Рішення диференціальних рівнянь.
Методи і комп'ютерні технології інтерполяції.

Мета роботи: навчитися вирішувати диференційні рівняння з допомогою засобів MATLAB

Короткі теоретичні відомості:

Диференціальні рівняння.

Диференціальним рівнянням називається рівняння, що зв'язує шукану функцію однієї або декількох змінних, ці змінні та похідні різних порядків даної функції.

Якщо шукана функція залежить від однієї змінної, то диференціальне рівняння називається **звичайним**, якщо від декількох, то **рівнянням в приватних похідних**.

Приклади диференціальних рівнянь:

$\frac{dx}{dt} + t * x^2 + \sin x = 5$ $\qquad$ $y''' + 2y' - 8y * t = 0$

В загальному випадку рішення диференційного рівняння n-го порядку

$$y^n = f(x, y, y', ..., y^{n-1}) \qquad (1)$$

полягає у знаходженні функції y = f (x), при підстановці якого в рівнянні (1) остання звертається в тотожність. Порядок старшої похідної, що входить в диференціальне рівняння, називається порядком диференціального рівняння.

Кожне диференціальне рівняння має незлічену безліч рішень, тому для знаходження приватного рішення необхідно вказати початкові умови, а саме, задати значення $y, y', \ldots, y^{n-1}$ при $x=x_0$:

$$y_0 = y(x_0),\ y_0' = y'(x_0),\ \ldots,\ y_0^{n-1} = y^{n-1}(x_0). \qquad (2)$$

Завдання відшукання рішення рівняння (1) при заданих значеннях початкових умов (2) називається задачею Коші для звичайного диференційного рівняння.

Рівняння (1) зводиться до системи n звичайних диференціальних рівнянь першого порядку заміною на невідому функцію p. Наприклад, рівняння другого порядку

$y'' - 4y' + 5y = 3x$ можна записати у вигляді системи з двох рівнянь:

$$\begin{cases} p = y', \\ p' = 4p - 5y + 3x. \end{cases}$$

Диференціальні рівняння вирішують **наближеними методами**. Існують дві групи наближених методів: **аналітичні та чисельні**.

Аналітичні методи:
1. Метод послідовного диференціювання (розкладання в ряд Тейлора).
2. Метод послідовних наближень.
(Для отримання більш докладних відомостей дивіться лекції)

Чисельні методи:
1. Метод Ейлера
2. Удосконалений метод Ейлера (Метод Ейлера - Коші)
3. Метод Рунге - Кутта.

Метод Рунге - Кутта:

Даний метод має більш високою точністю, ніж методи Ейлера за рахунок зменшення методичних помилок. Ідея методу полягає в наступному.

За методом Ейлера рішення диференціального рівняння першого по-рядка визначається зі співвідношення:

$$y_{i+1} = y_i + \Delta y_i$$

Тоді приріст Δy може бути знайдений шляхом інтегрування :

$$\Delta y_i = \int_{x_i}^{x_{i+1}} y'(x, y)dx = \int_{x_i}^{x_{i+1}} f(x, y)dx .$$

Або у завершеному вигляді $y_{1+i} = y_i + \int_{x_i}^{x_{i+1}} f(x, y)dx$

Потім обчислюється інтеграл за методом прямокутників.

У методі Рунге - Кутта шуканий інтеграл представляється у вигляді такої кінцевої суми:

$$y_{i+1} = y_i + \int_{x_i}^{x_{i+1}} f(x, y)dx = y_i + \sum_{i=1}^{q} p_i K_i(h)$$

де p_i — деякі числа, що залежать від q;

K (h) - функції, що залежать від виду підінтегральної функції f (x, y) і кроку інтегрування h.

Система MATLAB дозволяє вирішувати диференціальні рівняння і системи високого порядку з табличним і графічним представленням результатів.

Найбільш часто диференціальними рівняннями описуються процеси, що протікають у часі. Тоді змінної інтегрування є час. **Функції ode() реалізують чисельні методи Рунге - Кутта** третього, четвертого, п'ятого і шостого порядків з автоматичним вибором шагу.

Функціями рішення диференціальних рівнянь є **ode23()** і **ode45()**, що мають вигляд:

[t, y]=ode23('fun' t_0, t_f, y_0)
[t, y]=ode45('fun' t_0, t_f, y_0)
[t, y]=ode23('fun' t_0, t_f, y_0, tol, trace)
[t, y]=ode45('fun', t_0, t_f, y_0, tol, trace)

де:

'fun' - ім'я m-файлу, в якому містяться праві частини системи диференціальних рівнянь;

t0 - початкове значення аргументу;

tf - кінцеве значення аргументу;

y0 - вектор початкових умов;

tol- задається точність, за замовчуванням, для ode23() - 10^{-3}, для ode45() – 10^{-6};

trace — видача проміжних результатів.

Технологія рішення диференціальних рівнянь в системі MATLAB наступна:

1. Створення нової функції, що представляє собою m-файл обчислення правих частин системи диференціальних рівнянь.
2. Введення функції ode().
3. Отримання рішення натисканням клавіші <Enter>.

Приклад1: вирішити диференційне рівняння $\frac{dx}{dt} + x = \sin x * t$

x(0) = 1,5-початкове значення; інтервал інтегрування [0, 35]

1. Створюєм М-файл

```
function y=D(t,x)
y=-x+sin(x*t)
end
```

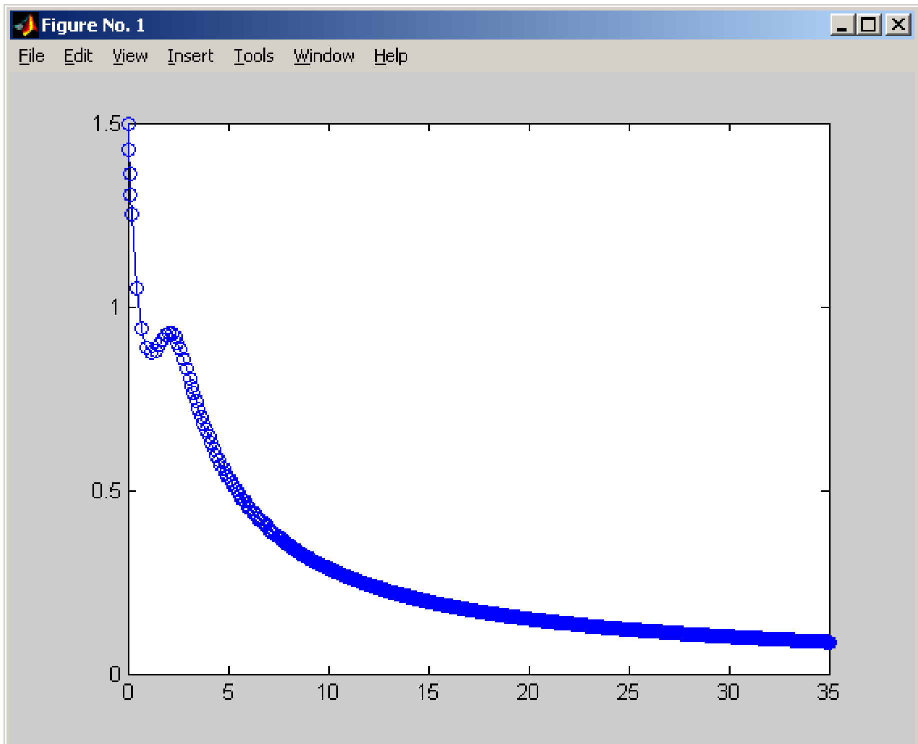
2. Викликаємо функцію а) ode23('D', [0, 35], 1.5)
   б) ode45('D', [0, 35], 1.5)

в) ode45(@D, [0, 35], 1.5)

в) [T,Y] = ode45(@D, [0, 35], 1.5)
plot (T,Y)

@D-це посилання на створений нами M-файл який містить функцію f(x,t), тобто пру частину нашого диференціального рівняння.

Видається множина рішень і будується наступний графік.



Інтерполяція функцій.

Одним з найважливіших завдань у процесі математичного моделювання є обчислення значень функцій, що входять в опис моделі. Для складних моделей подібні обчислення можуть бути вельми трудомісткими навіть при використанні ЕОМ.

Використовувані в математичних моделях функції задаються як аналітичним способом, так і табличним, при якому значення функції відомі тільки при дискретних значеннях аргументів.

Зазначені проблеми вирішуються шляхом наближеною **заміни функції f (x) більш простою функцією φ(x)**, яку неважко обчислювати при будь-якому значенні аргументу x в заданому інтервалі його зміни.

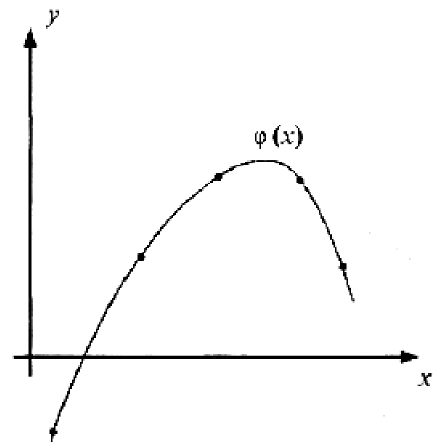
Функцію φ(x), **називають апроксимуючою функцією**, можна використовувати не тільки для наближеного визначення чисельних значень f (x) але й для проведення аналітичних викладок при теоретичному дослідженні моделі.

Припустимо у нас є таблиця значень функції при конкретних значеннях аргументу. У найпростішому випадку під **інтерполяцією** розуміється відшукання значень функції f (x), відповідних проміжним значенням аргументу, відсутнім у таблиці, тобто «Читання між рядками».

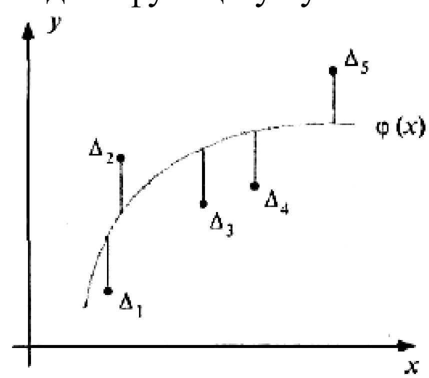
Основними видами інтерполяції є: точна у вузлах і наближена у вузлах:

При інтерполяції, **точної у вузлах**, значення апроксимуючої функції збігаються зі значеннями вихідної функції у вузлах інтерполяції.

.



При інтерполяції, наближеної у вузлах, значення апроксимуючої функції не збігаються зі значеннями вихідної функції у вузлах інтерполяції.



1. Інтерполяція точна у вузлах.

Сплайн-інтерполяція.

Інтерполяція кубічними сплайнами в середовищі MATLAB здійснюється за допомогою функції **spline()**. Функція має вигляд:

Yi = spline (x, y, xi)

де:

- x - вектор вузлів інтерполяції;
- y - вектор значень функції у вузлах інтерполяції;
- xi- вектор аргументів функції y = f (x) з області її визначення, яка задана користувачем**.**

Замість вектора функція y = f (x) може бути задана у вигляді формули.

Функція spline () не дозволяє отримати функцію інтерполяції у вигляді формули. У цьому її істотний недолік

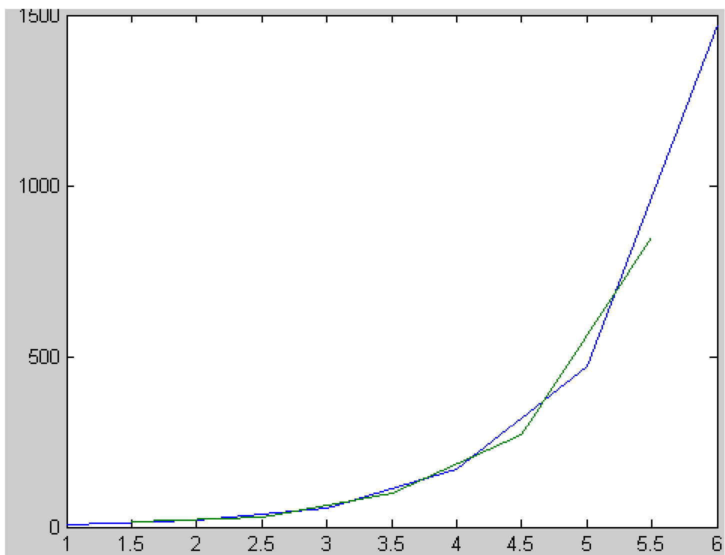
Приклад 1:

Нехай функція задана із допомогою таблиці:

X	1	2	3	4	5	6
y	6.5	20	53.5	167	473	1470

Знайти значення функції при x = 1.5, 2.5, 3.5, 4.5, 5.5.

```
x=[1, 2, 3, 4, 5, 6];
y = [6.5, 20, 53.5, 167, 473, 1470];
xi = [1.5, 2.5, 3.5, 4.5, 5.5];
yi = spline ( x, y, xi)
plot(x,y,xi,yi)
        yi   =   15.2333   29.7667   98.7000  270.9958  847.7542
```



Приклад 2.

Нехай функцією є **y=sin x,** задана при x = 1, 2, 3, 4, 5. Необхідно знайти її значення при x=1.5, 1.8,2.3,3,3.5.

```
x = [1, 2, 3, 4, 5];
xi = [1.5, 1.8, 2.3, 3, 3.5];
yi = spline (x, sin(x), xi)

        yi =   1.0222    0.9843    0.7358    0.1411   -0.3414.
```

З відповіді видно похибки інтерполяції: значення sin x не може бути більш за одиницю.

Інтерполяція точна у вузлах.

Функція interp() дозволяє вирішувати завдання інтерполяції декількома методами. Вона має вигляд:
yi = interp1 (x, y, xi, метод)

де:
- x, y - вектори значень вузлів і функції;
- xi - вектор значень аргументів, що задається користувачем;
- метод - аргумент, що дозволяє користувачеві вибрати метод інтерполяції.

Методами інтерполяції є:
- 'nearest'-ступінчаста;
- 'linear' - лінійна;
- 'cubic' - кубічна;
- 'spline' - кубічними сплайнами.

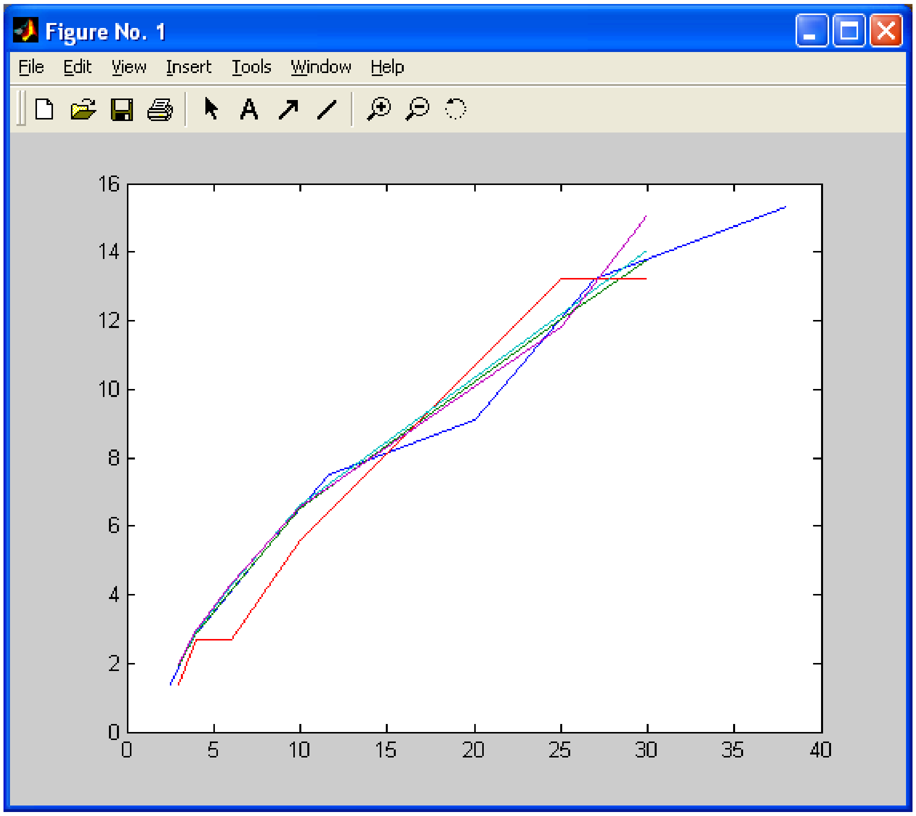
Якщо метод не зазначений, то реалізується лінійна інтерполяція. Наведемо приклад вирішення задачі цими методами.

Приклад1.
Функція y = f (x) задана у вигляді таблиці:

x	2.5	3.7	8.4	11.7	20	27	38
y	1.4	2.7	5.6	7.5	9.1	13.2	15.3

Необхідно визначити значення функції при значеннях аргументів
x = 3, 4, 6, 10, 25, 30, використовуючи різні методи.

```
x=[2.5, 3.7, 8.4, 11.7, 20, 27, 38];
y=[1.4, 2.7, 5.6, 7.5, 9.1, 13.2, 15.3];
xi=[3,4, 6, 10, 25, 30];
y1=interp1(x, y, xi, 'linear');
y2=interp1(x, y, xi, 'nearest');
y3=interp1(x, y, xi, 'cubic');
y4=interp1(x, y, xi, 'spline1');
plot (x,y,xi,y1,xi,y2, xi,y3,xi,y4)
```



2. Інтерполяція, наближена у вузлах

Поліноміальна апроксимація

Апроксимація поліномами в середовищі MATLAB здійснюється за допомогою функції polyfit (), яка має вигляд:

polyfit (x, y, n)

де:

- x - вектор вузлів інтерполяції;
- y - вектор значень функції у вузлах інтерполяції;
- n - ступінь полінома.

Відгуком при реалізації функції polyfit () є вектор коефіцієнтів полінома. Функція y = f (x) може бути також представлена в аналітичному вигляді. Наведемо приклад інтерполяції за допомогою функції polyfit ().

Приклад1.

Функція задана з допомогою таблиці

S	30	40	50	60	70	80	90	100	110	120
G	6.36	6.85	7.34	7.84	8.08	8.32	8.57	8.70	8.82	8.94

Тоді процедура інтерполяції буде мати вигляд:
x=[30,40,50,60,70,80,90,100,110,120];
y =[6.36, 6.85, 7.34, 7.84, 8.08, 8.32, 8.57, 8.7, 8.82, 8.94] ;
P=polyfit(x, y, 2)

Після натиснення клавіші <Enter> отримуємо відповідь у наступному вигляді:
P = -0.003 0.0719 4.4747

Туді функцією інтерполяції буде наступний поліном другого степеню: $\varphi(G) = 4.4747 + 0.0719S - 0.003S^2$.

Перевіримо правильність отриманих результатів із допомогою функції polyval (P,x).

F= polyval (P,x)

F = 6.3695 6.884 7.34 7.7373 8.0761 8.3564 8.5780 8.7412 8.8457 8.8917.

Інтерполяція кубічними поліномами.

Інтерполяція кубічними поліномами реалізується за допомогою функції icubic(), що має вигляд:
yi = icubic(x, y, xi)
де:

- x, y - аргумент і функція, задані у вигляді векторів, або x у вигляді вектора, а y- у вигляді формули;
- xi - вектор значень аргументу, для яких обчислюється значення функції; xi має бути в діапазоні значень x.

Приклад1.
Функція задана у вигляді таблиці

X	1	3	6	9	12	15	18
y	1	2	5	8	10	14	19

Необхідно знайти значення функції при значеннях аргументу x = 2, 5, 7, 13, 17.
x=[1 3 6 9 12 15 18];
y=[1 2 5 8 10 14 19];
xi=[2 5 7 13 17];
yi=icubic(x, y, xi)

Після натиснення клавіші <Enter> отримуємо рішення в наступного вигляду:
yi = 1.1246 3.0928 5.3590 10.7824 17.1211