ЛЕКЦІЯ 6: ЧИСЕЛЬНЕ ДИФЕРЕНЦІЮВАННЯ

Теоретичні засади чисельного диференціювання. Чисельні методи розв'язування диференціальних рівнянь та їх систем. Практичні засади чисельного диференціювання та розв'язування диференціальних рівнянь та їх систем у Scilab.

Задача чисельного диференціювання полягає в знаходженні значень похідних від функції $y = f(x)$ в заданих точках у випадках, коли її аналітичний вигляд невідомий (задана неявно), дуже складний або функція задана таблично. Привабливість чисельного підходу пояснюється наявністю простих формул, за допомогою яких похідні в заданих точках можна з достатньою точністю наближено обчислити з допомогою кількох значень функції $f(x)$ в цих та близьких до них точках.

Теоретичні засади чисельного диференціювання

Нехай функція $y = f(x)$ задана таблицею своїх значень $y_i = f(x_i)$ у рівновіддалених вузлах $x_i = x_0 \pm ih$ $(i = 0, \pm 1, \pm 2, ...)$. Для зручності нумерацію точок проводимо відносно точки x_0, у якій потрібно обчислити похідну. Виберемо множину n суміжних з x_0 вузлів x_i $(i = \pm 1, \pm n/2)$ і побудуємо на них інтерполяційний многочлен $I_n(x)$. Користуючись тим, що похідну від многочлена брати легко, можемо побудувати наближені формули чисельного диференціювання функції для похідних до порядку n. Тобто шукана похідна першого порядку від функції може бути наближено обчислена за формулою $f'(x_0) \cong I'_n(x_0)$, другого – $f''(x_0) \cong I''_n(x_0)$ і т.д. Як $I_n(x)$ зручно вибирати інтерполяційний поліном Лагранжа або Ньютона. Обмежуючи кількість суміжних вузлів у залежності від порядку похідної та потрібної точності наведемо кілька формул:

диференціювання вперед:

$$f'(x_0) = \frac{f(x_1) - f(x_0)}{h};$$

$$f'(x_0)=\frac{-f(x_2)+4f(x_1)-3f(x_0)}{2h};\qquad (6.1)$$

$$f''(x_0)=\frac{f'(x_1)-f'(x_0)}{h}=\frac{f(x_2)-2f(x_1)+f(x_0)}{h^2};$$

$$f''(x_0)=...=\frac{f(x_3)+4f(x_2)-5f(x_1)+2f(x_0)}{h^2};\ldots\qquad (6.2)$$

диференціювання назад:

$$f'(x_0)=\frac{f(x_0)-f(x_{-1})}{h};$$

$$f'(x_0)=\frac{3f(x_0)-4f(x_{-1})+f(x_{-2})}{2h};\qquad (6.3)$$

$$f''(x_0)=...=\frac{f(x_0)-2f(x_{-1})+f(x_{-2})}{h^2};$$

$$f''(x_0)=...=\frac{2f(x_0)-5f(x_{-1})+4f(x_{-2})-f(x_{-3})}{h^2};\qquad (6.4)$$

диференціювання симетричне:

$$f'(x_0)=\frac{f(x_1)-f(x_{-1})}{2h};$$

$$f'(x_0)=\frac{-f(x_2)+8f(x_1)-8f(x_{-1})+f(x_{-2})}{12h};\qquad (6.5)$$

$$f''(x_0)=...=\frac{f(x_1)-2f(x_0)+f(x_{-1})}{h^2};$$

$$f''(x_0)=...=\frac{-f(x_2)+16f(x_1)-30f(x_0)+16f(x_{-1})-f(x_{-2})}{h^2}\qquad (6.6)$$

У всіх наведених формулах, побудованих на двох сусідніх до x_0 вузлах, точність наближення складає $O(h)$, на трьох – $O(h^2)$, $O(h^3)$, на чотирьох – $O(h^4)$. Зрозуміло, що чим більше сусідніх до точки x_0 вузлів зберігати у інтерполяційному поліномі $I_n(x)$, тим точнішим буде чисельне диференціювання. Вищезазначені формули отримані шляхом зберігання відповідної кількості вузлів за вибору в якості $I_n(x)$ інтерполяційного поліному Лагранжа або Ньютона. Подібним шляхом можна отримати формули чисельного диференціювання довільних порядків.

Розв'язування диференціальних рівнянь

Диференціальним рівнянням n-го порядку називають співвідношення виду

$$H\left(t, x, x', x'', ..., x^{(n)}\right) = 0 \qquad (6.8)$$

Розв'язком такого диференціального рівняння є функція $x(t)$, яка перетворює рівняння в тотожність. Під *системою диференціальних рівнянь* n-го порядку будемо розуміти систему виду:

$$\begin{cases} x_1' = f_1\left(t, x_1, x_2, ..., x_n\right), \\ x_2' = f_2\left(t, x_1, x_2, ..., x_n\right), \\ ... \\ x_n' = f_n\left(t, x_1, x_2, ..., x_n\right). \end{cases} \qquad (6.9)$$

Розв'язком цієї системи є вектор

$$x(t) = \begin{cases} x_1(t) \\ x_2(t) \\ ... \\ x_n(t) \end{cases}, \qquad (6.10)$$

який перетворює рівняння системи (6.9) у тотожності. Оскільки диференціальні рівняння і системи мають нескінчену кількість розв'язків, що відрізняються один від одного константами, то для однозначного визначення розв'язку необхідно задати додаткові початкові чи граничні умови. Кількість таких умов визначається порядком диференціального рівняння чи системи. Залежно від їх вигляду розрізняють: *задачу Коші* – всі додаткові умови задані в одній (початковій) точці інтервалу; *крайову задачу* – додаткові умови задані на границях інтервалу. Чисельні методи розв'язування дифрівнянь застосовують у двох випадках: 1) аналітичний розв'язок не знайдено; 2) аналітичний розв'язок відомий, але потрібно знайти його значення при певних заданих вхідних даних.

Практичні засади чисельного диференціювання в пакеті матричної математики Scilab

В Scilab є кілька засобів чисельного диференціювання функцій, заданих таблично. Наприклад, команда-функція **dy=diff(y[,n]),** де **y** множина значень функції $y = f(x)$ у вузлових точках як вектор дійсних чисел; **n** – порядок диференціювання. Функція **diff(y)** повертає як результат вектор розділених різниць суміжних елементів масиву **y**: [y(2)–y(1), y(3)–y(2), … y(n)–y(n–1)], у якого кількість елементів на одиницю менша, ніж у вихідного вектора **y**. Якщо **y** – матриця, то diff(**y**) повертає матрицю різниць стовпчиків: [y(2:m,:)–y(1:m–1,:)]. Другий аргумент функції **diff(y,n)** оголошує порядок диференціювання. Так **diff(y,2)** це те саме, що **diff(diff(y)).** При розрахунках використовується рекурентне співвідношення **diff(y,n)** = **diff(y,n-1).**

Результатом є **dy** – вектор дійсних чисел, що є різницевими величинами порядку **n**

Приклад: 6.1

```
-->x=[1 2 4 6 7 9 3 45 6 7]
x  =
 1   2   4   6   7   9   3   45   6   7
 -->dy=diff(x)                    //Крок = 1
 dy  =
 1   2   2   1   2   -6   42   -39   1
 -->d2y=diff(x,2)                     //Крок = 2
 d2y  =
 1   0   -1   1   -8   48   -81   40
```

Приклад 6.2: Побудувати графік похідної від функції *cos(x)*

```
-->h=0.05; x=0:h:10;
-->C=cos(x);
-->D=diff(C);    //Побудова розділених різниць
```

```
-->xx=x(1:$-1);  //Зменшення розмірності вихідного вектора
-->plot(xx, D/h);
```

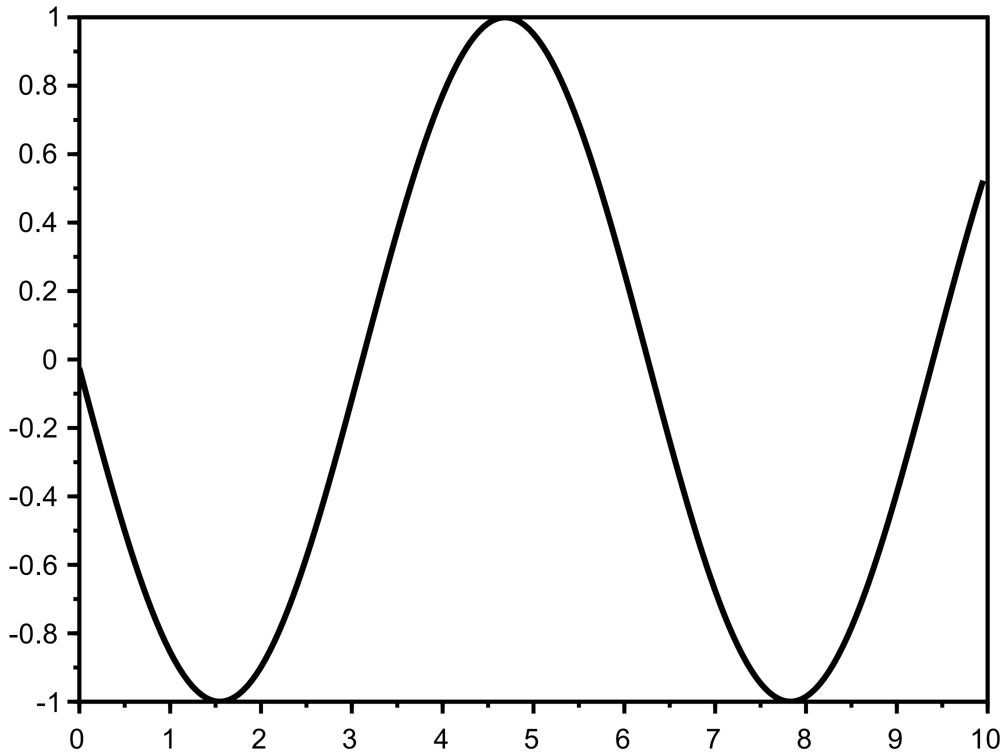

Рис. 6.1.

На рис. 6.1 добре помітна схожість графіка з графіком *-sin(x),* що власне і повинно бути. Рядок лістінгу перед командою побудови графіку зменшує розмірність вектора X на одиницю для узгодження з розмірністю вектора результатів.

Засобом чисельного диференціювання функції, аналітичний вираз якої відомий, в Scilab є команда **g=numderivative(fun, x),** де **fun** – ім'я функції, що задає вираз для диференціювання; **x** – змінна по якій відбувається диференціювання. Результатом обчислення буде матриця $g_{ij} = \dfrac{df_i}{dx_j}$.

Приклад 6.3: Обчислити $f'(x)$ в точках 0, 1, 2, 3, якщо $f(x) = (x+2)^3 + 5x$.

```
-->function F=a(x), F=(x+2)^3+5*x, endfunction;
```

```
-->xx=0:3;
-->numderivative(a, xx)
 ans  =
 17.   0.    0.    0.
  0.   32.  0.    0.
  0.    0.   53.  0.
  0.    0.    0.   80.
-->// Перевірка
-->function FF=b(x), FF=3*(x+2)^2+5, endfunction;
-->b(xx)
 ans =
 17. 32. 53. 80.
```

Цією функцією можна також скористатися при потребі порахувати частинні похідні функції кількох змінних.

Приклад 6.4: Нехай потрібно обчислити $\frac{dy}{dx_1}, \frac{dy}{dx_2}, \frac{dy}{dx_3}$ функції $y(x_1, x_2, x_3) = x_1 x_2{}^{x_3} + x_1^2 x_3$ в точці (1; 2;3)

```
-->function [Y]=f(X), Y=X(1)*X(2)^X(3)+X(1)^2*X(3),endfunction
-->X=[1 2 3];
-->numderivative(f,X)
 ans =
 14. 12. 6.5451775
-->// Перевірка
-->function [Y]=f1(X),
Y(1)=X(2)^X(3)+2*X(1)*X(3),
Y(2)=X(1)*X(3)*X(2)^(X(3)-1),
Y(3)=X(1)*X(2)^X(3)*log(X(2))+X(1)^2,
endfunction
-->f1(X)
```

```
ans =
14.
12.
6.5451774
```

Для розв’язування диференціальних рівнянь та систем у Scilab передбачена основна функція:

[y,w,iw]=ode([type],y0,t0,t [,rtol [,atol]],f [,jac] [,w,iw])

де обов'язковими вхідними параметрами є:

y0 – вектор початкових умов;

t0 – початкова точка інтервалу інтегрування;

t – координати вузлів сітки, у яких відбувається пошук розв’язку;

f – зовнішня функція, що визначає праву частину рівняння чи системи рівнянь (6.9);

y – вектор розв’язків (6.10).

Отже, щоб вирішити звичайне диференціальне рівняння виду

$$\frac{dy}{dt} = f(t, y),\ y(t_0) = y_0$$

викликаємо функцію **y=ode(y0,t0,t,f),** де

type – параметр вибору методу розв’язування або тип розв'язуваної задачі:

"adams" - застосовують якщо потрібно використати метод прогнозу-корекції Адамса;

"stiff" - вказується, якщо задача «жорстка»;

"rk" - вказується для застосування методу Рунге-Кутта четвертого порядку;

"rkf" - те саме, але за вибору п’ятиетапного методу Рунге-Кутта четвертого порядку;

"fix" - метод Рунге-Кутта, але з фиксованим кроком;

rtol,atol – відносна та абсолютна похибки обчислень, вектор, розмірність якого співпадає з розмірністю вектора **y** (за замовчуванням rtol=0.00001, atol=0.0000001, а при використанні параметрів "rkf" и "fix" – rtol=0.001,atol=0.0001) ;

jac – матрица-якобіан правої частини жорсткої системи диференціальних рівнянь, задається у вигляді зовнішньої функції **J=jak(t,y)**;

w, iw – вектори, призначені для збереження інформації про параметри інтегрування, які застосовують для того, щоб наступні обчислення виконувались з тими самими параметрами

Приклад 6.5: Розв'язати задачу Коші

$$\frac{dy}{dt} + y = \sin(yt),\ y(0) = 1.5$$

Перепишемо рівняння в такий спосіб:

$$\frac{dy}{dt} = -y + \sin(yt),\ y(0) = 1.5$$

Далі представимо його у вигляді зовнішньої функції, так як показано в лістингу прикладу 6.5 і застосуємо функцію **Y=ode(y0,t0,t,f),** де параметрами будуть

- f – посилання на попередньо створену функцію $f(t, y)$;
- t – координати сітки;
- y0, t0 – початкова умова $y(0) = 1.5$;
- YY – результат роботи функції.

```
-->function yd=f(t,y),yd=-y+sin(t*y),endfunction;
-->y0=1.5;t0=0;t=0:1:35;
-->YY=ode(y0,t0,t,f);
-->plot(t,YY)
```

Графік, що моделює процес, описаний заданим рівнянням, представлений на рис.6.2.

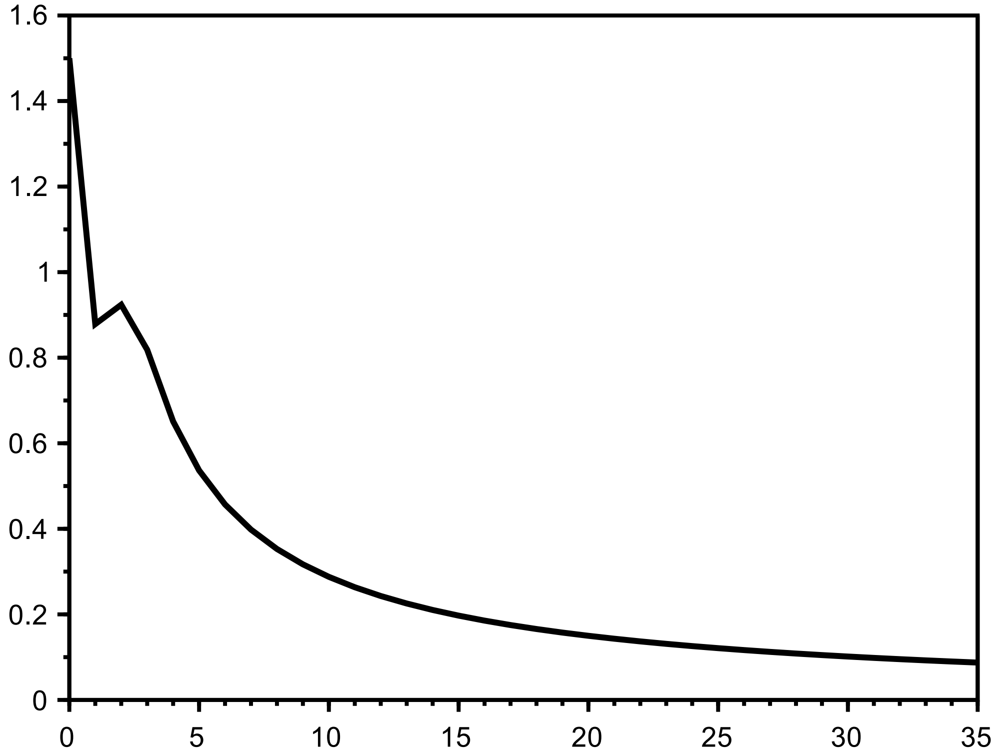

Рис. 6.2.

Приклад 6.6: Розв'язати задачу Коші

$$\begin{cases} x' = \cos(xy), \\ y' = \sin(x + ty), \\ x(0) = 0, \quad y(0) = 0 \end{cases}$$

на інтервалі [0; 10].

```
-->function dy=syst(t,y), dy=zeros(2,1), dy(1)=cos(y(1)*y(2)), dy(2)=sin(y(1)+y(2)*t), endfunction
-->x0=[0;0]; t0=0; t=0:1:10;
-->YY=ode(x0, t0, t, syst);
-->plot(t,YY’)
```

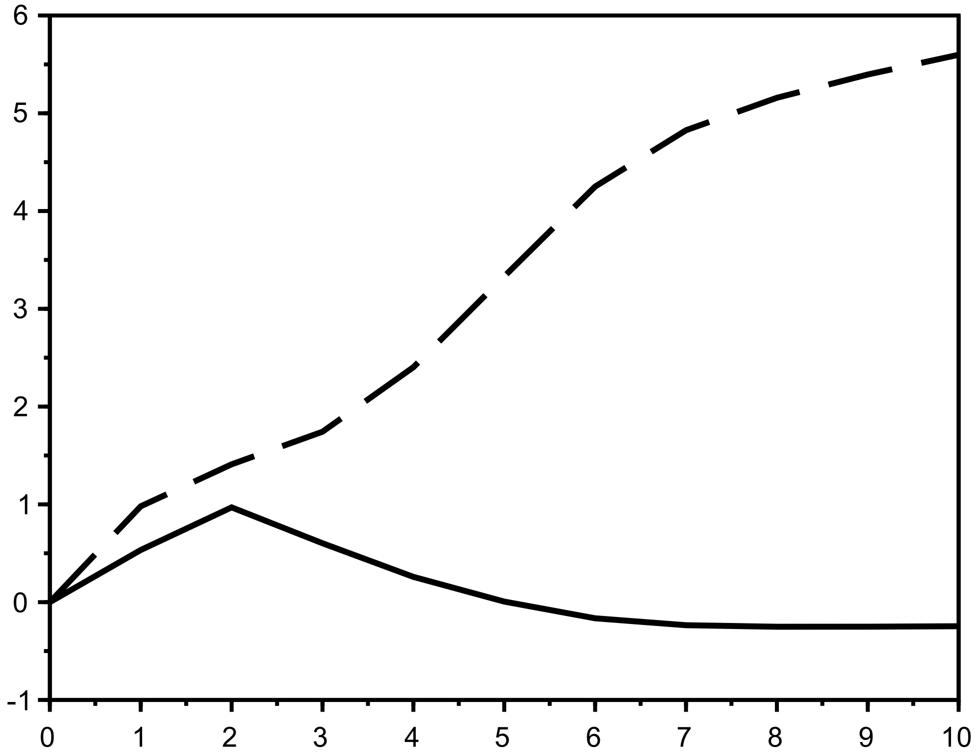

Рис. 6.3.

Приклад 6.7. Знайти розв'язок задачі Коші для жорсткої системи:

$$\frac{dX}{dt}=\begin{pmatrix}119.46 & 185.38 & 126.88 & 121.03\\ -10.395 & -10.136 & -3.636 & 8.577\\ -53.302 & -85.932 & -63.182 & 54.211\\ -115.58 & -181.75 & -112.8 & -199\end{pmatrix}X, \quad X(0)=\begin{pmatrix}1\\1\\1\\1\end{pmatrix}$$

```
-->B=[119.46 185.38 126.88 121.03;-10.395 -10.136 -3.636 8.577;
-->-53.302 -85.932 -63.182 -54.211;-115.58 -181.75 -112.8 -199];
-->function dx=syst1(t,x), dx=B*x,endfunction
-->function J=Jac(t,y),J=B,endfunction
-->x0=[1;1;1;1]; t0=0; t=0:0.01:5;
-->y=ode("stiff",x0,t0,t,syst1,Jac);
-->plot(t,y)
```

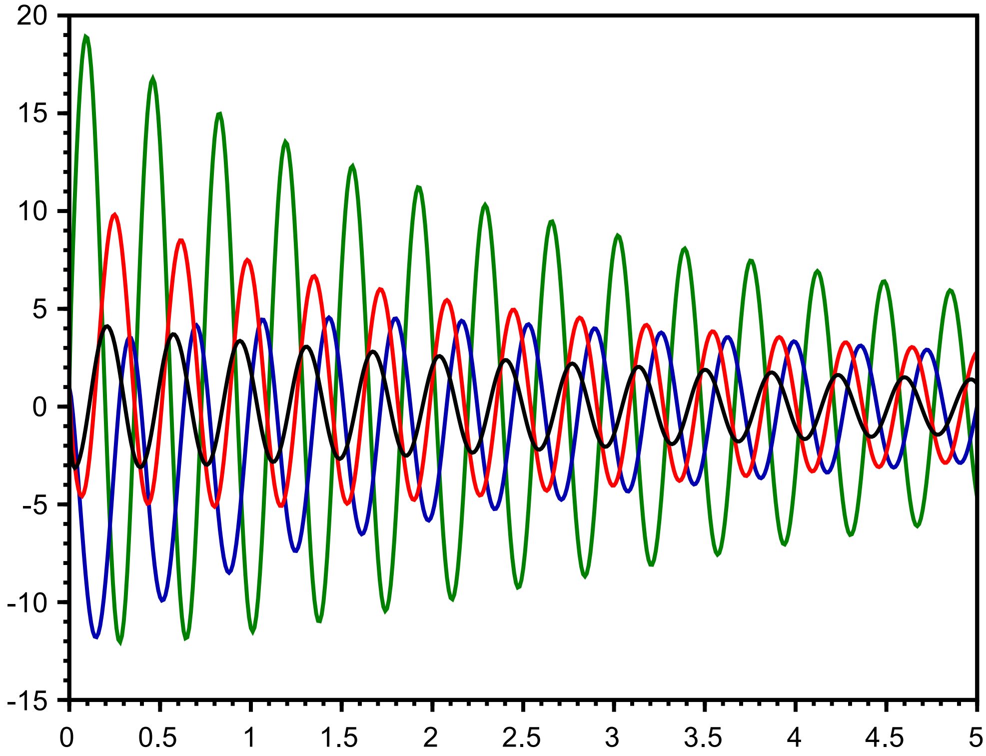

Рис. 6.4.

Приклад 6.8. Розв’язати нелінійну жорстку систему диференціальних рівнянь:

$$\begin{cases} \dfrac{dx_1}{dt} = -7x_1 + 7x_2 \\ \dfrac{dx_2}{dt} = 157x_1 - 1.15x_2x_3 \\ \dfrac{dx_3}{dt} = 0_3.96x_1x_2 - 8.36x \end{cases} \quad \text{при} \quad X(0) = \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} = \begin{Bmatrix} -1 \\ 0 \\ 1 \end{Bmatrix}.$$

На рис. 6.5 демонструється результат розв’язування системи на інтервалі [0; 2].

```
-->>function dx=syst2(t,x), dx=zeros(3,1),
dx(1)=-7*x(1)+7*x(2), dx(2)=157*x(1)+x(2)-1.15*x(1)*x(3),
dx(3)=0.96*x(1)*x(2)-8.36*x(3), endfunction
-->>//Розв’язок
-->x0=[-1;0;1]; t0=0; t=0:0.01:2;y=ode("stiff",x0,t0,t,syst2);
-->plot(t,y)
```

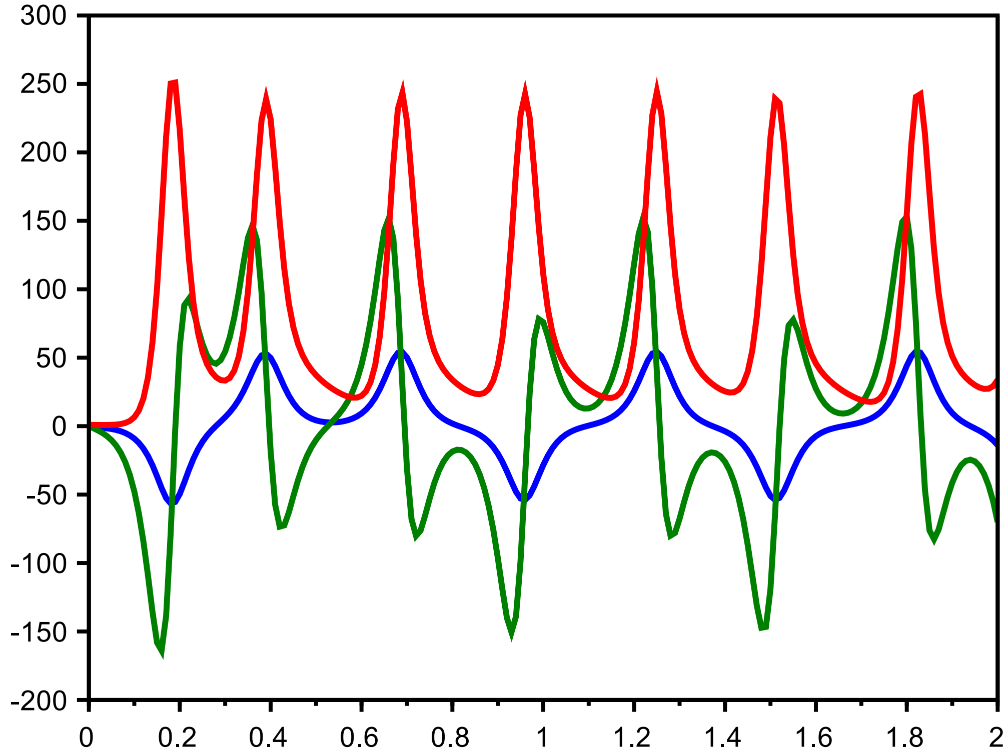

Рис. 6.5.

Приклад 6.9. Розв'язати наступну крайову задачу на інтервалі [0.25; 2].

$$\frac{d^2x}{dt^2}+4\frac{dx}{dt}+13=e^{\sin(t)},\ \ x(0.25)=-1,\ \ x'(0.25)=1$$

Перетворимо рівняння на систему, зробивши заміну $y=\dfrac{dx}{dt}$:

$$\begin{cases}\dfrac{dy}{dt}=-4y-13x+e^{\sin(t)},\\ \dfrac{dx}{dt}=y,\end{cases}\qquad \text{при}\ \ y(0.25)=1,\ \ x(0.25)=-1$$

Складемо функцію обчислення системи та розв'яжемо її у Scilab:

```
-->function F=FF(t,x), F=[-4*x(1)-13*x(2)+exp(t); x(1)],  endfunction
-->//Решение системы дифференциальных уравнений
-->X0=[1;-1]; t0=0.25; t=0.25:0.05:2;
```

```
-->y=ode("stiff", X0, t0, t, FF);
-->plot(t,y)
```

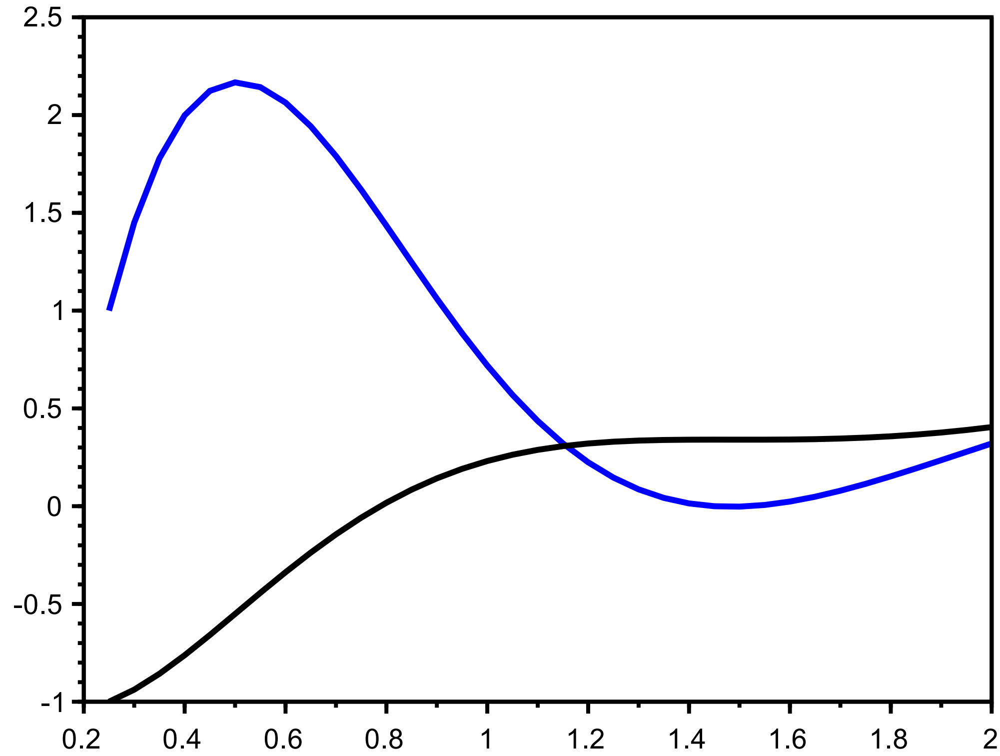

Рис. 6.6.

Крім згаданої функції **ode** у Scilab є низка функцій для розв’язування різних часткових та особливих випадків диференціальних рівнянь та їх систем – **ode_discrete, ode_optional_output, ode_roots, odedc**. Спосіб застосування цих функцій практично не відрізняється від розглянутого для функції **ode**.

Практичні завдання

Знайдіть чисельно першу та другу похідну в точці x = 3 функції, заданої таблицею з використанням різних засобів::

x	2	2.5	3	3.5	4	4.5
y	5.5	4.5	3	2	4.5	7