9. Лабораторна робота. Чисельні обчислення у системі комп'ютерної алгебри Scilab

Мета: ознайомлення з використанням можливостей роботи СКА Scilab для чисельних обчислень.

Теоретичні відомості та методичні рекомендації

Scilab – це вільно поширювана система комп'ютерної математики (computer algebra system, CAS) та технічних обчислень, яку можна розглядати як аналог MATLAB із відкритим кодом. Інтерфейс Scilab має доволі простий вигляд. Вікно містить меню, панель інструментів і консоль з знаком запрошення --> (див. рис. 9.1). Ввід команд у Scilab виконують з клавіатури. Натиснення клавіші Enter заставляє систему виконати команду і вивести результат. Вона орієнтована насамперед на чисельні обчислення, але має й певні можливості для символьної математики через підключення зовнішніх пакетів (наприклад, SciMax чи інтерфейс до Maxima). Файл зберігається з розширенням «sod». Слід зазначити, що можливості збереження всього тексту сесії команда «Зберегти середовище» не дає.

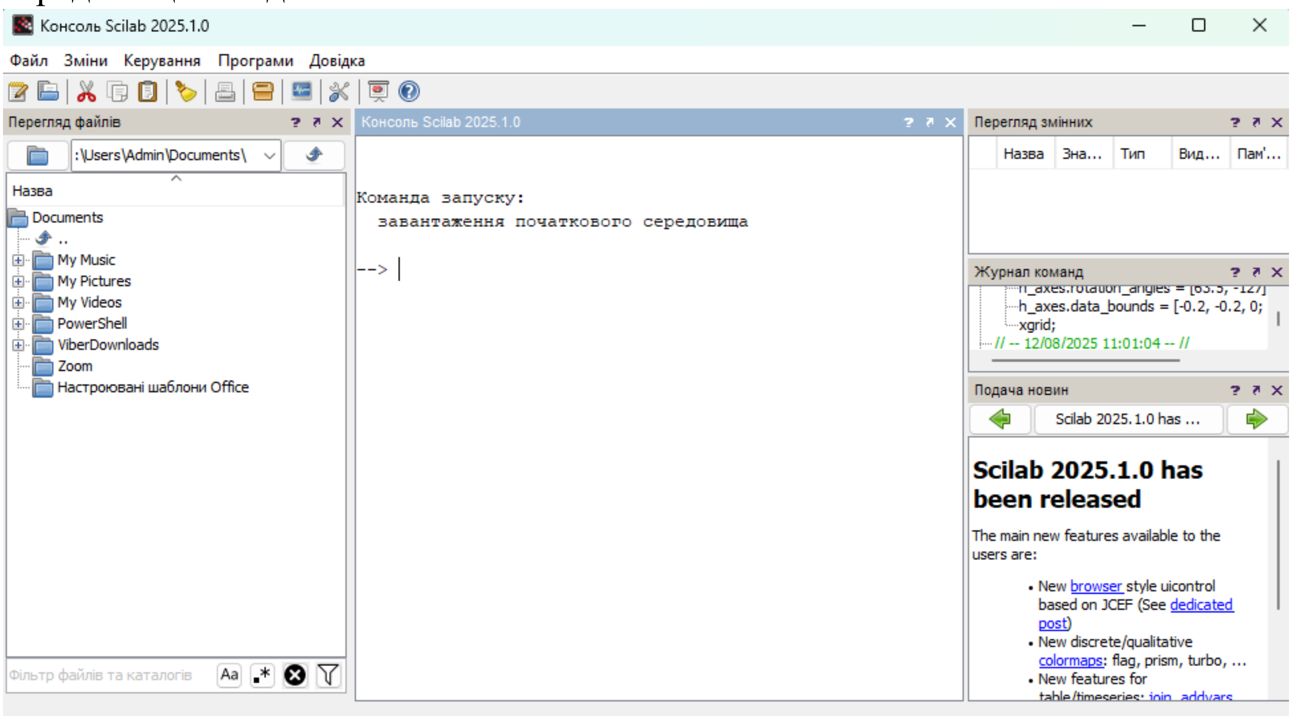

Рисунок 9.1 – Інтерфейс Scilab

Основні можливості СКА Scilab:

1. чисельні обчислення:

– лінійна алгебра (матриці, вектори, розв'язання систем рівнянь, власні значення/вектори);
– нелінійна оптимізація та рівняння;
– диференціальні рівняння;

– статистичні обчислення;
– обробка сигналів та зображень;
2. графіка та візуалізація:
– 2D та 3D графіки;
– анімації та інтерактивні побудови;
– візуалізація даних із файлів і сенсорів;
3. моделювання та симуляція
– Xcos (див. рис. 9.2) – аналог Simulink, дозволяє створювати блок-схеми для моделювання динамічних систем;
– підтримка моделювання у реальному часі (через спеціальні модулі);
4 символьна математика:
– базово відсутня (Scilab – це більше про чисельні методи), але можна підключити: SciMax – модуль, який інтегрує систему Maxima.

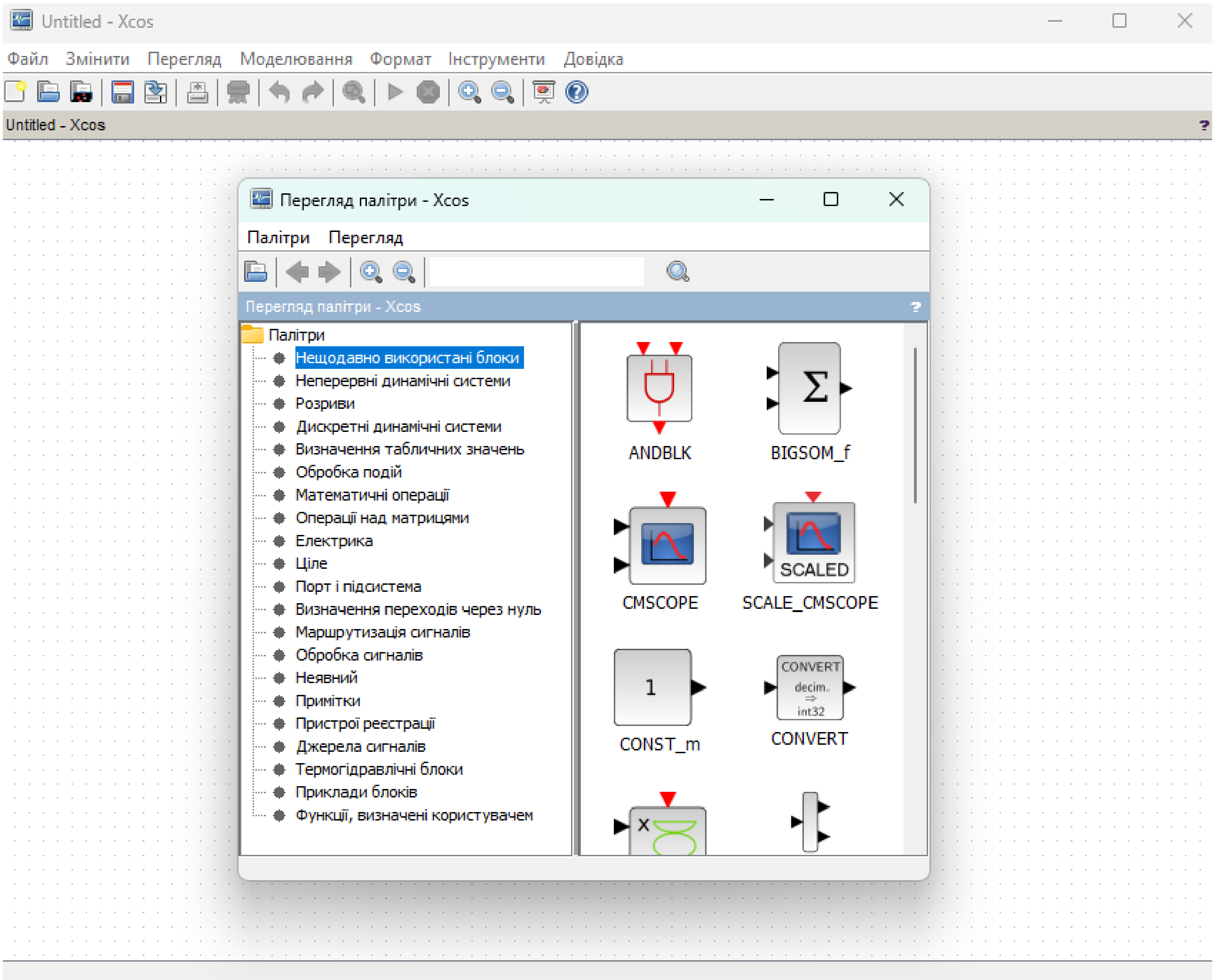

Рисунок 9.2 – Інтерфейс Xcos

У Scilab введення команд має кілька особливостей, які відрізняють його від інших систем комп’ютерної математики:
– команди можна вводити пакетом (декілька команд в одному рядку), розділяючи їх комою «,» або крапкою з комою «;»;

– якщо команда закінчується «;», то результат не виводиться;
– якщо «;» немає, результат відображається у консолі;
– однорядковий коментар починається з //;
– багаторядкові коментарі пишуться між /* та */;
– для визначення змінної треба набрати ім'я змінної, знак «=» і значення змінної. Для очистки значення змінної можна виконати команду `clear name`, де `name` – ім'я змінної.

Наприклад, див. рис. 9.3:

```
--> // Кілька команд в одному рядку

--> x = 5; y = x^2, z = y + 10

 y = 

   25.

 z = 

   35.

--> |
```

Рисунок 9.3 – Виконання команд пакетом

У Scilab є набір вбудованих системних констант (починаються з символу `%`), які можна використовувати без додаткового визначення. Вони стосуються як математичних значень, так і системних параметрів середовища і наведені у таблиці 9.1.

Таблиця 9.1 – Вбудовані системні константи

Константа	Значення	Призначення
`%pi`	π	Число пі
`%e`	e	Основа натуральних логарифмів
`%i`	$\sqrt{-1}$	Уявна одиниця
`%nan`	Not-a-Number	Результат некоректних обчислень, наприклад, $\frac{0}{0}, \frac{\infty}{\infty}$
`%inf`	∞	Нескінченність
`%eps`	$2.220D-16$	Машинний епсилон або умовний нуль (межа точності `double`)
`%t`	true	Логічне «істина»
`%f`	false	Логічне «хибний»

Наприклад, див. рис. 9.4:

```
--> %t
 ans  =
  T
--> %eps
 %eps  =
    2.220D-16
  "Environment saved."
--> %inf
 %inf  =
   Inf
```
Рисунок 9.4 – Приклади деяких вбудованих системних констант

У таблиці 9.2 наведено елементарні математичні функції, що підтримуються системою Scilab:

Таблиця 9.2 – Елементарні математичні функції у Scilab

Функція	Опис
`sin(x)`	Синус
`cos(x)`	Косинус
`tan(x)`	Тангенс
`cotg(x)`	Котангенс
`asin(x)`	Арксинус
`acos(x)`	Арккосинус
`atan(x)`	Арктангенс
`exp(x)`	Експонента e^x
`sqrt(x)`	Корінь квадратний з x
`round(x)`	Округлення
`log(x)`	Натуральний логарифм з числа x
`log10(x)`	Десятковий логарифм з числа x
`log2(x)`	Логарифм за основою 2 від числа x
`abs(x)`	Модуль числа x
`sinh(x)`	Синус гіперболічний
`cosh(x)`	Косинус гіперболічний
`tanh(x)`	Тангенс гіперболічний

Наприклад, див. рис. 9.5:

```
--> x = %pi/3;

--> y1 = sin(x) + cos(x);

--> y2 = log10(100) + sqrt(16);

--> disp("sin+cos:", y1);

  "sin+cos:"

   1.3660254

--> disp("log10+sqrt:", y2);

  "log10+sqrt:"

   6.

--> 
```

Рисунок 9.5 – Приклади з функціями

У Scilab функції, визначені користувачем (user-defined functions) – це спосіб створити власні команди, які можна багаторазово викликати з різними аргументами. Вони дуже схожі на функції у MATLAB або у мовах програмування, але мають свої особливості синтаксису.

Загальний синтаксис:

```
function [вихідні_змінні] = ім’я_функції(вхідні_змінні)
    // тіло функції
endfunction
```

- вхідні_змінні – список параметрів, які передає користувач;
- вихідні_змінні – значення, які функція повертає (може бути одне або кілька);
- імена функцій не повинні збігатися з іменами вбудованих команд;
- можна створювати анонімні функції (без збереження в окремому файлі) або зберігати їх у файлах з розширенням .sci.

Розглянемо приклади.

Функція з одним аргументом і одним результатом:

```
function y = square(x)
    y = x^2;
endfunction

// Виклик функції
res = square(5)
```

Результат див. рис. 9.6:

```
--> res = square(5)

res =                res =

 25.                  25.
```

Рисунок 9.6 – Результат роботи функції з одним аргументом

Функція з кількома аргументами і кількома результатами:

```
function [summa, diff] = sum_and_diff(a, b)
    summa = a + b;
    diff  = a - b;
endfunction

[s, d] = sum_and_diff(7, 3)
```

Результат див. рис. 9.10:

```
--> [s, d] = sum_and_diff(7, 3)

s =

  10.

d =

  4.
```

Рисунок 9.10 – Результат роботи функції з кількома аргументами

Функція без аргументів:

```
function msg = hello()
    msg = "Привіт, Scilab!";
endfunction

disp(hello());
```

Результат див. рис. 9.11:

```
--> disp(hello());

 "Привіт, Scilab!"
```

Рисунок 9.11 – Результат роботи функції без аргументів

Функція для обчислення факторіала:

```
function f = factorial_sc(n)
    f = 1;
    for k = 1:n
        f = f * k;
    end
endfunction

disp(factorial_sc(5))
```

Результат див. рис. 9.12:

```
--> disp(factorial_sc(5))

    120.
```

Рисунок 9.12 – Результат роботи функції для обчислення факторіала

У Scilab команда `deff` – це скорочений спосіб швидко створити функцію, визначену користувачем, без окремого файлу і без повного синтаксису. Загальний синтаксис:

```
deff('вихідні_змінні = ім’я_функції(вхідні_змінні)',
                          'тіло_функції')
```

- перший аргумент – рядок, який містить заголовок функції;
- другий аргумент – рядок з виразом або кількома командами тіла функції;
- якщо тіло функції складається з кількох команд, їх розділяють крапкою з комою «;».

Наприклад, написати функцію, за допомогою якої можна обчислити площу трикутника за формулою Герона:

```
deff('S=G(a,b,c)','p=(a+b+c)/2;S=sqrt(p*(p-a)*(p-b)*(p-c))');
```

Результат див. рис. 9.13:

```
--> G(3,4,5)

 ans =

    6.
```

Рисунок 9.13 – Результат роботи функції для обчислення площі трикутника за формуло. Герона

Запам’ятайте, що `deff` визначає анонімну функцію, яка зберігається в пам’яті на час роботи сесії.

У Scilab для розв’язання алгебраїчного рівняння існує команда:

```
r = roots(p)
```

де `p` – вектор коефіцієнтів многочлена, починаючи з найстаршого степеня, `r` – вектор усіх коренів (дійсних і комплексних).

Наприклад, знайти корні рівняння $x^3 - 2x - 7 = 0$. Коефіцієнти многочлена: $[1,0,-2,-7]$, тоді див. рис. 9.14:

```
--> p = [1 0 -2 -7];

--> r = roots(p)

 r = [3x1 double]

    2.2582589 + 0.i
   -1.1291294 + 1.3508515i
   -1.1291294 - 1.3508515i
```

Рисунок 9.14 – Знаходження коренів алгебраїчного рівняння

Але команда `roots(p)` не підходить для трансцендентних і нелінійних рівнянь та їх систем. У цьому випадку використовується команда:

```
x = fsolve(x0, f)
```

де `x0` – початкове наближення кореня (може бути число або вектор), `f` – функція, яка повертає значення рівняння $f(x) = 0$, `x` – знайдений корінь.

Наприклад, розв'яжемо систему:

$$\begin{cases} x^2 + y^2 - 4 = 0, \\ e^x + y - 1 = 0. \end{cases}$$

```
deff('F = sys(X)', 'F(1) = X(1)^2 + X(2)^2 - 4; F(2) =
exp(X(1)) + X(2) - 1')

// Початкове наближення
X0 = [1; 1];

// Виклик fsolve
sol = fsolve(X0, sys)
disp(sol)
```

Результат див. рис. 9.15:

```
--> disp(sol)

 -1.8162641
  0.8373678
```

Рисунок 9.15 – Знаходження розв'язку системи

Початкове наближення критично важливе – від нього залежить, чи знайде метод корінь і який саме (особливо, якщо коренів кілька). Якщо функція погано поводиться, можна:

– побудувати графік, щоб оцінити, де знаходиться корінь;
– використати кілька різних початкових наближень.

У Scilab чисельне інтегрування використовується, коли неможливо (або складно) знайти інтеграл аналітично. Система має кілька вбудованих функцій для цього, головна з яких:

```
integrate('вираз', 'змінна', нижня_межа, верхня_межа)
```

Якщо Scilab може знайти аналітичний результат – отримаєте точний інтеграл. Якщо ні – використає чисельний метод. Межі можна задавати числами або `%inf` / `-%inf`. Наприклад:

```
integrate('sin(x)', 'x', 0, %pi)
```

Результат див. рис. 9.16:

```
ans =

  2.
```

Рисунок 9.16 – Результат знаходження визначеного інтеграла

У Scilab `inttrap` – це функція чисельного інтегрування методом трапецій. Вона підходить, коли ми вже маємо табличні значення функції (тобто не формулу, а масив точок). Той же самий інтеграл $\int_0^{\pi} \sin x \, dx$ розв'яжемо чисельним методом трапецій:

```
// Масив точок
x = 0:%pi/10:%pi;
y = sin(x);

// Чисельне інтегрування методом трапецій
S = inttrap(x, y);
disp(S);
```

Результат див. рис. 9.17:

```
--> disp(S);

  1.9835235
```

Рисунок 9.17 – Результат знаходження визначеного інтеграла методом трапецій

Ще є функція `intg()`, яка виконує лише чисельний розрахунок.

Отже, СКА Scilab має такі переваги та недоліки.

Плюси:

– безкоштовність та відкритий код;
– добра підтримка матричних і чисельних методів;
– Xcos для моделювання без програмування;
– кросплатформеність.

Мінуси:

– менша популярність порівняно з MATLAB – менше прикладів і готових бібліотек;
– символьні обчислення потребують окремого встановлення;
– інтерфейс іноді менш зручний, ніж у комерційних аналогів.

Завдання до лабораторної роботи

1. За допомогою оператора `deff` створити функцію і обчислити її значення у заданій точці:

1) $f(x) = e^{\frac{x}{y}} \cdot \sqrt{\left|\frac{-x^2-7y-120}{\ln 3x}\right|}, x = 5, y = 7.$
2) $f(x) = \sin(xy) \cdot \log_2(x^2 + y^2 + 5), x = -2, y = -1.$
3) $f(x) = \mathrm{ch}(x - y) \cdot \sqrt{\left|\ln\frac{x^3-5}{y^4}\right|}, x = 1, y = 8.$
4) $f(x) = e^{x^2+y^8} \cdot \mathrm{tg}\left(\cos\left(\frac{x^3+2x-8}{141}\right)\right), x = -3, y = 6.$
5) $f(x) = \cos\left(\frac{x-y}{10}\right) \cdot \ln\left|\ln\frac{x^4-1}{y^4+1}\right|, x = -5, y = -1.$
6) $f(x) = \mathrm{ctg}\left(\left|\frac{x-y}{xy}\right|\right) \cdot e^{x^{12}+x^6+3}, x = -6, y = 6.$

7) $f(x) = \mathrm{sh}\left(\frac{x^3}{x+y}\right) \cdot \log_2(10x^2 + 10y^2), x = 2, y = 9.$

8) $f(x) = \mathrm{ch}(x^3 - y^3) \cdot \sqrt{\left|\ln\frac{x+y}{x^2+y^2}\right|}, x = -1, y = \frac{1}{2}.$

9) $f(x) = e^{-x^2-xy^3} \cdot \mathrm{tg}\left(\sin\left(\frac{x^3+3x}{\sqrt{x+y}}\right)\right), x = 3, y = 11.$

10) $f(x) = \mathrm{ctg}\left(\left|\frac{e^{xy}}{xy}\right|\right) \cdot e^{x^2-9x^6-5}, x = -5, y = 5.$

2. Обрати з геометрії або стереометрії довільну формулу і написати функцію, яка реалізує обчислення за обраною формулою.

3. Розв'язати рівняння СКА Scilab:

1) $e^x - 3x = 0.$
2) $\sin x - \frac{x^2}{4} = 0.$
3) $\ln x + x^2 - 3 = 0.$
4) $\mathrm{tg}\, x - x^3 = 0.$
5) $\cos x - |x| = 0.$
6) $e^{-x} - \cos x = 0.$
7) $\ln x - \sin x = 0, x > 0.$
8) $e^{x^2} - 5 = 0.$
9) $\sin^2 x - 0.5x = 0.$
10) $xe^x - 4 = 0.$

4. Обчислити чисельно визначений інтеграл, використовуючи функції `integrate`, `intg` і `inttrap`. Порівняйте отримані результати:

1) a) $\int_0^{\frac{\pi}{2}} x\cos x\, dx$; b) $\int_1^{+\infty} \frac{dx}{\sqrt{x}}$.

2) a) $\int_{\frac{\pi}{6}}^{\frac{\pi}{4}} \frac{\cos x}{x} dx$; b) $\int_0^{+\infty} \frac{\sin x \cos x dx}{x}$.

3) a) $\int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} \frac{dx}{1+\cos x}$; b) $\int_1^{+\infty} \frac{dx}{1+x^2}$.

4) a) $\int_0^{\frac{\pi}{2}} x^3 \cos x\, dx$; b) $\int_0^{+\infty} \frac{1-e^{-2x^2}}{xe^{x^2}} dx$.

5) a) $\int_0^{\frac{\pi}{2}} \frac{dx}{5-4\sin x+3\cos x}$; b) $\int_1^{+\infty} \frac{\mathrm{arctg}\, 4x}{x^2} dx$.

6) a) $\int_0^{\frac{\pi}{2}} \frac{\sin x \cos x}{(\cos^2 x+\sin^2 x)^2} dx$; b) $\int_1^{+\infty} \frac{\ln(x^2+1)}{x} dx$.

7) a) $\int_0^{\pi} x^3 \sin x\, dx$; b) $\int_0^{+\infty} e^{-2x} dx$.

8) a) $\int_0^{e} \ln^3 x\, dx$; b) $\int_0^{+\infty} \frac{dx}{x^2+4}$.

9) a) $\int_4^{9} \frac{x-1}{\sqrt{x}+1} dx$; b) $\int_0^{+\infty} e^{-x}\sqrt{x} dx$.

10) a) $\int_{\frac{\pi}{4}}^{\frac{\pi}{3}} \frac{x}{\sin^2 x} dx$; b) $\int_1^{+\infty} \frac{x^3+1}{x^4} dx$.

Питання для самоконтролю

1. Охарактеризуйте можливості системи комп’ютерної алгебри Scilab.
2. Чи можна виконувати у Scilab символьні обчислення?
3. Як задаються вбудовані системні константи у Scilab?
4. Наведіть приклади синтаксису елементарних математичних функцій у Scilab.
5. Для чого потрібен Xcos?
6. Розкажіть про особливості введення команд у Scilab.
7. Охарактеризуйте різні способи створення функції у Scilab.
8. Яка команда підходить для розв’язання трансцендентних і нелінійних рівнянь та їх систем?
9. Які функції Ви знаєте для чисельного інтегрування у Scilab?