

Тарас Кобильник, аспірант

Національного педагогічного університету
імені М. П. Драгоманова
м. Київ

ВИВЧЕННЯ ЕЛЕМЕНТІВ ПРОГРАМУВАННЯ В СИСТЕМІ КОМП'ЮТЕРНОЇ МАТЕМАТИКИ MAXIMA

У статті коротко охарактеризовані основні можливості вільно поширюваної системи комп'ютерної математики Maxima, основна увага звертається на вивчення елементів програмування у системі Maxima.

Постановка проблеми. В останні десятиріччя сформувався та розвивається науковий напрям на межі математики та інформатики – комп'ютерна математика [2], основними засобами якої є системи комп'ютерної математики (СКМ), що створювались для фахівців-математиків. СКМ належать до засобів прикладної, практичної інформатики – галузі діяльності людей, спрямованої на впровадження та застосування інформаційних технологій на практиці. На сьогодні СКМ використовуються не тільки для науково-дослідницької роботи, а й для інтенсифікації навчального процесу (див., наприклад у [5], [6]).

Аналіз останніх досліджень та публікацій. Різні аспекти використання СКМ як технічні, так і дидактичні розглядаються в працях таких науковців, як В. Аладьєв, М. Шишаков, В. Дьяконов, Т. Капустіна, Ю. Лазарев, Ю. Лотюк, Б. Манзон, В. Очков, Г. Прохоров, В. Потьомкін, С. Раков, Ю. Триус та інших. Вони, як правило, звертали увагу на вивчення та використання комерційних СКМ таких як Maple [4], MathCAD [5], MatLab [3], Mathematica [1] та інших. Однією з об'єктивних причин недостатнього використання в навчальному процесі педагогічного університету СКМ є недостатнє фінансування ВНЗ державою, наслідком чого є відсутність коштів на придбання ліцензійних програмних продуктів. Вихід з такої ситуації – вивчення, впровадження та використання вільно поширюваних, некомерційних СКМ таких як Maxima.

Мета статті: розглянути основні можливості та елементи програмування в системі комп'ютерної математики Maxima.

Виклад основного матеріалу. Система Maxima серед математичних пакетів володіє досить широкими можливостями при виконанні символічних обчислень. Це, по суті, єдина з вільно поширюваних відкритих систем, яка не поступається комерційним СКМ Mathematica та

Maple. Система Maxima розповсюджується під ліцензією GPL і є доступною як користувачам операційних систем Linux, так і Windows.

Для системи Maxima розроблено кілька графічних інтерфейсів: *xmaxima*, *etamaxima*, *imaxima* та інші. Робота в будь-якому з цих інтерфейсів системи Maxima відбувається в діалоговому режимі.

Розглянемо роботу з системою Maxima під управлінням операційної системи Windows з графічним інтерфейсом *wxMaxima*, який базується на *wxWidgets*.

Після запуску системи вікно програми буде мати приблизно вигляд, поданий на Рис.1, і буде очікуватися введення команди. Зазначимо, що додаткової панелі інструментів може і не бути або набір кнопок на ній може бути змінений.

Символ *(%i1)* – це запрошення до введення першої команди. Команда в системі Maxima – це будь-яка комбінація математичних виразів та вбудованих функцій, що закінчуються крапкою з комою (;) або символом \$ – команда буде виконана, проте результат обчислення не виводиться. Для арифметичних операцій використовуються традиційні позначення +, -, *, /, ** або ^.

У системі Maxima розрізняються регістри введених символів в іменах вбудованих констант та функцій. Запис *sin(x)* нееквівалентний запису *SIN(X)*. Імена вбудованих функцій задаються малими літерами. Регістр букв також важливий при використанні змінних, наприклад *X* та *x* – різні змінні.

Вказівка про надання значення деякій змінній описується за допомогою двокрапки (:), а символ рівності (=) використовується у записі рівнянь та при підстановках. Символ := використовується для задання функцій. Для стандартних математичних констант використовуються наступні позначення: *%e* (число *e*), *%pi* (число π), *%i* (уявна одиниця $\sqrt{-1}$).

ВИВЧЕННЯ ЕЛЕМЕНТІВ ПРОГРАМУВАННЯ В СИСТЕМІ КОМП'ЮТЕРНОЇ МАТЕМАТИКИ MAXIMA

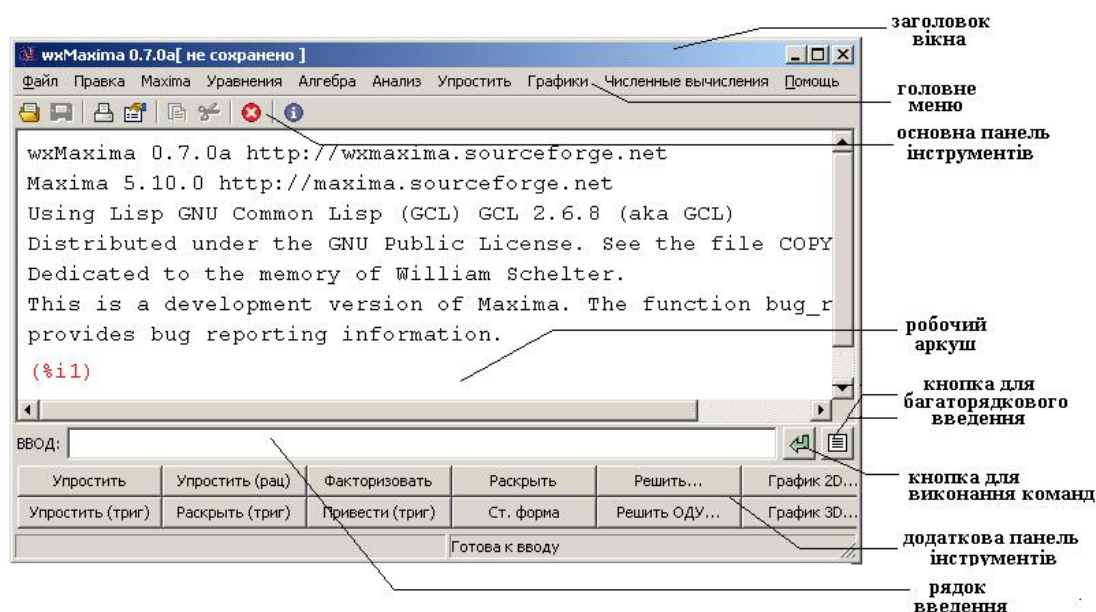


Рис. 1

Вивчення системи Maxima студентам педагогічного університету можна проводити в курсі “Системи комп’ютерної математики” після того, як студенти вже знайомі з принципами роботи хоча б однієї СКМ (наприклад, Maple чи Mathematica). При цьому основну увагу доцільно звернути на вивчення елементів програмування у системі.

Послуги пунктів головного меню – *Уравнения*, *Алгебра*, *Анализ*, *Упростить*, *Графики*, *Численные вычисления* – або відповідні їм команди використовуються для виконання певних математичних дій. Послуги пункту *Уравнения* головного меню призначені для розв’язування рівнянь, систем рівнянь, диференціальних рівнянь до другого порядку включно. Послуги пункту *Алгебра* головного меню призначені для розв’язування задач лінійної алгебри та для роботи зі списками. Послуги пункту *Анализ* головного меню призначені для розв’язування задач математичного аналізу та для деяких задач алгебри (ділення поліномів, обчислення найбільшого спільного дільника, найменшого спільного кратного). Послуги пункту *Упростить* головного меню призначені для виконання різноманітних перетворень над виразами, а саме розклад на множники, розкриття дужок, виконання підстановок, перетворення тригонометричних та комплексних виразів. Послуги пункту *Графики* головного меню призначені для побудови дво- та тривимірних графічних об’єктів. Послуги пункту *Численные вычисления* головного меню використовуються для задання точності чисельних обчислень.

Використання послуг пунктів головного меню не повинно викликати ніяких труднощів, оскільки користувачу залишається тільки правильно ввести вираз та вибрати послугу.

Детальніше розглянемо елементи програмування у системі Maxima. Розглянемо спочатку умовний оператор *if*. Умовний оператор в Maxima починається з зарезервованого слова *if*. Цей оператор має наступну форму опису:

if <умова> then <вираз_1> else <вираз_2>

За цією командою якщо значення умови – істинно, то обчислюється *вираз_1*, якщо хибно – *вираз_2*. В якості *вираз_1* та *вираз_2* може бути будь-який набір команд Maxima. Наприклад:

```
(%i13) x:13$ y:6$ if x>y then y else x;
(%o15) 6
```

Зазначимо, що блок *else* є необов’язковим (неповна форма умовного оператора *if*). Для конструювання умови використовуються операції відношення (<, <=, >, >=, = (тотожність синтаксична), *equal* (тотожність значення), *notequal*) та логічних операторів (*and*, *or*, *not* (або #)). Відмінність між операціями = та *equal* відношення проілюструємо наступним прикладом. Для цього використаємо функцію *is(expr)*, за допомогою якої визначається, чи вираз істинний (*true*), чи хибний (*false*).

```
(%i12) is((a+b)*(a-b)=a^2-b^2);
(%o12) false

(%i14) is(equal((a+b)*(a-b), a^2-b^2));
(%o14) true
```

ВИВЧЕННЯ ЕЛЕМЕНТІВ ПРОГРАМУВАННЯ В СИСТЕМІ КОМП'ЮТЕРНОЇ МАТЕМАТИКИ MAXIMA

Розглянемо створення власних функцій. Для визначення функції в Maxima використовується оператор `:=`. Наприклад, за вказівкою

```
(%i1) f(x):=x^2;
```

```
(%o1) f(x) := x^2
```

визначається функція f піднесення числа до квадрату. Слід зазначити, що в якості аргументу функції f можна задавати і список (аргумент поміщений у квадратні дужки). Наприклад:

```
(%i4) f([x]):=x^3;
```

```
(%o4) f([x]) := x^3
```

```
(%i5) f(1,2,3,4,5,8,6);
```

```
(%o5) [1, 8, 27, 64, 125, 512, 216]
```

Аналогічно можна визначати функцію кількох змінних. Наприклад:

```
(%o9) z(x,y):=x+y^2
```

```
(%i10) z(2,3);
```

```
(%o10) 11
```

Слід зазначити, що, визначаючи функцію кількох змінних, можна останній аргумент задавати як список ($[y]$), причому праву частину функції потрібно також задавати у вигляді списку. Наприклад:

```
(%o7) g(x,[y]):=[x,y^3]
```

```
(%i8) g(1,5,89,a,45);
```

```
(%o8) [1, [125, 704969, a^3, 91125]]
```

Функцію можна задавати наступним чином (використовуючи деякі оператори чи команди):

```
(%i9) z(x,y):=if x=y then x
      else (if x>y then x-y else x+y);
```

```
(%o9) z(x,y):=if x=y then x else if x>y
      then else x+y
```

```
(%i13)[z(2,2),z(9,4),z(13,5)];
```

```
(%o13)[2,5,8]
```

Для організації обчислень, що повторюються, у Maxima використовується оператор циклу `do`, який має дві форми *for-from* та *for-in*. Спочатку опишемо форму *for-from* організації циклу `do`, яка аналогічна до організації таких циклів в інших

мовах програмування (Algol, Fortran, системи Maple і т.п.).

```
for var:val1 thru val2 step val3
```

```
while unless bool
```

```
do тіло_циклу
```

Є три варіанти задання цієї форми, які відрізняються тільки умовами завершення циклу. Ними є:

1) *for параметр:початкове_значення thru кінцеве_значення_параметру do тіло_циклу*

Наприклад:

```
(%i13) s:0 $ for i:=-5 thru
```

```
10 step 3 do s:=s+i;
```

```
(%i14) done
```

```
(%i15) s;
```

```
(%o15)
```

Слово *done* у рядку виведення означає, що цикл виконано.

2) *for параметр:початкове_значення step крок_зміни_параметру while умова, яка накладається на параметр do тіло_циклу*

Наприклад:

```
(%i16) s:0 $ for i:1 step 5 while
```

```
i<25 do s:=s+i^2;
```

```
(%i17) done
```

```
(%i18) s;
```

```
(%o18) 855
```

3) *for параметр: початкове_значення step крок_зміни_параметру unless умова, яка накладається на параметр do тіло_циклу*

Наприклад, складемо процедуру, за якою отримується сім членів розвинення функції e^x в ряд Тейлора в околі точки $x=0$:

```
(%i1) series:1;
```

```
(%o1) 1
```

```
(%i2) func:exp(x);
```

```
(%o2) %e^x
```

```
(%i3) for i:=1 unless i>7 do
```

```
(func:diff(func,x)/i,
```

```
series:series+subst(x=0,func)*x^i)$
```

```
(%i4) series;
```

```
(%o4) x^7/5040 + x^6/720 + x^5/120 + x^4/24 + x^3/6 + x^2/2 + x + 1
```

Умову *unless i>7* можна замінити на *while i<=7* або *thru 7*. За допомогою операторів *while* або *unless* накладання умов можливе тільки на

ВИВЧЕННЯ ЕЛЕМЕНТІВ ПРОГРАМУВАННЯ В СИСТЕМІ КОМП'ЮТЕРНОЇ МАТЕМАТИКИ MAXIMA

параметр циклу. Початкове значення, крок зміни параметру, кінцеве значення можуть бути будь-якими виразами. Слід зазначити, що параметр циклу – локальна змінна. Оператор *step* (крок зміни параметру) може бути опущений – за замовчуванням крок зміни параметру дорівнює 1.

Пояснимо роботу циклу *for-from*. Спочатку надається початкове значення параметру циклу. Потім перевіряється одна з наступних умов:

1) якщо значення параметру циклу перевищило кінцеве значення *thru* або умова *while* набуває значення хибно, або умова *unless* набуває значення істинно, тоді робота циклу завершується. При цьому, якщо початкове значення параметру циклу перевищує кінцеве значення, то тіло циклу не виконується жодного разу (коли крок зміни параметра додатний);

2) виконується тіло циклу;

3) значення параметра циклу змінюється на значення кроку зміни.

Кроки 1)-3) виконуються доти, доки одна з умов завершення циклу не задовольняється.

При використанні форми *for-from* циклу обов'язковим є тільки ключове слово *do*. Конструкція з використанням тільки оператора *do* визначає нескінченний цикл, зупинити який можна за допомогою оператора *return()*, який використовується разом з умовним оператором *if*.

При цьому оператор *return()* повертає значення змінної, яка міститься в дужках. Наприклад:

```
(%i26) i: 45$ do if (i) > 32 then i := i - 2 else
return (i);
(%o27) 31
```

Якщо в циклі відсутні всі ключові слова, за винятком слова *while (unless)*, то отримується класичний цикл з передумовою: спочатку перевіряється істинність значення булевого виразу в умові, а потім, залежно від результатів перевірки, або виконуються оператори тіла циклу (умова істинна), або цикл завершується (умова хибна). При використанні циклу *while (unless)* треба слідкувати, щоб у тілі циклу змінювались змінні, що входять у вираз умови, оскільки у протилежному випадку цикл буде нескінченним. Наприклад:

```
(%i5) k:10$while k>5 do k:k-2;
(%o6) done
(%i7) k;
(%o7) 4
або
```

```
(%i8) k:10$unless k<=5 do k:k-2;
(%o9) done
(%i10) k;
(%o10) 4
```

Для виконання ітераційних обчислень, у яких параметр циклу не використовується, можна задати тільки умову закінчення циклу (наперед задати кількість ітерацій або накласти певну умову для завершення обчислень). Наприклад, нехай потрібно розв'язати рівняння $e^x - x - 2 = 0$, використовуючи метод простої ітерації (задамо рівняння у вигляді $x = e^x - 2$). Складемо для цього програму з

початковим наближенням $x = -1,1$ перший раз з наперед заданою кількістю ітерацій 10 (*thru 10*):

```
(%i109) x: -1.1$ thru 10 do (x: exp(x) - 2);
(%o110) done
(%i111) x;
(%o111) -1.841405648132845
```

другий варіант – з накладанням умови зупинки процесу ітерації $|x_{k+1} - x_k| < 10^{-10}$:

```
(%i114) x: -1.1$ do
(y: exp(x) - 2);
if (abs(y - x) ) < 10^(-10) then return(y),
x: y);
(%o115) -1.84140566049177
```

Для розв'язування цього ж рівняння утворимо функцію *met_dot* (метод дотичних), яка залежить від двох аргументів f – ліва частина рівняння та x – початкове наближення. Задамо функцію $f = e^x - x - 2$ та початкове наближення $x = -1,1$:

```
(%i4) met_dot(f, x) := (df: diff(f('x), 'x),
do (y: ev(df), x: x - f(x)/y,
if abs.(f(x)) < 10e-10 then return (x)))$
```

```
(%i5) riv (x) := exp(x) - x - 2.0$
(%i6) met_dot (riv, -1.1);
(%o6) -1.841405660436962
```

Розглянемо іншу форму циклу – *for-in*. За цією формою циклу *for-in* організовується цикл за елементами списку. Опис циклу має вигляд:

ВИМОГИ ДО ЗМІСТУ ТА ОФОРМЛЕННЯ СТАТЕЙ У НАУКОВО-ПЕДАГОГІЧНИЙ ЖУРНАЛ “МОЛОДЬ І РИНОК”

1.Приймаються одноосібні статті (співавтори). Текст обсягом 6 – 10 друкованих сторінок із двома-трьома ілюстраціями (рисунок, фотографія). У статтях повинно бути чітко і стисло, без зайвих математичних формул, викладено те нове та оригінальне, що досягнуто авторами в їх практичній діяльності. Потрібно уникати повторів, зайвих подробиць та загальновідомих положень, на які можна посилатися, вказуючи відповідний номер у списку літератури, що додається.

2.Рукопис статті надсилається у одному примірнику (обов'язково перший), надрукованих через півтора інтервали на одній сторінці стандартного паперу, з пронумерованими сторінками.

ДО РУКОПІСУ ДОДАЮТЬСЯ:

- анотація статті на окремій сторінці у одному примірнику українською мовою (у випадку подання статті російською, польською чи іншими мовами, додається анотація на мові оригіналу статті та українською мовою);
- рисунки, фотографії з підписаними підписами;
- список літератури, оформлений у відповідності з діючим ДСТ-ом (за абеткою; у тексті в дужках позначається позиція та сторінка [3, 47]);
- відомості про автора (авторська картка: прізвище, ім'я та по-батькові, посада, місце праці, вчений ступінь, наукове звання, адреса (службова, домашня), телефони).

3. До статей додається рекомендація кафедри (відділу) установи, де автор працює, і рецензія доктора чи кандидата наук; для статей докторів та кандидатів наук рецензій не потрібно.

4. Таблиці повинні мати назви та порядковий номер. Одночасне використання таблиць та графіків для пояснення одних і тих же положень не рекомендується.

5. Статті підписуються всіма авторами із зазначенням домашньої адреси, номерів домашнього та службового телефонів автора, який буде листуватися з редакцією з приводу цієї статті.

Редколегія відхиляє статті з порушенням цих вимог:

Відповідно до вимог ВАК України (Постанова №7-06. 1 від 15 січня 2003 р.) необхідно дотримуватися таких елементів написання статей: постановка проблеми у загальному вигляді та її зв'язок із важливими науковими чи практичними завданнями, аналіз основних досліджень і публікацій, в яких започатковано розв'язання даної проблеми і на які спирається автор, виділення невирішених раніше частин загальної проблеми, яким присвячується дана стаття; формування цілей статті (постановка завдання); виклад основного матеріалу дослідження з повним обґрунтуванням отриманих наукових результатів; висновки з даного дослідження і перспективи подальших розвідок у даному напрямку.

Сфера розповсюдження та категорія читачів:

- загальнодержавна, зарубіжна;
- викладачі університетів, ВНЗ всіх рівнів акредитації, студенти, аспіранти, науковці, спеціалісти, які підвищують свої професійні компетенції в галузі освіти та педагогічних наук.

Програмні цілі (основні принципи) або тематична спрямованість:

Ознайомлення із питаннями наукових досліджень педагогіки, ринкової економіки, інфраструктури ринку, історії, філософії, психології, формування професійних компетенцій у студентів ВНЗ, організації навчального процесу, використання інформаційних технологій у навчанні, виховання учнівської молоді та студентства у високих християнських цінностях і моральних та культурних засадах, української та романо-германської філології, тощо.

Головний редактор Мирон Вачевський

Комп'ютерний набір та верстка Іван Василюк