

Fizică experimentală

Laborator 8

Victor E. Ambruș

*Facultatea de fizică, Universitatea de Vest din Timișoara,
Bd. Vasile Pârvan nr. 4, Timișoara, RO 300223, România*

16 ianuarie 2018

Rezumat

Introducere în limbajul **Maxima**.

1 Instalarea

Maxima este un sistem de calcul algebric gratuit, compatibil cu majoritatea sistemelor de operare (Linux, Unix, BSD, OS X, Microsoft Windows, etc), pentru care documentația se găsește gratuit pe internet la adresa <http://maxima.sourceforge.net/docs/manual/maxima.html>. În mod tradițional, **Maxima** se folosește din linia de comandă, însă în momentul de față, există o mulțitudine de programe care oferă o interfață grafică peste motorul **Maxima**. Noi vom folosi **wxMaxima**, care se poate instala de la adresa <http://wxmaxima.sourceforge.net/>.

2 Operații algebrice simple

În **wxMaxima**, comenzile se execută folosind **Shift + Enter**. Astfel, adunarea, scăderea, înmulțirea, împărțirea, ridicarea la putere, aplicarea funcțiilor elementare, etc, se poate efectua natural:

```
(%i1) 9+7;  
(%o1) 16;  
(%i2) 10/2;  
(%o2) 5;  
(%i3) log(exp(3));  
(%o3) 3;
```

Maxima ține minte toate rezultatele returnate, care pot fi accesate folosind sintaxa **%oX**, unde **X** este numărul de ordine al rezultatului returnat. În particular, sintaxa **%** face referire la ultimul rezultat returnat de **Maxima**:

```
(%i4) 13^26;
(%o4) 91733330193268616658399616009;
(%i5) %^(1/26);
(%o5) 13;
```

Constantele uzuale definite în **Maxima** sunt: **%pi** (raportul dintre circumferința și raza cercului), **%e** (baza logaritmului natural), **%phi** (secțiunea sau raportul de aur), **%i** (unitatea imaginară), **inf** (plus infinit), **minf** (minus infinit), **infinity** (infinitul complex), ș.a. Uneori, **Maxima** refuză să evalueze expresiile primite, preferând să le păstreze în format simbolic [de exemplu, **sin(1.0)**]. Putem forța evaluarea unei expresii folosind comanda **float**:

```
sin(1);
=> sin(1);
float(%);
=> 0.8414709848078965;
```

3 Variabile și funcții

Maxima permite definirea de variabile folosind operatorul de atribuire **:**, astfel:

```
kb:1.38064852e-23;
qe:1.60217662e-19;
me:9.10938356e-31;
hplanck:6.62607004e-34;
eps0:8.854187817e-12;
kc:float(1/(4*%pi*eps0));
cvac:299792458;
```

Exercițiul 1. Folosind constantele de mai sus, să se calculeze lungimea de undă Compton a electronului:

$$\Lambda_c = \frac{h}{m_e c}. \quad (1)$$

[R: $\Lambda_c \simeq 2,43 \times 10^{-12}$ m]

Exercițiul 2. Folosind constantele de mai sus, să se calculeze constanta structurii fine

$$\alpha = \frac{K_C q_e^2}{\hbar c}, \quad (2)$$

unde $\hbar = h/2\pi$ este constanta redusă a lui Planck. Având valoarea lui α , să se calculeze α^{-1} .

[R: $\alpha^{-1} \simeq 137$]

4 Calcul simbolic: rezolvarea ecuațiilor

Comanda cu ajutorul căreia pot fi rezolvate ecuațiile se numește **solve**:

```
solve(x^2 - 4 = 0, x)
=> [x = -2, x = 2]
```

Sistemele de ecuații se pot rezolva înlocuind primul parametru (mai sus, e vorba de $x^2 - 4 = 0$) cu lista de ecuații de rezolvat, definită folosind [și]. Al doilea parametru (x în exemplul anterior) se înlocuiește cu lista de necunoscute care trebuie aflate din sistem. Pentru rezolvarea sistemului de mai jos:

$$\begin{cases} x + y = 3, \\ 2x - y = 4. \end{cases}$$

se poate utiliza următoarea comandă:

```
solve([x + y = 3, 2*x - y = 4], [x, y]);
=> [[x=7/3, y=2/3]];
```

În cazul în care dorim să aflăm mai multe despre o comandă, putem apăsa tasta F1, având comanda respectivă selectată.

Exercițiul 3. Presupunând că forța de atragere electrostatică $F_e = K_C q_1 q_2 / r^2$ joacă rolul de forță centripetă $F_{cp} = mv^2 / r$, să se găsească impulsul $p = mv$ al unui electron în orbită circulară în jurul unui proton (considerat fix). Se presupune că lungimea de undă de Broglie $\lambda = h/p$ a electronului în atomul de hidrogen este egală cu circumferința $2\pi r$ a orbitei sale.

- a) Folosind comanda **solve**, să se găsească raza r a orbitei care satisface aceste postulate, precum și impulsul p al electronului pe această orbită rezolvând sistemul de ecuații:

$$\begin{cases} \frac{K_C q_e^2}{r^2} = \frac{p^2}{m_e r}, \\ \frac{h}{p} = 2\pi r. \end{cases}$$

- b) Să se definească variabilele **re** și **pe** cu valorile obținute mai sus.
- c) Să se definească variabilele **ec** și **ep** cu semnificația de energie cinetică E_c , respectiv energie potențială E_p , în baza formulelor:

$$E_c = \frac{p_e^2}{2m_e}, \quad E_p = -\frac{K_C q_e^2}{r_e}. \quad (3)$$

- d) Să se exprime E_c , E_p și energia totală $E_t = E_c + E_p$ în eV și să se compare cu energia de ionizare a atomului de hidrogen (13,6 eV).
- [R: $E_t = -E_c = E_p/2 = -13,6 \text{ eV}$]

5 Reprezentarea grafică

Maxima permite reprezentarea graficelor bi- și tri-dimensionale folosind comenzile `plot2d` și `plot3d`.

```
plot2d(sin(cos(tan(x))), [x, -%pi, %pi]);
```

Pe prima poziție apare fie o funcție, fie o listă conținând mai multe funcții. Al doilea parametru se dă sub formă de listă și reprezintă variabila reprezentată pe axa absciselor, respectiv limitele inferioară și superioară ale acestei axe. Funcțiile `plot2d` și `plot3d` acceptă feluriti parametri, despre care putem afla mai multe consultând help-ul.

```
plot3d(sin(x) + cos(y), [x, -2*%pi, 2*%pi], [y, -2*%pi, 2*%pi]);
```

6 Simplificarea expresiilor

În documentația Maxima, la capitolul 9, apare următoarea listă de funcții utile la simplificarea expresiilor algebrice:

- `combine(expr)`: combină termenii cu același numitor.

```
1*f/2*b + 2*c/3*a + 3*f/4*b + c/5*b*a;  
=> (5*b*f)/4 + (a*b*c)/5 + (2*a*c)/3  
combine (%);  
=> (4*(3*a*b*c + 10*a*c) + 75*b*f)/60
```

- `demoivre(expr)` folosește relația lui de Moivre $e^{a+ib} = e^a(\cos b + i \sin b)$:

```
demoivre(exp(a + %i*b));  
=> %e^a*(%i*sin(b) + cos(b))
```

- `distrib`, `multthru` și `expand` sunt funcții utile la desfacerea parantezelor. `expand` desface toate parantezele la orice nivel (și în argumentul funcțiilor), pe când `distrib` și `multthru` acționează nerecursiv, însă comanda `expand` se execută mai lent.

```
expand((a+b)*(c+d));  
=> b*d + a*d + b*c + a*c
```

- `exponentialize` schimbă funcțiile trigonometrice și hiperbolice în exponențiale:

```
exponentialize(%i*sin(x+y));  
=> (%e^(%i*(x+y)) - %e^(-%i*(x+y)))/2
```

- `ratsimp` simplifică expresia trimisă ca argument, precum și toate subexpresiile acesteia.

7 Derivate, integrale și ecuații diferențiale

Funcția care efectuează derivarea în Maxima se numește `diff`, în timp ce integrarea se face cu comanda `integrate`:

$\frac{d}{dx} e^{-x}$	<code>diff(exp(-x),x);</code> <code>=> -%e^(-x)</code>
$\int dx e^{-x}$	<code>integrate(-%e^(-x),x);</code> <code>=> %e^(-x);</code>

Integrarea definită se face specificând după variabila de integrare limitele inferi-

$\int_0^1 \frac{du}{1+u^2}$	<code>integrate(1/(1+u^2),u,0,1);</code> <code>=> %pi/4</code>
oare și superioare: $\int dx e^{-x}$	<code>assume(x > 0);</code> <code>integrate(exp(-z^2),z,0,x);</code> <code>=> (sqrt(%pi)*erf(x))/2;</code>

Mai sus am folosit comanda `assume(x>0)` pentru a indica faptul că Maxima poate presupune că $x > 0$. Pentru anularea ulterioară a acestei presupunerii, se poate folosi comanda `forget(x>0)`.

Pentru integrarea improprie, limitele de integrare se pot înlocui cu `inf` (∞) și `minf` ($-\infty$):

$\int_0^\infty dx e^{-x^2}$	<code>integrate(exp(-x^2),x,0,inf);</code> <code>=> sqrt(%pi)/2;</code>
$\int_{-\infty}^\infty dx e^{-x^2}$	<code>integrate(exp(-x^2),x,minf,inf);</code> <code>=> sqrt(%pi);</code>

Funcția `integrate` încearcă integrarea simbolică a expresiei trimisă ca argument, chiar și pentru integralele definite. În cazul în care o valoare numerică este suficientă, se pot folosi metodele de integrare numerică disponibile în Maxima, dintre care amintim metoda `romberg`:

$\int_0^{0.5} dx e^{-x^2}$	<code>romberg(2/sqrt(%pi)*exp(-x^2),x,0,0.5);</code> <code>=> 0.5204998731606023;</code>
----------------------------	--

Integrala de mai sus poartă numele de *funcție de eroare*, care poate fi evaluată în Maxima folosind comanda `erf`:

```
erf(0.5);
=> 0.5204998778130465;
```

Ecuațiile diferențiale se pot rezolva analitic folosind comanda `ode2`.

$\frac{dy}{dx} = -2y + 1$	<code>ec: 'diff(y,x) = -2*y+1;</code> <code>ode2(ec, y, x);</code> <code>=> y=%e^(-2*x)*(%e^(2*x)/2+%c)</code>
---------------------------	---

Mai întâi s-a definit o variabilă `ec` căreia i s-a atribuit ecuația, a cărei evaluare a fost evitată folosind apostroful din fața comenzii `diff`. Soluția obținută conține o constantă `%c` care poate fi fixată folosind condițiile inițiale, care se specifică

cu ajutorul comenzii `ic1`. De exemplu, următoarea comandă impune ca $y = 1$ când $x = 0$:

```
sol:ic1(% , x=0, y=1);
=> y=(%e^(-2*x)*(1+%e^(2*x)))/2
```

Soluția astfel obținută a fost salvată în variabila `sol` și poate fi reprezentată folosind comanda:

```
plot2d(rhs(sol), [x, -1, 1]);
```

Mai sus, `rhs` selectează doar membrul drept al variabilei `sol`, care conține soluția sub forma `y=...`

Uneori, este necesar să specificăm anumite constrângeri pentru parametrii utilizați în ecuațiile diferențiale. Să considerăm ecuația oscilatorului armonic:

$$\ddot{x} + \omega^2 x = 0, \quad (4)$$

unde $\omega > 0$ reprezintă frecvență unghiulară a oscilațiilor. Pentru ω complex, soluțiile ec. (4) nu sunt neapărat periodice. Pentru evitarea includerii acestui tip de soluții, se poate impune $\omega > 0$ folosind comanda `assume`:

```
assume(omega > 0);
ode2('diff(x,t,2) + omega^2 * x = 0, x, t)
=> x=%k1*sin(omega*t)+%k2*cos(omega*t)
```

Fixăm constantele de integrare cu ajutorul comenzii `ic2`, impunând ca $x = 0$ și $\dot{x} = \omega$ când $t = 0$:

```
ic2(% ,t=0,x=0,'diff(x,t)=omega);
=> x=sin(omega*t);
```

Exercițiul 4. Să se rezolve ecuația diferențială

$$\ddot{x} + 2\lambda\dot{x} + (\omega^2 + \lambda^2)x = 0, \quad (5)$$

pentru cazul $\lambda > 0$ și $\omega > 0$, știind că $x(t = 0) = 1$ și $\dot{x}(t = 0) = 0$. Folosiți funcția `ratsimp` pentru a simplifica rezultatul. Atenție! În cazul în care ați definit deja variabila `lambda` (de exemplu, pentru exercițiul 3), aceasta poate fi reabilitată folosind comanda `kill(lambda)`.

8 Limite și dezvoltarea în serie

Maxima permite calcularea limitelor folosind comanda `limit`:

```
limit(sin(x)/x,x,0);
=> 1;
```

Uneori însă, e mai util să solicităm o dezvoltare în serie:

```
taylor(sin(x)/x,x,0,2);
=> 1-x^2/6+...
```

Exercițiul 5. Să se găsească dezvoltarea în serie în jurul lui $x = 0$ a funcției $\text{erf}(x)$, definită prin:

$$\text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x dz e^{-z^2}. \quad (6)$$

$$[\text{R: } \frac{2x}{\sqrt{\pi}} - \frac{2x^3}{3\sqrt{\pi}} + \dots]$$

9 Calcul matricial

```
m: entermatrix (3, 3);
=> Is the matrix 1. Diagonal 2. Symmetric 3. Antisymmetric 4. General
...
=> matrix([0,1,1],[1,0,1],[1,1,0])
```

Comanda **entermatrix** ne permite să specificăm dimensiunea matricei pe care urmează să o introducem. Mai departe, **Maxima** ne va întreba ce tip de matrice dorim să introducem, iar în funcție de răspuns, ne va solicita elementele necesare pentru construcția matricei. Alternativ, se poate folosi direct comanda **matrix** pentru definirea matricei:

```
m: matrix([0,1,1],[1,0,1],[1,1,0])
```

Se poate obține transpusa unei matrici **m** folosind comanda **transpose**, iar determinantul se calculează cu **determinant**. Matricea inversă se obține apelând **invert**:

```
minv: invert(m);
=> matrix([-1/2,1/2,1/2],[1/2,-1/2,1/2],[1/2,1/2,-1/2]);
```

Pentru a verifica rezultatul, putem efectua o multiplicare matricială folosind operatorul **.**:

```
m . minv;
=> matrix([1,0,0],[0,1,0],[0,0,1])
```

Vectorii și valorile proprii ale unei matrici **m** se găsesc cu ajutorul comenzii **eigenvectors**, care dă în primul șir lista de valori proprii, urmată de un șir reprezentând multiplicitățile acestora, după care vin vectorii proprii:

```
eigenvectors(m);
[[[2,-1],[1,2]],[[1,1,1]],[[1,0,-1],[0,1,-1]]]
```

Putem selecta vectorul propriu corespunzător primei valori proprii cu autorul sintaxei:

```
part(%,2,1,1)
```

Cei doi vectori proprii corespunzători celei de-a doua valoare proprie (care are multiplicitate 2) se pot obține folosind `part(%,2,2,1)` și `part(%,2,2,2)`.

Exercițiul 6. Să se calculeze comutatorul $[X, Y] = XY - YX$ corespunzător matricilor Pauli σ_1 , σ_2 și σ_3 , definite prin:

$$\sigma_1 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \sigma_2 = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad \sigma_3 = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \quad (7)$$

$$[\text{R: } [\sigma_1, \sigma_2] = 2i\sigma_3, [\sigma_2, \sigma_3] = 2i\sigma_1, [\sigma_3, \sigma_1] = 2i\sigma_2]$$