

Lucrarea de laborator nr. 3

I. Scopul lucrării

Introducere în MAPLE – mediu de programare pentru calcule simbolice și numerice.

II. Conținutul lucrării

1. Structura internă. Categoriile de comenzi MAPLE.
2. Operatori, constante și funcții predefinite în MAPLE. Expresii.
3. Comenzi de calcul în MAPLE.
4. Reprezentări grafice în MAPLE.

III. Prezentarea lucrării

III.1. Structura internă .Categoriile de comenzi MAPLE.

MAPLE este un mediu de programare pentru calcule numerice și simbolice. *Calculul simbolic* este calculul cu variabile și constante care respectă regulile algebrice, ale analizei și ale altor ramuri ale matematicii. MAPLE-ul permite manipularea formulelor care utilizează simboluri, necunoscute și operații formale, în comparație cu limbajele de programare tradiționale care utilizează doar date numerice, caractere și șiruri de caractere. Se încadrează în aceeași clasă de produse software ca și Mathematica, MathCAD, MATLAB și TKSolver.

MAPLE are trei componente de bază: *nucleul* (kernel), *biblioteca standard* (library) și *interfață cu utilizatorul* (interface). Nucleul este scris în C și realizează cea mai mare parte a calculelor făcute de sistem. Biblioteca standard este automat încărcată în memorie la deschiderea unei sesiuni MAPLE. În afara acestei biblioteci există o bibliotecă extinsă cu rutine destinate rezolvării unor probleme mai complicate, ca de exemplu, rezolvarea sistemelor de ecuații, probleme de statistică sau algebră liniară. Această

biblioteca nu este încărcată automat în memorie, ci trebuie accesată, atunci când este necesar. Interfața cu utilizatorul este scrisă în C. Interfața pentru sistemul de operare Windows este bazată pe ferestre. O **foaie (formular) de programare MAPLE** (fișier MAPLE, fișier cu extensia **.mws**) existentă poate fi încărcată selectând **Open** din meniul **File**, iar o foaie nouă de programare MAPLE poate fi creată selectând **New** din meniul **File**. Salvarea foii de programare MAPLE se realizează selectând **Save** sau **Save as** (pentru salvarea sub un alt nume) din meniul **File**. Foia de programare se poate salva sub forma unui fișier text sau latex dacă se selectează **Export as** din meniul **File**. Încheierea sesiunii MAPLE se face selectând **Exit** din meniul **File**, sau prin clic pe butonul de închidere X.

Foile de programare MAPLE constau în cinci tipuri de zone: **text**, **input** (intrare), **ouput** (ieșire), **2D graphics** (grafică 2D), **3D graphics** (grafică 3D), și **animation** (animație). În zona text se introduce textul necesar documentării. Zona input este zona în care se introduc comenzile MAPLE și este recunoscută după promptul > prezent în marginea din stânga. Întinderea zonei input sau a zonei text este arătată printr-o bară verticală în partea stângă. Comutarea între cele două zone se poate face cu ajutorul tastei funcționale F5 sau din bara de meniu. Zona output este generată automat la furnizarea răspunsului. Colecția de butoane și informația afișată în **bara de context** (sub **bara de instrumente**) depind de conținutul curent definit tipul de zonă în care se găsește cursorul. Informația despre foia de programare curentă este afișată în **bara de stare**, în partea de jos a ecranului.

MAPLE este un mediu interpretat. Explicăm în continuare ce înseamnă aceasta. Pentru ca un program (indiferent de limbajul în care este scris) să poată fi executat de calculator este necesar să fi tradus în limbaj mașină. Există trei modalități principale pentru a obține această **traducere**: **interpretarea**, **asamblarea** și **compilarea**. Programele **asamblate** și cele **compilate** sunt traduse în limbaj mașină înainte ca să fie utilizate. **Interpretarea** este un tip special de traducere, în care programul este tradus în instrucțiuni în limbaj mașină “din mers”, adică în timpul execuției sale. Mai precis, programul care trebuie interpretat (să-l numim P) este preluat de un program de interpretare (interpretorul I). Când se utilizează programul P, calculatorul rulează de fapt interpretorul I, iar interpretorul I execută pașii programului P. Interpretorul verifică textul programului P și îndeplinește instrucțiunile acestuia pas cu pas. Interpretarea este flexibilă deoarece ce un program interpretat poate fi adaptat, schimbat sau revizuit din mers. Sigur interpretarea are și dezavantaje asupra cărora nu insistăm aici (de exemplu, programele interpretate nu pot fi executate dacă nu există și un interpretor corespunzător).

Fiind un mediu interpretat MAPLE permite realizare de rutine interactive. Apariția promptului > în fereastra MAPLE semnifică faptul că se

poate introduce o comandă. Fiecare comandă (cu excepția comenzii ?) trebuie încheiată cu punct și virgulă (;) sau două puncte (:). Omiterea acestora generează o eroare de sintaxă. Rectificarea se face prin tipărire ; sau : pe o linie nouă. Fiecare comanda este executată după apăsarea tastei ENTER. Dacă s-a utilizat : pentru încheierea comenzii, comanda este executată fără a se afișa rezultatele, iar cazul utilizării ; se afișează și rezultatele.

MAPLE dispune de peste 2000 de funcții predefinite și comenzi. Fiecare **comandă** este introdusă, în zona input, în felul următor:

```
> nume_comanda(param1,param2,...);
```

Numele comenzilor a fost ales astfel încât pe de o parte să fie apropiat de funcționalitatea comenzii și pe de altă parte să fie cât mai scurt posibil. MAPLE este un mediu “case-sensitive” (se face distincție între literele mari și literele mici). Cele mai multe comenzi încep cu literă mică și au în corespondență o aceeași comandă care începe cu literă mare. Aceasta din urmă poartă denumirea **de comandă inertă** și rolul ei este doar de afișare matematică a unei expresii. Cele mai multe comenzi MAPLE necesită o listă de parametri la intrare. Această listă poate conține de exemplu, numere, expresii, mulțimi, etc., sau poate să nu conțină nici un parametru. Indiferent de numărul de parametri specificați, ei trebuie incluși între paranteze rotunde (). Toate comenzile au număr minim de parametri de tip precizat, de cele mai multe ori într-o ordine precizată. Multe comenzi pot fi utilizate cu un număr de parametri mai mare strict decât acest număr minim de parametri. Acești “extra” parametri reprezintă de obicei opțiuni de control al funcționării comenzii respective. Comenzi MAPLE pot fi folosite ca parametri. Acestea sunt evaluate și rezultatele lor sunt inserate în lista de parametri.

Comenzile MAPLE se pot clasifica în trei categorii:

1. Comenzi care se încarcă automat la deschiderea unei se încarcă automat la deschiderea unei sesiuni MAPLE. Acestea pot fi apelate direct așa cum s-a precizat mai sus.

2. Comenzi din biblioteca extinsă. Înainte de a le folosi acestea trebuie mai întâi încărcate în memorie cu ajutorul comenzii **readlib** sub forma

```
> readlib(nume_comanda);
```

3. Comenzi care aparțin unor pachete specializate. Există două modalități de utilizare a acestor comenzi:

- prin specificarea pachetului sub forma:

```
> nume_pachet[nume_comanda](param1,param2,...);
```

- cu ajutorul comenzii **with**. Un apel de forma

```
>with(nume_pachet);
```

are ca urmare încărcarea în memorie și afișarea în zona ouput a tuturor comenzilor din pachet. Până la încheierea sesiunii MAPLE acestea pot utilizate ca și cele din prima categorie.

Din cele de mai sus rezultă că nu este întotdeauna suficient să se cunoască numele unei comenzi. Uneori ea trebuie încărcată din bibliotecă sau dintr-un pachet. Dacă nu s-a făcut acest lucru și s-a introdus comanda, MAPLE nu generează un mesaj de eroare, ci afișează în zona output, comanda introdusă în zona input. În acest caz trebuie verificat dacă este scrisă corect comanda (inclusiv dacă literele mari și mici se potrivesc), sau trebuie încărcată în memorie. Informații asupra modului corect de introducere a unei comenzi se pot obține cu ajutorul comenzii **help**. Există mai multe modalități de utilizare a acestei comenzi. Este recomandabilă, următoarea formă:

```
> ? nume_comanda
```

O comandă de forma:

```
> ?
```

afișează informații generale despre structura help-ului.

Altă variantă presupune un apel de forma

```
> help('nume_comanda');
```

De remarcat faptul că numele comenzii este inclus între apostrofuri întoarse (backquotes).

III. 2. Operatori, constante și funcții predefinite în MAPLE. Expresii.

O expresie este o combinație validă de operatori și variabile, constante, și apeluri de funcții.

Operație	Operator	Exemple
Adunare	+	$x+y$
Scădere	-	$x-y$
Opus	-	$-x$
Înmulțire	*	$x*y$
Împărțire	/	x/y
Ridicare la putere (x^y)	** sau ^	$x**y$ sau x^y

Tabelul precedent conține operatorii aritmetici de bază din MAPLE. Precedența operatorilor este aceeași ca în majoritatea limbajelor de programare. Mai întâi sunt evaluate expresiile din paranteze. În lista următoare prioritatea cade de sus în jos:

1. - (operator unar)
2. **, ^
3. *, /
4. +, -(scădere)

De remarcat faptul că exponențierea succesivă nu e validă. Astfel MAPLE nu poate evalua x^y^z . O expresie de acest fel trebuie introdusă sub forma $x^(y^z)$. Ori de câte ori există ambiguități trebuie utilizate $()$.

Următorul tabel prezintă funcțiile de bază din MAPLE ce pot interveni în expresiile aritmetice.

Notăție MAPLE	Semnificație
abs(x)	x (modulul)
iquo(x,y)	partea întreagă a împărțirii x/y
irem(x,y)	restul împărțirii lui x la y
trunc(x)	cel mai mare număr întreg $\leq x$, dacă $x \geq 0$, sau cel mai mic număr întreg $\geq x$, dacă $x < 0$
frac(x)	$x - \text{trunc}(x)$
round(x)	rotunjește pe x la cel mai apropiat întreg
floor(x)	cel mai mare număr întreg $\leq x$
ceil(x)	cel mai mic număr întreg $\geq x$
sqrt(x) sau $x^{(1/2)}$	\sqrt{x}
exp(x)	e^x
ln(x) sau log(x)	lnx (logaritm natural)
sin(x)	sinx
cos(x)	cosx
tan(x)	tgx

Facem câteva remarci asupra funcțiilor irem și iquo (deoarece nu respectă întocmai teorema împărțirii cu rest). Astfel dacă m și n sunt două numere întregi, n este nenul și r este numărul întreg returnat de irem, atunci este satisfăcută relația:

$$m = n * q + r, \text{abs}(r) < \text{abs}(n) \text{ și } m * r \geq 0.$$

Dacă m și n nu sunt amândouă numere întregi, atunci irem rămâne neevaluată. Ambele funcții pot fi utilizate și cu câte trei parametri. Dacă al treilea parametru este prezent în funcția irem, atunci lui i se asignează câtul, iar în cazul funcției iquo i se asignează restul împărțirii.

Exemple:

```
> irem(29, 4, 'q');
1
> q;
7
> iquo(29, 4, 'r');
```

```

> r;
7
> irem(-29,4);
1
> irem(29,-4);
-1
> irem(-29,-4);
1
> iquo(-29,4);
-1
> iquo(29,-4);
-7
> iquo(-29,-4);
7

```

Funcțiile **rem** și **quo** se aplică polinoamelor și reprezintă analogele funcțiilor irem și iquo. Acestea cer obligatoriu al treilea parametru ce indică nedeterminata în raport cu care se consideră polinomul. Opțional admit al patrulea parametru, cu același rol cu parametru 3 din funcțiile irem și iquo. Astfel dacă a și b sunt două polinoame, b este nenul, r restul returnat de rem și q este câtul returnat de quo, atunci este satisfăcută relația:

$$a = b * q + r, \text{grad}(r) < \text{grad}(b)$$

Exemple:

```
> rem(x^5+2*x+1, x^2+x+1, x, 'q');
```

x

```
> q;
```

$$x^3 - x^2 + 1$$

```
> quo(x^5+2*x+1, x^2+x+1, x);
```

$$x^3 - x^2 + 1$$

```
> quo(x^5+2*y+z, x^2+x+1, x, 'r');
```

$$x^3 - x^2 + 1$$

```
> r;
```

$$2y + z - 1 - x$$

Funcția **factorial(k)** calculează k! (k factorial, 12...k). Același efect în are și **k!**, după cum rezultă din exemplele de mai jos:

```
> factorial(4);
24
> 4!;
24
> 6!;
720
> factorial(factorial(3))=3!!;
720 = 720
```

Tabelul de mai jos conține câteva constante MAPLE:

Constantă	Notăția matematică
Pi	π
infinity	∞
I	$i, i^2 = -1$
gamma	constanta lui Euler
true	adevăr, în cazul evaluării booleene
false	fals, în cazul evaluării booleene

De reținut că pi (scris cu literă mică) se referă la litera grecească π .

Tipul boolean în MAPLE are două valori: **true** și **false**. Expresiile booleene (logice) pot fi formate cu ajutorul operatorilor de comparație și operatorilor logici. Următoarele două tabele conțin acești operatori.

Operator	Simbol	Exemple
egal	=	$x=y$
diferit	\diamond	$x \diamond y$
mai mare	>	$x > y$
mai mare egal	>=	$x >= y$
mai mic	<	$x < y$
mai mic egal	<=	$x <= y$

Operator	Simbol	Exemple
Negație (non) – unar	not	not x
Conjunție (și)	and	x and y
disjunție (sau)	or	x or y

Ordinea de efectuare a operațiilor este: not, and, or.

În MAPLE există expresii similare cu expresiile din C formate cu operatorul virgulă. Astfel o *secvență de expresii* în MAPLE este un șir de expresii separate între ele prin virgulă. Cele mai multe funcții din MAPLE cer la intrare o secvență de expresii, și întorc un rezultat ce conține o secvență de instrucțiuni. Cel mai simplu mod de a crea o secvență de instrucțiuni este:

```
> 1, 2, 3, 4, 5;
      1, 2, 3, 4, 5
> a=1,b=a+2,c+2;
      a = 1, b = a + 2, c + 2
```

Alternativ, există alte două moduri de a crea secvențe de instrucțiuni în MAPLE: cu ajutorul operatorului \$ sau cu ajutorul comenzii seq. Următoarele exemple sunt edificatoare:

```
> a$5;
      a, a, a, a, a
> $2..7;
      2, 3, 4, 5, 6, 7
> i^2$i=1..3;
      1, 4, 9
> seq(i!,i=1..4);
      1, 2, 6, 24
> seq(i!!,i=1..4);
      1, 2, 720, 620448401733239439360000
```

Secvență vidă este desemnată prin NULL.

III.3. Comenzi de calcul în MAPLE.

Tabelul de mai jos conține comenzile din MAPLE pentru diferențiere, integrare și însumare.

Notăție MAPLE	Semnificație	Notăție matematică
diff(f(x),x)	derivată parțială	$\frac{\partial f}{\partial x}$
int(f(x),x)	integrală indefinită	$\int f(x)dx$
sum(f(n),n)	suma seriei	$\sum_{n=1}^{\infty} f(n)$

int(f(x),x=a..b)	integrală definită	$\int_a^b f(x)dx$
sum(f(k),k=a..b)	sumă de la a la b	$\sum_{k=a}^b f(k)$

Diff, Int, Sum, reprezintă comenzile inerte corespunzătoare.

Exemple:

> diff(sin(x), x);

$$\cos(x)$$

> diff(cos(x), y);

$$0$$

> diff(x*sin(cos(x)), x);

$$\sin(\cos(x)) - x \cos(\cos(x)) \sin(x)$$

> diff(ln(x), x);

$$\frac{1}{x}$$

> Diff(ln(x), x);

$$\frac{\partial}{\partial x} \ln(x)$$

> Diff(ln(x), x) = diff(ln(x), x);

$$\frac{\partial}{\partial x} \ln(x) = \frac{1}{x}$$

> Diff(sin(x)*tan(y), x, y) = diff(sin(x)*tan(y), x, y);

$$\frac{\partial^2}{\partial yx} \sin(x) \tan(y) \sin(x) \tan(y) = \cos(x) (1 + \tan(y)^2)$$

> int(sin(x), x);

$$-\cos(x)$$

> Int(sin(x), x);

$$\int \sin(x) dx$$

> int(sin(x), x=0..Pi);

$$2$$

> Int(x^2*ln(x), x=1..3) = int(x^2*ln(x), x=1..3);

$$\int_1^3 x \ln(x) dx = 9 \ln(3) - \frac{26}{9}$$

>Int(Int(exp(-x^2-y^2), x=0..infinity), y=0..infinity) =

```
int(int( exp(-x^2-y^2), x=0..infinity ),
y=0..infinity);
```

$$\int_0^{\infty} \int_0^{\infty} e^{(-x^2-y^2)} dx dy = \frac{1}{4} \pi$$

```
> sum(k^2, k=1..4);
```

30

```
> Sum(k^2, k=1..4);
```

$$\sum_{k=1}^4 k^2$$

```
> Sum(k^2, k=1..n)=sum(k^2, k=1..n);
```

$$\sum_{k=1}^n k^2 = \frac{1}{3} (n+1)^3 - \frac{1}{3} (n+1)^2 + \frac{1}{6} n + \frac{1}{6}$$

```
> sum(1/k^2, k=1..infinity);
```

$$\frac{1}{6} \pi^2$$

```
> Sum(1/k!, k=0..infinity)=sum(1/k!, k=0..infinity);
```

$$\sum_{k=0}^{\infty} \frac{1}{k!} = e$$

Prezentăm în continuare câteva exemple cu comenzile **expand**, **factor** și **simplify**. Principalul rol al comenzii **expand** este aplicarea distributivității produsului față de adunare. Comanda **factor** se aplică pentru descompunerea în factori ireductibili a polinoamelor de mai multe variabile. Iar comanda **simplify** aplică regulile de simplificare într-o expresie.

```
> expand((X^2-Y^2)^2*(X^2+Y^2)^2);
```

$$X^8 - 2 X^4 Y^4 + Y^8$$

```
> factor(X^6-Y^6);
```

$$(X - Y) (X + Y) (X^2 + X Y + Y^2) (X^2 - X Y + Y^2)$$

```
> simplify((X^6-Y^6)/(X^2+X*Y+Y^2));
```

$$X^4 - Y X^3 + Y^3 X - Y^4$$

III.4. Reprezentări grafice în MAPLE.

Comenzile destinate reprezentărilor grafice sunt incluse în pachetul **plots**. Numele pachetului trebuie să preceadă fiecare comandă. Altă variantă presupune încărcarea întregului pachet în memorie cu ajutorul comenzii `with`:

```
> with(plots);
[animate, animate3d, changecoords, complexplot, complexplot3d, conformal,
contourplot, contourplot3d, coordplot, coordplot3d, cylinderplot,
densityplot, display, display3d, fieldplot, fieldplot3d, gradplot, gradplot3d,
implicitplot, implicitplot3d, inequal, listcontplot, listcontplot3d,
listdensityplot, listplot, listplot3d, loglogplot, logplot, matrixplot, odeplot,
pareto, pointplot, pointplot3d, polarplot, polygonplot, polygonplot3d,
polyhedraplot, replot, rootlocus, semilogplot, setoptions, setoptions3d,
spacecurve, sparsematrixplot, sphereplot, surfdata, textplot, textplot3d,
tubeplot]
```

Prezentăm câteva exemple cu comenzile **plot**, **plot3d** și **animate3d**.

Plot este destinată reprezentărilor grafice în plan și poate fi folosită sub mai multe forme. Prezentăm de fiecare dată numărul minim de parametri ceruți.

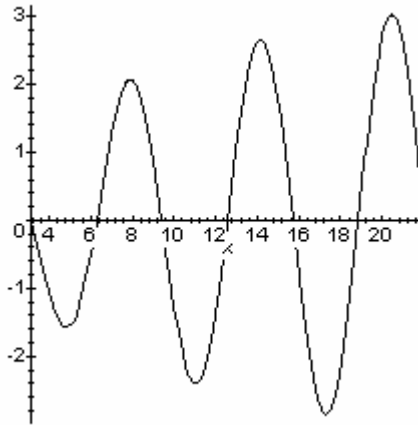
Notăție MAPLE	Curba/Curbele reprezentate
<code>plot(f(x),x = a..b)</code>	$y = f(x), x \in [a, b]$
<code>plot([f(x),g(x),...],x = a..b)</code>	$y = f(x), y = g(x), \dots, x \in [a, b]$
<code>plot([f(t),g(t),t = a..b])</code>	$\begin{cases} x = f(t) \\ y = g(t) \end{cases} t \in [a, b]$

Reprezentarea grafică se face conform cu opțiunile (de stil, culoare, axe, coordonate, rezoluție ...) indicate în comandă sau în raport cu cele implicite. Unele din aceste opțiuni se pot stabili și din meniul contextual: se introduce comanda de reprezentare grafică a curbei, iar apoi se selectează din bara de context, sau prin clic cu butonul drept al mouse-ului pe grafic, opțiunile dorite.

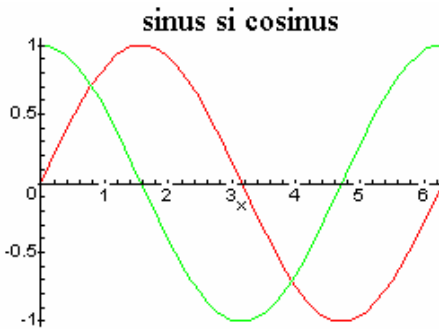
Implicit se folosesc coordonatele carteziene. Dacă se dorește utilizarea altor coordonate, acestea trebuie specificate, prin introducerea în lista de opțiuni sub forma `coords = nume_coordonate`. O opțiune de forma `discont=true`, determină apelul comenzii `Discont` pentru determinarea punctelor de discontinuitate a funcției ce se reprezintă grafic.

Exemple:

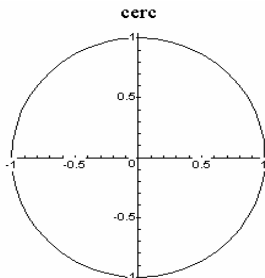
```
> plot(sin(x)*ln(x), x=Pi..8*Pi);
```



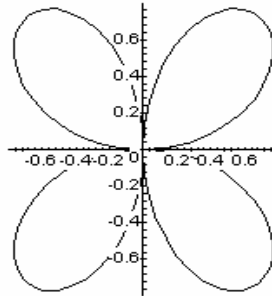
```
> plot([sin(x), cos(x)], x=0..2*Pi, title=`sinus si
cosinus`);
```



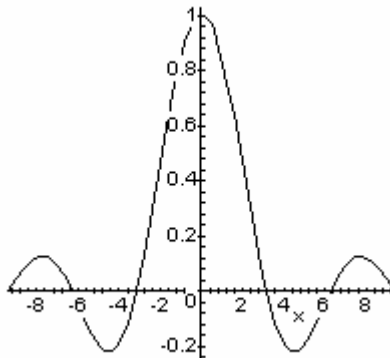
```
>plot([sin(t), cos(t), t=0..2*Pi], title=`cerc`);
```



```
> plot(sin(2*t), t=0..2*Pi, coords=polar,  
color=black);
```



```
>plot(sin(x)/x, x=-3*Pi..3*Pi,discont=true);
```



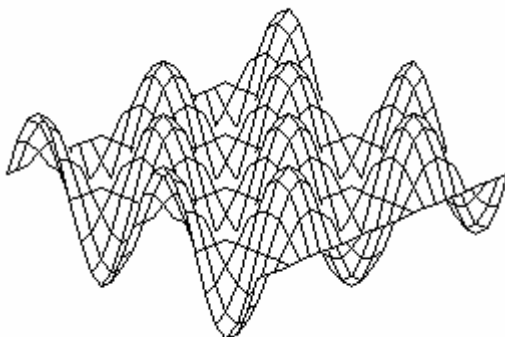
Comanda `plot3d` este destinată reprezentării grafice a suprafețelor în spațiu tridimensional. Ca și în cazul comenzii `plot` reprezentarea grafică se face conform cu opțiunile indicate în comandă sau în raport cu cele implicite. Unele din aceste opțiuni se pot stabili și din meniul contextual.

Comanda `plot3d` poate fi folosită sub mai multe forme. Prezentăm de fiecare dată numărul minim de parametri ceruți.

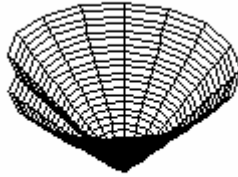
Notăție MAPLE	Suprafața/Suprafețele reprezentate
<code>plot3d(f(x,y),x = a..b,y=c..d)</code>	$z = f(x,y),$ $(x,y) \in [a,b] \times [c,d]$
<code>plot({f(x,y),g(x,y)},x = a..b,y=c..d)</code>	$z = f(x,y), z = g(x,y)$ $(x,y) \in [a,b] \times [c,d]$
<code>plot([f(u,v),g(u,v),h(u,v)],u=a..b,v=c..d)</code>	$\begin{cases} x = f(u, v) \\ y = g(u, v) \\ z = h(u, v) \end{cases}$

Exemple

```
>plot3d(cos(x)*sin(y), x=-2*Pi..2*Pi, y=-2*Pi..2*Pi);
```



```
>plot3d([v*cos(u), v*sin(u), v*ln(u)], u=Pi..4*Pi, v=0..1);
```



Comenzile **animate** și **animate3d** sunt destinate animației în plan și spațiu. Comanda

```
animate3d(f(x,y,t),x=a..b,y=b..c,t=t1..t2)
```

crează animație cu ajutorul cadrelor obținute prin reprezentarea grafică a suprafețelor

$$z_t = f(x, y, t), \quad (x, y) \in [a, b] \times [c, d]$$

pentru valori ale lui t în intervalul $[t_1, t_2]$.

Numărul de cadre poate fi stabilit cu ajutorul opțiunii `frames` (implicit sunt 8).

Exemplu:

```
>animate3d(cos(x)*sin(t*y),x=-Pi..Pi,y=-  
Pi..Pi,t=1..2);
```

