u'(b) = \gamma_2, \quad (5.14)$$

де $p(x), q(x), f(x)$ — задані обмежені та неперервні зі своїми 2-ми похідними функції; $\alpha_i, \beta_i, \gamma_i$ ($i = 1, 2$) — задані дійсні числа, причому $|\alpha_i| + |\beta_i| \neq 0$.

Припустимо, що ця крайова задача має єдиний розв'язок, і введемо на $[a, b]$ сітку $a = x_0 < x_1 < x_2 < \dots < x_N = b$, причому для спрощення — рівномірну, тобто

$$x_{n+1} - x_n = h \equiv \frac{b-a}{N}.$$

Замінивши похідні в звичайного диференціального рівняння на симетричні різницеві співвідношення, які мають похибку $O(h^2)$, а в крайових умовах — на вирази $u'(a) \approx \frac{y_1 - y_0}{h}$ і $u'(b) \approx \frac{y_N - y_{N-1}}{h}$ з похибкою $O(h)$, матимемо таку різницеву схему (РС):

$$\begin{aligned} & \frac{y_{n-1} - 2y_n + y_{n+1}}{h^2} + \frac{p_n(y_{n+1} - y_{n-1})}{2h} + q_n y_n = f_n \\ & \left[p_n = p(x_n), q_n = q(x_n), f_n = f(x_n); n = \overline{1, N-1} \right]; \\ & \alpha_1 y_0 + \frac{\beta_1(y_1 - y_0)}{h} = \gamma_1, \quad \alpha_2 y_N + \frac{\beta_2(y_N - y_{N-1})}{h} = \gamma_2. \end{aligned} \quad (5.15)$$

Одержану систему лінійних алгебричних рівнянь перепишемо в такому вигляді:

$$\begin{aligned} & (\alpha_1 h - \beta_1) y_0 + \beta_1 y_1 = \gamma_1 h; \\ & \left(1 - \frac{hp_n}{2} \right) y_{n-1} - (2 - q_n h^2) y_n + \left(1 + \frac{hp_n}{2} \right) y_{n+1} = h^2 f_n \quad (n = \overline{1, N-1}); \\ & -\beta_2 y_{N-1} + (\alpha_2 h + \beta_2) y_N = \gamma_2 h. \end{aligned} \quad (5.16)$$

Оскільки матриця системи (5.16) тридіагональна, то систему лінійних алгебричних рівнянь краще розв'язувати *методом алгебричної прогонки*. Для підвищення точності розв'язку слід скористатися правилом Рунге.

Приклад 5.2. Визначимо різницевим методом числовий розв'язок КЗ лінійного ЗДР 2-го порядку зі змінними коефіцієнтами вигляду

$$u''(x) + 2xu'(x) + 2u(x) = 4x; \quad u(0) = 1; \quad u(0.5) = 1.279,$$

точний розв'язок якого $u(x) = x + e^{-x^2}$.

Лістинг 5.2. Файл *rmlinbp*:

```
// rmlinbp
// Обчислення ЧР КЗ лінійного ЗДР 2-го порядку зі
// змінними коефіцієнтами (5.16) різницевим методом
function [u,y,x] = rmlinbp(N,a,b,a1,b1,g1,a2,b2,g2);
// N - число підвідрізків; a, b - відрізок [a,b];
// a1,b1,g1,a2,b2,g2 - коефіцієнти КУ (5.16).
// u - ТР, y - ЧР, x - вузли сітки
// Коефіцієнти і ПЧ ЗДР та точний розв'язок u(x)
deff('pp = p(x)', 'pp = 2*x');
deff('qq = q(x)', 'qq = 2');
deff('ff = f(x)', 'ff = 4*x');
deff('uu = ux(x)', 'uu = x+exp(-x^2)');
// Крок сітки
h=(b-a)/N; h1=h/2; h2=h*h;
// Розв'язання (5.16) алгебричною прогонкою
x(1)=a; k(1)=0; e(1)=0;
// Знаходження коефіцієнтів СЛАР (5.16)
xx=a-h; n1=N+1;
for n=1:N
xx=xx+h; x(n+1)=xx+h;
if(n<2) then
aa=0; bb=b1-a1*h; cc=b1; dd=g1*h
else
aa=h1*p(xx); cc=aa+1; aa=1-aa; bb=2-h2*q(xx);
dd=f(xx)*h2;
end;
// Прямий хід АП
zn=bb-aa*k(n);
k(n+1)=cc/zn; e(n+1)=(aa*e(n)-dd)/zn
end;
aa=-b2; dd=g2*h; bb=-(b2+a2*h);
y(n1)=(aa*e(n1)-dd)/(bb-aa*k(n1));
// Завершення знаходження ЧР y(n) - зворотний хід АП
for n=1:N
ii=n1-n+1; y(ii-1)=k(ii)*y(ii)+e(ii)
end;
// Обчислення точного розв'язку заданого ЗДР
for n=1:N+1
```



```

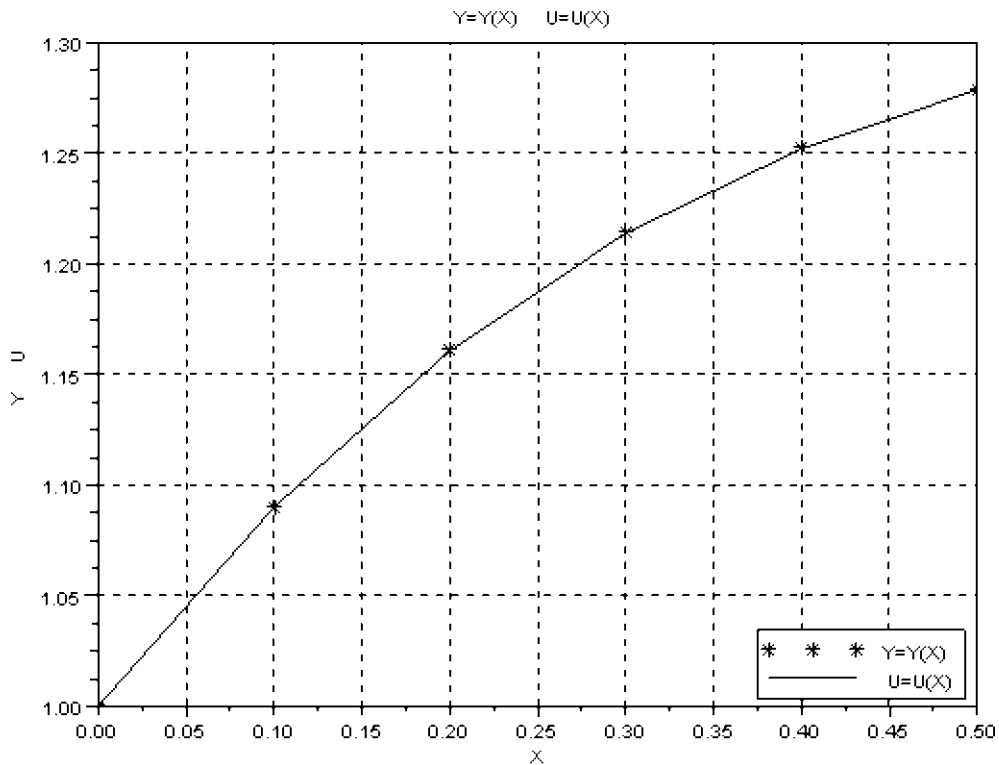
    xn=x(n); u(n)=ux(xn);
    end;
// Перетворення векторів на матриці-рядки
x = x.'; y = y.'; u = u.';
endfunction;
// main program
// Коментар до фактичних параметрів функції rmlinbp
//[u,y,x] = rmlinbp(N,a, b ,a1,b1,g1,a2,b2, g2);
[u,y,x] = rmlinbp(5,0,0.5, 1, 0, 1, 1, 0, 1.279)
// Побудова графіків числового та точного розв'язків
plot(x,y,'k*','x,u','k-');
xgrid();
xtitle(' Y=Y(X)      U=U(X) ', 'X', 'Y      U');
legend(' Y=Y(X) ', ' U=U(X) ', 4, %t);

```

Результати виконання *rmlinbp*
у табличному вигляді:

x	= 0.	0.1	0.2	0.3	0.4	0.5
y	= 1.	1.0902945	1.1611714	1.2143439	1.25249	1.279
u	= 1.	1.0900498	1.1607894	1.2139312	1.2521438	1.27884

графічному вигляді:



Задачі *на власні значення для звичайних диференціальних рівнянь* — це крайова задача для системи звичайних диференціальних рівнянь 1-го порядку такого вигляду:

$$u'(x) = f(x, u; \lambda) \quad (5.18)$$

$$\left[f = (f_1, f_2, \dots, f_p); u = (u_1, u_2, \dots, u_p); \lambda = (\lambda_1, \lambda_2, \dots, \lambda_t) \right],$$

праві частини яких залежать від t параметрів λ , їх значення невідомі і визначатимуться із самої задачі. Тут додаткових умов повинно бути $p + t$. До таких задач зводиться багато задач механіки.

Значення λ , за яких крайова задача має ненульовий розв'язок, називають власними значеннями (ВЗ), а відповідні їм розв'язки $u_k(x)$ ($k = \overline{1, p}$) — власними функціями (ВФ) крайової задачі.

Прикладом задачі на власні значення для звичайних диференціальних рівнянь є класична задача для рівняння 2-го порядку за нульових крайових умов (задача Штурма-Ліувілля - ЗШЛ):

$$u''(x) + p(x)u'(x) + [\lambda + q(x)]u(x) = 0; u(a) = u(b) = 0. \quad (5.19)$$

Це рівняння має одне власне значення λ .

Отже, задача потребує трьох додаткових умов, але, зважаючи на лінійність та однорідність рівняння, його розв'язок визначають з точністю до сталого множника. Це і є задання третьої додаткової умови. Наприклад, за додаткову можна взяти умову $u'(a) = 1$ або якусь іншу.

За використання в задачах на власні значення балістичного методу (БМ) природним визначальним параметром є λ . Як правило, балістичним методом послуговуються, коли параметр λ один.

Приклад 5.3. *Визначимо балістичним методом у сукупності з методом січних числовий розв'язок безрозмірного рівняння стійкості пружного стержня*

$$u''(x) + \lambda u(x) = 0, \quad u(0) = u(1) = 0 \quad \left(\lambda = \frac{l^2 P}{EI_z}; x = \frac{z}{l} \right).$$

Це рівняння має точний розв'язок (ТР)

$$u_m(x) = C \sin m\pi x, \quad \lambda_m = m^2 \pi^2 \quad (m = 1, 2, \dots) \quad (\lambda_1 = \pi^2 \approx 9,8696),$$

але на ньому просто проілюструвати розв'язування задачі на власні значення.

Задане звичайне диференціальне рівняння зведемо до задачі Коші для такої системи звичайних диференціальних рівнянь 1-го порядку:

$$u'(x) = u_1(x), \quad u_1'(x) = -\lambda u(x); \quad u(0) = 0, \quad u_1(0) = 1.$$

Розв'язуючи одержану задачу Коші, використовуємо метод Рунге-Кутти 4-го порядку з визначенням точності за параметром λ до виконання умови $u(1) = 0$. За початкові значення беремо $\lambda_1^{(0)} = 9$ та $\lambda_1^{(1)} = 10$. Ітерації завершуємо, коли різниця між сусідніми значеннями λ стане меншою за 10^{-3} . У результаті маємо таке

значення найменшого власного значення: ($\lambda_1 \approx 9,91965$) Зауважимо, що хоча вихідна крайова задача тут лінійна, це не спрощує розв'язування, бо розв'язок $u(x;\lambda)$ залежить від λ *нелінійно*.

Лістинг 5.3. Файл *bmvzdr*:

```
// bmvzdr
// Числове розв'язання задачі на ВЗ для ЗДР (5.18)
// (ЗШЛ) БМ у сукупності з методом січних та МРК4
function [yy1] =fyu(lmd,N,u10,u20,h,x);
// Розв'язання задачі Коші МРК4
// Коефіцієнти ЗДР
deff('pp = p(t)', 'pp = 0');
deff('qq = q(t)', 'qq = 0');
deff('ff1 = f1(t,y1,y2)', 'ff1 = y2');
deff('ff2 = f2(t,y1,y2)', 'ff2=-p(t)*y2-(lmd+q(t))*y1');
// Початкові умови ЗК ЗДР
y1=u10; y2=u20; hh=h/2;
// ЧР задачі Коші МРК4
for n=1:N
    t=x(n); yy1(n)=y1;
    f11 =f1(t,y1,y2); f21 =f2(t,y1,y2);
    f12=f1(t+hh,y1+hh*f11,y2+hh*f21);
    f22=f2(t+hh,y1+hh*f11,y2+hh*f21);
    f13=f1(t+hh,y1+hh*f12,y2+hh*f22);
    f23=f2(t+hh,y1+hh*f12,y2+hh*f22);
    f14=f1(t+h,y1+h*f13,y2+h*f23);
    f24=f2(t+h,y1+h*f13,y2+h*f23);
    y1=y1+h*(f11+2.*f12+2.*f13+f14)/6;
    y2=y2+h*(f21+2.*f22+2.*f23+f24)/6
end;
endfunction;
// Функція bmvzdr розв'язання задачі на ВЗ для ЗДР
function [lam,s,dvz,u,y,x] = ...
bmvzdr(N,a,b,l,u10,u20,lam1,lamn,smax,eps);
// N - число вузлів сітки, a, b - відрізок [a,b]
// l - логічна змінна,u10 - початкове значення u1,u20 -
// початкове значення u2, lam1 - перше наближення до
// ВЗ, lamn - друге наближення до ВЗ, smax - допустиме
// число ітерацій, eps - точність. lam - знайдене ВЗ, s
// - число виконаних ітерацій,dvz - ДСКВ ЧР,u - ТР,
// y - ЧР, x - вузли сітки
// Функція точного розв'язку задачі
deff('uu = ff(t)', 'uu = sin(%pi*t)');
// Крок сітки та формування вузлів сітки
```

```

h=(b-a)/(N-1);
for n=1:N
    x(n)=a+(n-1)*h;
end;
// Формування даних для методу січних
[ay] =fyy(lamn,N,u10,u20,h,x);
y10=ay(N);
[ay1] =fyy(lam1,N,u10,u20,h,x);
y11 =ay1(N); day1=lam1-lamn; dd1=abs(day1) ;
// Початок ітераційного процесу
s=0;
while ((s<=smax)|(dd1>eps)) then
    s=s+1;
    dy1=y11-y10;
if(dy1==0) then break end;
    da=dy1/day1;
    dd0=dd1;
if(da==0) then break end; ad2=lam1-y11/da;
lamn=lam1; lam1=ad2;
[ay2] =fyy(ad2,N,u10,u20,h,x);
y12=ay2(N);
y10=y11;
y11=y12;
for n=1:N
    ay(n) =ay1(n);
    ay1(n)=ay2(n)
end;
day1=lam1-lamn; dd1=abs(day1);
end;
[ay] =fyy(lamn,N,u10,u20,h,x);
anm=ay((N-1)/2+1);
for n=1:N
    ay(n)=ay(n)/anm
end;
lam=lamn;
for n=1:N
    xn = x(n);
    y(n)=ay(n);
    u(n)=ff(xn);
end;
// ДСКВ dvz ЧР від точного
dvz=0;
for k=1:N
    dvz=dvz+(u(k)-y(k))^2;
end;

```

```

    dvz=sqrt (dvz/N);
// Перетворення векторів на матриці-рядки
x = x.'; y = y.'; u = u.';
endfunction;
// main program
[lam,s,dvz,u,y,x] = ...
    bmvzdr( 5,0,1,%t, 0, 1, 9, 10, 99,0.00001)
//bmvzdr( N,a,b, l, u10,u20,lam1,lamn,smax, eps)
// Коментар до фактичних параметрів функції rmlinbr
// Побудова графіків числового та точного розв'язків
plot(x,y,'k*',x,u,'k-'); xgrid();
xlabel(' Y=Y(X)      U=U(X)      ','X','Y      U');
legend(' Y=Y(X) ',' U=U(X) ');

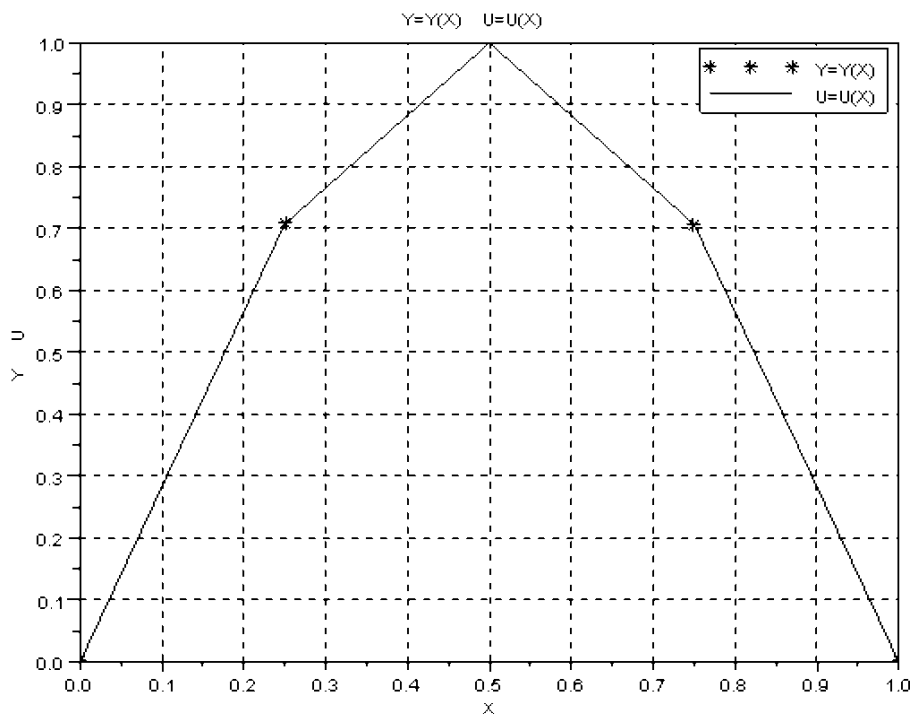
```

Результати виконання *bmvzdr*:

у табличному вигляді:

x = 0.	0.25	0.5	0.75	1.
y = 0.	0.7081888	1.	0.7060265	2.631D-16
u = 0.	0.7071068	1.	0.7071068	1.225D-16
dvz =	0.0006838	s = 9.	lam =	9.9196503

графічному вигляді:



6. Рівняння в частинних похідних

Зазвичай задачі механіки суцільного середовища зводяться до розв'язання диференціальних рівнянь у частинних похідних (РЧП).

Загальний вигляд *РЧП* з однією невідомою функцією u , яка залежить від часу t , координати r та інших змінних, такий: $F(t, r, u, u_t, u_{,1}, \dots) = 0$, відшукуватимемо ж його розв'язок в області $G(t, r, \dots) = g(r, \dots) \times [t_0, T]$, тобто в деякій просторовій зоні $g(r, \dots)$ на відрізку часу $[t_0, T]$. У цьому разі додаткові умови, задані за $t = t_0$, називають *початковими*, а додаткові умови на границі $\gamma(r)$ області $g(r)$ — *крайовими*.

Задачу Коші (задано лише початкові умови) для рівняння теплопровідності можна побудувати лише в необмеженому просторі

$$u_t(r, t) = k\Delta u(r, t) + f(r, t), \quad u(r, t_0) = \mu(r). \quad (6.1)$$

Задачу з початковими і крайовими умовами називають *нестационарною крайовою задачею*. Додаткові умови такої задачі мають вигляд:

$$u(r, t_0) = \mu(r) \quad [r \in g(r)]; \quad u(r, t)|_{\gamma} = \mu_1(r, t) \quad [t_0 \leq t \leq T]. \quad (6.2)$$

Далі розглянемо диференціальні рівняння у частинних похідних лише у разі *коректно поставлених задач*.

У загальному випадку коефіцієнти рівняння 2-го порядку з двома змінними

$$Au_{,11} + 2Bu_{,12} + Cu_{,22} + Du_{,1} + Eu_{,2} + F = 0 \quad (6.3)$$

можуть залежати від x_1 , x_2 та шуканої функції u . Якщо один з коефіцієнтів A , B або C відрізняється від нуля, рівняння залежно від знака дискримінанта $d = B^2 - AC$ може належати до одного з трьох типів: *еліптичного* ($d < 0$), *параболічного* ($d = 0$) або *гіперболічного* ($d > 0$).

Для застосування *різницевого методу* (РМ) під час розв'язування диференціальних рівнянь у частинних похідних в області $G(r, t)$ або $g(r)$ вводять сітку, в основному прямокутну (для просторових областей сітка складається з прямокутних паралелепіпедів).

Розглянемо розв'язання диференціального рівняння у частинних похідних *еліптичного типу*, до якого належать, наприклад, рівняння в задачі кручення стержня, стаціонарного розподілу тепла, в яких відсутні похідні за часом, тобто стаціонарні задачі. Отже, розглянемо *двовимірну задачу Діріхле* в довільній області $g(x)$ [$x = (x_1, x_2)$] з границею γ для рівняння Пуассона (крайову задачу)

$$\Delta u(x) \equiv u_{,11}(x) + u_{,22}(x) = -f(x); \quad u(x)|_{\gamma} = \mu(x) \quad [x \in g(x)]. \quad (6.4)$$

Для розв'язання цієї задачі *різницевим методом* введемо в області g прямокутну сітку, причому для спрощення — рівномірну за обома координатами. Візьмемо шаблон (рис. 6.1) в околі точки (x_{1n}, x_{2m}) і замінимо похідні в задачі (6.4) на різницеві співвідношення

$$(u_{,11})_{n,m} = (u_{n-1,m} - 2u_{n,m} + u_{n+1,m}) / h_1^2 + O(h_1^2);$$

$$(u_{,22})_{n,m} = (u_{n,m-1} - 2u_{n,m} + u_{n,m+1}) / h_2^2 + O(h_2^2).$$

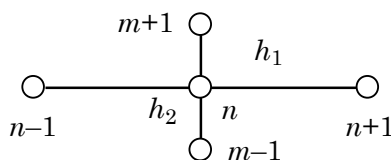


Рис. 6.1. Шаблон для побудови РС еліптичного рівняння

У результаті прийдемо до системи рівнянь — різницевої схеми

$$(y_{n-1,m} - 2y_{nm} + y_{n+1,m})h_1^{-2} + (y_{n,m-1} - 2y_{nm} + y_{n,m+1})h_2^{-2} = -f_{nm},$$

де $u(x_{1n}, x_{2m}) \approx y_{nm}, f_{nm} = f(x_{1n}, x_{2m})$.

Крім того, потрібно врахувати задані крайові умови на γ , що досить просто зробити, коли область g прямокутна. Це система лінійних алгебричних рівнянь відносно невідомих наближених значень y_{nm} шуканої функції $u(x)$ у внутрішніх вузлах області g . Її можна розв'язувати методом Гаусса. Якщо існують неперервні $u_{,1111}$ і $u_{,2222}$, то порядок одержуваного різницевого розв'язку у дорівнює $O(h_1^2 + h_2^2)$.

За великого числа невідомих одержану систему лінійних алгебричних рівнянь розв'язують ітераційними методами. Розглянемо найпростіший із них — процес Лібмана (усереднення), який має таку розрахункову формулу – різницеву схему:

$$y_{nm}^{(s+1)} = \left[y_{n-1,m} + y_{n+1,m} + (y_{n,m-1} + y_{n,m+1})\alpha^2 + h_1^2 f_{nm} \right]^{(s)} / \left[2(1 + \alpha^2) \right] \quad \left(\alpha = \frac{h_1}{h_2}; s = 0, 1, \dots \right). \tag{6.5}$$

У разі рівномірного кроку сітки ($h = \text{const}$) маємо

$$y_{nm}^{(s+1)} = \left(y_{n-1,m} + y_{n+1,m} + y_{n,m-1} + y_{n,m+1} + h^2 f_{nm} \right)^{(s)} / 4, \tag{6.6}$$

Різницеві схеми (6.5) та (6.6) — абсолютно (безумовно) стійкі.

За нульові наближення $y_{nm}^{(0)}$ можна взяти значення, які належать до відрізка $\left[\min u(x)|_\gamma, \max u(x)|_\gamma \right]$, або значення функції $u(x)$ у відповідних точках $g(x)$.

Приклад 6.1. Знайдемо різницевим методом, удаючись до процесу Лібмана, розв'язок крайової еліптичної задачі — задачі Діріхле для рівняння Лапласа $u_{,11}(x) + u_{,22}(x) = 0$; що визначає стаціонарне розподілення температури в прямокутній області $\{g(x) = [a_i \leq x_i \leq b_i]; i = 1, 2\}$ з такими КУ:

$$u(x_1, 0) = 25 x_1 \sin(\pi x_1 / 2), u(x_1, b_2) = 25 x_1, u(0, x_2) = 45 x_2 (1 - x_2), u(b_1, x_2) = 25.$$

Лістинг 6.1. Файл *ellipseq*:

```
// ellipseq
// Процес Лібмана розв'язання задачі Діріхле для
// рівняння Лапласа
function [yt,s]=ellipseq(a1,b1,a2,b2,N,M,eps,smax);
// a1,b1 та a2,b2 - початки і кінці відрізків області
// по осях x1 і x2 відповідно, N та M - числа
//підвідрізків по осях x1 і x2 відповідно, eps -
// точність, smax допустима кількість ітерацій.
//yt - розв'язок задачі Діріхле для рівняння Лапласа,
// s - число ітерацій
// КУ задачі Діріхле для рівняння Лапласа
deff('сс = fd(x1)', 'сс = 25*x1*sin(%pi*x1/2)');
deff('сс = fu(x1)', 'сс = 25*x1');
deff('сс = fl(x2)', 'сс = 45*x2*(1-x2)');
deff('сс = fr(x2)', 'сс = 25');
// Кроки по осях x1, x2
h1=(b1-a1)/N; h2=(b2-a2)/M;
// Обчислення сталих
ba1=b1-a1; ba2=b2-a2; a1=(h1/h2)^2; a11=(1+a1)*2;
// Обчислення координат вузлових точок по осях
for n=1:N+1
    x1(n)=a1+(n-1)*h1;
end;
for m=1:M+1
    x2(m)=a2+(m-1)*h2;
end;
// Формування першого і останнього рядків по X1
for n=1:N+1
    x1n=x1(n); y(1,n) = fd(x1n); y(M+1,n)=fu(x1n)
end;
// Формування першого і останнього стовпчиків по X2
for m=1:M+1
    x2m=x2(m); y(m,1) =fl(x2m); y(m,N+1)=fr(x2m);
end;
// Формування початкових наближень розв'язку по X1
for m=2:M
    for n=2:N
        yp=y(m,n-1); y1(m,n)=yp+(y(m,N+1)-yp)*x1(n)/ba1;
    end;
end;
// Формування початкових наближень розв'язку по X2
for n=2:N
    for m=2:M
        yp=y(m-1,n);
```



```

        y2(m,n)=yp+(y(M+1,n)-yp)*x2(m)/ba2;
    end;
end;
// Усереднення початкових наближень розв'язку
for n=2:N
    for m=2:M
        y(m,n)=(y1(m,n)+y2(m,n))/2;
    end;
end;
// Ітераційний процес (6.5)
for s=1:smax
    ym=0; sp=s;
    for m=2:M
        for n=2:N
            w=((y(m,n-1)+y(m,n+1))+...
            (y(m-1,n)+y(m+1,n))*a1)/a11;
            d=abs(w-y(m,n)); y(m,n)=w;
        // Обчислення модуля максимальної різниці
        // між попереднім і наступним наближеннями ЧР
            if (d>ym) then ym=d; end;
        end;
    end;
// Завершення ітерацій (7.21) за точністю
    if(ym<=eps) then break; end;
end;
// Формування значень ЧР для друку
for m=2:M+2
    for n=2:N+2
        yt(m,n)=y(m-1,n-1);
    end;
end;
//Розміщення вузлів по X1 у 1-му рядку таблиці
for n=2:N+2
    yt(1,n)=x1(n-1);
end;
// Розміщення вузлів по X2 у 1-му стовпчику таблиці
for m=2:M+2
    yt(m,1)=x2(m-1);
end;
yt(1,1)=0;
// Слід замінити це значення в таблиці на X2/X1 вручну
endfunction;
// main program
N=5; M=5;
// Коментар до фактичних параметрів функції ellipseq

```

```

//[yt,s]=ellipseq(a1,b1,a2,b2,N,M,eps,smax)
[yt,s]=ellipseq( 0, 1 ,0, 1,N,M,0.01,100)
// Формування значень для побудови поверхні ЧР
for m=2:M+2
for n=2:N+2
y(m-1,n-1)=yt(m,n);
end;
end;
for n=2:N+2
x1(n-1)=yt(1,n);
end;
for m=2:M+2
x2(m-1)=yt(m,1);
end;
mesh(x1,x2,y); title('Z=Y(X1,X2)');
xlabel('X1'); ylabel('X2');

```

Результати виконання *ellipseq*

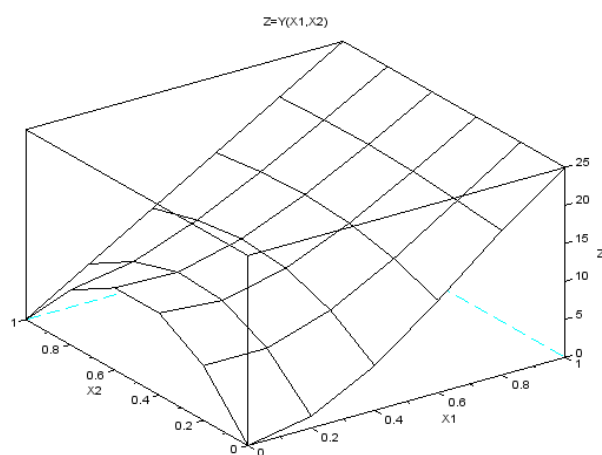
у табличному вигляді:

s = 14.

yt =

X2\X1	0.	0.2	0.4	0.6	0.8	1.
0.	0.	1.5451	5.8779	12.1353	19.0211	25.
0.2	7.2	7.1831	9.8579	14.3187	19.6268	25.
0.4	10.8	10.1448	12.0680	15.6661	20.1714	25.
0.6	10.8	10.5444	12.6192	16.1168	20.3959	25.
0.8	7.2	8.6247	11.75	15.7927	20.2972	25.
1.	0.	5.	10.	15.	20.	25.

графічному вигляді:



Розглянемо розв'язання **параболічного рівняння (ПР)**, а саме **нестационарної задачі теплопровідності**, причому **одновимірної задачі**, тобто коли є лише одна просторова змінна:

$$u_t(x,t) = ku_{xx}(x,t) + f(x,t). \quad (6.7)$$

Розв'язок шукатимемо у прямокутній області $G = [0 \leq x \leq a, 0 \leq t \leq T]$ за таких додаткових умов:

$$u(x,0) = \mu(x) \quad (0 \leq x \leq a); \quad (6.8)$$

$$u(0,t) = \mu_1(t), \quad u(a,t) = \mu_2(t) \quad (0 \leq t \leq T). \quad (6.9)$$

Візьмемо для спрощення $f(x,t) = 0$ та побудуємо *найпростіші різницеві схеми*, використовуючи в околі точки (x_n, t_m) шаблони (рис. 6.2)

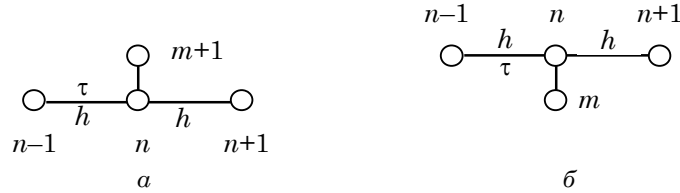


Рис. 6.2. Шаблиони для побудови найпростіших РС параболічного рівняння

Замінивши похідні у вихідному рівнянні на співвідношення (рис. 6.2, а)

$$u_t(x_n, t_m) = \frac{\hat{u}_n - u_n}{\tau} + O(\tau); \quad u_{xx}(x_n, t_m) = \frac{u_{n+1} - 2u_n + u_{n-1}}{h^2} + O(h^2),$$

прийдемо до явної різницевої схеми

$$\hat{y}_n = k\tau \left[y_{n-1} - \left(2 - \frac{h^2}{k\tau} \right) y_n + y_{n+1} \right] / h^2 \quad (n = \overline{1, N-1}; m = \overline{0, M-1}). \quad (6.10)$$

Якщо до цього додати початкові і крайові умови

$$y_n^0 = \mu(x_n) \quad (n = \overline{0, N}), \quad y_0^{m+1} = \mu_1(t_{m+1}), \quad y_N^{m+1} = \mu_2(t_{m+1}), \quad (6.11)$$

то задача знаходження наближених значень $u(x)$ легко розв'язується, бо на початковому шарі ($m = 0$) розв'язок відомий з початкових умов $y_n^0 = \mu(x_n)$, а за інших n ($n = \overline{1, N-1}$) (6.11) на першому шарі лише одне невідоме (y_0^1 і y_N^1 беремо з крайових умов). За відомим y_n^1 визначимо y_n^2 і т. д. Тому одержану різницеву схему називають *явною*.

У разі шаблону (див. рис. 6.2, б) різницєва схема має такий вигляд:

$$\frac{\hat{y}_n - y_n}{\tau} = k \frac{\hat{y}_{n+1} - 2\hat{y}_n + \hat{y}_{n-1}}{h^2} \quad (n = \overline{1, N-1}). \quad (6.12)$$

Ця різницєва схема містить у собі декілька невідомих на новому шарі, тому її називають *неявною*. Для її розв'язування перемістимо невідомі до лівої частини, унаслідок чого одержимо таку СЛАР, додаючи (6.11):

$$\hat{y}_{n+1} - \left(2 + \frac{h^2}{k\tau} \right) \hat{y}_n + \hat{y}_{n-1} = -\frac{y_n h^2}{k\tau} \quad (n = \overline{1, N-1}). \quad (6.13)$$

Оскільки ця СЛАР має матрицю тридіагональної структури, то доцільно скористатись алгебричною прогонкою.

Порядок одержуваного різницевого розв'язку u за існування похідних u_{tt} і u_{xxxx} дорівнює $O(\tau + h^2)$, тобто похибка зменшується з такими порядками, коли τ та h прямують до нуля.

Можна довести, що неявна РС *безумовно стійка*, тобто стійка за довільних співвідношень τ і h , а явна РС *умовно стійка*, тобто лише в разі виконання умови

$$\tau \leq \frac{h^2}{2k}. \quad (6.14)$$

Крім того, треба зауважити, що з використаних апроксимацій похідних випливає, що наближений розв'язок збігається до точного.

Приклад 6.2. Знайдемо різницевим методом розв'язок параболічної задачі — задачі нестационарної теплопровідності:

$$u_t(x,t) = u_{xx}(x,t) + \sin(xt), \quad G = [0 \leq x \leq 1, 0 \leq t \leq 0,03]$$

з ДУ $u(x,0) = \cos \frac{\pi x}{2} \quad (0 \leq x \leq 1); \quad u(0,t) = 1+t, \quad u(1,t) = t \quad (0 \leq t \leq 0,03).$

Скористаємося найпростішою явною РС, виконуючи умову стійкості $\tau \leq \frac{h^2}{2k}$.

Лістинг 6.2. Файл *parabeq*:

```
//parabeq
// Явна РС Розв'язання параболічного рівняння
// Права частина рівняння
function u=f(x,t);
    u=sin(x*t);
endfunction
// Початкова умова
function u=mu(x);
    u=cos(%pi*x/2);
endfunction
// Ліва КУ
function u=mu1(t);
    u=t+1;
endfunction
// Права КУ
function u=mu2(t);
    u=t;
endfunction
// ЧР ПР за допомогою явної РС
function [yt]=parabeq(N,M,a,T,k)
// N і M - числа підвідмірків по осях x і t
// відповідно, // a, T - довжини відрізків
// по осях x і t відповідно, k - стала ПР.
// yt - числовий розв'язок задачі
```

```

// Кроки за обома аргументами
h=a/N; tau=T/M;
// Сітка по X, заповнення першого рядка ПУ
for n=1:N+1
    x(n)=(n-1)*h;    y(1,n)=mu(x(n));
end
//Заповнення 1-го і останнього стовпчиків сітки по T,
for m=1:M+1
    t(m)=(m-1)*tau;
    y(m,1)=mu1(t(m));    y(m,N+1)=mu2(t(m));
end;
// Параметр явної РС
gam=k*tau/h^2;
// ЧР ПР за допомогою явної РС (7.33)
for m=1:M
    for n=2:N
        y(m+1,n)=gam*y(m,n-1)-(2*gam-1)*y(m,n)+ ...
            gam*y(m,n+1)+tau*f(x(n),t(m));end;
end;
// Формування ЧР для друку у вигляді таблиці
for m=2:M+2
    for n=2:N+2
        yt(m,n)=y(m-1,n-1)
    end
end;
// Розміщення вузлів по X у 1-му рядку таблиці
for n=2:N+2
    yt(1,n)=x(n-1);
end;
// Розміщення вузлів по t у 1-му стовпчику таблиці
for m=2:M+2
    yt(m,1)=t(m-1);
end;
yt(1,1)=0
// Замінити це значення в таблиці на X/T вручну
endfunction;
// main program
N=5;M=3;
// Коментар до фактичних параметрів функції
//[yt]=parabeq(N,M,a,    T,k)
    [yt]=parabeq(N,M,1,0.03,1)
// Формування значень для побудови поверхні ЧР
for m=2:M+2
for n=2:N+2
y(m-1,n-1)=yt(m,n);

```

```

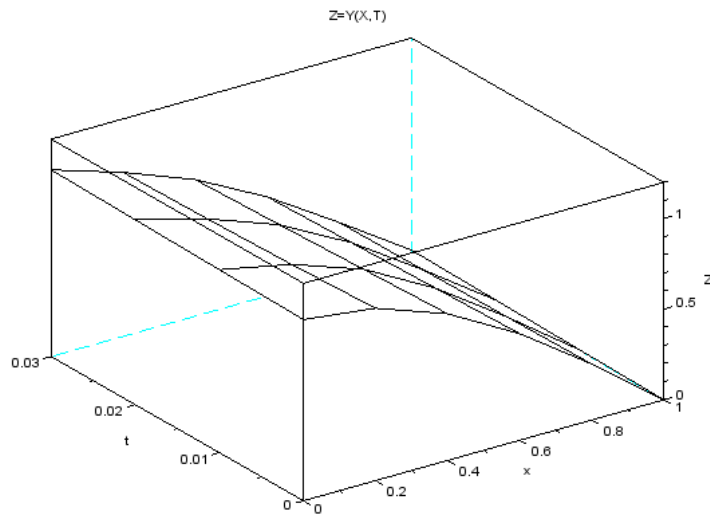
end;
end;
for n=2:N+2
    x(n-1)=yt(1,n);
end;
for m=2:M+2
    t(m-1)=yt(m,1);
end;
// Побудова поверхні ЧР
mesh(x,t,y); title('Z=Y(X,T)');
xlabel('x'); ylabel('t');

```

Результати виконання *parabeq*:
у табличному вигляді:

y(x, t)						
X\T	0.	0.2	0.4	0.6	0.8	1.0
0.	1.	0.9511	0.8090	0.5878	0.3090	0.
0.01	1.01	0.9278	0.7892	0.5734	0.3015	0.01
0.02	1.02	0.9137	0.7699	0.5594	0.2967	0.02
0.03	1.03	0.9044	0.7533	0.5465	0.2933	0.03

графічному вигляді:



Визначимо різницевий розв'язок *гіперболічного рівняння (ГР)*

$$u_{tt}(x,t) = c^2 u_{xx}(x,t) + f(x,t), \quad (6.15)$$

яке описує малі коливання натягнутої однорідної пружної струни із розподіленим навантаженням $f(x,t)$, у прямокутній області $G = [0 \leq x \leq a, 0 \leq t \leq T]$

за таких додаткових умов:

$$u(x,0) = \mu_1(x), \quad u_t(x,0) = \mu_2(x) \quad (0 \leq x \leq a); \quad (6.16)$$

$$u(0,t) = \mu_3(t), \quad u(a,t) = \mu_4(t) \quad (0 \leq t \leq T). \quad (6.17)$$

Тут шукана функція $u(x,t)$ — це відхилення струни від стану рівноваги.

Узявши шаблон у вигляді *хреста* (рис. 6.3) в околі точки (x_n, t_m) ,

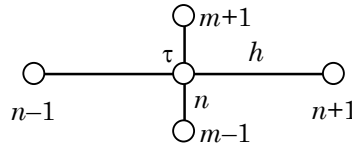


Рис. 6.3. Шаблон для побудови РС «хрест» гіперболічного рівняння

одержимо тришарову різницеву схему

$$\frac{1}{\tau^2} \left(\hat{y}_n - 2\overset{\vee}{y}_n + \underset{\vee}{y}_n \right) = \left(\frac{c}{h} \right)^2 (y_{n-1} - 2y_n + y_{n+1}) + f_n \quad (n = \overline{1, N-1}), \quad (6.18)$$

до якої потрібно додати крайові умови

$$y_0 = \mu_3(t), \quad y_N = \mu_4(t). \quad (6.19)$$

Розв'язок u на нульовому шарі знайдемо з першої початкової умови:

$$y_n^0 = \mu_1(x_n) \quad (n = \overline{0, N}), \quad (6.20)$$

а на першому шарі — із другої з урахуванням того, що

$$\frac{y_n^1 - y_n^0}{\tau} \approx u_t(x_n, 0) = \mu_2(x_n) \quad (n = \overline{1, N-1}).$$

Отже,

$$y_n^1 = \mu_1(x_n) + \tau \mu_2(x_n) \quad (n = \overline{1, N-1}). \quad (6.21)$$

Це явна різницєва схема, бо \hat{y} можна виразити через значення y на двох попередніх шарах, а починаючи з другого шару, знаходити за різницєвою схемою (6.18).

Описаний алгоритм показує, що різницєвий розв'язок існує і він єдиний. Якщо скористатись формулою (6.21), він матиме порядок $O(\tau + h^2)$.

Різницєва схема „хрест” стійка за виконання умови Куранта $c\tau \leq h$, тобто вона умовно стійка, а отже, забезпечує добру точність для розв'язків u , які мають неперервні четверті похідні за обома аргументами.

Приклад 6.3. Знайдемо різницєвим методом розв'язок гіперболічного задачі – задачі малих коливань однорідної пружної струни

$$u_{tt}(x,t) = u_{xx}(x,t) + \frac{t}{1+x}, \quad G = [0 \leq x \leq 0,5, 0 \leq t \leq 0,3]$$

з ДУ

$$u(x,0) = \sin \frac{\pi x}{2}, \quad u_t(x,0) = x^2 \quad (0 \leq x \leq 0,5);$$

$$u(0,t) = t^2, \quad u(0,5,t) = \frac{\sqrt{2}}{2} + t \quad (0 \leq t \leq 0,3).$$

Скористаємося явною різницєвою схемою «хрест», виконуючи умову стійкості Куранта $c\tau \leq h$.

Лістинг 6.3. Файл *giperbeq*:

```
// giperbeq
// Права частина рівняння
function u=f(x,t);
    u=t/(1+x);
endfunction;
// Перша початкова умова
function u=mu1(x);
    u=sin(%pi*x/2);
endfunction;
// Друга початкова умова
function u=mu2(x);
    u=x*x;
endfunction ;
// Ліва КУ
function u=mu3(t);
    u=t*t;
endfunction ;
// Права КУ
function u=mu4(t);
    u=sqrt(2)/2+t;
endfunction;
// Обчислення різницевого розв'язку гіперболічного
//рівняння за допомогою РС «хрест»
function [yt]=giperbeq(N,M,a,T,c);
// N і M - число підвідрізків по осях x і t відповідно,
// a, T - довжина відрізків по осях X і T відповідно,
// c - стала ГР. yt - розв'язок задачі
// Кроки за обома аргументами
h=a/N; tau=T/M;
// Сітка по X, заповнення 1-го й 2-го рядків ПУ
for n=1:N+1
    x(n)=(n-1)*h;    y(1,n)=mu1(x(n));
    y(2,n)=y(1,n)+tau*mu2(x(n));
end;
//Сітка по T, заповнення 1-го і останнього стовпчиків КУ
for m=1:M+1
    t(m)=(m-1)*tau;    y(m,1)=mu3(t(m));
    y(m,N+1)=mu4(t(m));
end;
// Параметр явної РС «хрест»
gam=(c*tau/h)^2;
// ЧР ГР за допомогою явної РС (6.18) - (6.21)
for m=2:M
    for n=2:N
```



```

        y(m+1,n)=-y(m-1,n)+gam*y(m,n-1)+2*(1-gam)*...
            y(m,n)+gam*y(m,n+1)+tau^2*f(x(n),t(m));
    end;
end;
// Формування значень ЧР для друку у вигляді таблиці
    for m=2:M+2
        for n=2:N+2
            yt(m,n)=y(m-1,n-1)
        end;
    end;
// Розміщення вузлів по X у першому рядку таблиці
    for n=2:N+2
        yt(1,n)=x(n-1);
    end;
// Розміщення вузлів по T у першому стовпчику таблиці
    for m=2:M+2
        yt(m,1)=t(m-1);
    end;
yt(1,1)=0;
// замінити це значення в таблиці на t/x вручну
endfunction;
//main program
N=5; M=3;
// Коментар до фактичних параметрів функції giperbeq
//[yt]=giperbeq(N,M,a, T ,c)
[yt]=giperbeq(N,M,0.5,0.3,1)
// Формування значень ЧР для побудови поверхні ЧР
for m=2:M+2
    for n=2:N+2
        y(m-1,n-1)=yt(m,n);
    end;
end;
    for n=2:N+2
        x(n-1)=yt(1,n);
    end;
        for m=2:M+2
            t(m-1)=yt(m,1);
        end;
// Побудова поверхні ЧР
mesh(x,t,y);
title('Z=Y(X,T)');
xlabel('x');
ylabel('t');

```

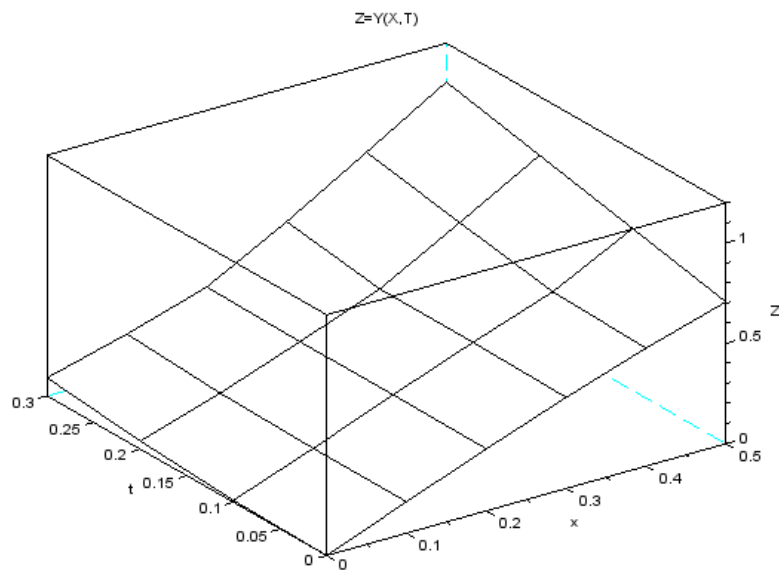
Результати виконання *giperbeq*:

у табличному вигляді:

$y(x, t)$

t \ x	0.	0.1	0.2	0.3	0.4	0.5
0.	0.	0.1564	0.3090	0.4540	0.5878	0.7071
0.1	0.01	0.1574	0.3130	0.4630	0.6038	0.8071
0.2	0.04	0.1675	0.3122	0.4636	0.6830	0.9071
0.3	0.09	0.1966	0.3197	0.5338	0.7683	1.0071

графічному вигляді:



7. Інтегральні рівняння

Якщо шукана (невідома) функція $u(x)$ знаходиться під знаком інтеграла, то рівняння називають *інтегральним (ІР)*. До ІР зводиться багато задач механіки. Найбільш вивченими є такі лінійні інтегральні рівняння:

а) *Фредгольма 2-го роду*:

$$u(x) - \lambda \int_a^b K(x, \xi) u(\xi) d\xi = f(x), \quad (7.1)$$

ядро якого задане на квадраті $a \leq x \leq b, a \leq \xi \leq b$,

б) *Вольтерра 2-го роду*:

$$u(x) - \lambda \int_a^x K(x, \xi) u(\xi) d\xi = f(x), \quad (7.2)$$

ядро якого задане на трикутнику $a \leq \xi \leq x \leq b$.

У разі розв'язування інтегральних рівнянь (7.1) *різницевим методом* (метод скінченних сум або квадратур) інтеграл замінюють на квадратурну формулу, що приводить до такої системи лінійних алгебричних рівнянь:

$$y_n - \lambda \sum_{m=1}^N C_m K_{nm} y_m = f_n \left[f_n = f(x_n); n = \overline{1, N} \right]. \quad (7.3)$$

Одержану систему лінійних алгебричних рівнянь з N до 150 розв'язують методом Гаусса. Система (7.3) має єдиний розв'язок, якщо λ не збігається з жодним характеристичним числом інтегрального рівняння Фредгольма 2-роду, і різниця між наближеним та точним розв'язками мала. Отже, цей алгоритм *добре обумовлений*.

Приклад 7.1. Знайдемо різницевим методом розв'язок ІР Фредгольма 2-го роду

вигляду $u(x) + \int_0^1 x e^t u(t) dt = e^{-x}$, яке має точний розв'язок $u(x) = e^{-x} - x/2$.

Лістинг 7.1. Файл *ief2*:

```
// ief2
// Розв'язання ІР Фредгольма 2-го роду РМ
function [dirf,u,y,x] = ief2(N,M,a,b,lam)
// Визначення ядра, ПЧ ІРФ та точного розв'язку
deff('KK = K(x,ksi)', 'KK = x*exp(ksi)')
deff('FF = f(x)', 'FF = exp(-x)')
deff('FU = fu(x)', 'FU = exp(-xx)-xx*0.5')
h=(b-a)/N; // Крок сітки
// Формування ваг та вузлів КФ
for n=1:N
    c(n)=h;
    ax(n)=a+(n-0.5)*h
end;
// Формування значень ПЧ ІР r та матриці d СЛАР (7.3)
for n=1:N
    xn=ax(n); r(n)=f(xn);
    for m=1:N
        d(n,m)=-lam*c(m)*K(xn,ax(m))
    end;
    d(n,n)=d(n,n)+1
end;
// Розв'язання (7.3) методом Гаусса із пакета Scilab
C=rref([d r]); [NN,MM]=size(C); R=C(:,MM);
// Обчислення ЧР ІР в М точках відрізка інтегрування
xx=a; hh=(b-a)/(M-1);
for m=1:M
    yy=f(xx);
    for n=1:N
        yy=yy+lam*c(n)*K(xx,ax(n))*R(n)
    end;
// Обчислення значень u(x) та ЧР
u(m)=fu(xx); y(m)=yy; x(m)=xx; xx=xx+hh;
```

```

end;
// Обчислення ДСКВ dirf ЧР від u(x)
dirf=0;
for k=1:M
    dirf=dirf+(u(k)-y(k))^2;
end;
dirf=sqrt(dirf/M);
// Перетворення векторів на матриці-рядки
x = x.'; u = u.'; y = y.';
endfunction;
// main program
// Коментар до фактичних параметрів функції ief2
//[dirf,u,y,x] = ief2(N ,M,a,b,lam)
[dirf,u,y,x] = ief2(11,5,0,1, -1)
// Побудова графіків заданої функції та ЧР
plot(x,y,'k*','x,u','k-'); xgrid();
xlabel(' Y=Y(X)          U=U(X)          ', 'X', 'Y          U');
legend(' Y=Y(X)  ', ' U=U(X) ');

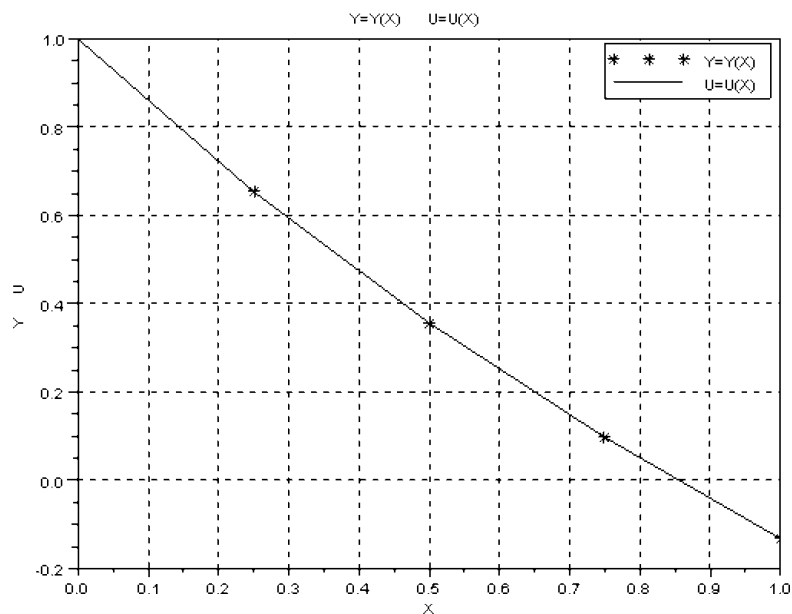
```

Результати виконання *ief2*:

у табличному вигляді:

x = 0.	0.25	0.5	0.75	1.
y = 1.	0.6537053	0.3563396	0.0970800	- 0.1325026
u = 1.	0.6538008	0.3565307	0.0973666	- 0.1321206
dirf =	0.0002340			

графічному вигляді:



Числовий розв'язок цього інтегрального рівняння Фредгольма 2-роду у разі застосування квадратурної формули середніх з кроком інтегрування $h = 0.25$ має середнє квадратичне відхилення від точного близько 0.0234 %.

Список рекомендованої літератури

- Мусяка, В.Г. Основи чисельних методів [Текст]: Підручник /В.Г. Мусяка — К.: Вища освіта, 2004. — 240 с.
- Поршнев, С.В. Вычислительная математика[Текст]: Курс лекцій /С.В. Поршнев. — СПб.: БХВ — Петербург, 2004. — 320 с.
- Алексеев, Е.Р. Scilab: Решение инженерных и математических задач [Текст]: пособие /Е.Р. Алексеев, О.В. Чеснокова, Е.А.Рудченко. — М.: ALT Linux; Бинوم. Лаборатория знаний, 2008. — 260 с.
http://window.edu.ru/window/library/pdf2txt?p_id=28168
- Павлова, М.И. Руководство по работе с пакетом Scilab [Текст] /М.И. Павлова. — http://www.csa.ru/~zebra/my_scilab/
- Тропин, И.С. Численные и технические расчеты в среде Scilab (ПО для решения задач численных и технических вычислений) [Текст]: Учеб. пособие /И.С. Тропин, О.И. Михайлова, А.В. Михайлов. — М.: 2008. — 65 с.
- Vaudin, M. Введение в Scilab [Текст]/ M. Vaudin; перевод с англ.:А. Glebov. 2010 г.
<http://forge.scilab.org/upload/docintrotoscilab/files/introscilab-v1.3-ru.pdf> .
<http://www.scilab.org>.Ехес-файл scilab-5.2.2.exe.

Список скорочень

БМ — балістичний метод	МД — метод дотичних
ВВ — власний вектор	МН — метод Ньютона
ВЗ — власне значення	МНК — метод найменших квадратів
ВФ — власна функція	МРК — метод Рунге—Кутти
ГР — гіперболічне рівняння	МРК2 — МРК 2-го порядку
ДСКВ — дискретне середнє квадратичне відхилення	ПР — параболічне рівняння
ДСКН — дискретне середнє квадратичне наближення	ПУ — початкова умова
ДУ — додаткові умови	РМ — різницевий метод
ЗДР — звичайні диференціальні рівняння	РС — різницева схема
ЗК — задача Коші	РЧП— рівняння в частинних похідних
ЗШЛ — задача Штурма-Ліувілля	СЛАР — система лінійних алгебричних рівнянь
ІПН — інтерполяційний поліном Ньютона	СНР — система нелінійних рівнянь
ІР — інтегральне рівняння	ТР — точний розв’язок
КЗ — крайова задача	ФП — формула парабол
ВКС — вільний кубічний сплайн	ФС — формула середніх
КУ — крайові умови	ФТ — формула трапецій
КФ — квадратурна формула	ХП — характеристичний поліном
МГ — метод Гаусса	ХР — характеристичне рівняння
МГЗ — метод Гаусса—Зейделя	ЧІ — числове інтегрування
	ЧР — числовий розв’язок

Основні елементи мови пакета *Scilab*

Пакет *Scilab* – діалогова система, яка дозволяє вводити команди й відстежувати результати їх виконання. Після пуску пакета *Scilab* в OS Windows за допомогою *scilab-5.2.2.exe* на екрані дисплея з'явиться наведене нижче командне вікно (консоль):

```
scilab-5.2.2
```

```
Консорціум Scilab (DIGITEO)
Авторські права належать INRIA, ©1989-2010
Авторські права належать ENPC, ©1989-2007
```

Команда запуску:
завантаження початкового середовища
-->

Символ `-->` є запрошення до введення команд. Після цього символу вводять команду або програму. Їх введення завершується натисненням на клавішу `<Enter>`.

Прикладом найпростішої програми мовою *Scilab* є програма *gauss*, яка реалізує метод Гаусса під час розв'язання СЛАР:

$$\begin{pmatrix} 1 & 2 & -3 \\ 4 & -1 & 1 \\ 2 & 6 & -1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 4 \\ 7 \end{pmatrix}.$$

```
// gauss
A=[1 2 -3; 4 -1 1; 2 6 -1]; b=[ 0; 4; 7];
C=rref([A b]); [NN,MM]=size(C); x=C(:,MM); x=x.'
```

Результат виконання програми *gauss*: `x = 1. 1. 1.`

Перший рядок – коментар, другий – формування матриці *A* та вектора *b* СЛАР. Перші три команди 3-го рядка саме й реалізують метод Гаусса розв'язання СЛАР, а 4-й – транспонує матрицю стовпчик (вектор) на матрицю-рядок для його друку у вигляді рядка.

Якщо команда завершується крапкою з комою (;), то результат роботи команди не виводиться на консоль, у іншому ж випадку результат виводиться. Довгі командні рядки можна розбивати на два, додаючи в місці розбиття три крапки (...).

Математичні вирази, як і знаки дій, записують подібно до інших мов програмування.

Як і в інших мовах програмування, об'єктами *Scilab* є скалярні величини, вектори (одновимірні масиви) та матриці (багатовимірні масиви).

Визначені в *Scilab* стандартні скалярні величини починаються з % (`%pi` - число π , `%e` – число $e=2.71828 \dots$).

Порядок формування матриць і векторів у *Scilab* показано на прикладі програми *gauss*. Індeksi елементів векторів та матриць починаються з одиниці. Імена змінних завжди починаються з букв.

У пакеті *Scilab* є вбудовані функції (*sin(x)*, *cos(x)*, ...), а втім їх можуть будувати і користувачі двома способами:

```
deff('f = ff(x)', 'f = log(1+x)'); y=ff(2)
або
function [f] =ff(x);
f = log(1+x)
endfunction; y=ff(2)
```

Результат виконання обох програм однаковий: $y = 1.0986123$.

У пакеті *Scilab* прийнята загальновідома конструкція циклу. Цикл, як і умовний оператор, завершується командою *end*, наприклад:

```
for n=n1:step:n2
// n - змінна циклу, n1 і n2 початкове і кінцеве значення змінної циклу n,
// step - крок змінювання i (якщо крок циклу step=1, його можна опускати)
  a=n*n; // тіло цикла
end
```

Приклад циклу *while*

```
x=0;
while x<10
x=x^2+1
end
```

Виконання циклів *while* і *for* можна перервати командою *break*, наприклад,

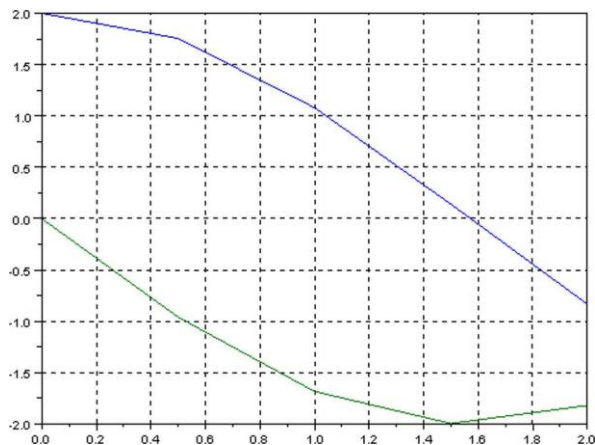
```
for i=1:n
  if (xj)>=xx(i) & xj<xx(i+1)) then
    k=i;
    break;
  end
end;
```

Застосування команди *plot* для побудови графіка показано у процесі розв'язання задачі коливання математичного маятника у пакеті *Scilab*

$$u''(x)+u(x)=0, \quad u(0)=2, \quad u'(0)=0$$

```
function dy = syst(t,y)
dy=zeros(2,1); dy(1) = y(2); dy(2) = - y(1);
endfunction;
x0=[2;0];t0=0;t=0:0.5:2;
y=ode(x0,t0,t,syst)
plot(t,y); xgrid();
```

```
y =
  2.      1.7551653      1.0806047      0.1414745      - 0.8322937
  0.      - 0.9588513      - 1.6829421      - 1.9949901      - 1.818595
```



На графіку верхня крива – розв'язок $u(x)$, а нижня крива – похідна від нього

Завдання для лабораторних робіт із застосуванням числових методів розв'язання задач

Вихідні дані для всіх завдань слід брати із відповідних таблиць згідно з одержаним шифром. Наприклад, якщо шифр 27964, то дані 1-го стовпчика необхідно брати з 2-го рядка, дані 2-го стовпчика – з 7-го рядка й т.д. Якщо у відповідній таблиці стовпчиків більше ніж 5, то потрібно взяти необхідну кількість перших цифр із шифру і доповнити ними заданий шифр. Наприклад, якщо у таблиці 8 стовпчиків, то новий шифр буде таким: 27964279.

Текст робіт слід набирати на комп'ютері за допомогою текстового редактора Word і роздруковувати на одному боці аркуша білого паперу формату А4 (210×297 мм), дотримуючись таких розмірів полів: верхнє, ліве і нижнє – не менше 20 мм, праве – не менше 10 мм. Розмір шрифту повинен відповідати 14 (або 12) пунктам Times New Roman, міжрядковий інтервал при цьому має бути полуторний.

Завдання 1. Інтерполяція функції поліномом Ньютона

Побудувати інтерполяційний поліном $P_n(x)$ у формі Ньютона для функції

$$y(x) = k_1 f_1(x) + k_2 f_2(x) + C$$

у точках x_i , значення яких слід шукати за співвідношеннями

$$a = 0.08 |k_1|, \quad b = 0.135 |k_2|, \quad \zeta = a + k_3 (b-a), \quad x_i = a + i h \quad [i = \overline{0, n}; h = (b-a)/n].$$

Необхідні дані взяти із табл. 1 відповідно до шифру. Значення сталої C підібрати так, щоб знак $y(x)$ на відрізку $[a, b]$ не змінювався й на кінцях відрізка функція не дорівнювала нулю.

Одержаний ПН $P_n(x)$ застосувати для обчислення наближених значень $y(x)$ у M точках, зокрема, серед них повинні бути вузли інтерполяції та точка ζ , у якій до того ж необхідно обчислити значення абсолютної похибки $P_n(\xi) - y(\xi)$ та похибки інтерполяції $r_n(\zeta)$.

Побудувати графіки $y(x)$ та $P_n(x)$ за обчисленими їх M значеннями.

Завдання 2. Інтерполяція функції вільними кубічними сплайнами

Побудувати вільні кубічні сплайни для функції з попереднього завдання, взявши за вузли точки $x_i = a + i h \quad [i = \overline{0, N}; h = (b-a)/N]$, й застосувати їх для обчислення значень $y(x)$ у тих же точках, що й у завданні 1, та побудувати за обчисленими M значеннями графіки $y(x)$ та $s(x)$.

Завдання 3. Наближення функції методом найменших квадратів

Визначити коефіцієнти алгебричного полінома

$$\varphi(x; C) = \sum_{k=0}^n C_k x^k,$$

який апроксимує в середньому таблично задану функцію $y(x)$ (із завдання 1) за знайденими її значеннями у точках $x_i = a + (i-1)h \quad [i = \overline{1, N}; h = (b-a)/(N-1)]$.

Побудовану функцію $\varphi(x)$ застосувати для обчислення наближених значень $y(x)$ у тих же точках, що й у завданні 1, визначити середнє квадратичне відхилення $\varphi(x)$ від $y(x)$ та побудувати графіки функцій $y(x)$ і $\varphi(x)$.

Завдання 4. Числове інтегрування

Для заданої функції $y(x)$ із завдання 1 обчислити значення інтеграла $F = \int_a^b y(x) dx$,

застосовуючи КФ на рівномірних сітках $x_i = a + i h \quad [i = \overline{0, n}; h = (b-a)/n]$.

Завдання 5. Розв'язання нелінійного рівняння з одним невідомим

Визначити розв'язок x рівняння $f(x) \equiv k_1 f_1(x) + k_2 f_2(x) + D = 0$ методом дотичних та одним із ітераційних методів (МПД, МПІ, МС) на відрізку $[a, b]$ з точністю ε (необхідні дані взяти із завдання 1). Значення сталої D підібрати так, щоб на кінцях відрізка $[a, b]$ знак $y(x)$ змінювався.

Завдання 6. Розв'язання систем нелінійних рівнянь

Застосовуючи метод Ньютона та метод простих ітерацій, визначити один із розв'язків $x = (x_1, x_2)$ системи нелінійних рівнянь з точністю ε

$$f_1(x_1, x_2) = 0, \quad f_2(x_1, x_2) = 0. \quad (1)$$

Нульове наближення розв'язку знайти графічним способом. Функції $f_1(x_1, x_2), f_2(x_1, x_2)$, що входять у (1), взяти з табл. 2 відповідно до шифру. Якщо графіки функцій $f_1(x_1, x_2), f_2(x_1, x_2)$ не перетинаються, то, як і в завданні 5, до однієї з них слід додати сталу D , щоб вони перетнулись.

Завдання 7. Розв'язання СЛАР методом Гаусса—Зейделя

Побудувати СЛАР з матрицею A та правою частиною – вектором b за співвідношеннями

$$A = \begin{pmatrix} 3 & 2 & 1 \\ 2 & 6 & 2 \\ 1 & 2 & 2 \end{pmatrix} + 0.1 \text{ I N} \begin{pmatrix} 1 & -1 & 1 \\ -1 & 1 & 0 \\ 1 & 0 & 2 \end{pmatrix}, \quad b = \begin{pmatrix} 3 \\ 6 \\ 2 \end{pmatrix} + 0.1 \text{ I N} \begin{pmatrix} 1 \\ -1 \\ 1 \end{pmatrix}$$

та розв'язати її ітераційним методом Гаусса—Зейделя. Тут I - порядковий номер студента у журналі групи, а N – номер групи.

Завдання 8. Розв'язання алгебричної проблеми власних значень

Застосувати ітераційний (степеневий) метод для визначення найбільшого ВЗ матриці A із попереднього завдання і відповідний йому ВВ.

Завдання 9. Розв'язання задачі Коші для ЗДР методом Рунге—Кутти

За допомогою МРК 2-го порядку ($\alpha=1$) розв'язати задачу Коші для ЗДР

$$u''(x) = f(x, u(x)), \quad u(\xi) = \eta_1, \quad u'(\xi) = \eta_2 \quad \text{на відрізку } [\xi, X].$$

Для побудови умов завдання скористатися даними табл. 1 та співвідношеннями

$$f(x, u) = f_1(x) f_2(u); \quad \xi = |k_1|/10, \quad \eta_1 = k_1, \quad \eta_2 = k_2, \quad X = (1 + |k_2|) \xi.$$

Побудувати графік числового розв'язку $u(x)$.

Завдання 10. Розв'язання крайової задачі для ЗДР різницевим методом

Розв'язати різницевим методом лінійну крайову задачу 2-го порядку

$$u''(x) + p(x)u'(x) + q(x)u(x) = f(x) \quad (x \in [a, b]), \quad u(a) = A, \quad u'(b) = B.$$

Для побудови умов завдання скористатися даними табл. 1 та співвідношеннями

$$p(x) = f_1(x), \quad q(x) = f_2(x), \quad f(x) = f_1(x) f_2(x); \\ a = |k_1|/10, \quad b = (1 + |k_2|)a, \quad A = k_1, \quad B = k_2, \quad x_i = a + i h \quad [i = \overline{0, n}; \quad h = (b-a)/n].$$

Побудувати графік числового розв'язку $u(x)$.

Завдання 11. Розв'язання задачі на ВЗ для ЗДР балістичним методом

Розв'язати задачу на ВЗ 2-го порядку для ЗДР балістичним методом

$$u''(x) + p(x)u'(x) + (\lambda + q(x))u(x) = 0 \quad (x \in [a, b]), \quad u(a) = 0, \quad u'(b) = 0.$$

Для побудови умов завдання скористатися даними табл. 1 та співвідношеннями

$$p(x) = f_1(x), \quad q(x) = f_2(x); \quad a = |k_1|/10, \quad b = (1 + |k_2|)a, \quad x_i = a + i h \quad [i = \overline{0, n}; \quad h = (b-a)/n].$$

Побудувати графік числового розв'язку $u(x)$.

Завдання 12. Розв'язання крайової задачі для рівняння еліптичного типу

Знайти різницевим методом, скориставшись процесом Лібмана, розв'язок крайової еліптичної задачі – задачі Діріхле для рівняння Лапласа $u_{,11}(x) + u_{,22}(x) = 0$, що визначає

стаціонарне розподілення температури в прямокутній області $\{g(x) = [a_i \leq x_i \leq b_i]; i = 1, 2\}$ з КУ

$$u(x_1, 0) = f_d(x_1), \quad u(x_1, b_2) = f_u(x_1), \quad u(0, x_2) = f_l(x_2), \quad u(b_1, x_2) = f_r(x_2).$$

За нульові наближення розв'язку у внутрішніх вузлах сітки взяти середнє арифметичне його значень у напрямку осей (x_1, x_2) , знайдених лінійним інтерполюванням, скориставшись заданими крайовими умовами. Побудувати поверхню числового розв'язку $u(x_1, x_2)$.

Вирази для $f_d(x_1), f_u(x_1), f_l(x_2), f_r(x_2)$ та значення $[a_i, b_i]$ ($i = 1, 2$) взяти з табл. 3.

Завдання 13. Розв'язання рівняння теплопровідності різницевим методом

Використати різницевий метод до розв'язання нестационарної задачі теплопровідності з однією просторовою змінною $u_t(x, t) = k u_{xx}(x, t) + f(x, t)$ у прямокутній області $G = [0 \leq x \leq a, 0 \leq t \leq T]$ за таких додаткових умов:

$$u(x, 0) = \mu(x) \quad (0 \leq x \leq a); \quad u(0, t) = \mu_1(t), \quad u(a, t) = \mu_2(t) \quad (0 \leq t \leq T).$$

Вирази для функцій $f(x, t)$, $\mu(x)$, $\mu_1(t)$, $\mu_2(t)$ та значення k , a , T взяти з табл. 4. Побудувати поверхню числового розв'язку $u(x, t)$.

Завдання 14. Розв'язання рівняння коливань струни різницевою методом

Застосувати різницевий метод до розв'язання рівняння малих коливань натягнутої однорідної пружної струни $u_{tt}(x, t) = c^2 u_{xx}(x, t) + f(x, t)$, у прямокутній області $G = [0 \leq x \leq a, 0 \leq t \leq T]$ за таких додаткових умов:

$$u(x, 0) = \mu_1(x), \quad u_t(x, 0) = \mu_2(x) \quad (0 \leq x \leq a); \quad u(0, t) = \mu_3(t), \quad u(a, t) = \mu_4(t) \quad (0 \leq t \leq T).$$

Вирази для функцій $f(x, t)$, $\mu_1(x)$, $\mu_2(x)$, $\mu_3(t)$, $\mu_4(t)$ та значення c^2 , a , T взяти з табл. 5. Побудувати поверхню числового розв'язку $u(x, t)$.

Завдання 15. Розв'язання інтегрального рівняння Фредгольма 2-го роду різницевою методом

$$\text{Розв'язати інтегральне рівняння Фредгольма 2-го роду } u(x) - \lambda \int_a^b K(x, \xi) u(\xi) d\xi = f(x),$$

застосовуючи різницевий метод. Побудувати графік числового розв'язку $u(x)$.

Вирази для $K(x, \xi)$ і $f(x)$ та значення сталих λ , a , b визначають так:

$$K(x, \xi) = f_1(x) f_2(\xi), \quad f(x) = f_2(x) \quad a = |k_1|, \quad b = |k_2|, \quad \lambda = k_3.$$

Функції $f_1(x)$ і $f_2(\xi)$ та значення сталих взяти з табл. 1.

Таблиця 1

Рядок	Стовпчик				
	1	2	3	4	5
	$f_1(x)$	$f_2(x)$	k_1	k_2	k_3
1	$1+x^2/2$	$1/x+1/6$	-1.5	6.5	1/3
2	$1-x^2/2$	$1/x-1/6$	2.0	-7.0	1/4
3	$1+x^2/3$	$1/x^2+1/3$	-2.5	7.5	1/5
4	$1-x^2/3$	$1/x^2-1/3$	3.0	-8.0	1/6
5	$1+x^3/2$	$1/x+1/2$	-3.5	8.5	1/7
6	$1-x^3/2$	$1/x-1/2$	4.0	-9.0	1/8
7	$1+x^3/3$	$1/x+1/3$	-4.5	9.5	1/9
8	$1-x^3/3$	$1/x-1/3$	5.0	-10.0	2/9
9	$1+x^2/6$	$1/x^2+1/2$	-5.5	10.5	2/3
0	$1-x^2/6$	$1/x^2-1/2$	6.0	-11.0	2/5

Таблиця 2

Рядок	Стовпчик			
	1	2	3	4
	$f_1(x_1, x_2)$	$f_2(x_1, x_2)$	a	b
1	$\sin(x_1+x_2) - x_1/b$	$x_1^2 + x_2^2 - b^2$	1.5	6.5
2	$\sin(x_1-x_2) - x_1x_2 - a/b$	$x_1^2 + x_2^2 - a^2$	2.0	7.0
3	$\cos(x_1+x_2) - x_2/a$	$x_1^2 + x_2^2/b - 1$	2.5	7.5
4	$\cos(x_1-1/a)+x_2-1/b$	$x_1^2 + x_2^2 a^2 - a/b$	3.0	8.0
5	$\sec(x_1x_2+a/b) - x_1 x_1$	$x_1^2/b + x_2^2/a - 1$	3.5	8.5
6	$\operatorname{cosec}(x_1+x_2/b) - x_2x_2$	$x_1^2 + x_2^2 - b/a$	4.0	9.0
7	$\sec(x_1 x_2 - a/b) - x_1 x_1$	$x_1^2/a^2 + x_2^2/b^2 - 1$	4.5	9.5
8	$\sin(x_1x_2+a/b) - a/b$	$x_1^2/b + x_2^2 - 1$	5.0	10.0
9	$\cos(x_1-a/b)+x_2-1$	$x_1^2/b^2 + x_2^2/a^2 - 1$	5.5	10.5
0	$\cos(x_1 x_2) - a$	$x_1^2/a + x_2^2/b - 1$	6.0	11.0

Таблиця 3

Рядок	Стовпчик							
	1	2	3	4	5	6	7	8
	$f_d(x)$	$f_u(x)$	$f_l(x)$	$f_r(x)$	a_1	b_1	a_2	b_2
1	x	$3x^2$	$3x^2$	x	1.25	4.75	1.25	6.25
2	x^2	$4x^3$	$4x^3$	x^2	1.50	4.00	1.50	6.50
3	x^3	$3x/2$	$3x/2$	x^3	1.75	4.25	1.75	6.75
4	$x/2$	x	x	$x/2$	2.00	4.50	2.00	7.00
5	$x^2/3$	x^2	x^2	$x^2/3$	2.25	5.25	2.25	7.25
6	$x^3/4$	x^3	x^3	$x^3/4$	2.50	5.50	2.50	5.50
7	$2x$	$x/2$	$x/2$	$2x$	2.75	5.75	2.75	5.75
8	$3x^2$	$x^2/3$	$x^2/3$	$3x^2$	3.00	6.00	3.00	5.00
9	$4x^3$	$x^3/4$	$x^3/4$	$4x^3$	3.25	6.25	3.25	4.25
0	$3x/2$	$x/2$	$x/2$	$3x/2$	3.50	6.50	3.50	4.50

Примітка. Для одержання остаточної виразів КУ їх слід помножити на $(x_i - a_i)(x_i - b_i)$ ($i = 1, 2$).

Таблиця 4

Рядок	Стовпчик						
	1	2	3	4	5	6	7
	$f(x, t)$	$\mu(x)$	$\mu_1(t)$	$\mu_2(t)$	k	a	T
1	xt	$x(x+1)$	$1-t$	0.1	1	0.2	0.1
2	$(xt)^2$	$(x+1)^2$	$2+t$	0.15	1.5	0.3	0.15
3	$\sin xt$	$\sin 2x$	$3t$	0.2	2	0.5	0.2
4	$\cos xt$	$\cos 3x$	$4t+1$	0.25	2.5	0.25	0.25
5	$tg xt$	$tg x$	$6(t+1)$	0.3	3	0.35	0.3
6	$lg xt$	$lg x$	$t-3$	0.35	3.5	0.55	0.35
7	$\ln xt$	$\ln x$	$2t$	0.4	4	0.4	0.4
8	$t \sin xt$	$t \sin x$	$3t+1$	0.45	4.5	0.45	0.45
9	$x \cos xt$	$x \cos x$	$t/2+3$	0.5	5	0.65	0.5
0	$xt \, tg \, xt$	$x \, tg \, x$	$t/3+4$	0.6	6	0.6	0.6

Таблиця 5

Рядок	Стовпчик							
	1	2	3	4	5	6	7	8
	$f(x, t)$	$\mu_1(x)$	$\mu_2(x)$	$\mu_3(t)$	$\mu_4(t)$	c^2	a	T
1	xt	$x(x+1)$	x^2+1	$1-t$	$0.1t^2$	1	0.2	0.1
2	$(xt)^2$	$(x+1)^2$	$2x+1$	$2+t$	$0.15t$	1.21	0.3	0.15
3	$\sin xt$	$\sin 2x$	$x^2(x+1)$	$3t$	$0.2t^2$	1.44	0.5	0.2
4	$\cos xt$	$\cos 3x$	$2x^2$	$4t+1$	$0.25t$	1.69	0.25	0.25
5	$tg xt$	$tg x$	x^2+x	$6(t+1)$	$0.3t^2$	1.96	0.35	0.3
6	$lg xt$	$lg x$	$1-x^2$	$t-3$	$0.35t$	2.25	0.55	0.35
7	$\ln xt$	$\ln x$	$1.2x+x^2$	$2t-1$	$0.4t^2$	2.56	0.4	0.4
8	$t \sin xt$	$\sin x$	$(x+1)\sin x$	$3t+1$	$0.45t$	2.89	0.45	0.45
9	$x \cos xt$	$x \cos x$	$x(1-x^2)$	$t/2+3$	$0.5t^2$	3.24	0.65	0.5
0	$xt \, tg \, xt$	$x \, tg \, x$	x^2+2x	$t/3+4$	$0.6t$	3.61	0.6	0.6

Структура лабораторної роботи та приклад її оформлення

(Окрема стор. - 1)

Міністерство освіти і науки України
Дніпропетровський національний університет
імені Олеся Гончара

Механіко-математичний факультет
Кафедра обчислювальної механіки та міцності конструкцій

ЛАБОРАТОРНА РОБОТА
з числових методів
Інтерполяція функції поліномом Ньютона

Виконавець студент групи (код групи) Шифр 88888
Керівник (посада викладача) (Прізвище й ініціали)
(Прізвище й ініціали)

Дніпропетровськ
Рік виконання (наприклад, 2012)

=====

(З нової стор. - 2)

З М І С Т

Постановка задачі	3
Реферат	4
Вступ	
1. Інтерполяційний поліном Ньютона	
2. Опис обчислювального алгоритму	
3. Обговорення результатів	
Висновки	
Список посилань	
Додатки:	
А. Опис вихідних даних та результатів розрахунку	
Б. Лістинг програми	
В. Схема обчислювального алгоритму	

=====

(З нової стор. - 3)

Постановка задачі

Знайти наближене значення функції $y(x) = 5 \cdot \sin x - 10 \cdot 3^{-x}$ у точках $\xi = 0.60$ та $x = 0.6375$,
1.1125 за допомогою ПН (1.1), побудованого за її відомими трьома значеннями у точках
 $x_0 = a = 0.4$, $x_1 = 0.875$, $x_2 = b = 1.35$ ($k_1 = 5$, $k_2 = -10$, $k_3 = 2/9$).

Оцінити похибку інтерполяції в точці $\xi = 0.60$ за формулою (1.2).

Окрім того, знайти $y(\xi) - P_2(\xi)$ і порівняти з $|r_2(\xi)|$.

=====

(З нової стор. - 4)

РЕФЕРАТ

Задача наближення функцій взагалі полягає в заміні аналітично або таблично заданої функції $y(x)$ зручною для обчислень апроксимувальною функцією $\varphi(x)$, яка б для всіх необхідних нам значень аргументу задовольняла співвідношення $y(x) \approx \varphi(x)$.

Цієї мети досягають введенням вектора вільних параметрів (невдомих коефіцієнтів) $C = (C_1, \dots, C_n)$ в $\varphi(x)$ [тобто $\varphi(x) = \varphi(x; C)$] і його визначенням з прийнятої умови близькості функцій $y(x)$ та $\varphi(x)$. Звичайно, вважається, що у випадку табульованої функції $y(x)$ її графік є плавна крива [1].

Інтерполяція

Якщо за умову близькості $\varphi(x)$ і $y(x)$ взяти їх збіжність в n точках (вузлах інтерполяції) $x_i (i=\overline{1,n})$, то C визначають як результат розв'язання такої системи алгебричних рівнянь (САР):

$$\varphi(x_i; C) = y(x_i) \quad (i=\overline{1,n}). \quad (1)$$

Це так звана лагранжова інтерполяція.

Якщо компоненти вектора $C_i (i=\overline{1,n})$ входять до виразу $\varphi(x_i; C)$ лінійно, тоді інтерполяцію називають лінійною, у протилежному випадку - нелінійною.

Лінійна інтерполяція

Задача істотно спрощується, якщо $\varphi(x)$ має вигляд узагальненого полінома

$$\varphi(x; C) = \sum_{k=1}^n C_k \varphi_k(x), \quad (2)$$

де $\varphi_k(x)$ - система лінійно незалежних функцій.

Підставивши (2) до (1), одержимо для визначення C таку СЛАР:

$$\sum_{k=1}^n C_k \varphi_k(x_i) = y(x_i) \quad (i=\overline{1,n}). \quad (3)$$

Для єдиності розв'язку задачі інтерполяції визначник цієї СЛАР (3) не повинен перетворюватись на нуль (природно, серед вузлів інтерполяції не повинно бути збіжних).

Найзручнішими для обчислень функцій є алгебричні поліноми $\varphi_k(x) = x^k \quad (i = \overline{0,n})$. Причому для зручності вузли інтерполяції нумерують від 0 до n .

Про застосування та збіжність інтерполяції

Окрім основної задачі інтерполяції (відновлення значення функції за відомими) її застосовують для тестування таблиць на наявність несистематичних похибок (друга ПР поводить тоді себе хаотично), субтабулювання (зменшення кроку існуючої таблиці) та оберненої інтерполяції (знаходження значення аргументу для заданого чи відомого значення функції).

Похибку алгебричної інтерполяції можна зменшити двома шляхами - зберегти степінь полінома, а крок зменшити, або зберегти крок, збільшивши кількість використовуваних вузлів. У практичних розрахунках збільшення n небажане (бо призводить до збільшення похибки), тому для підвищення точності інтерполяції доцільно обмежитися 3 - 5 вузлами (точніше, вільними параметрами) і зменшити крок таблиці, не змінюючи n .

Вступ

Необхідність застосування наближення функцій виникає в механіці для обробки експериментальних даних або числового розв'язання задачі, тобто для відновлення значень деякої функції у точках, цікавих для нас, за відомими її значеннями.

1. Інтерполяційний поліном Ньютона

Для побудови ППН вводять так звані поділені різниці (ПР)

$$y(x_i, x_j) = \frac{y(x_i) - y(x_j)}{x_i - x_j}, \quad y(x_i, x_j, x_k) = \frac{y(x_i, x_j) - y(x_j, x_k)}{x_i - x_k}.$$

Оскільки ПР мають вимірність похідних, то інколи їх використовують як наближені значення останніх. Якщо побудувати ПР полінома n -го степеня, можна, скориставшись умовами збіжності полінома з заданою функцією та її ПР у вузлах інтерполяції, одержати такий вираз для ППН:

$$y(x) \approx y(x_0) + \sum_{k=1}^n y(x_0, x_1, \dots, x_k) \prod_{i=0}^{k-1} (x - x_i). \quad (1.1)$$

Наведений тут ППН використовують у випадку таблиць зі змінним кроком. Похибку інтерполяції в точці x оцінюють за формулою

$$|r_n(x)| \leq |y(x) - P_n(x)| = M_{n+1} |\omega(x)| / (n+1)!, \quad (1.2)$$

де $M_{n+1} = \max |y^{(n+1)}(x)|$ на відрізку $[a, b]$.

Для обчислення ПР складають таблицю ПР функції $y(x)$, а потім її наближені значення знаходять за формулою (1.1).

2. Опис обчислювального алгоритму

Реалізуючи алгоритм знаходження ПН за формулою (1.1) для зменшення необхідної пам'яті ЕОМ, розміщуємо значення ПР на місці попередньо знайдених значень вихідної функції $y(x)$. Крім того, витрати машинного часу зменшаться, якщо ПН знаходити за схемою Горнера. Оскільки значення ПН обчислюють неодноразово, то цей процес оформлюємо у вигляді функції.

3. Обговорення результатів

Із порівняння наведених у лістингу точних і наближених значень функції $y(x)$ видно, що вони відрізняються третіми значущими цифрами [$y(\xi) - P_2(\xi) = 0.014$], а значення похибки наближення $|r_2(\xi)|$ визначаємо за формулою (1.2), вона дорівнює 0.027 .

=====
(З нової стор.)

Висновки

Отже, ураховуючи п. 2 і 3, на практиці слід надавати перевагу ПН, оскільки формула (1.1) дозволяє одержати прийнятні результати, а за збільшення кількості вузлів не потрібно перерахувати всі доданки у (1.1), до того ж порядок нумерації вузлів тут не має значення.

=====
(З нової стор.)

Список посилань

1. Мусіяка, В.Г. Основи чисельних методів механіки [Текст]: підручник / В.Г. Мусіяка — К.: Вища освіта, 2004. — 240 с.
2. Мусіяка, В.Г. Практикум з обчислювальної механіки [Текст] / В.Г. Мусіяка. — Д.: РВВ ДНУ, 2006. — 64 с.

=====
(З нової стор.)

Додатки

А. Опис вихідних даних і результатів розрахунку

Вихідні дані для побудови ПН (вузли і значення в них заданої функції)
 $\mathbf{xx} = [0.4, 0.875, 1.35]$; $\mathbf{yy} = [-4.4968, 0.0137, 2.6093]$;
та координати точок для побудови таблиці й графіків заданої функції й ПН
 $\mathbf{xd} = [0.4, 0.6, 0.6375, 0.875, 1.1125, 1.35]$.

Результати виконання програми друкують в такому вигляді:

```
 $\mathbf{x} =$  0.4 0.6 0.6375 0.875 1.1125 1.35  
 $\mathbf{y} =$  - 4.4968 - 2.3496 - 1.9881 0.01371 1.5382 2.6093  
 $\mathbf{p} =$  - 4.4968 - 2.3643 - 2.0022 0.01371 1.5509 2.6093
```

(З нової стор.)

Б. Лістинг програми

```
// ipn  
function [pln] = plnwt(n, xp, xx, yy);  
y=yy(n+1);  
for k=1:n  
    n1=n-k+1;  
    y=yy(n1)+y*(xp-xx(n1));  
end;  
pln=y  
endfunction;  
function [p, y, x] = ipn(n, M, xx, yy, xd);  
deff('c = f(x)', 'c = 5*sin(x) - 10/exp(x*log(3))');  
m=n+1;
```

```

for k=1:m
    nt=n-k+1;
    for i=1:nt
        n1=n-i+1;  n2=n1+1;
        yy(n2)=(yy(n2)-yy(n1))/(xx(n2)-xx(n1-k+1));
    end
end;
for k=1:M
    xp=xd(k);  p(k)=plnwt(n,xp,xx,yy);
    x(k)=xp;  y(k)=f(xp);
end;
x=x.'; y=y.'; p=p.';
endfunction;
// main program
xx=[ 0.4, 0.875, 1.35];
yy=[-4.4968484,0.0137085,2.6093461];
xd=[0.4,0.6,0.6375,0.875,1.1125,1.35];
//[p,y,x] = ipn(n,M,xx,yy,xd)
[p,y,x] = ipn(2,6,xx,yy,xd)
plot(x,p,'k*',x,y,'k-');
xgrid();
xlabel(' P=P(X)      Y=Y(X) ', 'X', 'P      Y');
legend(' P=P(X) ', ' Y=Y(X) ', 4, %t);

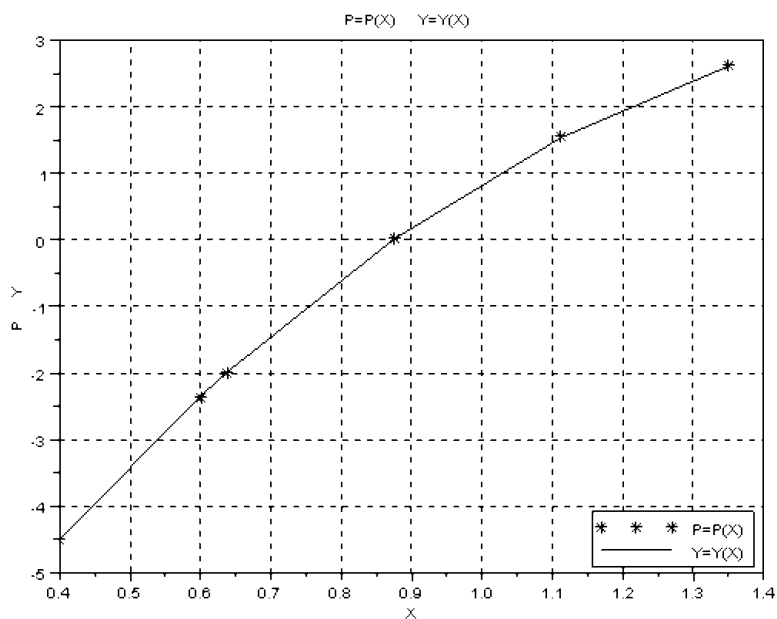
```

Результати виконання *ipn*:

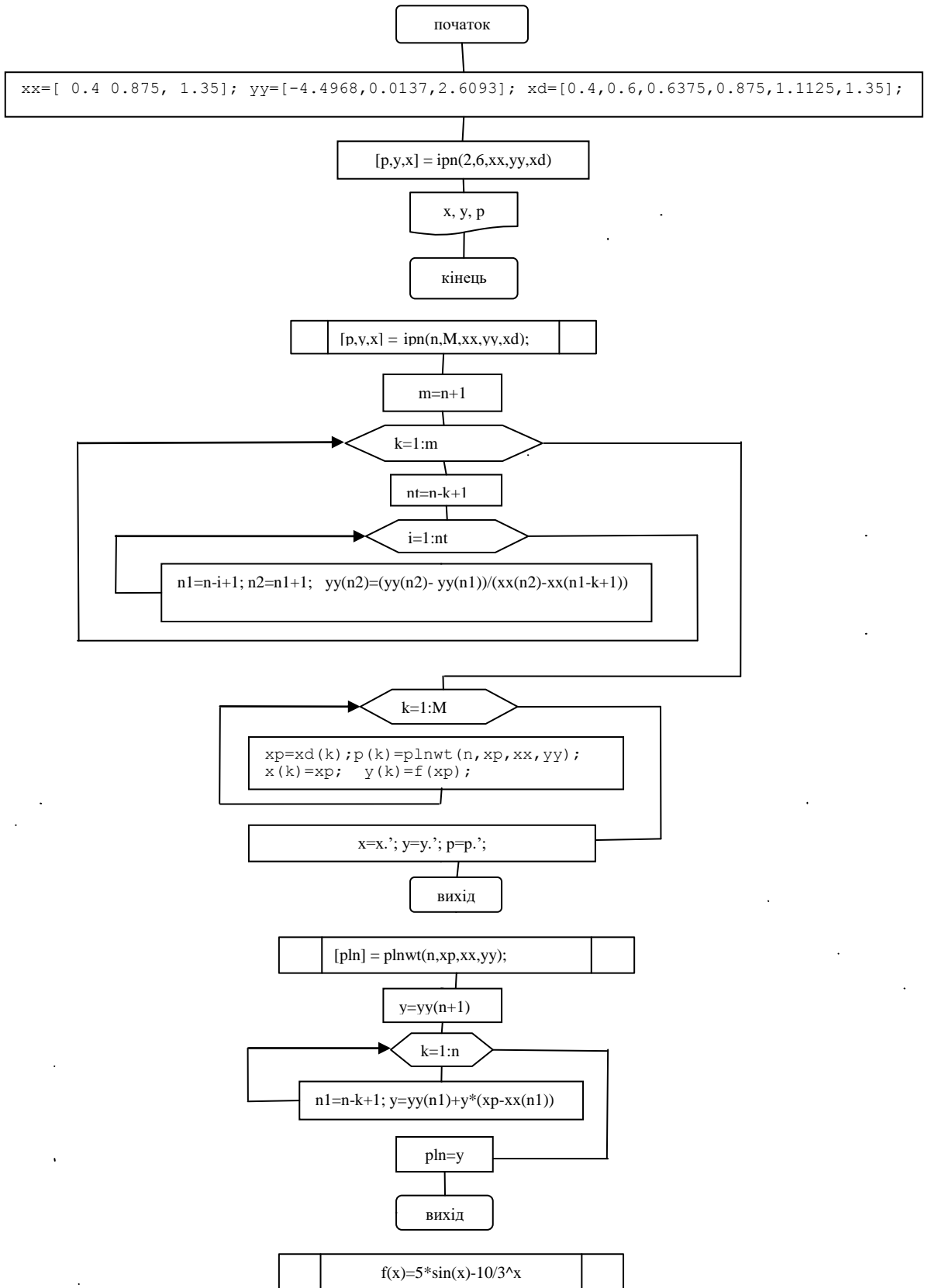
у табличному вигляді:

x =	0.4	0.6	0.6375	0.875	1.1125	1.35
y =	- 4.4968	- 2.3496	- 1.9881	0.01371	1.5382	2.6093
p =	- 4.4968	- 2.3643	- 2.0022	0.01371	1.5509	2.6093;

графічному вигляді:



В. Схема обчислювального алгоритму



З М І С Т

Вступ	3
1. Наближення функцій	4
2. Числове інтегрування	12
3. Нелінійні алгебричні рівняння	15
4. Задачі лінійної алгебри	18
5. Звичайні диференціальні рівняння	25
6. Рівняння в частинних похідних	36
7. Інтегральні рівняння	49
Список рекомендованої літератури	52
Список скорочень	52
Додатки	53
Додаток 1. Основні елементи мови пакета <i>Scilab</i>	53
Додаток 2. Завдання для лабораторних робіт із застосуванням числових методів розв'язання задач	55
Додаток 3. Структура лабораторної роботи та приклад її оформлення	59