MAPLE, UN STANDARD PENTRU MATEMATICA COMPUTERIZATĂ

Dana Petcu

December 2, 1996

Moto:

„Principala utilitate a matematicii este aceea de a rezolva probleme, iar cea mai bună cale de a învăţa matematica este prin rezolvarea de probleme."

... Din colecţia de texte de pe Internet

Prefaţă

Cum se poate învăţa un limbaj de programare? Majoritatea persoanelor învaţă studiind exemple şi experimentând. Acest document este dedicat lor şi conţine un număr mare de exemple, exerciţii şi probleme în scopul învăţării mediului Maple.

De ce Maple? Maple este unul dintre numeroasele medii care permit calcule matematice simbolice şi numerice. Alături de Mathematica, Matlab şi MathCad este, în acest moment, unul dintre cele mai cunoscute CAS-uri (Computer Algebra System). Diferitele versiuni (astăzi cea mai nouă fiind Maple V Release 4.0) s-au dezvoltat simultan pe platforme UNIX, DOS, Windows, iar sursele programelor utilizatorilor sunt portabile. Facilităţile oferite sunt similare cu cele ale mediilor Mathematica şi Matlab, astfel încât odată cunoscut mediul Maple, utilizatorii pot uşor trece de la unul la altul.

Primele trei capitole constituie parţial suportul cursurilor de software matematic de la secţiile facultăţii de matematică. Următoarele două sunt dedicate orelor de laborator. Acest document nu are pretenţia de a explica şi exemplifica toate facilităţile mediului Maple, ci numai o parte dintre acestea.

De menţionat este faptul că în limba română există un număr extrem de redus de cărţi ce prezintă software-ul matematic existent în acest moment pe piaţa mondială.

Autoarea documentului îşi exprimă speranţa ca parcurgerea acestor pagini să ajute cititorul atât în învăţarea mediului Maple, cât şi în întărirea cunoştinţelor dobândite la cursurile de matematică. Poate că în viitorul destul de apropiat un asemenea mediu ar putea să constituie un instrument valoros pentru elevi, studenţi şi profesori în înţelegerea obiectelor matematice şi a modului de operare cu acestea.

Timişoara, 1.12.1996

Autoarea

Capitolul 1

Introducere

1.1 Ce este Maple?

Manipularea sofisticată a simbolurilor şi expresiilor pentru rezolvarea de ecuaţii şi investigarea unor funcţii poate fi realizată cu ajutorul pachetelor software numite *sisteme de calcul algebric* (Computer Algebra System -CAS). Un CAS poate fi utilizat pentru a genera soluţii simbolice pe care utilizatorul le poate determina utilizând aparatul matematic actual, pentru a oferi aproximaţii numerice similar utilizării unui calculator de buzunar, sau pentru a realiza o serie de reprezentări grafice. Maple este unul din numeroasele CAS-uri; alte exemple sunt Mathematica, Macsyma, Derive, MathCad, Axiom, Reduce etc.

Maple este un program care manipulează expresii matematice simbolice. Este capabil, de exemplu, să rezolve sisteme de ecuaţii algebrice sau diferenţiale, liniare sau neliniare. Maple este un rezolvitor de probleme care suportă o varietate de operaţii matematice precum analiză simbolică, analiză numerică şi grafică. Permite calcul aritmetic cu precizie mare. În plus, poate fi programat să execute aproape orice sarcină (task) dorită de utilizator. Maple cuprinde un limbaj complet de programare care utilizează o sintază similară cu C, Pascal, Basic sau Fortran.

1.2 Cine utilizează Maple?

Acest limbaj este indispensabil celor care vor să determine soluţiile unor probleme de dimensiuni mari.

Principala forţă a Maple-ului constă în aritmetica exactă. Lucrând cu fracţii şi radicali, evită erorile de rotunjire. Maple permite manipularea unor numere în virgulă flotantă cu exponenţi pozitivi foarte mari sau negativi foarte mici. De exemplu, poate calcula 123.456e789·987.654e3210. Limita mărimii exponenţilor depinde de versiunea Maple şi de tipul de calculator pe care este implementat.

Având peste 2500 de funcţii, Maple este un asistent perfect în matematică. Este un utilitar care permite educatorilor şi cercetătorilor să fie mai exacţi, mai creativi, mai productivi şi mai eficienţi. Este de asemenea un utilitar esenţial pentru ingineri şi oameni de ştiinţă care lucrează cu formule, ecuaţii şi date.

Maple poate fi utilizat în diferite moduri. O asemenea modalitate este

asistarea la efectuarea unor calcule lungi şi încurcate. Aceasta nu înseamnă că utilizatorul nu trebuie să ştie să rezolve problemele singur; înseamnă doar că există un utilitar care poate fi utilizat atunci când calculele devin prea complicate. Maple nu ne permite să rezolvăm orice problemă care nu poate fi rezolvată cu creionul şi hârtia; este un utilitar care simplifică lucrul implicat în rezolvarea unei probleme, dacă este utilizat adecvat.

Maple este un sistem de calcul algebric care poate fi utilizat în asistarea unor calcule lungi care intervin în procesul de rezolvarea a ecuaţiilor diferenţiale. În acest sens, Maple este pentru ecuaţiile diferenţiale ceea ce este un calculator pentru aritmetică.

Maple poate fi utilizat de asemenea în programarea metodelor numerice pentru aproximarea soluţiilor ecuaţiilor unui model matematic la fel de bine ca pentru generarea reprezentărilor grafice ale acestor soluţii.

1.3 Când ne ajută Maple?

Răspunsul este simplu. Când rezolvăm o problemă.

Ce este o problemă? În cele ce urmează, o problemă este acel ceva care poate fi formulată ca o întrebare a cărui răspuns implică anumite soluţii matematice.

Problemele apar de obicei într-un anumit context. Odată ce contextul este bine înţeles, problema poate fi formulată şi pusă, astfel încât probabil se poate alege o metodă de găsire a soluţiei. În procesul de rezolvare a problemei pot să apară alte probleme.

Procesul de rezolvare a unei probleme este un proces activ, care poate fi însă blocat în momentul în care nu se cunoaşte pasul următor al rezolvării.

Care sunt etapele rezolvării unei probleme?

1. Specificarea problemei care presupune mai multe etape:
   (a) citirea cu atenţie a problemei în scopul înţelegerii semnificaţiei tuturor termenilor ce intervin („o problemă înţeleasă este pe jumătate rezolvată”);
   (b) trasarea unei diagrame, a unui grafic, a unei imagini reprezentative (putem utiliza în această etapă facilităţile grafice ale Maple-ului);
   (c) stabilirea dimensiunilor şi variabilelor;
   (d) Determinarea modelului matematic (relaţii, ecuaţii, funcţii);
2. Rezolvarea ecuaţiilor (această fază poate fi realizată în totalitate sau parţial de către Maple sau un alt mediu de calcul simbolic sau numeric);
3. Interpretarea rezultatelor (dacă soluţiile obţinute sunt realiste, dacă sunt toate soluţiile etc.)

Fie următorul exemplu foarte simplu.

Problema: Un bazin este de trei ori mai lung decât lat. De asemenea este cu 40 de m mai lung decât lat. Determinaţi dimensiunile bazinului.

Model matematic: Se consideră l lungimea bazinului şi w lăţimea sa. Din ipotezele problemei aflăm că

```
> eq1 := l = 3*w:
```

$$eq1 := l = 3\,w$$

```
> eq2 := l = w + 40;
```

$$eq2 := l = w + 40$$

Soluţia 1: Utilizăm funcţii Maple de rezolvare a sistemelor de ecuaţii:

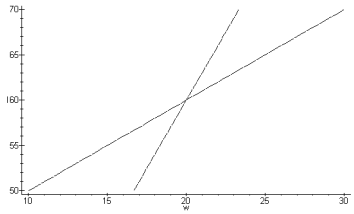


Figura 1.1:

```
> solve({eq1,eq2},{l,w});
```

$$\{\, w = 20, l = 60 \,\}$$

Soluţia 2: Ghicim rezultatul şi îl verificăm. Dacă bazinul are 25 metri lăţime şi 75 metri lungime,

```
> subs({w=25,l=75},[eq1,eq2]);
```

$$[\, 75 = 75, 75 = 65 \,]$$

Încercăm alte valori...

```
> subs({w=24,l=72},[eq1,eq2]);
```

$$[\, 72 = 72, 72 = 64 \,]$$

Soluţia 3: Prin substituire:

```
> eq3 := subs(l=solve(eq1,l),eq2); sol1 := w = solve(eq3,w);
```

$$eq3 := 3\,w = w + 40$$

$$sol1 := w = 20$$

```
> sol2 := l = solve(subs(sol1,eq1),l);
```

$$sol2 := l = 60$$

```
> sol := {sol1, sol2};
```

$$sol := \{\, w = 20, l = 60 \,\}$$

Soluţia 4: Rezolvăm problema grafic:

```
> with(plots): implicitplot({eq1,eq2},w=10..30,l=50..70);
```

Grafic: vezi figura 1.1

Capitolul 2

Maple, limbaj de programare

2.1 Structura unui document Maple

Un `document` Maple este compus din maxim patru tipuri de câmpuri:

1. comenzi (instrucţiuni) Maple, pentru introducerea opţiunilor utilizatorilor (câmp marcat prin prompter-ul >);

2. rezultate Maple, pentru afişarea calculelor efectuate pe baza comenzilor;

3. grafice, pentru afişarea unor informaţii grafice în ferestre separate. Aceste grafice pot fi copiate şi pasate documentului Maple pentru a crea un doument complet.

4. text, pentru note explicatorii (câmp marcat prin lipsa prompter-ului de intrare sau prin caracterul #);

2.2 Reguli de scriere a informaţiilor într-un document Maple

Enumerăm mai jos o serie de reguli de bază în scrierea unui document Maple:

1. Orice comandă trebuie să se termine cu caracterul ; sau : (în caz contrar, Maple presupune că linia de comandă se continuă pe linia următoare). În cazul al doilea, rezultatul comenzii nu este afişat.

2. Asignarea valorilor unei variabile se face cu :=. Prin aceasta se atribuie valoarea din dreapta simbolului (`rhs`) numelui din stânga (`lhs`). Numele de variabile trebuie să difere de numele procedurilor şi constantelor Maple. Variabilele cărora li s-au asignat valori nu pot fi utilizate ca variabile la diferenţiere, integrare, limite, cicluri, grafice, etc. Pentru ştergerea valorii unei variabile, de exemplu x, se execută `x:='x'`.

3. Maple face diferenţa între litere mici şi litere mari. Numele x este diferit de X. Constantele comune sunt predefinite în Maple cu numele: `Pi, E`, şi `I`. `Pi` indică constanta 3.14159..., pe când `pi` reprezintă o variabilă (desemnată printr-o literă grecească în câmpul rezultatelor).

4. Pentru argumente de funcţii se utilizează parantezele rotunde, iar pentru

indici de matrice, vectori, liste, tablouri, mulţimi, parantezele drepte (indicii vor fi separaţi prin virgulă).

5. Prezenţa acoladelor indică o mulţime, parantezele drepte o listă, iar parantezele rotunde o secvenţă.

6. Operatorul de multiplicare trebuie introdus explicit, deci utilizatorul va scrie $2 * x$ şi nu $2x$.

7. Ghilimelele " au o semnificaţie specială, permiţând referirea la comanda anterioară.

8. Nu se permite spaţiu într-un câmp domeniu .. sau în semnul de asignare ,,:=". ". ." între a şi b indică toate elementele x de tipul lui a şi b, pentru care $a \leq x \leq b$.

9. Dacă Maple răspunde prin ecou la o comandă, înseamnă că nu o poate interpreta. Fie numele comenzii a fost introdus greşit, fie funcţia se găseşte într-o bibliotecă care trebuie încărcată cu `readlib` (pentru funcţii independente) sau cu `with` (pentru funcţii din pachete speciale).

10. ? sau `help` este utilizat pentru o cerinţă de informare.

Există o serie de versiuni pe platforme diferite pentru UNIX, Windows, DOS sau Macintosh. Principala diferenţă este interfaţa.

2.3 Relaţia Maple-ului cu alte limbaje de programare

- Maple este un limbaj de programare procedural. În plus conţine un număr de constructori de programare funcţională. Un programator obişnuit cu C, Pascal, Basic, Algol, Fortran sau Lisp poate să scrie uşor proceduri numerice în Maple.
- Maple nu necesită declaraţii de structuri (este mai mult asemănător din acest punct de vedere cu Basic-ul şi Lisp-ul). Există însă o serie de structuri de date. Conversia de la un tip la altul este posibilă prin programare explicită.
- Maple este interactiv, iar limbajul de programare este de tip *interpretor*. Maple nu este indicat pentru programe care efectuează calcule numerice intensive datorită posibilităţii de depăşire a capacităţii interpretorului. Este indicat în special pentru calcule numerice cu mare precizie şi ca utilitar pentru generarea unor coduri numerice.

2.4 Evaluare

Trebuie subliniate diferenţele importante dintre Maple şi limbajele tradiţionale de programare. Dacă unui identificator nu i s-a asignat o valoare, el are înţeles în sine, adică este un *simbol*. Simbolurile sunt utilizate pentru a reprezenta necunoscute în ecuaţii, variabile în polinoame, indici de însumare, etc. Fie de exemplu instrucţiunea de asignare în Maple

```
> p:=x^2+4*x+4;
```

$$p := x^2 + 4\,x + 4$$

Identificatorului p îi este asignată formula $x^2 + 4x + 4$. Identificatorului x nu i

s-a asignat nici o valoare, este doar un simbol, o necunoscută. Identificatorul p are deja o valoare; precum o variabilă dintr-un limbaj de programare, valoarea sa poate fi utilizată în calculele următoare la fel ca o variabilă obişnuită. Care este valoarea lui p?

```
> p;
```

$$x^2 + 4\,x + 4$$

Care este valoarea lui x?

```
> x;
```

$$x$$

Este simbolul x. Dacă p trebuie evaluat, considerăm de exemplu:

```
> x:=3;
> p;
```

$$x := 3$$
$$25$$

Utilizarea identificatorilor atât ca variabile de programare cât şi ca necunoscute matematice este un atu al Maple-ului faţă de alte limbaje de programare. Evaluarea de mai sus poate afecta eficienţa şi semantica programului. O serie de probleme cu care se confruntă utilizatorii au legătură cu folosirea identificatorilor atât pentru simboluri cât şi ca variabile de programare. De exemplu, ce se întâmplă dacă încercăm să calculăm:

```
> int(p,x);
Error, (in int) wrong number (or type) of arguments
```

A apărut o eroare la integrarea cu funcţia `int`. În acest caz x este interpretat ca simbol, ca variabilă de integrare. Pe de altă parte, anterior i s-a asignat o valoare, întregul 3. Maple evaluează argumentele din funcţia `int` şi încearcă să integreze 25 relativ la 3, ceea ce nu are sens. Cum putem să-l transformăm pe x din nou în simbol? Dezasignarea unei variabile x în Maple se face prin:

```
> x:='x'; int(p,x);
```

$$x := x$$
$$\frac{1}{3}\,x^3\,y + \frac{3}{4}\,x^4\,z + 2\,x$$

2.5 Expresii

În Maple, formulele matematice, de exemplu $1/2+1/3$, $\sin(x+\pi/2)$ sau $x^3y^3-2/3$ sunt numite *expresii*. Toate formulele sunt construite din simboluri, numere, operatori aritmetici şi funcţii. Simboluri sunt, de exemplu, `sin, x, y, Pi` etc. Numere sunt 12, 2/3, 2.1 etc. Operatorii aritmetici sunt + (adunare), − (scădere), ∗ (multiplicare), / (împărţire) şi ^ (ridicare la putere). Funcţii sunt $\sin(x)$, $f(x,y)$, $\min(x1,x2,x3,x4)$. De exemplu, formula $p = x^2y + 3x^3 + 2$ (de fapt un polinom), este introdus astfel:

```
> p:=x^2*y+3*x^3*z+2;
```

$$p := x^2\,y + 3\,x^3\,z + 2$$

iar formula $\sin(x+\pi/2)e^{-x}$ se scrie în Maple

```
> sin(x+Pi/2)*exp(-x);
```

$$\cos(\,x\,)\,e^{(\,-x\,)}$$

Se observă că Maple simplifică $\sin(x+\pi/2)$ la $\cos(x)$. Formulele în Maple sunt

reprezentate ca arbori (DAG – Directed Acyclic Graphs). Când programăm funcţii Maple pentru a manipula formule, la bază manipulăm arbori de expresii. Maple are trei routine de bază pentru examinarea acestor arbori de expresii: `type, op` şi `nops`. Funcţia `type` cu apelul `type`(f, t) returnează valoarea `true` dacă expresia f este de tip t. Tipurile de bază includ `string, integer, float, +, *, ^ , function`. Funcţia `whattype` este de asemenea utilă pentru tipărirea tipului unei expresii. De exemplu,

```
> type(p, integer); whattype(p); type(p,`+`);
```

false

+

true

Atenţie la utilizarea caracterului `. Nu are aceeaşi semnificaţie ca şi apostroful. Primul este utilizat pentru şiruri de caractere în Maple. Alte tipuri numerice sunt `rational, float` sau `numeric`. Tipul `rational` indică numere raţionale, adică întregi şi fracţii de întregi. Tipul `float` indică numere în virgulă mobilă, numere ce conţin punctul zecimal. Tipul `numeric` indică orice tip menţionat mai sus, adică expresii de tip `integer, rational` sau `float`. Maple face distincţie între numerele exacte şi numerele aproximative. Prezenţa punctului zecimal este semnificativă. De exemplu,

```
> 2/3; 2/3.0;
```

$$\frac{2}{3}$$

.6666666667

Funcţia `nops` returnează numărul de operanzi ai unei expresii. Pentru o sumă indică numărul de termeni din sumă. De exemplu,

```
> nops(p);
```

3

Funcţia `op` este utilizată pentru a extrage unul dintre operanzii unei expresii. Are sintaxa `op`(i, f) însemnând că extrage al i-lea operand al expresiei f. În exemplul nostru,

```
> op(1,p), op(2,p);
```

$x^2\,y,\ 3\,x^3\,z$

Funcţia `op` poate fi utilizată în forma `op`$(i..j, f)$. Aceasta returnează secvenţa de operanzi ai lui f de la i la j. De exemplu,

```
> op(1..3,p);
```

$x^2\,y, 3\,x^3\,z, 2$

O prescurtare este `op`(f), echivalentă cu `op`$(1..$`nops`$(f), f)$, care are semnificaţia creării unei secvenţe cu toţi operanzii lui f.

```
> type(op(2,p),`*`), nops(op(2,p)), op(i,op(2,p)), op(op(2,p));
```

true, 3, op$(i, 3\,x^3\,z)$, $3, x^3, z$

Exerciţii:

```
> object:=x^3*exp(1)-34/Pi;
> nops(object);
> op(object);
> op(1,object);
> op(1,op(1,object));
> frc:=(a+b)/(c+d);
> subsop(1=2*op(1,frc),frc);
```

2.6 Variabile şi nume indexate

Maple are două tipuri de *variabile* sau *nume* (ultima este denumirea dată de Maple unei variabile). Acestea sunt şiruri de caractere precum `x, sin, Pi`, care sunt de tipul `string` şi nume indexate sau variabile indexate precum $A_1, A_{i,j}, A_{i_j}$, care sunt de tip `indexed` şi care pot fi introduse în Maple prin `A[i], A[i,j], A[i][j]`. Majoritatea funcţiilor din Maple acceptă ambele tipuri de variabile. Exemple:

```
> type(a,string); type(a,name);
```

true
true

```
> whattype(A[i]); type(A[i],indexed); type(A[i],name);
```

indexed
true
true

Dacă f este un nume indexat, atunci `nops` returnează numărul de indici şi `op` returnează al i-lea index.

```
> nops(A[i,j]); op(i,A[i,j]); op(0,A[i,j]);
```

$$2$$
$$\mathrm{op}\left(i, A_{i,j}\right)$$
$$A$$

```
> nops(A[i][j]), op(1,A[i][j]), op(0,A[i][j]);
```

$$1,\ j,\ A_i$$

Sintaxa unui apel de funcţie este $f(x_1, x_2, \ldots)$ unde f este numele funcţiei şi $x_1, x_2, \ldots$ sunt argumentele. Funcţia `nops` returnează numărul argumentelor, iar funcţia `op` returnează un anumit argument. În plus, `op(0,f)` returnează numele funcţiei. De exemplu,

```
> nops(f(x,y,z)), op(1..3,f(x,y,z)), op(0,f(x,y,z));
```

$$3,\ x, y, z,\ f$$

2.7 Instrucţiuni: asignare, condiţionare şi ciclare

Sintaxa Maple pentru asignare, `if, for` sau `while` este preluată de la Algol 60. O instrucţiune de asignare are forma *nume* := *expresie* unde *expresie* este o expresie oarecare. Definiţiile recursive sunt permise numai în anumite condiţii. De exemplu considerăm p dintr-un exemplu anterior:

```
> p:=x^2+4*x+4;
```

$$p := x^2 + 4\,x + 4$$

Ce se întâmplă dacă utilizăm acelaşi nume p în ambele părţi ale unei asignări?

```
> p:=p+x;
```

$$p := x^2 + 5\,x + 4$$

Numelui p îi este asignat rezultatul evaluării şi simplificării membrului drept

$p + x$. În evaluare p este polinomul $x^2 + 4x + 4$, iar x doar x. Deci rezultatul evaluării este $(x^2 + 4x + 4) + x$ şi cel al simplificării, rezultatul tipărit. Ce se întâmplă dacă lui p nu i-am fi asignat o valoare? De exemplu,

```
> q:=q+x;
Warning, recursive definition of name
```
$$q := q + x$$

Putem afirma că utilizatorul a uitat să asigneze o valoare lui q. Numelui q i s-a asignat o formulă care include q. Ce se întâmplă dacă dorim să evaluăm q? Maple încearcă evaluarea lui $q + x$. Dar această evaluare necesită evaluarea lui q. Rezultă un ciclu infinit. Astfel,

```
> q:=q+x^2;
Error, STACK OVERFLOW
```

Într-un limbaj de programare convenţional, această problemă a definiţiei recursive nu poate apărea deoarece toate variabilele trebuie să aibă valori. Depinzând de limbajul de programare, variabilele pot să primească o valoare implicită, sau deţin o valoare arbitrară aflată într-o locaţie de memorie pe care o desemnează prin numele său, sau limbajul detectează lipsa unui asignări.

Instrucţiunea de execuţie condiţionată în Maple are sintaxa:

```
if expr  then setinstr1
         [elif altexpr then altsetinstr]*
         [else setinstr2]
fi;
```

unde *setinstr*1, *altsetinstr*, *setinstr*2 sunt secvenţe de instrucţiuni separate prin caracterul ; , [ ] denotă o parte opţională, iar * denotă o parte care poate fi repetată de mai multe ori. Un exemplu tipic pentru instrucţiunea `if` este

```
> if x<0 then -1 elif x=0 then 0 else 1 fi;
```

Pentru instrucţiunea `for` există două variante. Prima este

```
[for nume] [from expr1] [by expr2] [to expr3] [while expr4]
                    do setinstruct od;
```

Fiecare din clauzele `for, from, by, to, while` poate fi omisă. Dacă sunt omise, valorile implicite sunt variabila `dummy`, 1, 1, `infinity` şi, respectiv, `true`. Un exemplu tipic pentru ciclul `for` este:

```
> for i to 3 do print(i^2) od;
```
$$1$$
$$4$$
$$9$$

Dacă sunt omise clauzele `for, from, by, to` ciclul este de tip `while`. De exemplu,

```
> i:=10^10+1: while not isprime(i) do i:=i+2 od: print(i);
```
$$10000000019$$

Combinaţia dintre `for` şi `while` este adesea utilă:

```
> for i from 10^10+1 by 2 while not isprime(i) do od;  print(i);
```
$$10000000019$$

A doua variantă de ciclul `for` este aşa numitul ciclu `for-in`. Această formă

provine dintr-o prescurtare pentru un ciclu care apare adesea

```
> for i to nops(s) do f(op(i,s)) od;
```

unde s poate fi o sumă, sau, mai general, orice expresie Maple sau structură de date precum o listă sau o mulţime. Forma ciclului `for-in` pentru exemplul dat este

```
for i in s do f(i) od;
```

Sintaxa unui ciclu `for-in` este

`[for` *nume*`] [in` *expr*`] [while` *expr*`]`
`do` *setinstr* `od;`

Exerciţii:

```
> x:=sqrt(sqrt(sqrt(2.))): y:=1.1:
> if x < y then 'x' else 'y' fi;
> '';
> for i from 1 by 2 to 11 do print(i!) od;
> for x from 1 by 4 while ithprime(x) < 100 do
> print(x,ithprime(x)); od;
> i:='i': x:='x': y:='y': for i in 4*x-3*y-6 do i/2 od;
```

2.8 Structuri de date

Programele mai complicate presupun manipularea şi stocarea datelor. Modul în care reprezentăm datele poate afecta algoritmii pe care îi scriem şi, în plus, timpul de execuţie. Maple are prevăzută o mulţime mare de structuri de date. Cele menţionate în continuare sunt secvenţele, listele, mulţimile, tablourile şi matricele. Maple nu are prevăzut explicit tipurile listă înlănţuită sau tipul înregistrare.

Expresiile Maple sunt clasificate în diferite tipuri de date. De exemplu, expresiile aritmetice sunt clasificate funcţie de tipul lor: sume ca tip "+", produse ca tip "*", etc. Cuvântul cheie Maple `whattype` ne oferă informaţii despre o expresie particulară:

```
> whattype(1/2);
```

fraction

```
> whattype(a + b);
```

+

```
> whattype(x^2 + x = 2*x - 1);
```

=

```
> whattype(a,b,3);
```

exprseq

2.8.1 Secvenţe

O *secvenţă* (de tip `exprseq`) este o înşiruire de expresii separate prin virgule. O secvenţă poate fi descrisă direct:

```
> s:=1,4,9,16,25;
```

$$s := 1, 4, 9, 16, 25$$

```
> t:=sin,cos,tan;
```

$$t := \sin, \cos, \tan$$

Secvenţele de secvenţe se simplifică într-o secvenţă mai mare, adică secvenţele sunt asociative. De exemplu,

```
> s:=1,(4,9,16),25;
```

$$s := 1, 4, 9, 16, 25$$

```
> s,s;
```

$$1, 4, 9, 16, 25, 1, 4, 9, 16, 25$$

Simbolul special `NULL` indică o secvenţă vidă. Secvenţele sunt adesea utilizate în Maple. Cu ajutorul lor se pot construi liste şi mulţimi. Apelul unei funcţii se bazează pe o secvenţă de argumente ale funcţiei. De exemplu funcţiile `min` sau `max` consideră un număr arbitrar de valori ca argumente, adică o secvenţă de argumente.

```
> max(s);
```

$$25$$

```
> min(s,0,s);
```

$$0$$

Funcţiile `op` şi `nops` nu pot fi utilizate pentru secvenţe. Explicaţia constă în aceea că secvenţa în sine devine listă de argumente pentru aceste funcţii. Astfel apelul `op(s)` este echivalent cu `op(1,4,9,16,25)`, rezultând o eroare. Pentru a utiliza aceste două funcţii este necesară transformarea secvenţei într-o listă:

```
>  nops(3,4,a);
Error, wrong number (or type) of parameters in function nops
> nops([3,4,a]);
```

$$3$$

Funcţia `seq` este extrem de utilă în crearea secvenţelor. Are două forme ce corespund celor două variante pentru cicluri `for`. Prima este de tipul

`seq(f(i),i=m..n)`

În această formă, rezultatul `seq` este echivalent cu rezultatul utilizării ciclului

```
> s:=NULL; for i from m by 1 to n do s:=s, f(i) od;
```

De exemplu,

```
> seq(i^2,i=1..5);
```

$$1, 4, 9, 16, 25$$

```
> s:=NULL; for i from 1 to 5 do s:=s,i^2 od;
```

$$s :=$$
$$s := 1$$
$$s := 1, 4$$
$$s := 1, 4, 9$$
$$s := 1, 4, 9, 16$$

$$s := 1, 4, 9, 16, 25$$

Se observă că utilizarea `seq` este mai eficientă deoarece nu crează secvenţe intermediare.

A doua variantă pentru `seq` este

```
seq(f(i),i=a)
```

care este echivalentă cu

```
seq(f(op(i,a)),i=1..nops(a))
```

Prezentăm câteva exemple (funcţia `coeff` calculează coeficientul termenului de grad i al polinomului în x, iar funcţia `D` în Maple este operatorul derivată):

```
> a:=3*x^3+y*x-11;
```

$$a := 3\,x^3 + y\,x - 11$$

```
> seq(coeff(a,x,i),i=0..degree(a,x));
```

$$-11, y, 0, 3$$

```
> seq(D(f),f=[sin,cos,tan,exp,ln]);
```

$$\cos, -\sin, 1 + \tan^2, \exp, a \to \frac{1}{a}$$

2.8.2 Liste şi mulţimi

Listele, mulţimile şi funcţiile sunt construite pe bază de secvenţe. O *listă* este a structură de date care colectează mai multe expresii (obiecte). Pentru crearea listelor se utilizează parantezele drepte. Astfel o listă poate fi definită ca o secvenţă de expresii încadrate de paranteze drepte.

```
> l:=[x,1,1-z,x];
```

$$l := [\,x, 1, 1 - z, x\,]$$

```
> whattype(l);
```

list

Lista vidă este desemnată prin [ ].

Aplicaţia 1: Putem construi un grafic dintr-o listă de puncte. De exemplu, pentru a trasa un pătrat cu vârfurile (1,1), (3,1), (3,3), (1,3) se crează o listă:

```
> ab := [1,1,3,1,3,3,1,3,1,1];
```

$$ab := [\,1, 1, 3, 1, 3, 3, 1, 3, 1, 1\,]$$

```
> plot(ab);
```

Grafic: vezi figura 2.1.a

Se observă că originea nu este inclusă în domeniul vizualizat. Coordonatele pot fi restricţionate:

```
> plot(ab,x=0..4,y=0..4);
```

Grafic: vezi figura 2.1.b

Dacă considerăm o curbă în plan descrisă parametric prin $x = f(t),\ y = g(t)$, cu un parametru $t \in [a, b]$, atunci această curbă se poate trasa plecând de la o listă cu câte 3 valori pentru un punct. De exemplu, pentru jumătatea superioară a cercului de rază 4 şi centru la (1,5):

```
>  plot( [1+4*cos(t),5+4*sin(t),t=0..Pi],scaling=constrained);
```

Grafic: vezi figura 2.2

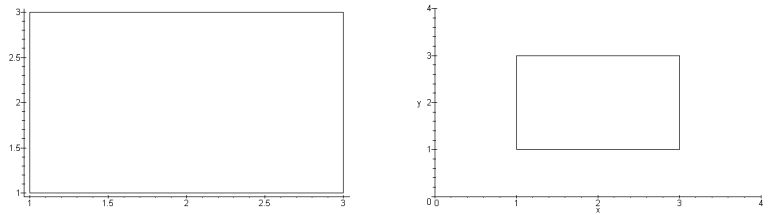


Figura 2.1:

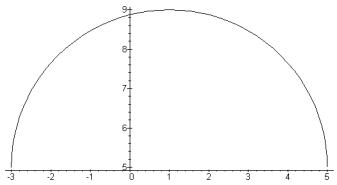


Figura 2.2:

O *mulţime* poate fi de asemenea utilizată pentru a colecţiona mai multe expresii. Diferenţa este aceea că duplicatele sunt eliminate. Se utilizează acoladele pentru marcare.

```
> s:={x,1,1-z,x};
```

$$s := \{\, 1 - z, 1, x \,\}$$

```
> whattype(s);
```

$$set$$

Mulţimea vidă este desemnată prin { }. Funcţia `nops` returnează numărul elementelor unei liste sau a unui mulţimi, iar funcţia `op` extrage un element de rang dat. Se poate utiliza de asemenea notaţia cu indici pentru a accesa un element de indice dat al unei secvenţe, liste sau mulţimi. De exemplu,

```
> op(1,s); s[1]; op(1..3,s); s[1..3];
```

$$1 - z$$
$$1 - z$$
$$1 - z, 1, x$$
$$1 - z, 1, x$$

Pentru a analiza dacă o listă sau o mulţime conţine un element x putem utiliza ciclul de mai jos, rezultând `true` în caz afirmativ, `false` în caz contrar:

```
> for i to nops(s) while s[i]<>x do od:
> if i>nops(s) then print(false) else print(true) fi;
```

$$\text{true}$$

Funcţia predefinită `member` efectuează această operaţie. `member`(x, s) returnează `true` dacă elementul x este în lista sau mulţimea s. Adăugarea unui nou element la lista l se face prin

```
> l:=[op(l),x];
```

$$l := [\, x, 1, 1 - z, x, x \,]$$

Se poate elimina al i-lea element al unei liste prin

```
> i:=2: l:=[l[1..i-1],l[i+1..nops(l)]];
```

$$l := [\, x, 1-z, x, x \,]$$

sau utilizând funcţia `subsop`

```
> l:=subsop(3=NULL,l);
```

$$l := [\, x, 1-z, x \,]$$

Funcţia `subsop` operează asupra oricărui tip de expresie.

Pentru mulţimi, există operatorii reuniune: `union`, intersecţie: `intersect`, scădere: `minus` etc.

```
> t:={u,x,z}; s union t; s intersect t; s minus t;
```

$$t := \{\, x, z, u \,\}$$
$$\{\, 1-z, 1, x, z, u \,\}$$
$$\{\, x \,\}$$
$$\{\, 1-z, 1 \,\}$$

Maple ordonează elementele mulţimilor într-un mod straniu, inprevizibil, la prima vedere. Algoritmii utilizaţi de Maple pentru a elimina duplicatele, pentru reuniune, intersecţie şi diferenţă funcţionează sortând prima dată elementele mulţimilor de intrare. Maple sortează pe baza ,,adresei maşină", adică în ordinea în care elementele apar în memoria calculatorului. Poziţia lor este imprecizabilă. Acest mod de abordare se datorează condiţiei ca operaţia să fie realizată cât mai rapid posibil.

Maple nu are lista înlănţuită ca structură de dată explicită. Listele în Maple sunt matrice de pointeri la intrările lor. O listă, precum şi o mulţime, în Maple, diferă de o matrice prin aceea că este o structură de tip `read-only`. Astfel nu se poate asigna o valoare unui element de listă (precum în cazul matricelor). Listele înlănţuite sunt structuri de date recursive. Considerăm exemplul reprezentării unui polinom $a(x) = \sum_{i=0}^{n} a_i x^i$. Una din posibilităţi este aceea de a stoca coeficienţii $a_0, \ldots, a_n$ într-o listă. Polinomul $p = x^4+3x^2+2x+11$ poate fi astfel reprezentat prin `[11, 2, 3, 0, 1]`. Putem utiliza însă lista înlănţuită `[1, [0, [3, [2, [11, NIL]]]]]`. În acest caz structura este recursivă şi constă din două valori, a doua indicând un pointer la altă listă ce conţine restul datelor sau valoarea specială `NIL`, lista înlănţuită vidă. De ce este mai bine să reprezentăm polinoame ca liste înlănţuite în loc de liste simple? Raţiunea principală este aceea că adăugarea unui nou element într-o listă înlănţuită se face în timp constant în loc de timp liniar. Adăugarea termenului $5x^5$ la polinomul p presupune operaţia `[op(p),5]`. Operatorul `op` crează secvenţa de intrări în timp constant, rezultatul fiind secvenţa (11, 2, 3, 0, 1), 5. Această secvenţă este transformată în 11, 2, 3, 0, 1, 5. Pentru a realiza această netezire sunt necesare 6 variabile intermediare de memorie. În general, adăugarea lui $a_{n+1}x^{n+1}$ la polinomul de grad n necesită un timp şi spaţiu de ordinul $\mathcal{O}(n)$. În cazul unei liste înlămnţuite, adăugarea se face prin `[5,p]` care necesită doar o listă de lungime 2. Timpul şi spaţiul necesar sunt de ordinul $\mathcal{O}(1)$.

Exerciţii:

```
> {3,2,4,5,6};
> [3,2,4,5,6];
> {a,b,a,b,b,c,a};
> [a,b,a,b,b,c,a];
> [seq(i^2,i=1..15)];
> primele10prime:=[seq(ithprime(i),i=1..10)];
> member(x^2,[1,2,y+3,x^2,5/7]);
```

```
> lis:=[seq(6^i+1,i=1..100)]: select(isprime,lis);
```

2.8.3 Şiruri de caractere

Sunt delimitate de caractere ‘. De exemplu,

```
> ‘Acesta este un sir‘;
```

Acesta este un sir

```
> ‘I love ‘.‘you‘;
```

I love you

```
> substring(",3..9);
```

love yo

```
> readlib(search): search(‘ov‘,"""");
```

false

```
> x:=5: y:=4:
> convert(x,string); convert(y,string);
```

5

4

```
> z:=cat(x,y);
```

$z := 54$

```
> whattype(z);
```

string

```
> length(z);
```

2

2.8.4 Tabele

Tabelele sunt extrem de utile pentru scrierea unor programe eficiente. Un tabel este o relaţie unu-la-mai mulţi între două seturi discrete de date. De exemplu, considerăm un tabel de translaţie a culorilor din engleză în franceză, germană şi română.

```
> COLOUR[red]:=rouge, rot, rosu;
```

$COLOUR_{red} := rouge, rot, rosu$

```
> COLOUR[blue]:=bleu, blau, albastru;
```

$COLOUR_{blue} := bleu, blau, albastru$

```
> COLOUR[yellow]:=jaune, gelb, galben;
```

$COLOUR_{yellow} := jaune, gelb, galben$

Domeniul tabelului `COLOUR` este numele culorii în engleză. Rangul tabelului este

secvenţa de nume de culori în franceză, germană şi română. Valorile domeniului în tabel sunt numite *chei* sau *indici*. Funcţia Maple `indices` returnează secvenţa indicilor. Valorile rangului din tabel sunt numite *valori* sau *intrări*. Funcţia Maple `entries` returnează secvenţa intrărilor.

```
> indices(COLOUR);
```

$$[\,yellow\,], [\,red\,], [\,blue\,]$$

```
> entries(COLOUR);
```

$$[\,jaune, gelb, galben\,], [\,rouge, rot, rosu\,], [\,bleu, blau, albastru\,]$$

Se observă că ordinea în care funcţiile `indices` şi `entries` returnează indicii şi intrările nu este în mod necesar aceea în care au fost introduşi în tabel. Acest fapt se explică prin tehnica Maple-ului de a construi tabele pentru a permite o căutare extrem de rapidă (tehnica „hashing").

Ce se poate face cu un tabel? Facilitatea de bază este aceea că dând o cheie sau un indice de tabel putem găsi rapid intrarea corespunzătoare. De exemplu,

```
> COLOUR[red];
```

$$rouge, rot, rosu$$

returnează rapid numele culorii în alte limbi. Cât de rapid? În acelaşi timp, indiferent de lungimea intrările tabelului.

Putem de asemenea testa dacă o intrare se află în tabel:

```
> assigned(COLOUR[green]);
```

$$\text{false}$$

şi putem şterge intrări din tabel prin dezasignare

```
> COLOUR[blue]:='COLOUR[blue]'; print(COLOUR);
```

$$COLOUR_{blue} := COLOUR_{blue}$$

$$\begin{aligned}&\text{table}([\\ &\quad yellow = (\,jaune, gelb, galben\,)\\ &\quad red = (\,rouge, rot, rosu\,)\\ &\quad])\end{aligned}$$

2.8.5 Vectori şi matrice

O `matrice` unidimensională poate fi creată utilizând comanda `array`: `array(`m `..`n`)`. Aceasta crează o matrice cu indicii $m,\ m+1, \ldots, n$. Adesea m este 1. Pot fi inserate intrări în matrice prin asignări precum la tabele:

```
> restart; n:=3; v:=array(1..n); v[1]:=a;
```

$$\begin{aligned}n &:= 3\\ v &:= \text{array}(\,1..3, [\,]\,)\end{aligned}$$

```
> for i from 2 to n do v[i]:=a*v[i-1] mod n od;
```

$$\begin{aligned}v_1 &:= a\\ v_2 &:= a^2\\ v_3 &:= a^3\end{aligned}$$

Matricele unidimensionale se aseamănă cu tabelele cu un indice restricţionat la un rang fix. Sunt mult mai eficiente în accesare decât tabelele şi în plus prevăd verificarea apartenenţei indicelui în rang. Considerăm exemplul de uti-

lizare a unei matrice unidimensionale pentru sortarea prin metoda bulelor a unei secvenţe de numere.

```
> a:=0.5:
> for i to n-1 do
>    for j from i+1 to n do
>         if v[i]>v[j] then temp:= v[i]; v[i]:=v[j]; v[j]:=temp fi
>    od
> od;
> evalm(v);
```

$$[\,.125\;.25\;.5\,]$$

Matrice bi- sau multi-dimensionale se pot construi în mod similar. O matrice bidimensională poate fi creată cu `array(`$c..d$`,`$m..n$`)`. Fie, de exemplu, 4 polinoame simetrice în variabilele x_1, x_2, x_3:

```
> v:=array(1..4): v[1]:=1: v[2]:=x[1]+x[2]+x[3]:
> v[3]:=x[1]*x[2]+x[1]*x[3]+x[2]*x[3]:v[4]:=x[1]*x[2]*x[3]:
```

Construim matricea bidimensională M a cărui element M_{ij} este derivata lui v_i de x_j:

```
> M:=array(1..4,1..3):
> for i to 4 do for j to 3 do M[i,j]:=diff(v[i],x[j]) od od:
> M; eval(M);
```

$$M$$

$$\begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 1 \\ x_2+x_3 & x_1+x_3 & x_1+x_2 \\ x_2\,x_3 & x_1\,x_3 & x_1\,x_2 \end{bmatrix}$$

Se observă că valoarea lui M este numele matricei M. Regulile de evaluare pentru matrice şi tabele sunt speciale. Explicaţia constă în aceea că pot exista matrice cu intrări neasignate.

2.8.6 Articole

Maple nu are o structură de tip `record` ca în Pascal sau `struct` ca în C. Printr-o structură de tip înregistrare se înţelege o structură de date care reuneşte o colecţie heterogenă de obiecte, adică o mulţime de obiecte nu în mod necesar de acelaşi tip.

Se consideră două exemple. Primul se referă la înregistrarea informaţiei legate de un cuaternion. Un cuaternion este un număr de forma $a + bi + cj + dk$ unde a, b, c, d sunt numere reale. Pentru a reprezenta un cuaternion, este necesară stocarea celor patru cantităţi a, b, c, d. Cel de-al doilea exemplu se referă la factorizarea unui polinom în $Q[x]$. Factorizarea lui $a(x)$ este $a(x) = uf_1^{e_1} \cdots f_n^{e_n}$ unde fiecare din factorii $f_i \in Q[x]$ sunt monoame ireductibile. Este necesară în acest caz stocarea factorilor f_i, a exponenţilor e_i şi a valorii $u \in Q$.

Există mai multe posibilităţi de reprezentare a structurii `record` în Maple. Cea mai simplă este utilizarea unui liste. Cuaternionul $a + bi + cj + dk$ poate fi reprezentat prin lista $[a, b, c, d]$. Utilizăm indici pentru a ne referi la o componentă a structurii de date. Se poate utiliza `macro` pentru a defini un identificator

ca fiind egal cu o constantă dacă se preferă utilizarea unui simbol pentru a referi o componentă. De exemplu:

```
> a:=[-1/2,[[x+1,2],[x-1,1]]];
```

$$a := \left[\frac{-1}{2}, [[x+1, 2], [x-1, 1]]\right]$$

```
> macro(unit=1,factori=2,baza=1,exponent=2);
> a[unit]; a[factori][1][baza];
```

$$\frac{-1}{2}$$
$$x+1$$

```
> a[unit]*a[factori][1][baza]^a[factori][1][exponent]*a[2][2][1]^
>   a[2][2][2];
```

$$-\frac{1}{2}(x+1)^2(x-1)$$

O a doua posibilitate pentru reprezentarea unui articol în Maple este utilizarea unui apel de funcţie. De exemplu putem reprezenta $a + bi + cj + dk$ prin `CUATERNION(i,j,k,1)`. Un avantaj al acestei reprezentări este acela că putem spune Maple-ului cum să efectueze operaţii diverse asupra funcţiilor.

```
> CUATERNION(2,3,0,1);
```

$$\text{CUATERNION}(2, 3, 0, 1)$$

```
> `print/CUATERNION`:=proc(a,b,c,d) a+b*'i'+c*'j'+d*'k' end:
> CUATERNION(2,3,0,1);
```

$$2 + 3\,i + k$$

Am definit o procedură de tipărire. Apostroful a fost introdus în procedură pentru a obţine identificatorii i, j, k la rezultat, şi nu valorile variabilelor i, j, k care pot fi asignate în program (de obicei le folosim ca variabile de ciclare).

O a treia posibilitate de reprezentare a unui `record` este de a-l gândi ca un polinom multivariabile în câmpul numelor, şi de a-i stoca valorile coeficienţilor. Acest mod este util când câmpurile sunt numerice şi se doreşte efectuarea unor operaţii cu aceste câmpuri. De exemplu, putem reprezenta un cuaternion ca un polinom de variabilele $i,\ j,\ k$:

```
> restart; z1:=2+3*i+k; z2:=2-3*i+2*j+2*k; coeff(z1,i); z1+z2;
```

$$z1 := 2 + 3\,i + k$$
$$z2 := 2 - 3\,i + 2\,j + 2\,k$$
$$3$$
$$4 + 3\,k + 2\,j$$

2.9 Funcţii şi proceduri

2.9.1 Proceduri, variabile locale, return, error

O procedură Maple are sintaxa

```
proc   (secvnume1)
       [local secvnume2;]
       [options secvnume3;]
       secvinstruct
end;
```

unde *secvnume1, secvnume2, secvnume3* sunt secvenţe de simboluri separate prin virgule. De exemplu o procedură simplă este

```
> proc(x,y) x^2+y^2 end;
proc(x,y) x^2+y^2 end
```

Această procedură are doi parametrii x şi y. Valoarea returnată este x^2+y^2. În general valoarea returnată de o procedură este ultima valoare calculată, dacă nu există o valoare returnată explicit.

Următoarea procedură returnează `true` dacă x este în lista L şi `false` în caz contrar.

```
> MEMBER:= proc(x,L) local v;
>   for v in L do if v=x then RETURN(true) fi od;
>   false
> end;
MEMBER :=
    proc(x,L)
    local v;
        for v in L do  if v = x then RETURN(true) fi od; false
    end
```

Procedura are o variabilă locală v pentru a nu interfera cu variabila globală v a utilizatorului. Variabilele care apar într-o procedură, şi nu sunt parametrii sau variabile locale, sunt variabile globale.

Funcţia `ERROR` poate fi utilizată pentru a genera un mesaj de eroare într-o procedură.

```
> MEMBRU :=proc(x,L) local v;
>   if not type(L,list) then ERROR(‘Argument 2: o lista‘) fi;
>   for v in L do if v=x then RETURN(true) fi od;
>   false
> end:
```

2.9.2 Operatori săgeată

Pentru proceduri simple care utilizează numai formule, există o sintaxă alternativă, numită sintaxa *săgeată*:

simbol`-> [ local` *secvnume;*`]` *expr*

În cazul unei funcţii de un parametru, sau pentru doi sau mai mulţi parametrii

(*sevcsimbol*)`->[local` *secvnume;*`]` *expr*

De exemplu, procedura pentru x^2+y^2 poate fi rescrisă `(x,y)-> x^ 2+y^ 2`.

2.9.3 Operatori

Maple permite specificarea unui operator în două modalităţi:

```
> f := x -> x^3-4;
```

$$f := x \to x^3 - 4$$

```
> f(2);
```
$$4$$
```
> f:=<x^3-4|x>;
```
$$f := \langle x^3 - 4 \,|\, x \rangle$$
```
> f(2);
```
$$4$$

O expresie neevaluată poate fi transformată într-un operator utilizând funcţia `unapply`:

```
> f:=unapply(z^3-4,z);
```
$$f := z \rightarrow z^3 - 4$$
```
> f(2);
```
$$4$$

De exemplu, Maple are operatorul predefinit de diferenţiere:

```
> D(sin);
```
$$\cos$$
```
> D(ln);
```
$$a \rightarrow \frac{1}{a}$$

Operatorul `D` primeşte ca parametru o funcţie şi returnează o funcţie. În primul caz, funcţia `cos` există. În al doilea caz, nu există funcţie specifică în Maple, astfel încât `D` returnează un operator săgeată care transformă numere în inversele lor.

2.9.4 Relaţii

O relaţie între două variabile x şi y este specificată printr-o ecuaţie de forma:

```
> y = x^2 + 1;
```
$$y = x^2 + 1$$

Pentru fiecare valoare a lui x, putem utiliza ecuaţia pentru a calcula valorile corespunzătoare a lui y. Curba de dependenţă poate fi trasată cu comanda Maple `implicitplot`. Eliminaţi caracterul # pentru a obţine graficul.

```
> #plots[implicitplot](y=x^2+1,x=-3..3,y=-10..10);
```

Pentru a testa dacă un asemenea grafic reprezintă o funcţie, putem aplica următorul test: ,,Dacă este posibil să găsim o linie verticală care traversează graficul în mai mult de un punct, atunci relaţia nu este o funcţie.” Aplicând această regulă, graficul anterior este graficul unei funcţii, la fel ca următoarele:

```
> eq1 := y=x^2+1: eq2 := y=x^3-3*x^2-2:
> #implicitplot({eq1, eq2}, x=-3..3, y=-10..10);
```

Următoarele relaţii nu sunt însă funcţii (vezi grafic):

```
> eq1 := x=y^2+1: eq2 := x=y^3-3*y^2-2:
> plots[implicitplot]({eq1, eq2}, x=-10..10, y=-3..3);
```

Grafic: vezi figura 2.3

Când y este funcţie de x, scriem adesea $y = f(x)$. $f(x)$ este numele expresiei dependente de x care este utilizată pentru a calcula valori diverse ale lui y. f

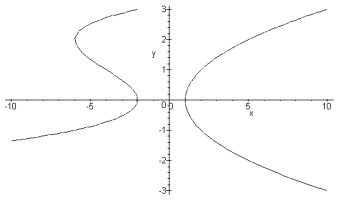


Figura 2.3:

este numele funcţiei. Astfel putem utiliza expresii dependente de x pentru a defini funcţia. Dată o valoare x, definiţia funcţiei indică cum să fie calculată $f(x)$. Spunem deci că x este transformat în $f(x)$ aplicând funcţia (sau regula) f.

```
> f := (x) -> x^2 + 3:
```

Putem calcula valori y din ecuaţia $y = f(x)$ aplicând regula f la diverse valori x:

```
> f(x);  f(3); f(a^2+b^2);
```

$$x^2 + 3$$
$$12$$
$$(a^2 + b^2)^2 + 3$$

2.9.5 Funcţii

O funcţie f este privită ca o regulă pentru asignarea oricărui argument x dintr-un set de numere a unui unic număr $f(x)$ numit valoarea lui x. Funcţiile pot fi definite în mai multe moduri.

1. Funcţii definite ca expresii: asignarea

```
> aria := Pi*r^2;
```

$$aria := \pi\, r^2$$

defineşte aria cercului ca funcţie de raza sa. Funcţia arie definită ca expresie poate fi evaluată cu **subs**.

```
> subs(r=3,aria);
```

$$9\,\pi$$

Funcţia poate fi trasată prin:

```
> # plot(aria,r=0..4);
```

rezultând o fereastră separată pentru graficul funcţiei pe intervalul $r = 0..4$.

2. Funcţii definite prin operatorul săgeată: asignarea

```
> aria := r -> Pi*r^2;
```

$$aria := r \rightarrow \pi\, r^2$$

defineşte de asemenea funcţia arie. Pentru a găsi aria cercului de rază 3, putem scrie simplu:

```
>  aria(3);
```

$$9\,\pi$$

Pentru a trasa funcţia:

```
> # plot(aria,0..4):
```

3. Utilizând **unapply** se pot trasnforma expresii de una sau mai multe variabile în funcţii definite cu operatorul săgeată. De exemplu:

```
> pol := x^2 + 4*x -1;
```

$$pol := x^2 + 4\,x - 1$$

```
> pol := unapply(pol,x);
```

$$pol := x \to x^2 + 4\,x - 1$$

4. Funcţii definite ca proceduri: asignarea

```
>  aria := proc(r) Pi*r^2 end;
aria := proc(r) Pi*r^2 end
```

defineşte de asemenea funcţia arie. Poate fi evaluată şi trasată precum în definiţia ca operator săgeată. Unul din avantajele definirii în acest mod a funcţiei este faptul că domeniul poate fi specificat. De exemplu, domeniul funcţiei arie este cel al numerelor reale pozitive. Această informaţie poate fi inserată în procedură utilizând cuvântul cheie ERROR. Mesajul de eroare este încadrat între caracterele ‘.

```
> aria := proc(r)
>   if r <= 0 then ERROR(‘Raza trebuie sa fie pozitiva‘)
>   else Pi*r^2 fi
> end;
aria :=
    proc(r)
        if r <= 0 then ERROR(‘Raza trebuie sa fie pozitiva‘)
        else Pi*r^2
        fi
    end
> aria(3);
```

$$9\,\pi$$

```
> aria(-3);
Error, (in aria) Raza trebuie sa fie pozitiva
```

Funcţii de două variabile pot fi de asemenea definite şi trasate. Fie de exemplu, volumul V al unui cilindru de înălţime h şi rază r:

```
> V := (r,h) -> Pi*r^2*h;
```

$$V := (\,r, h\,) \to \pi\, r^2\, h$$

Pentru a vizualiza graficul lui V se poate utiliza **plot3d**:

```
> plot3d(V,0..4,0..4);
```

Grafic: vezi figura 2.4

Toate funcţiile matematice standard se întâlnesc în Maple. Pentru o listă completă se pot cere informaţii suplimentare cu

```
> ?inifcns.
```

Operatorul de compunere a funcţiilor este @.

```
> y := sin@cos@(x->x^2+3);
```

$$y := \sin@\cos@(\, x \to x^2 + 3\,)$$

```
> y(3.);
```

$$.7472099577$$

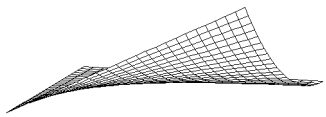


Figura 2.4:

2.9.6 Parametrii şi domeniul lor

Maple acceptă procedură în procedură. De exemplu putem scrie

```
> restart; f1:=proc(x) local g; g:=x->x+1; x*g(x) end:
```

Procedura f_1 are o variabilă locală g care este o procedură. Aceasta calculează $x(x+1)$. Procedura de mai sus nu este însă echivalentă cu

```
> f2:=proc(x) local g; g:=()->x+1; x*g() end:
```

deoarece x din procedura g nu se referă la parametrul x din procedura f_2. Se referă la variabila globală x. Astfel

```
> f1(a); f2(a);
```

$$a\,(\,a+1\,)$$
$$a\,(\,x+1\,)$$

```
> x:=7; f2(a);
```

$$x := 7$$
$$8\,a$$

2.9.7 Proceduri recursive şi opţiunea remember

Numerele Fibonacci F_n sunt definite prin relaţia recursivă $F_0 = F_1 = 1$, $F_n = F_{n-1} + F_{n-2}$. Putem construi o procedură recursivă:

```
> F:=proc(n) if n=0 or n=1 then 1 else F(n-1)+F(n-2) fi end:
> seq(F(i),i=0..10);
```

$$1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89$$

```
> F(200);
Error, (in F) STACK OVERFLOW
```

Acesta nu este un mod eficient de a calcula termenii şirului. De fapt, nu este posibilă calcularea lui `F(200)` nici pe cel mai rapid calculator. Dacă se numără apelurile procedurii `F` se observă că apeluri cu aceleaşi argumente se repetă. O posibilitate de evitare este aceea de a reţine valorile calculate. Se poate considera un ciclu:

```
> F:=proc(n) local fnm1, fnm2, f, i;
>   if n=0  or n= 1 then RETURN(1) fi;
>   fnm2:=1; fnm1:=1;
>   for i to (n-1) do f:=fnm1+fnm2; fnm2:=fnm1; fnm1:=f od;
```

```
>   fnm1
> end:
> F(200);
```

$$453973694165307953197296969697410619233826$$

O altă posibilitate este aceea de a utiliza opţiunea `remember`. Aceasta este utilizată pentru a stoca valorile care sunt calculate astfel încât pot fi folosite ulterior.

```
> restart; F:=proc(n) option remember;
>          if n=0 or n=1 then 1 else F(n-1)+F(n-2) fi end:
> F(200);
Error, (in F) STACK OVERFLOW
> F(80); F(150); F(200);
```

$$37889062373143906$$

$$1613053142490458141579790738634 9$$

$$453973694165307953197296969697410619233826$$

Fiecare procedură Maple are asociat un tabel `remember`. Indexul tabelului este secvenţa de argumente, iar intrarea tabelului este valoare funcţiei. Când F este calculat la n, Maple verifică în tabelul `remember` al procedurii F să vadă dacă $F(n)$ a fost deja calculat. Dacă da, returnează rezultatul din tabelul `remember` al funcţie F. În caz contrar, execută codul procedurii F şi automat stochează perechea $n, F(n)$ în tabelul `remember` al lui F.

Există şi posibilitatea de salvare explicită a valorilor în tabelul `remember` utilizând aşa numita asignare funcţională. Aceasta este mult mai flexibilă decât opţiunea `remember` deoarece permite salvarea numai a valorilor selectate în tabelul `remember`.

```
> F:=proc(n) F(n):=F(n-1)+F(n-2) end:
> F(0):=1: F(1):=1:
> F(200);
Error, (in F) STACK OVERFLOW
> F(80), F(150), F(200);
```

$$37889062373143906, 1613053142490458141579790738634 9,$$
$$453973694165307953197296969697410619233826$$

2.9.8 Funcţiile type şi map

Funcţia `type` poate fi utilizată pentru a opera, de exemplu, asupra unor parametrii de tipuri diferiţi ai unei proceduri. Astfel procedura `DIFF` diferenţiază expresii care sunt polinoame de o variabilă x.

```
> DIFF:=proc(a,x) local u,v;
>   if not type(a, algebraic)
>    then ERROR(‘Primul argument trebuie sa fie o formula‘) fi;
>   if not type(x,name)
>    then ERROR(‘Al doilea  argument trebuie sa fie un nume‘) fi;
>   if type(a, numeric) then 0
>   elif type(a,name) then if a=x then 1 else 0 fi
>   elif type(a,‘+‘) then map(DIFF,a,x)
>   elif type(a,‘*‘) then u:=op(1,a); v:=a/u; DIFF(u,x)*v+DIFF(v,x)*u
```

```
>   elif type(a,anything^integer) then
>     u:=op(1,a); v:=op(2,a); v*DIFF(u,x)*u^(v-1)
>   else ERROR(`Nu se poate diferentia`,a)
>   fi
> end:
```

Funcţia `map` utilizată mai sus are următoarea sintaxă la apel: `map(`$f, a, x_1, \ldots, x_n$`)`. Semnificaţia este aplicarea funcţiei f la operanzii expresiei a pasând argumentele $x_1, \ldots, x_n$ lui f. De obicei, argumentelor opţionale sunt neglijate. Formal, această situaţie este echivalentă cu cea întâlnită în calculul secvenţei

```
> seq(f,op(i,a),x1,...,n),i=1..nops(a));
```

De exemplu,

```
> p:=x^3+2*x+1: map(F,p);
```

$$\mathrm{F}(\,x^3\,) + \mathrm{F}(\,2\,x\,) + \mathrm{F}(\,1\,)$$

```
> map(x->x^2,p); map(degree,p,x);
```

$$x^6 + 4\,x^2 + 1$$

$$4$$

Funcţia `DIFF` utilizează tipul `structured`. Tipurile `anything, name, +` şi `*` sunt tipuri simple. Tipul `anything^ integer` este de tip `structured`, adică valoarea trebuie să fie o putere, baza poate fi orice, dar exponentul trebuie să fie întreg. Este echivalent cu a scrie

```
> if type(a,`^`) and type(op(2,a),integer) then
```

Tipurile structurate permit înlocuirea testelor lungi de tip cu teste concise. Fie două exemple comune. Tipul `nume=algebraic..algebraic` este tipul argumentului secund pentru integrarea definită şi sumarea definită, precum în

```
> int(exp(-2*t)*ln(t),t=0..infinity);
```

$$-\frac{1}{2}\ln(\,2\,) - \frac{1}{2}\,\gamma$$

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

$$\frac{3}{4}$$

Pe de altă parte, este permisă integrarea nedefinită şi sumarea nedefinită.

```
> int(exp(-2*x)*ln(x),x);
```

$$-\frac{1}{2}\,\mathrm{e}^{(\,-2\,x\,)}\ln(\,x\,) - \frac{1}{2}\,\mathrm{Ei}(\,1, 2\,x\,)$$

```
> sum(1/(n^2-1),n);
```

$$-\frac{1}{n} - \frac{1}{2}\,\frac{1}{n\,(\,n-1\,)}$$

Astfel tipul argumentului secund poate fi un nume de variabilă sau tipul mai sus descris. Putem descrie acest fapt prin {`name, name=algebraic..algebraic`}. Utilizarea acoladelor denotă alternativele.

Un alt exemplu comun este testarea unei multimi sau liste de nume sau ecuaţii. O mulţime de ecuaţii poate fi testată cu tipul `set(equation)`:

```
> type({x+y=2, x-y=3}, set(equation));
```

true

`Solve` acceptă ca argument o secvenţă de formule care sunt implicit egalate cu

zero:

```
> solve({x+y-2,x-y-3},{x,y});
```

$$\left\{y = \frac{-1}{2}, x = \frac{5}{2}\right\}$$

Deci tipul trebuie să fie {`set(algebraic),set(equation)`}. Acest tip nu este identic cu `set({algebraic, equation})`.

2.9.9 Număr variabil de argumente

Este posibil în Maple ca o funcţie să aibă un număr variabil de agumente. Un exemplu de asemenea funcţie a mediului este `max`. O procedură similară este următoarea:

```
> MAX:=proc(x1) local maximum, i;
>   maximum:=x1;
>   for i from 2 to nargs do
>     if args[i]>maximum then maximum:=args[i] fi od;
>   maximum
> end:
```

Variabila specială `nargs` indică numărul argumentelor, iar variabila `args` este o secvenţă de argumente, `args[i]` fiind al i-lea argument.

2.9.10 Returnarea apelului procedurii ca expresie neevaluată

Procedura `MAX` scrisă mai sus poate fi utilizată numai în cazuri numerice, pe când funcţia Maple `max` permite calcul simbolic.

```
> MAX(1,2,x);
Error, (in MAX) cannot evaluate boolean
> max(1,2,x);
```

$$\max(\,2, x\,)$$

Analog, funcţia `signum(x)` returnează -1 dacă $x < 0$, $+1$ dacă $x \geq 0$, altfel returnează `unevaluated`, adică returnează `signum(x)`:

```
> signum(sqrt(2)-1); signum(sqrt(2)-Pi); signum(a-b);
```

$$1$$
$$-1$$
$$\mathrm{signum}(\,a - b\,)$$

Maple nu poate executa procedura `MAX` pentru că nu poate evalua `args[i]<maximum` dacă nu se introduc valori numerice. Procedura ar trebui să lucreze cel puţin pentru a determina maximul dintre π şi $\sqrt{2}$. Utilizăm în acest scop funcţia `signum`.

```
> MAX:=proc() local a,i,j,n,s;
>   n:=nargs; a:=array(1..n); for i to n do a[i]:=args[i] od; i:=1;
>   while i<n do
>     j:=i+1;
>    while j<=n do
>      s:=signum(a[i]-a[j]);
>      if s=1 then a[j]:=a[n]; n:=n-1;
```

```
>      elif s=-1 then a[i]:=a[j]; a[j]:=a[n]; j:=n; n:=n-1; i:=i-1;
>      else j:=j+1 fi
>    od;
>    i:=i+1
>   od;
>   if n=1 then RETURN(a[1]) fi;
>   'MAX'(seq(a[i],i=1..n));
> end:
```

Caracterele apostrof indică faptul că funcţia `MAX` nu este reexecutată (altfel s-ar produce un ciclu infinit).

```
> MAX(x,1,sqrt(2),x+1);
```

$$\mathrm{MAX}\left(x+1, \sqrt{2}\right)$$

2.9.11 Simplificări şi reguli de transformare

De multe ori este necesară introducerea unor simplificări care să pot fi descrise algebric sau prin reguli de transformare. Dată o funcţie f putem afirma, de exemplu, că f este comutativă şi asociativă. `max` este o asemenea funcţie, adică `max(a,b)=max(b,a)` şi `max(a,max(b,c))=max(max(a,b),c)`. Astfel este necesară o formă canonică de a scrie expresiile ce implică funcţia `map`. Comutativitatea se poate implementa prin sortarea argumentelor. Pentru asociativitate vom transforma atât `max(max(a,b),c)` cât şi `max(a,max(b,c))` în `max(a,b,c)`. În plus `max(max(a))=max(a)` (idempotenţa).

```
> MAX:=proc() local a;
>   a:=[args]; a:=map(flatten,a, MAX); 'MAX'(op(sort(a)));
> end:
> flatten:=proc(x,f)
>   if type(x,function) and op(0,x)=f then op(x) else x fi
> end:
> MAX(a,MAX(c,b),a);
```

$$\mathrm{MAX}(a, a, b, c)$$

Să introducem şi proprietatea `max(a,a)=a`.

```
> MAX:=proc() local a;
>    a:={args}; a:=map(flatten,a,MAX); 'MAX'(op(a));
> end:
> MAX(a,MAX(c,b),a);
```

$$\mathrm{MAX}(a, b, c)$$

Am văzut într-o secţiune anterioară că pentru a urmări modul de execuţie al procedurii `MAX`, se poate modifica `printlevel`. În cazul dat putem astfel obţine ieşirile nu numai de la `MAX` ci şi de la `flatten`. Funcţia `trace` poate fi utilizată pentru a selecta urmărirea numai a unor procedurilor specificate.

```
> trace(MAX);
```

$$MAX$$

```
> MAX(a,MAX(b,a),c);
{--> enter MAX, args = b, a
```

$$a := \{\, a, b \,\}$$
$$a := \{\, a, b \,\}$$
$$\mathrm{MAX}(\, a, b \,)$$

```
<-- exit MAX (now at top level) = MAX(a,b)}
{--> enter MAX, args = a, MAX(a,b), c
```

$$a := \{\, a, c, \mathrm{MAX}(\, a, b \,) \,\}$$
$$a := \{\, a, b, c \,\}$$
$$\mathrm{MAX}(\, a, b, c \,)$$

```
<-- exit MAX (now at top level) = MAX(a,b,c)}
```

$$\mathrm{MAX}(\, a, b, c \,)$$

2.9.12 Urmărirea execuţiei unei proceduri

Considerăm o procedură de calcul al celui mai mare divizor comun a două numere naturale utilizând algoritmului lui Euclid. Funcţia `irem` calculează restul întreg al împărţirii a doi întregi.

```
> GCD:=proc(a,b) local c,d,r;
>   c:=a; d:=b; while d<>0 do r:=irem(c,d); c:=d; d:=r od;  c
> end:
```

Cea mai simplă facilitate de a urmări execuţia unei proceduri este `printlevel`. Variabila `printlevel` este o variabilă globală care iniţial este setată pe 1. Dacă se setează o valoare mai mare se tipăreşte urma tuturor asignărilor, intrările şi ieşirile procedurii.

```
> printlevel:=100:
> GCD(21,15);
{--> enter GCD, args = 21, 15
```

$$c := 21$$
$$d := 15$$
$$r := 6$$
$$c := 15$$
$$d := 6$$
$$r := 3$$
$$c := 6$$
$$d := 3$$
$$r := 0$$
$$c := 3$$
$$d := 0$$
$$3$$

```
<-- exit GCD (now at top level) = 3}
```

$$3$$

Procedura `GCD` operează în aritmetica întregilor de precizie arbitrară, astfel încât se poate lucra cu numere foarte mari:

```
> printlevel:=1: GCD(100!, 2^100);
```

$$158456325028528675187087900672$$

Procedura poate fi scrisă şi recursiv:

```
> GCD:=proc(a,b) if b=0 then a else GCD(b,irem(a,b))fi end:
> printlevel:=40: GCD(15,21);
{--> enter GCD, args = 15, 21
{--> enter GCD, args = 21, 15
{--> enter GCD, args = 15, 6
{--> enter GCD, args = 6, 3
{--> enter GCD, args = 3, 0
                                3

<-- exit GCD (now in GCD) = 3}
                                3

<-- exit GCD (now in GCD) = 3}
                                3

<-- exit GCD (now in GCD) = 3}
                                3

<-- exit GCD (now in GCD) = 3}
                                3

<-- exit GCD (now at top level) = 3}
                                3
```

Sursa unei proceduri de bibliotecă poate fi vizualizată modificând variabila `verboseproc`.

```
> interface(verboseproc=2):
> eval(isprime);
proc(n)
local btor,nr,p,r;
options remember,system,`Copyright 1993 by Waterloo Maple Software`;
  if not type(n,integer) then
      if type(n,numeric) then ERROR(`argument must be an integer`)
      else RETURN(îsprime(n)
      fi
  fi;
  if n < 2 then false
  elif has(`isprime/w`,n) then true
  elif igcd(230556796394551842475310214733175607 0,n) <> 1 then false
  elif n < 10201 then true
  elif igcd(
  849696948923341811053233990918734996592606258664893273661154 54\
  263422038932707693909090694773095091375097869171186680288614 99\
  333825097682386722983737962963066757674131126736578936440788 15\
  718696989373063311306647862044862494925732402262739543736363 90\
  387526081667586612559568346306972204475122988482222285500626 83\
  786342519960225996301315945644470064720696621750477244528915 92\
  7867113,n) <> 1 then
      false
  elif n < 1018081 then true
  else
      nr := igcd(408410100000,n-1);
      nr := igcd(nr^5,n-1);
      r := iquo(n-1,nr);
      btor := modp(power(2,r),n);
      if `isprime/cyclotest`(n,btor,2,r) = false or
          irem(nr,3) = 0 and `isprime/cyclotest`(n,btor,3,r) =
          false or irem(nr,5) = 0 and `isprime/cyclotest`
          (n,btor,5,r) = false or irem(nr,7) = 0 and
          `isprime/cyclotest`(n,btor,7,r) = false then
```

```
            RETURN(false)
        fi;
        for p from 3 while numtheory[jacobi](p^2-4,n) <> -1 do od;
        evalb(`isprime/TraceModQF`(p,n+1,n) = [2,p])
    fi
end
```

2.9.13 Argumente opţionale şi valori implicite

Multe din funcţiile Maple acceptă argumente opţionale. Utilizatorul poate evita specificarea parametrilor prin existenţa valorilor implicite. Exemple sunt funcţiile `plot, factor, collect, series`. Funcţia `degree` calculează gradul unui polinom univariat în variabila x, cu gradul total al polinomului multivariat. De exemplu,

```
> p:=x^3+2*x+1; degree(p);
```

$$p := x^3 + 2\,x + 1$$
$$3$$

```
> p:=3*x^2*y+2*y^2-x*z+7; degree(q); degree(q,x);
```

$$p := 3\,x^2\,y + 2\,y^2 - x\,z + 7$$
$$1$$
$$0$$

Argumentul secund este opţional.

Cum putem formula o asemenea procedură? Presupunem că intrarea este o formulă şi, dacă este dat un al doilea argument, atunci acesta este un nume sau o mulţime de nume pentru variabile.

```
> DEGREE:=proc(a,x) local s,t;
>    if nargs=1 then
>       s:=indets(a);
>       if not type(s,set(name))
>        then ERROR(`Intrarea nu este un polinom`) fi;
>       DEGREE(a,s)
>   elif type(a,constant) then 0
>   elif type(a,name) then
>      if type(x,name) then if a=x then 1 else 0 fi
>      elif type(x,set(name)) then if member(a,x) then 1 else 0 fi
>      else ERROR(`Argument 2: nume sau o multime de nume`)
>      fi
>   elif type(a,`+`) then max(seq(DEGREE(t,x),t=a))
>   elif type(a,`+`) then
>      s:=0; for t in a do s:=s+DEGREE(t,x) od; s
>   elif type(a, algebraic^integer) then DEGREE(op(1,a),x)*op(2,a)
>   else ERROR(`Nu se poate calcula gradul`)
>   fi
> end:
```

Funcţia `indets` utilizată mai sus returnează mulţimea tuturor nedeterminatelor (sau variabilelor) care apar ca intrare.

2.9.14 Returnarea rezultatelor prin parametrii

Numeroase funcţii în Maple returnează mai mult decât o valoare. Este posibilă returnarea unui număr oarecare utilizând o secvenţă sau o listă. Pe de altă parte este posibilă returnarea valorilor prin parametrii precum în alte limbaje de programare. De exemplu, să considerăm funcţia `divide` din Maple care permite efectuarea împărţirii dintre două polinoame. La apelul `divide(a,b)` se returnează `true` dacă şi numai dacă polinomul b divide polinomul a fără rest:

```
> divide(x^3-1,x-1);
```

true

De obicei, dacă b divide a, utilizatorul are nevoie în continuarea calculelor de câtul împărţirii, q. În Maple, acest lucru poate fi realizat dând în funcţia `divide` un al treilea parametru, care este un nume şi căruia i se va asigna câtul dacă b divide pe a:

```
> if divide(x^3-1,x-1,`q`) then print(q) fi;
```

$$x^2 + x + 1$$

Caracterul ` este utilizat pentru a pasa funcţiei `divide` numele q şi nu valoarea lui q.

Revenim la un exemplu anterior: am construit o funcţie `MEMBER` care testează dacă x aparţine listei L. Să modificăm această funcţie astfel încât `MEMBER(x,L,`p`)` să returneze în plus în p poziţia primei apariţii a lui x în L.

```
> MEMBER:=proc(x,L,p) local i;
>    for i to nops(L) do
>     if x=L[i] then if nargs=3 then p:=i fi;
>         RETURN(true)
>     fi
>    od;
>    false
> end:
> MEMBER(4,[1,3,5],`pozitie`);
```

false

```
> pozitie;
```

pozitie

```
> MEMBER(3,[1,3,5],`pozitie`); pozitie;
```

true

2

2.9.15 Interfaţa cu funcţiile Maple

Multe rutine Maple au interfeţe care permit utilizatorului să le „învăţe" cum să efectueze calcule asupra unei funcţii date de utilizator. Acestea includ `diff, evalf, expand, combine, simplify, series, latex` etc. Dacă utilizatorul doreşte să specifice cum se va face diferenţierea unei funcţii W trebuie să scrie o rutină `` `diff/W` ``. Dacă procedura `diff` este apelată într-o expresie $f(x)$ care conţine $W(g)$, atunci rutina `diff` va invoca `` `diff/W`(g,x) `` care va calcula

derivata lui $W(g)$ funcţie de x. Să presupunem că $W'(x) = W(x)/(1 + W(x))$. Atunci:

```
> `diff/W`:=proc(g,x) diff(g,x)*W(g)/(1+W(g)) end:
> diff(W(x),x), diff(W(x^2),x);
```

$$\frac{\mathrm{W}(x)}{1+\mathrm{W}(x)}, 2\,\frac{x\,\mathrm{W}(x^2)}{1+\mathrm{W}(x^2)}$$

Putem spune cum Maple trebuie să evalueze numeric o funcţie f definind procedura `` `evalf/f` `` astfel încât `` `evalf/f`(x) `` să calculeze numeric $f(x)$. De exemplu, să utilizăm iteraţiile Newton pentru a determina rădăcina pătrată a unei valori numerice.

```
> `evalf/Sqrt`:=proc(a) local x,xk, xkm1;
>   x:=evalf(a);
>   if not type(a,numeric) then RETURN(Sqrt(x)) fi;
>   if x<0 then ERROR(`Radical dintr-un numar negativ?`) fi;
>   Digits:=Digits+3; xkm1:=0; xk:=evalf(x/2);
>   while abs(xk-xkm1)>abs(xk)*10^(-Digits) do xkm1:=xk;
>         xk:=(xk+x/xk)/2; od;
>   Digits:=Digits-3; evalf(xk);
> end:
> x:=Sqrt(3): y:=sqrt(3):
> Digits:=60; evalf(x); evalf(y);
```

$$Digits := 60$$

1.73205080756887729352744634150587236694280525381038062805581
1.73205080756887729352744634150587236694280525381038062805581

2.10 Proceduri I/O

2.10.1 Proceduri de citire şi salvare

În cazul în care elaborăm programe mai lungi dorim de multe ori salvarea acestora într-un fişier.

Un program poate fi citit în Maple utilizând comanda `read`. De exemplu, dacă am scris procedura `MAX` în fişierul MAX, putem citi acest fişier prin

```
> read `MAX`;
```

Procedurile şi orice formulă de calcul dintr-o sesiune Maple poate fi salvată din linia de comandă utilizând comanda `save`, cu forma `save, f1, f2,...,`*`numefisier`*. Aceasta salvează datele f_1, $f_2, \ldots$ în format intern Maple, în aşa numitul format „.m".

Interfaţa mediului Maple V versiunea 3.0 pentru Windows 95 permite salvarea comenzilor, a datelor şi a rezultatelor calculelor într-un fişier tip Maple cu extensie ".ms", într-un fişier tip text cu extensie ".txt", sau într-un fişier tip LaTeX cu extensie ".tex".

Un fişier cu extensie ".ms" poate fi încărcat ulterior, afişându-se situaţia finală a worksheet-ului salvat. Pentru referirea la o anumită procedură sau dată din acest fişier, comenzile corespunzătoare trebuie reevaluate. Un fişier cu extensie ".txt" poate fi de asemenea încărcat cu condiţia ca: comenzile să fie precedate de caracterul ">", iar comentariile de caracterul "#".

În cazul în care se doreşte salvarea unei set restrâns de informaţii, este preferabilă utilizarea salvării din linia de comandă. În cazul în care salvarea se face într-un fişier cu extensia ".m", informaţia poate fi reutilizată.

```
> restart;
> FACTORIAL:= proc(number) option remember; local i, tmp1;
>  if number <= 1 then RETURN(1) fi;
>  i := 1; tmp1 := 1;
>  while i < number do tmp1 :=(i+1)*tmp1; i:=i+1 od;
>  RETURN(tmp1)
> end:
> save FACTORIAL, `fact.m`;
```

Prin restart toate informaţiile se pierd:

```
> restart;
> print(FACTORIAL);
```

$$FACTORIAL$$

```
> read `fact.m`;
> print(FACTORIAL);
proc(number)
local i,tmp1;
options remember;
    if number <= 1 then RETURN(1) fi; i := 1; tmp1 := 1;
    while i < number do  tmp1 := (i+1)*tmp1; i := i+1 od;
    RETURN(tmp1)
end
> alias(F=FACTORIAL);
```

$$I, F$$

```
> F(4); FACTORIAL(4); F(20);
```

$$24$$
$$24$$
$$2432902008176640000$$

```
> A:=plot([[0,1],[2,3]]):
> save A, `a.m`;
> restart;
> print(A);
```

$$A$$

```
> read `a.m`;
> op(A);
```

$$\text{CURVES}([[0, 1.], [2., 3.]], \text{COLOUR}(RGB, 0, 0, 0)),$$
$$\text{AXESTICKS}(DEFAULT, DEFAULT), \text{TITLE}(), \text{AXESLABELS}(,),$$
$$\text{VIEW}(DEFAULT, DEFAULT)$$

```
> NEWTON:=proc(f,start,n) local x1,x0,j,Df;
>  if not(type(f,procedure)) then ERROR(`Ceruta o functie`); fi;
>  Df:=unapply(diff(f(x),x),x); x0:=start;
>  for j from 1 to n do
>    x1:= x0-evalf(f(x0)/Df(x0)); x0:=x1; print(j,x0);
> od:
> end:
> save NEWTON, `/dana/newt.m`;
> with(DEtools);
```

$$[DEplot, DEplot1, DEplot2, Dchangevar, PDEplot, dfieldplot, phaseportrait]$$

```
> print(DEtools);
```

table([
$Dchangevar = \text{readlib}('DEtools/Dchangevar')$
$dfieldplot = \text{readlib}('DEtools/dfieldplot')$
$DEplot1 = \text{readlib}('DEtools/DEplot1')$
$DEplot2 = \text{readlib}('DEtools/DEplot2')$
$PDEplot = \text{readlib}('DEtools/PDEplot')$
$DEplot = \text{readlib}('DEtools/DEplot')$
$phaseportrait = \text{readlib}('DEtools/phaseportrait')$
])

Pentru crearea unui pachet propriu:

```
> restart;
> read `fact.m`;
> read `/dana/newt.m`;
> mypack:=table([fact=FACTORIAL,newt=NEWTON]);
```

$mypack := \text{table}([$
$fact = FACTORIAL$
$newt = NEWTON$
$])$

```
> save mypack,FACTORIAL,NEWTON, `mypack.m`;
> restart;
> read `mypack.m`; fact(4);
```

$$\text{fact}(4)$$

```
> with(mypack);
```

$$[fact, newt]$$

```
> fact(4);
```

$$24$$

```
> newt(x->cos(x)-x,0,5);
```

$$1, 1.$$

2, .7503638679
3, .7391128909
4, .7390851334
5, .7390851332

2.10.2 Apelul programelor externe

Dacă în Maple nu reuşim să realizăm ceva, apare normală ideea de a utiliza un alt program în loc de a reimplementa un algoritm în Maple. Pe de altă parte Maple este eficient în calcule simbolice şi nu este, în general, eficient pentru calcule numerice. Se preferă utilizarea unui alt limbaj de programare pentru calcule numerice asupra vectorilor sau integrări numerice multiple. Comunicaţiile de date în Maple se fac via fişiere. Programul Maple trebuie să scrie datele cerute de programul extern într-un fişier de intrare, utilizând, `writeto, appendto` şi `lprint`. Maple poate rula programul extern prin comanda `system`. Programul extern trebuie să citească datele din fişierul scris de Maple şi să scrie rezultatele într-un fişier de ieşire. După ce programul extern îşi termină execuţia, Maple citeşte rezultatele din fişierul de ieşire utilizând comanda `read`. O schiţă a codului Maple corespunzător este următoarea:

```
> interface(quiet=true);
> writeto(input);
> ... #scrie datele in fisierul input
> writeto(terminal);
> interface(quiet=false);
> system(...); #executa programul extern
> read output;
> ... #continua procesarea în Maple
```

Prima comandă `interface(quiet=true)` elimină toate diagnosticurile din Maple (octeţi utilizaţi, avertizări, etc) care nu trebuie să apară în fişierul de intrare. Comanda `writeto(input)` deschide fişierul `input` pentru scriere (cu suprascriere). Dacă se doreşte adăugarea unor date la un fişier deja existent se recomandă `appendto` în loc de `writeto`. Toate ieşirile Maple din acest punct se vor îndrepta spre fişier. Ieşirea este creată cu comanda `lprint`. După ce toate datele au fost scrise în fişier, acesta este implicit închis prin resetarea ieşirii către terminal prin comanda `writeto(terminal)`. Comanda `system` este utilizată pentru executarea unui program extern.

Una din problemele care intervin este legată de faptul că comunicaţiile au loc pe bază de fişiere tip text. Programul extern va trebui să ţină seama de forma comenzilor Maple în momentul în care încearcă să scrie date care urmează a fi citite de către Maple.

2.10.3 Legătura cu C-ul şi Fortran-ul

Este posibilă utilizarea comenzilor `C` şi `fortran` pentru a genera ieşiri ale formulelor Maple în forme acceptate de compilatoarele C şi Fortran. Să presupunem

că avem următorul polinom în Maple:

```
> f:=-3.9*x+1.8-1.7*x^3+3.4*x^2-.0003*x^7+.5*x^4-.09*x^5+.009*x^6:
```

Pentru a evalua polinomul într-un timp cât mai scurt îl scriem în forma Horner şi îl transcriem în C:

```
> h:=convert(f,horner);
```

$$h := 1.8+(-3.9+(3.4+(-1.7+(.5+(-.09+(.009-.0003\,x)\,x)\,x)\,x)\,x)\,x)\,x$$

```
> readlib(C):
> C(h);
  t0 = 0.18E1+(-0.39E1+(0.34E1+(-0.17E1+(0.5+(-0.9E-1+(0.9E-2-\
0.3E-3*x)*x)*x)*x)*x)*x)*x;
> C([r=h],filename=`temp.c`);
```

În fişierul `temp.c` regăsim expresia de mai sus, $t0$ fiind înlocuit cu r.

Presupunem că dorim ca în codul nostru să utilizăm determinantul derivatelor, Hessian-ul pentru expresia $e1$.

```
> restart; e1:=exp(-x^2) * (a*cos(y+t)+b*sin(y-t)):
> fortran(e1);
      t0 = exp(-x**2)*(a*cos(y+t)+b*sin(y-t))
> readlib(C): C(e1);
      t0 = exp(-x*x)*(a*cos(y+t)+b*sin(y-t));
```

Dorim să efectuăm calculele în Maple şi să convertim rezultatul în limbajul C.

```
>  with(linalg): h:=det(hessian(e1,[x,y,t]));
```

$$\begin{aligned} h := {} & -8\,(\mathrm{e}^{(-x^2)})^3\,a^2\cos(y+t)^2\,b\sin(y-t) - 8\,(\mathrm{e}^{(-x^2)})^3\,a\cos(y+t)\,b^2\sin(y-t)^2 \\ & + 16\,(\mathrm{e}^{(-x^2)})^3\,a^2\cos(y+t)^2\,b\sin(y-t)\,x^2 \\ & + 16\,(\mathrm{e}^{(-x^2)})^3\,a\cos(y+t)\,b^2\sin(y-t)^2\,x^2 \\ & + 16\,x^2\,(\mathrm{e}^{(-x^2)})^3\,a^2\sin(y+t)^2\,b\sin(y-t) \\ & + 16\,x^2\,(\mathrm{e}^{(-x^2)})^3\,b^2\cos(y-t)^2\,a\cos(y+t) \end{aligned}$$

```
> C([hessian=h]);
      hessian = -8.0*pow(exp(-x*x),3.0)*a*a*pow(cos(y+t),2.0)*b*si\
n(y-t)-8.0*pow(exp(-x*x),3.0)*a*cos(y+t)*b*b*pow(sin(y-t),2.0)+16.\
0*pow(exp(-x*x),3.0)*a*a*pow(cos(y+t),2.0)*b*sin(y-t)*x*x+16.0*pow\
(exp(-x*x),3.0)*a*cos(y+t)*b*b*pow(sin(y-t),2.0)*x*x+16.0*x*x*pow(\
exp(-x*x),3.0)*a*a*pow(sin(y+t),2.0)*b*sin(y-t)+16.0*x*x*pow(exp(-\
x*x),3.0)*b*b*pow(cos(y-t),2.0)*a*cos(y+t);
```

Subexpresiile care se repetă pot fi eliminate prin utilizarea opţiunii de optimizare.

```
> C([hessian=h],optimized);
      t1 = x*x;
      t2 = exp(-t1);
      t3 = t2*t2;
      t4 = t3*t2;
      t5 = a*a;
      t6 = t4*t5;
      t7 = y+t;
      t8 = cos(t7);
      t9 = t8*t8;
      t11 = y-t;
      t12 = sin(t11);
      t15 = t4*a;
```

```
      t16 = b*b;
      t18 = t12*t12;
      t29 = t1*t4;
      t32 = pow(sin(t7),2.0);
      t38 = pow(cos(t11),2.0);
      hessian = -8.0*t6*t9*b*t12-8.0*t15*t8*t16*t18+16.0*t6*t9*b*t*12\
 t1+16.0*t15*t8*t16*t18*t1+16.0*t29*t5*t32*b*t12+16.0*t29*t16*t38*a*t8;
```

Dacă dorim să calculăm inversa unei matrice simbolice:

```
> M := array(1..3,1..3,symmetric):
> M[2,3] := 0:
> M[1,1] := 18*cos(q3)*cos(q2)*m30*p^2-sin(q3)^2*j30y+ sin(q3)^2*
> j30z-9*sin(q3)^2*m30*p^2 + j10y+j30y+m10*p^2+18*m30*p^2:
> M[1,2] := 9*cos(q3)*cos(q2)*m30*p^2-sin(q3)^2*j30y+ sin(q3)^2*
> j30z-9*sin(q3)^2*m30*p^2+ j30y+9*m30*p^2:
> M[1,3] := -9*sin(q3)*sin(q2)*m30*p^2:
> M[2,2] := -sin(q3)^2*j30y+sin(q3)^2*j30z- 9*sin(q3)^2*m30*p^2+
> j30y+9*m30*p^2: M[3,3] := 9*m30*p^2+j30x:
> evalm(M):
```

O primă strategie este aceea de a calcula simbolic inversa matricei M. Maple poate efectua această operaţie, însă codul, chiar şi optimizat, va fi foarte lung.

```
> B := linalg[inverse](M):
> # C( B,optimized);
> readlib(cost):
> cost(optimize(B));
```

$$78\,\mathit{additions} + 181\,\mathit{multiplications} + \mathit{divisions} + 4\,\mathit{functions} + 9\,\mathit{subscripts} + 66\,\mathit{assignments}$$

O abordare mai bună presupune calculul inversei lui M înaintea asignării intrărilor.

```
> M := array(1..3,1..3,symmetric);
```

$$M := \mathrm{array}(\mathit{symmetric}, 1..3, 1..3, [\,])$$

```
> M[2,3] := 0:
> B := linalg[inverse](M):
```

Definim acum intrările matricei M (vezi relaţiile de mai sus).

```
> B:=map(eval,B):
> cost(optimize(B));
```

$$23\,\mathit{additions} + 53\,\mathit{multiplications} + \mathit{divisions} + 4\,\mathit{functions} + 9\,\mathit{subscripts} + 33\,\mathit{assignments}$$

Rezultatul este mai bun. În general este mai bine ca inversarea să nu se facă simbolic, ci numeric.

```
> C(M,optimized);
      t4 = p*p;
      t5 = m30*t4;
      t6 = cos(q3)*cos(q2)*t5;
      t7 = sin(q3);
      t8 = t7*t7;
      t9 = t8*j30y;
```

```
        t10 = t8*j30z;
        t12 = t8*m30*t4;
        t15 = 9.0*t6-t9+t10-9.0*t12+j30y+9.0*t5;
        t18 = t7*sin(q2)*t5;
        M[1][1] = 18.0*t6-t9+t10-9.0*t12+j10y+j30y+m10*t4+18.0*t5;
        M[1][2] = t15;
        M[1][3] = -9.0*t18;
        M[2][1] = t15;
        M[2][2] = -t9+t10-9.0*t12+j30y+9.0*t5;
        M[2][3] = 0.0;
        M[3][1] = -9.0*t18;
        M[3][2] = 0.0;
        M[3][3] = 9.0*t5+j30x;
    > cost( optimize(M) );
```

17 *additions* + 23 *multiplications* + 4 *functions* + 9 *subscripts* + 19 *assignments*

2.10.4 Legătura cu editoare de texte

Subrutinele `latex` şi `eqn` sunt destinate pentru prezentări ulterioare ale rezultatelor din Maple într-o formă apropiată de stilul articolelor şi cărţilor. Acestea constituie baza unui interpretor matematic pentru elaborarea de documente matematice.

```
> e2:=Int(Int(exp(x^2+y^2),x=-infinity..infinity),y=-Pi/2..zeta);
```

$$e2 := \int_{-1/2\,\pi}^{\zeta} \int_{-\infty}^{\infty} \mathrm{e}^{(x^2+y^2)}\,dx\,dy$$

```
> latex(e2);
\int _{-{\frac {\pi }{2}}}^{\zeta}\!\int _{-\infty }^{
\infty }\!{e^{{x}^{2}+{y}^{2}}}{dx}\,{dy}
> e3:=diff(f(x,y,z),x,y);
```

$$e3 := \frac{\partial^2}{\partial x\,\partial y}\mathrm{f}(x, y, z)$$

```
> latex(e3);
{\frac {\partial ^{2}}{\partial y\partial x}}f(x,y,z)
> readlib(eqn): eqn(e1);
{{e sup {-^ {  "x"  sup 2 }}}^{( {{ "a" ^{cos ( { "y" ^+^ "t" })}}}\
^+^{
 "b" ^{sin ( { "y" ^-^ "t" })}}} )}}
> eqn(e2);
{int from {{-^  pi  over 2 }} to {{size 13 zeta}} {int from {{-^  \
inf}} to { inf } {e sup {{  "x"  sup 2 }^+^{  "y"  sup 2 }}}\~\ d \
"x" }\~ d "y" }
> eqn(e3);
{d over {d "x" }{d over {d "y" }{f ( { "x" ,\~\  "y" ,\~\  "z" })}}}
```

Rezultatele pot fi salvate într-un fişier, adăugând numele fişierului ca parametru secund:

```
> #latex(e1,‘foo.tex‘);
```

Capitolul 3

Calcule cu Maple

3.1 Calcule aritmetice, exacte sau aproximative

Diferenţa dintre Maple şi un ,,calculator de buzunar" este aceea că Maple afişează rezultatele exacte, lucrează cu fracţii şi variabile. Maple lucrează cu fracţii atât cât este posibil, schimbându-le în numere zecimale numai la cerere:

```
> 100/12;
```

$$\frac{25}{3}$$

```
> 2-3+4/5*6^7;
```

$$\frac{1119739}{5}$$

```
> 1/3 + 1/2;
```

$$\frac{5}{6}$$

```
> evalf(5/6);
```

$$.8333333333$$

Maple acceptă şi numere complexe. I este constanta pentru $\sqrt{-1}$.

```
> (3+2*I)*(2-I);
```

$$8 + I$$

Numele pentru π, aria cercului de rază 1, în Maple este `Pi`.

```
> Pi*3^2;
```

$$9\,\pi$$

Maple nu are o limită a numărului de cifre a unui întreg:

```
> big:=100!;
```

$$big := 933262154439441526816992388562667004907159682643816214\backslash$$
$$68592963895217599993229915608941463976156518286253697\backslash$$
$$92082722375825118521091686400000000000000000000000$$

Maple are capacitatea de a manipula obiecte matematice. De exemplu, permite factorizarea într-un timp scurt a unor numere întregi mari:

```
> ifactor(big);
```

$$(2)^{97}\,(3)^{48}\,(5)^{24}\,(7)^{16}\,(11)^{9}\,(13)^{7}\,(17)^{5}\,(19)^{5}\,(23)^{4}\,(29)^{3}\,(31)^{3}\,(37)^{2}$$
$$(41)^{2}\,(43)^{2}\,(47)^{2}\,(53)\,(59)\,(61)\,(67)\,(71)\,(73)\,(79)\,(83)\,(89)\,(97)$$

Un alt exemplu privind capabilităţile de calcul aritmetic este determinarea celui mai mare divizor comun a două numere întregi:

```
> igcd(3^500,5^300-1);
```

$$9$$

```
> ilcm(9615319,12345678909876543210);
```

$$16958234427147887654490570$$

Maple utilizează un test probabilitistic pentru a determina dacă un număr este prim. Un rezultat `false` va fi întotdeauna corect, pe când un rezultat `true` este foarte probabil de a fi exact.

```
> isprime(3195958637);
```

$$\mathtt{true}$$

```
> nextprime(3195958637);
```

$$3195958679$$

Maple poate calcula numărul de întregi mai mici decât n, argumentul funcţiei, numere relativ prime cu n:

```
> numtheory[phi](76458146187641);
```

$$75925851724800$$

Maple efectuează calcule în virgulă flotantă cu o precizie arbitrară. De exemplu, putem aproxima π cu 100 de zecimale:

```
> evalf(Pi,100);
```

$$3.14159265358979323846264338327950288419716939937510582097494\backslash$$
$$45923078164062862089986280348253421170 68$$

Numerele reale în Maple pot fi introduce în Maple fie utilizând numere ce conţin punctul zecimal, fie direct utilizând funcţia `Float`, unde `Float`(m,e)=$m\times 10^e$, iar mantisa m este un număr întreg cu precizie oarecare, şi exponentul e este restricţionat funcţie de dimensiunea întregilor ce se pot reprezenta pe un anumit calculator (tipic, 31 de biţi). Funcţia `op` poate fi utilizată pentru a extrage mantisa sau exponentul unui număr în virgulă mobilă. Să generăm o secvenţă de numere aleatoare din $[0,1)$, cu 6 cifre zecimale exacte.

```
> UniformInteger:=rand(0..10^6-1):
> UniformFloat:=proc() Float(UniformInteger(),-6) end:
> seq(UniformFloat(), i=1..6);
```

$$.669081, .693270, .073697, .143563, .718976, .830538$$

Funcţia `rand` returnează un număr generat aleator dintr-un interval specificat.

Calculele în virgulă mobilă se efectuează cu rotunjire. Precizia este controlată de variabila globală `Digits` cu valoarea implicită 10. Funcţia `evalf` este utilizată pentru a evalua o constantă simbolică exactă printr-o aproximare în virgulă mobilă.

```
> sin(1.); sin(1);
```

$$.8414709848$$

$$\sin(1)$$

```
> Digits:=25: sin(1.); sin(1);
```

$$.84147098480789650665250 23$$

$$\sin(1)$$

```
> evalf(");
```

$$.84147098480789650665250 23$$

Pe lângă funcţiile matematice pe care le întâlnim într-un limbaj de programare, Maple cunoaşte şi o serie de funcţii speciale precum funcţiile Bessel, funcţia Dirac, funcţia Γ etc. Acestea sunt calculate cu precizie înaltă prin însumarea unor serii. De exemplu, $\sqrt{a}$ poate fi calculat aplicând metoda Newton la rezolvarea ecuaţiei $x^2 - a = 0$, pornind cu valorea de start $x_0 = a/2$. Şirul iterativ este

$$x_{k+1} = \left(x_k + \frac{a}{x_k} \right) /2$$

```
> SqrtNewton:=proc(a) local xk,xkm1; Digits:=55; xkm1:=0;
>    xk:=evalf(a/2);
>    while abs(xk-xkm1)> abs(xk)*10^(-50) do
>      xkm1:=xk; print(xk); xk:=(xk+a/xk)/2; od;
>    Digits:=50; evalf(xk);
> end:
> SqrtNewton(2);
```

$$1.$$
$$1.5000000000000000000000000000000000000000000000000000$$
$$1.4166666666666666666666666666666666666666666666666667$$
$$1.4142156862745098039215686274509803921568627450980392 16$$
$$1.4142135623746899106262955788901349101165596221157440 45$$
$$1.4142135623730950488016896235025302436149819257761974 29$$
$$1.4142135623730950488016887242096980785696718753772340 02$$
$$1.4142135623730950488016887242096980785696718753769480 73$$
$$1.4142135623730950488016887242096980785696718753769$$

Deoarece în modelul Maple pentru virgulă mobilă se operează mai încet decât cu aritmetica în virgulă mobilă dependentă direct de hardware-ul existent pe un calculator anume, Maple are o funcţie `evalhf` pentru evaluare în aritmetica în virgulă flotantă a hardware-ului (bazată pe rutinele C).

3.2 Simplificare şi calcul polinomial

Maple are posibilitatea de a simplifica expresii şi de a normaliza polinoame. Automat, Maple face următoarele simplificări: (a) sume, produse, puteri ale numerelor întregi şi raţionale; (b) cel mai mare divizor comun al numărătorului şi numitorului într-un număr raţional este eliminat; (c) factori sintactic identici în numărătorul şi numitorul unei expresii raţionale sunt eliminaţi; (d) produs de termeni rearanjaţi astfel încât termenul constant este primul termen în produs (e) termenii asemenea în produse şi sume sunt colectaţi (`x*x`→`x^2`, `x+x`→`2*x`); (f) dacă o expresie este produs de un număr şi o sumă de

monoame, numărul este distribuit în sumă (`1/2(x+y)`→`x/2+y/2`); (g) simplificarea funcţiilor matematice standard pentru anumite argumente (`cos Pi/4`→ $\sqrt{2}/2$, `ln 1`→ 0, `binomial(n,1)`→ n, dar nu şi `ln x^ 2`→ $2 \ln x$); (h) ordonarea alfabetică (`x+y+z, y+x+z, y+z+x` identice; ieşirea este `x+y+z`).

Normalizarea polinomială este o formă simplă de simplificare a expresiilor. Maple are comanda `normal` destinată pentru a rezolva această problemă.

```
> eq1:=expand((41*x^2+x+1)^2*(2*x-1))/expand((3*x+5)*(2*x-1));
```

$$eq1 := \frac{3362\,x^5 - 1517\,x^4 + 84\,x^3 - 79\,x^2 - 1}{6\,x^2 + 7\,x - 5}$$

```
> normal(eq1);
```

$$\frac{1681\,x^4 + 82\,x^3 + 83\,x^2 + 2\,x + 1}{3\,x + 5}$$

Factorul comun $2x - 1$ a fost eliminat din numărător şi numitor.

```
> eq2 := expand((x+y)^3*z+(x+y)^2*z^2) /expand(2*(x+y)^3+
>   5*(x+y)^4*z-(x+y)^3*z^2);
```

$$eq2 := (\,z\,x^3 + 3\,z\,x^2\,y + 3\,z\,x\,y^2 + z\,y^3 + z^2\,x^2 + 2\,z^2\,x\,y + z^2\,y^2\,)\,/(2\,x^3 + 6\,x^2\,y + 6\,x\,y^2 + 2\,y^3 + 5\,z\,x^4 + 20\,z\,x^3\,y + 30\,z\,x^2\,y^2 + 20\,z\,x\,y^3 + 5\,z\,y^4 - z^2\,x^3 - 3\,z^2\,x^2\,y - 3\,z^2\,x\,y^2 - z^2\,y^3)$$

```
> normal(eq2);
```

$$\frac{(\,x + z + y\,)\,z}{5\,z\,x^2 + 10\,z\,x\,y + 2\,x - x\,z^2 + 2\,y + 5\,z\,y^2 - z^2\,y}$$

O funcţie mai lentă este `simplify`. Aceasta pe lângă simplificarea polinoamelor are posibilitatea de simplificare a expresiilor trigonometrice sau a expresiilor cu radicali.

```
> eq3:=cos(x)^5 + sin(x)^4 + 2*cos(x)^2 - 2*sin(x)^2 - cos(2*x):
> simplify(eq3);
```

$$\cos(\,x\,)^5 + \cos(\,x\,)^4$$

Pentru anumite simplificări este necesar să facem o serie de presupuneri despre variabilele invocate.

```
> eq4:=sqrt(x^2);
```

$$eq4 := \sqrt{x^2}$$

```
> simplify(eq4);
```

$$\mathrm{csgn}(\,x\,)\,x$$

```
> assume(x<0);
> simplify(eq4);
```

$$-x\_$$

Caracterul ~ după x indică faptul că s-a făcut o presupunere asupra sa.

Putem indica simplificarea numai de un anumit tip.

```
> eq5 := (y^5+273*y^4+27502*y^3+1219766*y^2+20416065*y+
>         13760393)^(1/3):
> simplify(eq5,radical,assume=positive);
```

$$(\,y + 59\,)\,(\,y^2 + 96\,y + 67\,)^{1/3}$$

O altă funcţie este `combine`. Aceasta reduce numărul termenilor dintr-o expre-

sie.

```
> assume(a>0,b>0);
> eq5:=log(a)+log(b);
```

$$eq5 := \ln(a\_) + \ln(b\_)$$

```
> simplify(eq5);
```

$$\ln(a\_) + \ln(b\_)$$

```
> combine(eq5,ln);
```

$$\ln(a\_\, b\_)$$

Cel mai mare divizor comun poate fi calculat şi pentru polinoame. Se poate aplica testul de ireductibilitate a polinoamelor:

```
> irreduc(x^3+1);
```

false

```
> p6 := x^8 + x^6 - 3*x^4 - 3*x^3 + 8*x^2 + 2*x - 5:
> p7 := 3*x^6 + 5*x^4 - 4*x^2 - 9*x + 21:
> gcd(p6,p7);
```

1

Maple permite aceleaşi operaţii şi asupra multinomialelor:

```
> p8 := x^4*y^4+2*x^3*y^3*z+4*x^3*y^3+2*x^2*y^2*z^2+
> 4*x^2*y^2*z+5*x^2*y^2+2*x*y*z^3+4*x*y*z^2+2*x*y*z+
> 4*x*y+z^4+4*z^3+5*z^2+4*z+4:
> p9 := x^4*y^2-x^2*y^4+x^2*y^2*z^2-x^2*y^2+x^2*z^2+x^2-y^2*z^2
> -y^2+z^4-1:
> gcd(p8,p9);
```

$$1 + z^2 + x^2\, y^2$$

Maple poate factoriza polinoame peste mulţimea numerelor raţionale sau în Z_p.

```
> p1 := 72*x^3+60*x^2-192*x-180:
> factor(p1);
```

$$12\,(3\,x-5)\,(2\,x+3)\,(x+1)$$

```
> p2 := x^12-1:
> factor(p2,I);
```

$$(1-x+x^2)\,(x^2+x+1)\,(x^2+I\,x-1)\,(x^2-I\,x-1)\,(x+I)\,(x-I)\,(x-1)\,(x+1)$$

```
> factor(p2,sqrt(3));
```

$$(1-x+x^2)\,(x^2+x+1)\,(x^2+1)\,\left(x^2+\sqrt{3}\,x+1\right)\,\left(x^2-\sqrt{3}\,x+1\right)\,(x-1)\,(x+1)$$

```
> p4:=x^6+x^5+x^4+x^3+1:
> Factor(p4) mod 2;
```

$$(x^4+x+1)\,(x^2+x+1)$$

```
> factor(p4);
```

$$x^6 + x^5 + x^4 + x^3 + 1$$

Putem reprezenta serii ca fracţii continue:

```
> p3 := taylor(tan(x),x,10);
```

$$p10 := x + \frac{1}{3}\,x^3 + \frac{2}{15}\,x^5 + \frac{17}{315}\,x^7 + \frac{62}{2835}\,x^9 + \mathrm{O}(\,x^{10}\,)$$

```
> convert(p3,confrac);
```

$$\cfrac{x}{1+\cfrac{x^2}{-3+\cfrac{x^2}{5+\cfrac{x^2}{-7+\frac{1}{9}\,x^2}}}}$$

Considerăm exemplul de calcul a normei $\sum_{i=0}^{n} a_i^2$ a polinomului $a(x) = \sum_{i=0}^{n} a_i x^i$:

```
> Euclidian:=proc(a)
>    sqrt(convert(map(x->x^2,[coeffs(expand(a))]),`+`))
> end:
```

Prima operaţie este aceea de expandare a polinomului. Funcţia `coeffs` returnează secvenţa coeficienţilor, care ulterior este pusă într-o listă. Fiecare element al listei este ridicat la pătrat, rezultând o nouă listă; apoi lista de pătrate este convertită la o sumă. Funcţiile `coeff` şi `coeffs` necesită expandarea în prealabil a polinomului argument:

```
> p:=x^3-(x-3)*(x^2+x)+1;
```

$$p := x^3 - (\,x-3\,)\,(\,x^2+x\,) + 1$$

```
> coeffs(p);
Error, invalid arguments to coeffs
```

```
> expand(p);
```

$$2\,x^2 + 3\,x + 1$$

```
> coeffs(expand(p));
```

$$1, 3, 2$$

```
> Euclidian(p);
```

$$\sqrt{14}$$

Dacă se doreşte determinarea coeficienţilor unui polinom în anumită variabilă, este necesară în prealabil expandarea polinomului în acea variabilă. Funcţia `expand` expandează polinomul în toate variabilele. Funcţia `collect` expandează însă polinomul numai după o variabilă specificată.

Procedura `Euclidian` lucrează cu polinoame multivariabile în sensul că calculează rădăcina pătrată a sumei pătratelor coficienţilor numerici. Adică pentru $p = ux^2 + y^2 + v$ `Euclidian` returnează $\sqrt{3}$. Dacă dorim să vedem acest polinom ca polinom de x şi y cu coeficienţi simbolici în u şi v, trebuie să-i specificăm

procedurii variabilele polinomiale. Putem face acest lucru printr-un parametru adiţional care poate fi o singură variabilă, o mulţime de variabile sau o listă de variabile (de tip {`name, set(name), list(name)`}). Funcţia `coeffs` acceptă un asemenea argument secund opţional. În plus dorim ca procedura să verifice că intrarea este un polinom.

```
> Euclidian:=proc(a,v)
>   if nargs=1 then
>       if not type(a, polynom) then ERROR(‘Arg.1: polinom?!‘,a) fi;
>      sqrt(convert(map(x->x^2,[coeffs(expand(a))]),‘+‘))
>   elif type(v,{name,set(name),list(name)}) then
>      if not type(a,polynom(anything,v)) then
>         ERROR(‘Arg.1: polinom în‘,v,‘?!‘) fi;
>      sqrt(convert(map(x->x^2,[coeffs(expand(a),v)]),‘+‘))
>   else ERROR(‘Al doilea argument este invalid (variabila)‘)
>   fi
> end:
```

Testăm dacă tipul unei expresii este `polynom` prin `polynom`(R, X). Obţinem `true` dacă expresia este un polinom a cărui coeficienţi sunt de tip R în variabilele X. Un rezultat `true` la testul de `polynom(rational,x)` indică un polinom univariat în x cu coeficienţi raţional, adică un polinom din $Q[x]$. Dacă R şi X nu sunt specificaţi, înseamnă că expresia trebuie să fie un polinom în toate variabilele sale.

Funcţiile de bază pentru calcul polinomial sunt `degree, coeff, expand, divide` şi `collect`. Pe lângă acestea există şi altele, precum cel mai mare divizor comun sau funcţia utilizată în procedura următoare care determină polinomul ireductibil a de forma $x^n + x^m + 1$ şi inelul $Z_2[x]$:

```
> trinomial:=proc(n) local i,t;
>    for i to iquo(n+1,2) do t:=x^n+x^i+1;
>      if Primitive(t) mod 2 then REURN(t) fi; od;
>    FAIL
> end:
```

Exerciţii:

```
> p:=x^3+2*x^2-5*x+6; q:=x^2*y-y^2*x+5/3*x*y-2/3*x+y-1;
> type(p,polynom(integer)); type(p,polynom(integer,y));
> type(q,polynom(rational)); type(q,polynom(integer));
> type(p,polynom(anything,[x,y]));
> lcoeff(p);
> degree(p);
> op(q);
> p1:=3*x^3-3*x^2-15*x-9: p2:=x^3-7*x-6:
> p3:=x^2+3*x+2:
> expand(p1*p2);
> divide(p2,p3,quotient);
> quotient;
> gcd(p1,p2);
```

```
> factor(x^3-2*x^2-5*x+6);
> expand('');
> simplify(exp(ln(b*sin(c))));
> convert(Pi/12,degrees);
> convert([[1,2,3],[4,5,6],[7,8,9]],matrix);
> subs(x^2=y^4,x^2*z-y^3+x^2-5);
> convert(7*x^5+x^3*y^4,mod2);
> ''expand((x-2)^3*(x-1))'';
> functie:='';
> '';
> x:=3;
> functie;
```

Presupunem că derivatele până la ordinul n ale lui $y = f(x)$ sunt definite în punctul $x = a$. Polinomul Taylor de ordin n la $x = a$ este:

```
> p[n] :=  x -> sum((D@@i)(f)(a)/i!*(x-a)^i,i=0..n);
```

$$p_n := x \to \sum_{i=0}^{n} \frac{D^{(i)}(f)(a)(x-a)^i}{i!}$$

Dacă a $(n+1)$ derivată a lui f este definită pe intervalul $[a, x]$, atunci restul este:

```
> R[n](x) = abs(f(x) - p[n]) , ' < ' , M/(n+1)!* (x-a)^(n+1);
```

$$R_n(x) = |f(x) - p_n|, < , \frac{M(x-a)^{(n+1)}}{(n+1)!}$$

unde M este margină superioară a derivatei de ordin $(n+1)$ pe $[a, b]$.

Determinăm de exemplu polinomul Taylor de ordin doi a lui $f(x) = \ln x \arctan e^x$ la $x = 1$. Determinăm, de asemenea, eroarea maximă.

```
> f := x -> ln(x)*arctan(exp(x)):
> p[2]  := x -> sum((D@@i)(f)(1.)/i!*(x-1.)^i,i=0..2);
```

$$p_2 := x \to \sum_{i=0}^{2} \frac{D^{(i)}(f)(1.)(x-1.)^i}{i!}$$

```
> p[2](x);
```

$$1.218282905\,x - 1.218282905 - .2851143156\,(x-1.)^2$$

Putem estima eroarea trasând a a treia derivată:

```
> #est := plot((D@@3)(f),1..1.5):  est;
> M := (D@@3)(f)(1.);
```

$$M := .724152878$$

Astfel restul $R_2 = |f(x) - p_2(x)|$ este mărginit de

```
> M/3!*(1.5-1)^3;
```

$$.01508651830$$

Eroarea maximă este întâlnită la $x = 1.5$:

```
> (f-p[2])(1.5);
```

$$.0100268555$$

Exerciţiu: Găsiţi cel mai mic n pentru care polinomul Taylor de ordin n al funcţiei e^x la $x = 0$ aproximează e^x cu eroarea 10^{-12} în [0,1].

3.3 Limite şi diferenţiale

Maple calculează limitele unei expresii examinând expansiunile în serii de puteri ale expresiei. Se pot calcula limite în puncte singulare sau limite laterale în puncte de discontinuitate.

```
> eq1 := sin(x)/x: limit(eq1,x=0);
```

$$1$$

```
> eq2 := (2*x+3)/(7*x+5): limit(eq2,x=infinity);
```

$$\frac{2}{7}$$

```
> eq3 := (x^2-2*x+1)/(x^4+3*x^3-7*x^2+x+2):
> subs(x=1,eq3);
Error, division by zero
> limit(eq3,x=1);
```

$$\frac{1}{8}$$

În cazul următor regula l'Hospital nu poate fi aplicată: limita fiecărui termen al diferenţei este infinit şi dacă termenii sunt combinaţi într-o singură fracţie, numitorul are derivatele de orice ordin 0 la 0.

```
> Limit((2/(a^2-b^2)^3/r^6-1/(a^2-b^2)^2/r^4)/exp(b^2*r^2)-
>    (2/(a^2-b^2)^3/r^6+1/(a^2-b^2)^2/r^4)/exp(a^2*r^2),r=0);
```

$$\lim_{r \to 0} \frac{2\,\dfrac{1}{(a^2-b^2)^3 r^6} - \dfrac{1}{(a^2-b^2)^2 r^4}}{\mathrm{e}^{(b^2 r^2)}} - \frac{2\,\dfrac{1}{(a^2-b^2)^3 r^6} + \dfrac{1}{(a^2-b^2)^2 r^4}}{\mathrm{e}^{(a^2 r^2)}}$$

```
> value(");
```

$$\frac{-1}{6}$$

Limitele sunt definite implicit pe spaţiul real. Dacă limita se doreşte în spaţiul complex, trebuie specificată opţiunea `complex`.

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

$$\mathit{undefined}$$

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

$$\infty$$

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

$$\infty$$

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

$$-\infty$$

Exerciţii:

```
> L := Limit( tan(theta), theta=Pi/2 );
> value(L);
> limit( tan(theta), theta=Pi/2, right );
> y := (x-2)/(x^2-4);
> limit(y,x=2);
> limit(abs(x)/x,x=0,left);
> y := (x^2-4)/((x-2)*(x+3));
> simplify(y);
> limit(y,x=infinity);
```

Comanda Maple `limit` poate fi utilizată pentru calculul derivatelor utilizând definiţia:

```
>  f := x -> x^2 + x -3;
```

$$f := x \to x^2 + x - 3$$

```
> fp := limit((f(x+h)-f(x))/h,h=0);
```

$$fp := 2\,x + 1$$

```
> fp := diff(f(x),x);
```

$$fp := 2\,x + 1$$

Pentru definirea ca funcţie a derivatei unei expresii se poate utiliza `unapply`:

```
> y := x^2 + x;
> yp := diff(y,x);
```

$$yp := 2\,x + 1$$

```
> yp:=unapply(yp,x);
```

$$yp := x \to 2\,x + 1$$

```
> yp(3);
```


Pentru o funcţie dată, putem defini funcţia derivată cu ajutorul operatorului `D`:

```
> f:=x->tan(x*a):
> fp:=D(f);
```

$$fp := x \to \left(1 + \tan(\,x\,a\,)^2\right)\,a$$

```
> fp(4);
```

$$\left(1 + \tan(\,4\,a\,)^2\right)\,a$$

`D` primeşte ca intrare o funcţie şi prezintă ca ieşire diferenţiala funcţiei.

3.4 Serii

Maple poate calcula serii Taylor, serii Laurent şi serii Chebyshev.

```
> eq5:=x*sin(x):
```

Dacă funcţia este analitică, se calculează seria Taylor.

```
> series(eq5,x=0);
```

$$x^2 - \frac{1}{6}\,x^4 + \mathrm{O}(\,x^6\,)$$

Ordinul de acurateţe poate fi specificat.

```
> series(eq5,x=0,10);
```

$$x^2 - \frac{1}{6}\,x^4 + \frac{1}{120}\,x^6 - \frac{1}{5040}\,x^8 + \mathrm{O}(x^{10})$$

Dacă funcţia are un pol în punctul de expansiune, se determină seria Laurent.

```
> eq6:=1/x^5*sin(x):
> series(eq6,x=0);
```

$$x^{-4} - \frac{1}{6}\,x^{-2} + \frac{1}{120} + \mathrm{O}(x^2)$$

```
> eq7 := (x^2 + 2*x - 3) / (x^4 - 11*x^3 + 42*x^2 - 68*x + 40);
```

$$eq7 := \frac{x^2 + 2\,x - 3}{x^4 - 11\,x^3 + 42\,x^2 - 68\,x + 40}$$

```
> series(eq7,x=2);
```

$$-\frac{5}{3}\,(x-2)^{-3} - \frac{23}{9}\,(x-2)^{-2} - \frac{32}{27}\,(x-2)^{-1} - \frac{32}{81} - \frac{32}{243}\,(x-2) - \frac{32}{729}\,(x-2)^2 + \mathrm{O}\left((x-2)^3\right)$$

3.5 Sume

Puterea Maple-ului constă, de fapt, în abilitatea de calcul simbolic. De exemplu, să calculăm suma $\sum_{i=1}^{n} i^20$.

```
> sum(i^20,i=1..n);
```

$$\frac{1}{21}\,(n+1)^{21} - \frac{1}{2}\,(n+1)^{20} + \frac{5}{3}\,(n+1)^{19} - \frac{19}{2}\,(n+1)^{17} + \frac{1292}{21}\,(n+1)^{15} - 323\,(n+1)^{13} + \frac{41990}{33}\,(n+1)^{11} - \frac{223193}{63}\,(n+1)^9 + 6460\,(n+1)^7 - \frac{68723}{10}\,(n+1)^5 + \frac{219335}{63}\,(n+1)^3 - \frac{174611}{330}\,n - \frac{174611}{330}$$

```
> simplify(");
```

$$\frac{219335}{63}\,n^3 - \frac{68723}{10}\,n^5 + 6460\,n^7 - \frac{223193}{63}\,n^9 + \frac{41990}{33}\,n^{11} - 323\,n^{13} + \frac{1292}{21}\,n^{15} - \frac{19}{2}\,n^{17} + \frac{5}{3}\,n^{19} + \frac{1}{2}\,n^{20} + \frac{1}{21}\,n^{21} - \frac{174611}{330}\,n$$

```
> factor(");
```

$$\frac{1}{6930}\,n\,(2\,n+1)\,(n+1)(165\,n^{18} + 1485\,n^{17} + 3465\,n^{16} - 5940\,n^{15} - 25740\,n^{14} + 41580\,n^{13} + 163680\,n^{12} - 266310\,n^{11} - 801570\,n^{10} + 1335510\,n^9 + 2806470\,n^8 - 4877460\,n^7 - 6362660\,n^6 + 11982720\,n^5 + 7591150\,n^4 - 17378085\,n^3 - 1540967\,n^2 + 11000493\,n - 3666831)$$

Maple lucrează şi cu noţiunea de infinit. De exemplu, putem calcula $\sum_{i=1}^{\infty} \frac{1}{i^3}$

```
> sum(1/i^3,i=1..infinity);
```

$$\zeta(3)$$

Maple reduce răspunsul în termenii funcţiei Riemann Zeta cunoscută în analiza matematică. Biblioteca Maple conţine peste 2000 de subrutine care încapsulează o parte mare din cunoştiinţele matematice actuale. De asemenea, observăm că Maple permite scrierea cu caractere greceşti.

Pentru a ne forma o idee despre valoarea răspunsului anterior (răspunsul exact, fără erori de rotunjire), cerem o aproximare în virgulă flotantă:

```
> evalf(");
```

$$1.202056903$$

Maple acceptă funcţia sumă în forma nedefinită:

```
> Sum((-15*n^4+40*n^3-74*n^2-51*n+(-10)) / (25*n^6-5*n^5+29*n^4-
>     81*n^3+35*n^2-133*n+55),n);
```

$$\sum_n \frac{-15\,n^4+40\,n^3-74\,n^2-51\,n-10}{25\,n^6-5\,n^5+29\,n^4-81\,n^3+35\,n^2-133\,n+55}$$

```
> value(");
```

$$\frac{3\,n^2-7\,n+9}{5\,n^3-8\,n^2+9\,n-11}$$

Exerciţii:

```
> eq8:=(-3*n^2+28*n+57)*13^n / ((n+1)!*(3*n+8)*(3*n+5));
> sum(eq8,n);
> Sum(i^12, i=0..n-1);
> value(");
> sum(i^2*binomial(n,i)*x^i, i=0..n);
> Sum(i^4*7^i, i=0..n-1);
> value(");
> sum(1/i, i=1..100);
> Sum(i, i=1..100)=sum(i, i=1..100);
> sum(4+i*3,i=0..99);
> evalf(sum(sqrt(i),i=1..100)) ;
> sum((1/2)^i,i=0..n);
> limit(sum((1/2)^i,i=0..n),n=infinity);
```

3.6 Integrare

Maple poate rezolva o serie de probleme de calcul integral şi diferenţial. Poate determina primitivele unei funcţii:

```
> int(f(x),x);
```

$$\int \mathrm{f}(x)\,dx$$

```
> f := 1 / (exp(x) + 1): int(f,x);
```

$$-\ln(\mathrm{e}^x+1)+\ln(\mathrm{e}^x)$$

```
> f1:=x^7-4*x^3+3: int(f1,x);
```

$$\frac{1}{8}\,x^8 - x^4 + 3\,x$$

```
> f2:= (x^3-x) / (x^5 + x^4 + x^3 - x^2 - x - 1): int(f2,x);
```

$$-\frac{1}{3}\,\frac{2\,x+1}{x^2+x+1} + \frac{2}{9}\,\sqrt{3}\arctan\left(\frac{1}{3}\,(\,2\,x+1\,)\,\sqrt{3}\right)$$

De asemenea poate trata integrale definite:

```
> f4 := 1/(x+3)^3 + 1/x: r4:=int(f4,x=1..2);
```

$$r4 := \frac{9}{800} + \ln(\,2\,)$$

```
> evalf(r4);
```

$$.7043971806$$

Există însă funcţii a căror integrale nu pot fi reprezentate în termenii unor funcţii elementare. De exemplu, funcţia Gauss de eroare:

```
> f6 := exp(-x^2): int(f6,x);
```

$$\frac{1}{2}\,\sqrt{\pi}\,\mathrm{erf}(\,x\,)$$

```
> f7 := sqrt(x)/exp(x): int(f7,x);
```

$$-\frac{\sqrt{x}}{\mathrm{e}^x} + \frac{1}{2}\,\sqrt{\pi}\,\mathrm{erf}\left(\sqrt{x}\right)$$

Maple utilizează anumite funcţii predefinite în termeni de integrale.

Se pot trata şi integralele improprii:

```
> Int( exp(-t^2), t=-infinity..infinity );
```

$$\int_{-\infty}^{\infty} \mathrm{e}^{(\,-t^2\,)}\,dt$$

```
> value(");
```

$$\sqrt{\pi}$$

```
> Int(exp(-t)/t^(1/3), t=0..infinity);
```

$$\int_0^{\infty} \frac{\mathrm{e}^{(\,-t\,)}}{t^{1/3}}\,dt$$

```
> value(");
```

$$\Gamma\left(\frac{2}{3}\right)$$

unde $\Gamma(n) = \int_0^{\infty} e^x x^{(n-1)} dx$.

```
> Int(x/(x^3-x^2+1),x);
```

$$\int \frac{x}{x^3 - x^2 + 1}\,dx$$

```
> value(");
```

$$\sum_{\_R=\%1} \_R \ln\left(x + \frac{23}{3}\,\_R^2 + \frac{1}{3}\right)$$

$$\%1 := \mathrm{RootOf}(\,23\,\_Z^3 + 3\,\_Z + 1\,)$$

În anumite cazuri, integralele depind parametrii. De exemplu:

```
> f5:= exp(-u*x)*ln(x)*sqrt(x): int(f5,x=0..infinity);
```

$$\int_0^{\infty} \mathrm{e}^{(-u\,x)} \ln(x)\,\sqrt{x}\,dx$$

Putem informa Maple despre o anumită calitate a lui u:

```
> assume(u<0); int(f5,x=0..infinity);
```

$$\infty$$

```
> assume(u>0);int(f5,x=0..infinity);
```

$$-\frac{1}{2}\,\frac{\sqrt{\pi}\ln(u\_)}{u\_^{3/2}} + \frac{\sqrt{\pi}}{u\_^{3/2}} - \frac{1}{2}\,\frac{\sqrt{\pi}\,\gamma}{u\_^{3/2}} - \frac{\sqrt{\pi}\ln(2)}{u\_^{3/2}}$$

Există însă integrale care nu pot fi rezolvate prin Maple. În asemenea cazuri există o serie de alternative pentru a obţine o aproximare numerică: de exemplu, prin generarea unor serii.

```
> f8 := exp(t^3): int(f8,t=0..x);
```

$$\int_0^{x} \mathrm{e}^{(t^3)}\,dt$$

```
> r8:=series(int(f8,t=0..x),x,10);
```

$$r8 := x + \frac{1}{4}\,x^4 + \frac{1}{14}\,x^7 + \mathrm{O}(x^{10})$$

Răspunsul este dat într-o formă specifică structurii de date tip serie; aceasta poate fi convertită la o funcţie polinomială:

```
> p8:=unapply(convert(r8,polynom),x);
```

$$p8 := x \to x + \frac{1}{4}\,x^4 + \frac{1}{14}\,x^7$$

```
> evalf(p8(1));
```

$$1.321428571$$

O cale mai scurtă este:

```
> evalf(int(f8,t=0..1));
```

$$1.341904418$$

Un avantaj al utilizării funcţiei `evalf` este acela că răspunsul este corect până la ultima zecimală. Dacă utilizăm soluţia tip serie trebuie să facem o analiză a ordinului de acurateţe a răspunsului.

```
> r1 := int( exp(-t) / sqrt(1-t^2),t = -1..1 );
```

$$r1 := \int_{-1}^{1} \frac{\mathrm{e}^{(-t)}}{\sqrt{1-t^2}}\,dt$$

```
> evalf(r1);
```

$$3.977463261$$

Exerciţii:

```
> Int( u/sin(u), u=0..Pi/2 ):
> " = value( " );
> evalc( " );
> int( exp(-u^2), u=0..infinity );
> int( exp(-a*u^2), u=0..infinity );
> assume( a>0 );
```

```
> about( a );
> int( exp(-a*u^2), u=0..infinity );
> a := 'a';
> about( a );
> restart; F := (x^2-26*x-47)/(x^5+5*x^3+3*x^4+11*x^2-20):
> Int( F, x );
> int( F, x );
> factor( denom( F ) );
> FORM := a/(x-1) + b/(x+2) + c/(x+2)^2 + (d+e*x)/(x^2+5);
> eqn := F = FORM;
> simplify( " );
> collect( ", x );
> eq1 := a+b+e=0; eq2 := 4*a+b+c+d+3*e;
> eq3 := 9*a+3*b-c+3*d = 1; eq4 := -4*e+5*b+5*c+20*a=-26;
> eq5 := -5*c-10*b-4*d+20*a = -47;
> COEFFS := solve( {eq1,eq2,eq3,eq4,eq5}, {a,b,c,d,e} );
> subs( COEFFS, FORM );
> simplify( " );
> int( ", x );
```

3.7 Utilizarea pachetelor speciale de funcţii

Maple conţine o varietate de rutine care pot asista utilizatorul în procesele calculatorii. Multe dintre acestea sunt grupate în pachete specifice (vezi `linalg, plots, student, geom` etc).

De exemplu, în pachetul `student` există o colecţie de comenzi ajutătoare pentru studiul integrării şi a diferenţierii.

```
> with(student);
```

[*D*, *Doubleint*, *Int*, *Limit*, *Lineint*, *Sum*, *Tripleint*, *changevar*, *combine*, *completesquare*, *distance*, *equate*, *extrema*, *integrand*, *intercept*, *intparts*, *isolate*, *leftbox*, *leftsum*, *makeproc*, *maximize*, *middlebox*, *middlesum*, *midpoint*, *minimize*, *powsubs*, *rightbox*, *rightsum*, *showtangent*, *simpson*, *slope*, *trapezoid*, *value*]

Să considerăm, de exemplu, funcţia f definită prin:

```
> f := (x) -> x^2 + 3:
```

Panta liniei tangente într-un punct oarecare $x = a$ poate fi calculată astfel:

```
> slope( [x,f(x)], [a,f(a)] );
```

$$\frac{x^2 - a^2}{x - a}$$

```
> Limit(",x=a);
```

$$\lim_{x \to a} \frac{x^2 - a^2}{x - a}$$

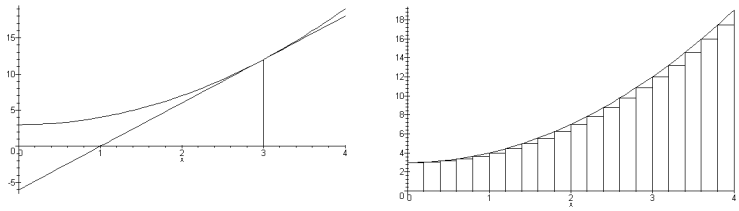


Figura 3.1:

```
> normal(");
```

$$\lim_{x \to a} a + x$$

```
> value(");
```

$$2\,a$$

De exemplu, linia tangentă în $x = 3$ are panta:

```
> subs(a=3,");
```

$$6$$

Acestă linie tangentă poate fi vizualizată cu comanda

```
> showtangent( f(x),x=3,x=0..4 );
```

Grafic: vezi figura 3.1.a

Pentru o secvenţă de valori x, se poate construi o secvenţă de imagini ale tangentelor:

```
> #for i to 8 do p.i := showtangent(f(x),x=2+i*.2,x=0..4) od:
> #plots[display]( [p.(1..8)], insequence=true );
```

Se poate aproxima, de asemenea, aria subîntinsă de o curbă. Fie aria de interes cuprinsă între curba $y = f(x)$, din exemplul anterior, între $x = 0$ şi $x = 4$. Această arie poate fi aproximată prin dreptunghiuri:

```
> leftbox( f(x),x =0..4,20 );
```

Grafic: vezi figura 3.1.b

```
> leftsum( f(x),x=0..4,20);
```

$$\frac{1}{5}\left(\sum_{i=0}^{19}\left(\frac{1}{25}\,i^2+3\right)\right)$$

```
> value(");
```

$$\frac{794}{25}$$

```
> evalf(");
```

$$31.76000000$$

Suma poate fi calculată cu un număr nedefinit n de dreptunghiuri:

```
> leftsum( f(x) , x = 0.. 4 , n );
```

$$4\,\frac{\sum_{i=0}^{n-1}\left(16\,\frac{i^2}{n^2}+3\right)}{n}$$

```
> expand(");
```

$$64\,\frac{\sum_{i=0}^{n-1} i^2}{n^3} + 12\,\frac{\sum_{i=0}^{n-1} 1}{n}$$

```
> value(");
```

$$64\,\frac{\frac{1}{3}\,n^3 - \frac{1}{2}\,n^2 + \frac{1}{6}\,n}{n^3} + 12$$

```
> expand(");
```

$$\frac{100}{3} - 32\,\frac{1}{n} + \frac{32}{3}\,\frac{1}{n^2}$$

Pentru a obţine aria exactă, trecem la limită:

```
> Limit(",n=infinity);
```

$$\lim_{n\to\infty} \frac{100}{3} - 32\,\frac{1}{n} + \frac{32}{3}\,\frac{1}{n^2}$$

```
> value(");
```

$$\frac{100}{3}$$

Acest proces stă la baza definiţiei funcţiei de integrare întâlnită drept comandă Maple:

```
> Int(f(x),x=0..4);
```

$$\int_0^4 x^2 + 3\,dx$$

```
> value(");
```

$$\frac{100}{3}$$

Alte exemple:

```
>  middlesum(x^2,x=0..1,10) = value(middlesum(x^2,x=0..1,10));
```

$$\frac{1}{10}\left(\sum_{i=0}^{9}\left(\frac{1}{10}\,i + \frac{1}{20}\right)^2\right) = \frac{133}{400}$$

```
>  evalf(middlesum(x^2,x=0..1,10));
```

.3325000000

```
>  middlebox(x^2,x=0..1,10);
```

Grafic: vezi figura 3.2

3.8 Grafică

3.8.1 Grafică 2D

Suplimentarea procedurilor de trasare se realizează prin:

```
> with(plots);
```

[*animate*, *animate3d*, *conformal*, *contourplot*, *cylinderplot*, *densityplot*, *display*, *display3d*, *fieldplot*, *fieldplot3d*, *gradplot*, *gradplot3d*, *implicitplot*,

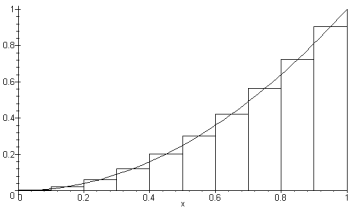


Figura 3.2:

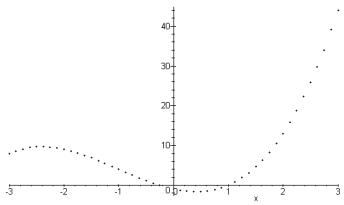


Figura 3.3:

implicitplot3d, *loglogplot*, *logplot*, *matrixplot*, *odeplot*, *pointplot*, *polarplot*, *polygonplot*, *polygonplot3d*, *polyhedraplot*, *replot*, *setoptions*, *setoptions3d*, *spacecurve*, *sparsematrixplot*, *sphereplot*, *surfdata*, *textplot*, *textplot3d*, *tubeplot*]

Procedura Maple de trasare pentru cazul unei informaţii din spaţiul bidimensional este `plot`.

```
> A:=plot(x^3+3*x^2-3*x-1,x=-3..3,style=POINT):
> A;
```

Grafic: vezi figura 3.3

Trasarea unor funcţii discontinue presupune anumite precauţii. Fie exemplul funcţiei tangente $y = \tan(x)$ care are asimptote verticale la multiplii impari de $\pi/2$.

```
> f :=  tan(x);
```

$$f := \tan(x)$$

```
> plot( f , x = -Pi .. Pi );
```

Grafic: vezi figura 3.4.a

Acest rezultat se datorează faptului că funcţia este discontinuă. În plus, poate atinge valori infinite pe intervalul $(-\pi, \pi)$. În asemenea situaţii putem limita scara verticală, de exemplu de la -10 la 10:

```
> plot( f,  x = -Pi .. Pi ,  -10 .. 10 , discont=true );
```

Grafic: vezi figura 3.4.b

Exerciţiu: Trasaţi funcţia $g(t) = (10 - 4t^3)/(1 - t^2)$ pe intervalul $[-3, 5]$.

Două funcţii pot fi trasate pe acelaşi grafic:

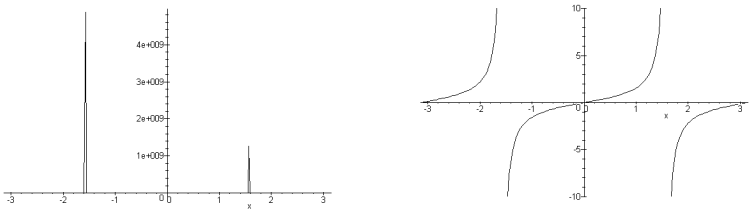


Figura 3.4:

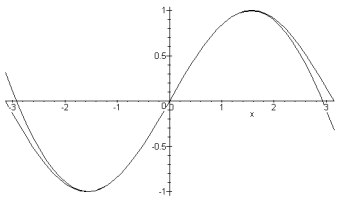


Figura 3.5:

```
> s := taylor(sin(x), x=0);
> approx := convert(s, ratpoly);
> plot({approx, sin(x)}, x = -Pi..Pi);
```

$$s := x - \frac{1}{6}\,x^3 + \frac{1}{120}\,x^5 + \mathrm{O}(\,x^6\,)$$

$$approx := \frac{-\frac{7}{60}\,x^3 + x}{1 + \frac{1}{20}\,x^2}$$

Grafic: vezi figura 3.5

Exerciţii:

```
> f:=sin(x);
> g1:=convert(series(f,x=0),polynom);
> g2:=convert(series(f,x=0,12),polynom);
> plot({f,g1,g2},x=-5..5,y=-2..2);
```

O curbă parametrică bidimensională poate fi trasată tot cu procedura `plot`:

```
> plot( [cos(3*t),sin(2*t),t=0..Pi], x=-1..1, y=-1..1 );
```

Grafic: vezi figura 3.6

Trasarea unei curbe bidimensionale în coordonate polare se poate realiza cu `polarplot`:

```
> pic1:=plot(x/2,x=-1..4):
> pic2:=plots[polarplot](1+2*cos(t),t=0..2*Pi): #cardiod
> pic3:=plots[polarplot](cos(t),t=0..Pi): #cerc
```

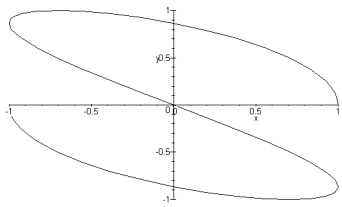


Figura 3.6:

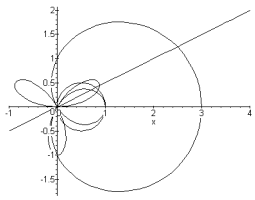


Figura 3.7:

```
> pic4:=plots[polarplot](sin(3*t),t=0..2*Pi): #trandafir
> plots[display]({pic1,pic2,pic3,pic4});
```

Grafic: vezi figura 3.7

Alte exemple:

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

Grafic: vezi figura 3.8.a

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

Grafic: vezi figura 3.8.b

```
> plot([3,3,6,0,3,-3,0,0,3,3],x=-2..10);
```

Grafic: vezi figura 3.8.c

```
> plot({x^2,exp(x),x},x=0..3);
```

Grafic: vezi figura 3.9.a

```
> plot({seq(cos(x*i),i=1..4)},x=-Pi..Pi);
```

Grafic: vezi figura 3.9.b

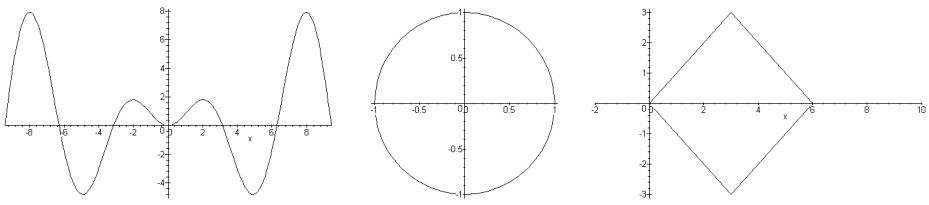


Figura 3.8:

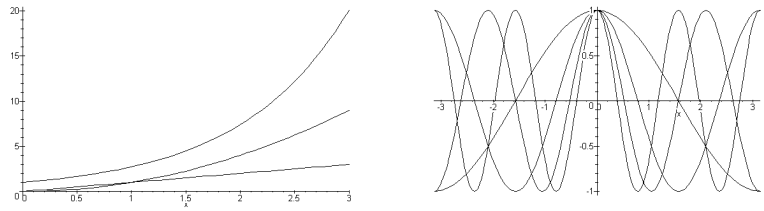


Figura 3.9:

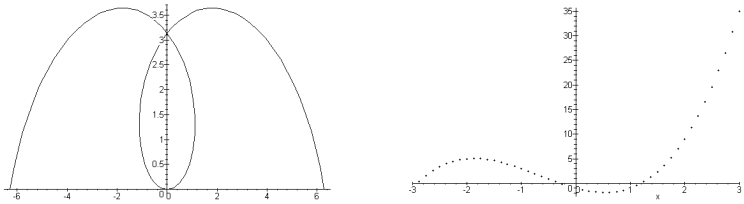


Figura 3.10:

```
> plots[polarplot](2*t);
```

Grafic: vezi figura 3.10.a

```
> plot(x^3+2*x^2-3*x-1,x=-3..3,style=POINT);
```

Grafic: vezi figura 3.10.b

Aplicaţia 2: Datele privind creşterea populatiei unei ţări de-a lungul anilor sunt date prin vectorul de mai jos. Să se traseze graficul evoluţiei populaţiei într-o scară logaritmică.

```
> pop:=[508,711,912,1131,1811,2015,2249,2509,3008,3610,3967]:
> yr:=[1650,1750,1800,1850,1900,1920,1930,1940,1950,1960,1970,1975]:
> map(ln,pop);
```

$[\ln(508), \ln(711), \ln(912), \ln(1131), \ln(1811), \ln(2015), \ln(2249), \ln(2509), \ln(3008), \ln(3610), \ln(3967)]$

```
> lnpop:= evalf ( " ):
> data:= zip( (a,b) -> op( [a,b] ) , yr, pop ):
> logdata:=  zip( (a,b) -> op( [a,b] ) , yr, lnpop ):
> plots[logplot]( data, style = LINE, thickness = 2);
```

Grafic: vezi figura 3.11

Aplicaţia 3: Ilustrarea grafică a efectului de transformare liniară în plan.

```
> with(linalg):
> transform := proc( A: matrix, V: list ) local i;
>   plot({V,[seq(convert(multiply(A,op(i,V)),list),i=1..
>     nops(V))]},scaling = CONSTRAINED) ;
> end:
```

Descrierea transformării se realizează cu ajutorul unei matrice.

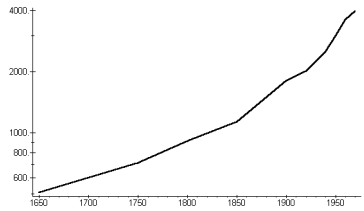


Figura 3.11:

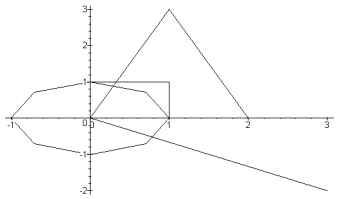


Figura 3.12:

```
> symm := matrix( [ [1,2], [2,1.5] ] ):
```

Obiectele care se vor transforma sunt:

```
> sq := [ [0,0], [1,0], [1,1], [0,1], [0,0] ]:
> segment:=  [  [0, 0] , [ 3, -2] ]:
> triang:=  [ [0,0], [2,0], [1,3], [0,0] ] :
> ngon:= proc( N: posint)  local j;
>  [seq([evalf(cos(2*Pi*j/N)),evalf(sin(2*Pi*j/N))],j=0..N)]:
> end:
> plot( {triang,sq,segment,ngon( 8 )} ) ;
```

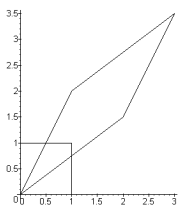
Grafic: vezi figura 3.12

```
> transform( symm, sq ); transform(symm,ngon(8));
```

Grafic: vezi figura 3.13

```
> flipx:= matrix(  [  [1, 0], [0, -1]  ] ):
> transform(flipx,triang); transform(flipx,ngon(8));
> transform(flipx,segment);
```

Grafic: vezi figura 3.14



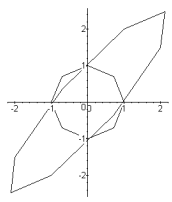


Figura 3.13:

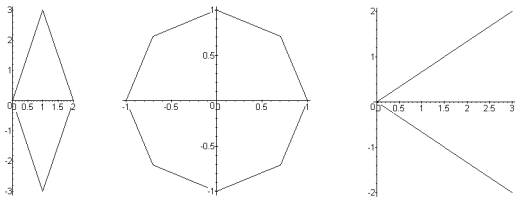


Figura 3.14:

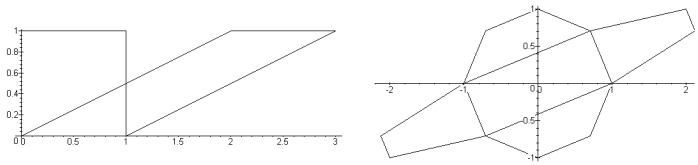


Figura 3.15:

```
> shear:= matrix([ [ 1, 2], [ 0, 1] ] ):
> transform(shear,sq); transform(shear,ngon(8));
```

Grafic: vezi figura 3.15

```
> rotate:=matrix([[1/sqrt(2),1/sqrt(2)],[-1/sqrt(2),1/sqrt(2)]]):
> transform(rotate,sq);transform(rotate,segment);
> transform(rotate, triang);
```

Grafic: vezi figura 3.16

Aplicaţia 4: Curba fulgului de zăpadă. A fost descrisă iniţial de Koch ca răspuns afirmativ la problema dacă există o curbă continuă care nu are linie tangentă în orice punct de pe curbă. Este definită ca limita secvenţei de curbe generate prin procedura `snowflake` dată mai jos.

Prima procedură, `basic`, efectuează o operaţie de bază asupra oricărui segment introdus ca dată de intrare prin capetele sale: segmentul de lungime d este înlocuit cu o secvenţă de 4 segmente de lungime $d/3$ obţinute astfel: se porneşte de la capătul stâng al segmentului, se parcurge $d/3$ din segment, după care are

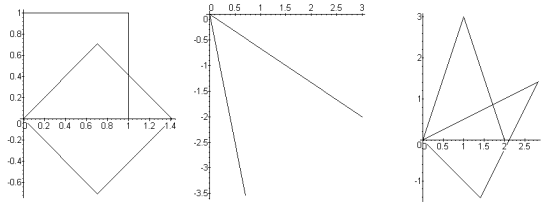


Figura 3.16:

loc o întoarcere cu 60 de grade la stânga, direcţie pe care se parcurge $d/3$, apoi din nou o rotaţie cu 60 de grade, însă înspre dreapta, şi distanţă $d/3$, şi din nou cu 60 de grade înspre stânga şi distanţă $d/3$. Curba rezultată are trei puncte în care nu are tangentă.

```
> basic := proc(p1,p2)
> local dx,dy, p3,p4,p5;
> dx := (p2[1]-p1[1])/3.;
> dy := (p2[2]-p1[2])/3.;
> p3 := p1[1]+dx,p1[2]+dy;
> p4 := p1[1]+2*dx,p1[2]+2*dy;
> p5 := p1[1]+1.5*dx-sqrt(3.)/2*dy,p1[2]+1.5*dy+sqrt(3.)/2.*dx;
> p3,p5,p4,op(p2);
> end:
```

De exemplu, dacă aplicăm `basic` asupra segmentului de lungime 1 cu capetele [0,0], [1,0] obţinem o secvenţă de numere:

```
>  basic([0,0],[1,0]);
```

$$.3333333333, 0, .5000000000, .2886751347, .6666666666, 0, 1, 0$$

Punctul din stânga a segmentului original a fost în mod intenţionat omis, pentru ca procedura `basic` să poată fi utilizată de următoarea. Procedura `flake` preia o listă de puncte ce reprezintă o secvenţă de segmente de linii legate cap la cap şi returnează o listă reprezentând de 4 ori mai multe segmente de linii, unde fiecare segment din lista originală este înlocuit cu cele 4 segmente furnizate de procedura `basic`.

```
> flake := proc(fl) local i,curve;
>  curve := [fl[1],fl[2]];
>  for i by 2 to nops(fl)-2 do
>   curve:=[op(curve),basic([fl[i],fl[i+1]],[fl[i+2],fl[i+3]])]
>  od
> end:
```

Fulgul de zăpadă se obţine pornind de la un triunghi echilateral:

```
> snowflake := proc(n)
>  local i,curve,ti;
>      curve := [0,0,1/2,1/2*sqrt(3),1,0,0,0];
>      for i from 2 to n do  curve := flake(curve) od;
>      plot(curve,style = LINE,axes = NONE,scaling = CONSTRAINED)
>  end:
> snowflake(5);
```

Grafic: vezi figura 3.17

Aplicaţia 5: Se consideră ecuaţia $f(x) = 0$ unde f este diferenţiabilă. Prin metoda grafică sau ghicind, putem ajunge la o primă estimare a soluţie. Fie x_0 această estimare. Probabil că punctul $(x_0, f(x_0))$ nu se află pe axa x (altfel am obţinut soluţia). Astfel parcurgem linia tangentă la graficul lui f până în punctul în care aceasta întâlneşte axa x. Fie $(x_1, 0)$ acest punct de intersecţie. x_1 poate fi exprimat astfel:

```
>  f:= eq :=  diff(f(x0),x0) = (f(x0)-0)/(x0-x1);
```

$$eq := \frac{\partial}{\partial x0}\,\mathrm{f}(\,x0\,) = \frac{\mathrm{f}(\,x0\,)}{x0 - x1}$$

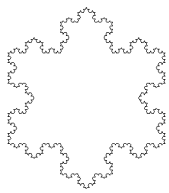


Figura 3.17:

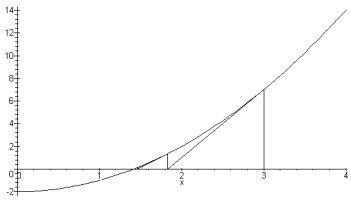


Figura 3.18:

```
> expand(solve(eq,{x1}));
```

$$\left\{ x1 = x0 - \frac{\mathrm{f}(\,x0\,)}{\frac{\partial}{\partial x0}\,\mathrm{f}(\,x0\,)} \right\}$$

Această ecuaţie este numită ecuaţia Newton. După calculul lui x_1, acesta poate fi redenumit x_0 şi se poate aplica metoda din nou. În anumite condiţii secvenţa de numere converge spre soluţia exactă. Procedura următoare oferă o imagine pentru modul de aplicare a metodei, fiind utilă în studiul convergenţei:

```
> vnewt := proc(f,start,a,b) local  i, x0, fp, p;
>   x0 :=  evalf(start); fp := D(f);  p := x0,0;
>   for i from 1 to 10 do
>     p := p, x0, f(x0); x0 := x0 - f(x0)/fp(x0);
>     p := p,x0,0;
>   od;
>   plot({f(x),[p]},x=a..b,style=LINE);
> end:
```

Testăm această procedură pentru rezolvarea ecuaţiei $x^2 = 2$ plecând de la estimarea $x_0 = 3$ şi utilizând intervalul de trasare de la 0 la 4.

```
> vnewt(x -> x^2 - 2, 3, 0,  4);
```

Grafic: vezi figura 3.18

Exerciţii: a) Utilizaţi procedura **vnewt** pentru a rezolva ecuaţia $\cos x - x = 0$ în x pentru diferite valori de start de la 2 la 3.05.

b) Utilizaţi procedura **vnewt** pentru a rezolva $x^2 + 0.01 = 0$. Porniţi de la 0.2.

c) Modificaţi definiţia procedurii **vnewt** pentru a controla numărul de iteraţii.

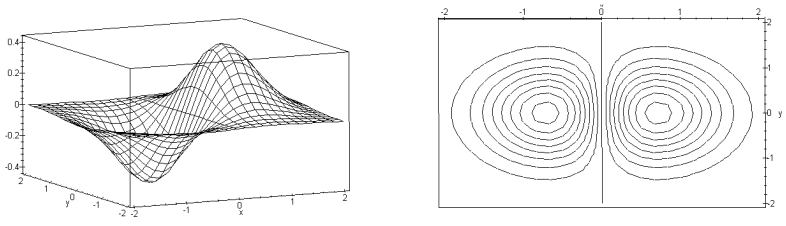


Figura 3.19:

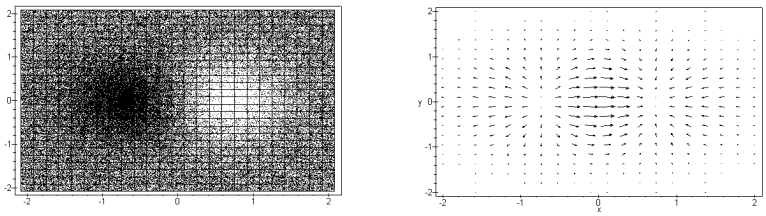


Figura 3.20:

3.8.2 Grafică 3D

Se consideră următoarea funcţie:

```
> f := x*exp(-x^2-y^2):
```

Graficul său arată astfel:

```
> plot3d(f,x=-2..2,y=-2..2,axes=BOX,orientation=[-116,75]);
```

Grafic: vezi figura 3.19.a

`Contourplot` este utilizat pentru a trasa contururile lui f în planul xy:

```
> contourplot(f,x=-2..2,y=-2..2,axes=BOX,shading=Z);
```

Grafic: vezi figura 3.19.b

Un grafic de densitate este o altă cale de a transpune informaţia tridimensională într-un plan.

```
> densityplot(f, x=-2..2, y=-2..2, axes=BOX );
```

Grafic: vezi figura 3.20.a

Putem genera de asemenea un grafic al gradientului lui f utilizând `gradplot`. Săgeţile sunt orientate în direcţia de creştere a lui f. Mărimea săgeţilor indică mărimea pantei.

```
> gradplot(f,x=-2..2,y=-2..2,style=LINE,arrows=SLIM, axes=BOX);
```

Grafic: vezi figura 3.20.b

Un câmp vectorial poate fi vizualizat cu ajutorul funcţiei `fieldplot3d`:

```
> F := [ y/(x^2+y^2),-x/(x^2+y^2),0]:
> fieldplot3d(F,x=-1..1,y=-1..1,z=-1/10..1/10,scaling=constrained);
```

Grafic: vezi figura 3.21

Suprafeţele descrise explicit pot fi trasate cu `plot3d`:

```
> plot3d((x^2-y^2)/(x^2+y^2),x=-2..2,y=-2..2,axes=BOXED);
```

Grafic: vezi figura 3.22

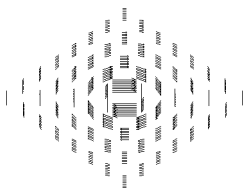


Figura 3.21:

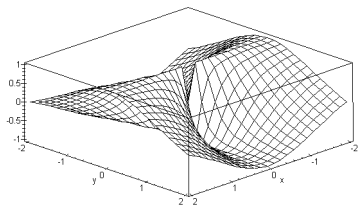


Figura 3.22:

O suprafaţă tridimensională descrisă implicit poate fi trasată cu:

```
> implicitplot3d( x^2+y^2+z^2-z*x*y=1, x=-2..2, y=-2..2, z=-2..2,
>   orientation=[9,63] );
```

Grafic: vezi figura 3.23.a

O suprafaţă parametrică tridimensională poate fi vizualizată cu plot3d:

```
> plot3d([cos(t)*(1+.2*sin(u)),sin(t)*(1+.2*sin(u)),.2*sin(t)*
>        cos(u)],t=0..2*Pi,u=-Pi..Pi);
```

Grafic: vezi figura 3.23.b

Suprafaţa Kuen este de asemenea o suprafaţă descrisă parametric:

```
> kuenx:=(u,v)->2*(cos(u)+ u*sin(u))*sin(v)/(1 +(u*sin(v))^2):
> kueny:=(u,v)->2*(sin(u)- u*cos(u))*sin(v)/(1 +(u*sin(v))^2):
> kuenz:=(u,v)->log(tan(v/2))+2*cos(v)/(1 +(u*sin(v))^2):
> plot3d([kuenx,kueny,kuenz], -4..4, .01 .. Pi-.01,grid=[35,35],
>    orientation=[134,70]);
```

Figura 3.23:

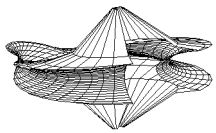


Figura 3.24:

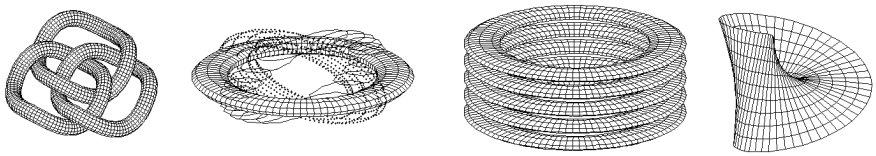


Figura 3.25:

Grafic: vezi figura 3.24

Inelele Borromean se pot construi utilizând funcţia `tubeplot`:

```
> k := 3.^(1/2):
> N := 15:
> knot := tubeplot({[1+k*cos(t),k*sin(t),.3*sin(3*t),
>   t=0..2*Pi,radius=.3,numpoints=trunc(6.4*N),tubepoints=N],
>   [-1/2+k*cos(t),k/2+k*sin(t),.3*sin(3*t),
>   t=0..2*Pi,radius=.3,numpoints=trunc(6.4*N),tubepoints=N],
>   [-1/2+k*cos(t),-k/2+k*sin(t),.3*sin(3*t),
>   t=0..2*Pi,radius=.3,numpoints=trunc(6.4*N),tubepoints=N]},
>   scaling=CONSTRAINED,orientation=[63,37]):
> knot;
```

Grafic: vezi figura 3.25.a

Stilurile de trasare ale obiectelor pot să difere:

```
> with(plots):
> opts := t=0..2*Pi,numpoints=100,radius=.2:
> p1 := tubeplot([cos(t),sin(t),0],opts, style=PATCH):
> p2 := tubeplot([0,cos(t),sin(t)],opts, style=POINT):
> p3 := tubeplot([sin(t),0,cos(t)],opts, style=CONTOUR):
> display({p1,p2,p3});
```

Grafic: vezi figura 3.25.b

Construirea unei proceduri poate simplifica trasarea şi retrasarea. De exemplu, o procedură de trasare a unui număr arbitrar n de tor-uri suprapuse este

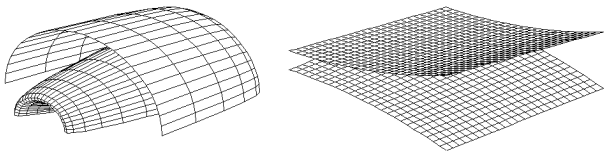


Figura 3.26:

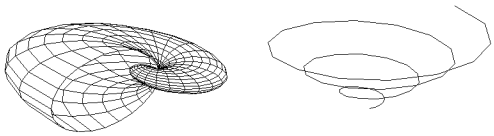


Figura 3.27:

următoarea:

```
> torii := proc(n)
>  local s,k;
>  s := seq(tubeplot([cos(t),sin(t),k],
>    t=0..2*Pi,numpoints=100,radius=.2),k=0..n-1);
>   display({s});
> end:
> torii(5);
```

Grafic: vezi figura 3.25.c

Pentru trasarea unei bande Moebius de lăţime oarecare:

```
> mob := proc(width)
> plot3d([(5+cos(1/2*t)*u)*cos(t), (5+cos(1/2*t)*u)*sin(t),
>   sin(1/2*t)*u],t=0..2*Pi,u=-width/2..width/2, grid=[60,10],
>   scaling=CONSTRAINED,orientation=[-106,70]);
> end:
> mob(15);
```

Grafic: vezi figura 3.25.d

Alte exemple:

```
> plot3d([x*sin(x),x*cos(y),x*sin(y)],x=0..2*Pi,y=0..Pi);
```

Grafic: vezi figura 3.26.a

```
> plot3d({x+y^2,-x-y^2},x=0..3,y=0..3);
```

Grafic: vezi figura 3.26.b

```
> plots[sphereplot]((1.3)^z*sin(theta),z=-1..2*Pi,theta=0..Pi);
```

Grafic: vezi figura 3.27.a

```
> plots[spacecurve]([t*cos(t),t*sin(t),t],t=0..7*Pi);
```

Grafic: vezi figura 3.27.b

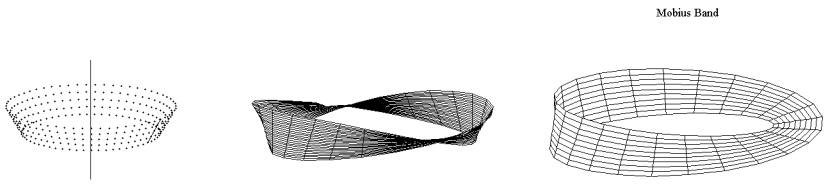


Figura 3.28:

Aplicaţia 6: Banda Moebius. Prin r desemnăm distanţa până la axa z. Se consideră un segment de linie în planul rOz, fie $r(t) = a + bt$, $z(t) = ct$, $-4 \leq t \leq 4$ care se roteşte în jurul axei z. Se obţine o suprafaţă de forma unei bande. Raportul c/b poate varia. Fie ϕ unghiul dintre linie şi axa r. Atunci $b = \cos(\phi)$ şi $c = \sin(\phi)$. Se consideră $\phi = 2\pi/3$.

```
> phi:=-2*Pi/3:  s:=ş  t:=ţ
> L1:=spacecurve({[a+cos(phi)*t, 0, sin(phi)*t, t=-2..2,color=red],
>  [0,0,t, t=-8..8, color=violet]}):
> L2:=plot3d([(a+cos(phi)*t)*cos(s),(a+cos(phi)*t)*sin(s),sin(phi)*t],
>   s=0..2*Pi,t=-2..2,scaling=constrained,grid=[60,5],style=POINT,
>   orientation=[-55, 75]):
> display3d({L1, L2}, orientation=[-55,75]);
```

Grafic: vezi figura 3.28.a

Presupunem că panta depinde de s, de exemplu $\phi = s$.

```
> phi:=s:
> plot3d([(a+cos(phi)*t)*cos(s),(a+cos(phi)*t)*sin(s),sin(phi)*t],
>   s=0..2*Pi,t=-2..2,scaling=constrained, style=patch,
>   orientation=[-25, 80]);
```

Grafic: vezi figura 3.28.b

Dacă $\phi = s/2$, atunci

```
> phi:=s/2:
> plot3d([(a+cos(phi)*t)*cos(s),(a+cos(phi)*t)*sin(s),sin(phi)*t],
>   s=0..2*Pi,t=-2..2,scaling=constrained, grid=[25,10],
>   style=patch, orientation=[-80, 75], title=`Mobius Band`);
```

Grafic: vezi figura 3.28.c

Aplicaţia 7: Cum construim un tor? Un tor standard poate fi văzut ca o suprafaţă generată prin rotirea cercului centrat în $x = a$, $z = 0$ cu rază b, în jurul axei z. Fie r distanţa de la axa z. Atunci acest cerc este $r - a = b\cos t$, $z = b\sin t$. Considerând o curbă de profil oarecare $(r(t), 0, h(t))$ în planul rOz şi dacă această curbă este rotită în jurul axei z, atunci suprafaţa rezultată are ecuaţiile parametrice $x(t,s) = r(t)\cos(s)$, $y(t,s) = r(t)\sin(s)$, $z(s,t) = h(t)$, pentru $0 \leq s \leq 2\pi$.

```
> a:=10: b:=1:
> circ:=spacecurve({[a+ b*cos(t),0,b*sin(t),t=0..2*Pi,color=red],
>   [0,0,t, t=-4..4]}):
```

Figura 3.29:

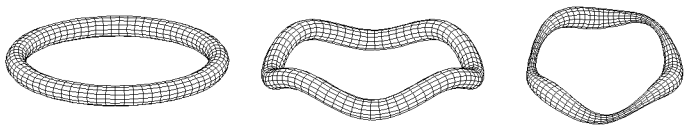


Figura 3.30:

```
> torus:=plot3d([(a + b*cos(t))*cos(s), (a + b*cos(t))*sin(s),b*
>   sin(t)], s=0..2*Pi, t=0..2*Pi,style=CONTOUR, scaling=CONSTRAINED):
> display3d({circ, torus}, orientation=[-60,80]);
```

Grafic: vezi figura 3.29.a

O altă posibilitate este aceea de a gândi torul ca un tub în jurul cercului $x^2 + y^2 = a^2$ din planul $z = 0$.

```
> Circ:=spacecurve({[a*cos(s), a*sin(s), 0, s=0..2*Pi],[0,0,t,
>   t=-4..4]},orientation=[-60,80]):
> Torus:=plot3d([(a + b*cos(t))*cos(s),(a + b*cos(t))*sin(s),
>  b*sin(t)],s=0..2*Pi,t=0..2*Pi,scaling=CONSTRAINED,
>  style=POINT,grid=[50,20]):
> display3d({Circ, Torus}, orientation=[-60,80]);
```

Grafic: vezi figura 3.29.b

Torul poate fi privit ca o suprafaţă tubulară:

```
> tubeplot([a*cos(s), a*sin(s), 0], s=0..2*Pi, radius=1, scaling=
>  constrained,numpoints=75, tubepoints=15, orientation=[45,70]);
```

Grafic: vezi figura 3.30.a

Dacă coordonata z oscilează:

```
> tubeplot([a*cos(s),a*sin(s),cos(4*s)],s=0..2*Pi,radius=1,scaling=
>  constrained,numpoints=75, tubepoints=15, orientation=[45,70]);
```

Grafic: vezi figura 3.30.b

Dacă variază raza tubului:

```
> tubeplot([a*cos(s),a*sin(s),cos(4*s)],s=0..2*Pi,radius=(2+
>  cos(3*s))/2,scaling=constrained,numpoints=75,tubepoints=15);
```

Grafic: vezi figura 3.30.c

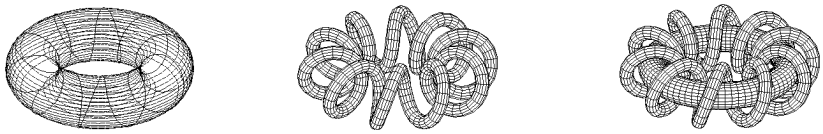


Figura 3.31:

Să construim un tub spiralat în jurul unui tor descris prin: $x(t,s) = (a + c\cos(t))\cos(s)$, $y(t,s) = (a + c\cos(t))\sin(s)$, $z(t,s) = c\sin(t)$ pentru $0 < s, t < 2\pi$. Pentru a obţine o curbă de pe tor, se consideră t ca funcţie de s, de exemplu $t = 10s$.

```
> t:=10*s: a:=10: c:=4:
> T1:=spacecurve([(a + c*cos(t))*cos(s),(a + c*cos(t))*sin(s),c*sin(t
> )],s=0..2*Pi,numpoints=200,scaling=constrained,orientation=[45,70]):
> T2:=tubeplot([a*cos(s),a*sin(s),0],radius=.98*c,s=0..2*Pi,
>  numpoints=75,tubepoints=25,style=CONTOUR,scaling=constrained,
>  orientation=[45,60]):
> display({T1,T2}, orientation=[45,60]);
```

Grafic: vezi figura 3.31.a

Această curbă este inclusă într-un tub:

```
> T3:=tubeplot([(a+c*cos(t))*cos(s),(a+c*cos(t))*sin(s),c*sin(t)],
>  s=0..2*Pi,radius=1,numpoints=200,tubepoints=15,scaling=
>  constrained): display({T3}, orientation=[45,65]);
```

Grafic: vezi figura 3.31.b

Introducem torul iniţial, dar cu o rază mai mică:

```
> T4:=tubeplot([a*cos(s),a*sin(s),0],radius=2,s=0..2*Pi,numpoints=
>    75, tubepoints=25, scaling=constrained, orientation=[45,60]):
> display({T3,T4}, orientation=[45,65]);
```

Grafic: vezi figura 3.31.c

Un nod toroidal se poate construi astfel:

```
> t:='t': a:=2:  b:=4/5:  c:=1:  m:=7: n:=4:
> r := a+b*cos(m*t):  z := c*sin(m*t):
> torus_knot := [r*cos(n*t),r*sin(n*t),z]:
> tubeplot(torus_knot,t=0..2*Pi,radius=1/4,
>  numpoints=200,tubepoints=20,orientation=[45,10],shading=XYZ);
```

Grafic: vezi figura 3.32

Un tub spiralat:

```
> f:= [10*sin(t)/3,10*cos(t)/3,(t-5*Pi)*.9,t=0..5*Pi]:
> TS1:=tubeplot(f,radius=1,orientation=[-37,81],tubepoints=25,
>  style=PATCH):
> f:= [(t-5*Pi)*sin(t)/3,(t-5*Pi)*cos(t)/3,(t-5*Pi)*.9,t=0..5*Pi]:
> TS2:= tubeplot(f,radius=1,orientation=[-37,81],tubepoints=25,
```

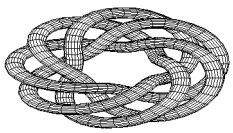


Figura 3.32:

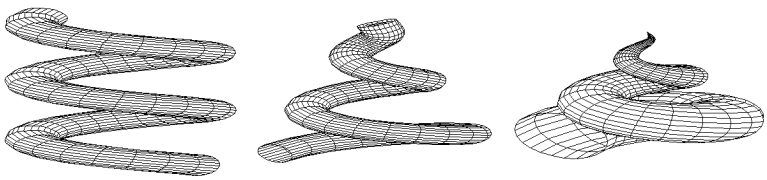


Figura 3.33:

```
>  style=PATCH):
> f:= [(t-5*Pi)*sin(t)/3,(t-5*Pi)*cos(t)/3,(t-5*Pi)*.9,t=0..5*Pi]:
> TS3:=tubeplot( [(t-5*Pi)*sin(t)/3,(t-5*Pi)*cos(t)/3,(t-5*Pi)*.9],
>  t=0..5*Pi,radius=(t-5*Pi)*.2,orientation=[-37,81],tubepoints=25,
>  style=PATCH):display(TS1); display(TS2); display(TS3);
```

Grafic: vezi figura 3.33

3.8.3 Grafică 4D

Maple V Release 3.0 permite realizarea animaţiei cu ajutorul funcţiilor `display` cu opţiunea `insequence=true`, `animate` sau `animate3d`.

Exerciţii:

```
> with(plots):
> animate(sin(Pi*x)*cos(Pi*t),x=0..1,t=0..2,frames=16);
> animate(sin(2*Pi*x)*cos(2*Pi*t),x=0..1,t=0..2,frames=16);
> animate(sin(3*Pi*x)*cos(3*Pi*t),x=0..1,t=0..2,frames=32);
> animate(sin(Pi*t*x),x=0..1,t=0.1..4,frames=64);
> animate3d([u*cos(v),u*sin(v),BesselJ(0,2.398*u)*cos(t)],u=0..1,
>  v=0..2*Pi,t=0...2*Pi,scaling=constrained,style=patch,frames=16);
> animate3d([u*cos(v),u*sin(v),BesselJ(1,3.821*u)*cos(v)*cos(1.5*t)],
> u=0..1,v=0..2*Pi,t=0..4,frames=16,scaling=constrained,style=patch);
> X := proc(u,v) a*cos(2*Pi*u)+b*cos(2*Pi*u)*cos(2*Pi*v) end:
> Y := proc(u,v) a*sin(2*Pi*u) + b*sin(2*Pi*u)*cos(2*Pi*v) end:
> Z := proc(u,v) b*sin(2*Pi*v) end:
> NX := proc(u,v) cos(2*Pi*u)*cos(2*Pi*v) end:
```

```
> NY := proc(u,v) -sin(2*Pi*u)*cos(2*Pi*v) end:
> NZ := proc(u,v) sin(2*Pi*v) end:
> a := 2.0: b := 0.4:
> animate3d([X(u,v),Y(u,v),Z(u,v)],v=0..1,t=0..1,u=0..1.,scaling=
> constrained);
> Mode1X:=(u,v,t)->X(u,v)+c*NX(u,v)*sin(Pi*u)*sin(Pi*v)*cos(Pi*t):
> Mode1Y:=(u,v,t)->Y(u,v)+c*NY(u,v)*sin(Pi*u)*sin(Pi*v)*cos(Pi*t):
> Mode1Z:=(u,v,t)->Z(u,v)+c*NZ(u,v)*sin(Pi*u)*sin(Pi*v)*cos(Pi*t):
> c := 0.2:
> animate3d([Mode1X(u,v,t),Mode1Y(u,v,t),Mode1Z(u,v,t)],u=0..1,
>  v=0..1,t=0..1,grid=[30,30],style=patch,scaling=constrained);
> Mode2X:=(u,v,t)->X(u,v)+c*NX(u,v)*sin(4*Pi*u)*sin(4*Pi*v)*cos(Pi*t):
> Mode2Y:=(u,v,t)->Y(u,v)+c*NY(u,v)*sin(4*Pi*u)*sin(4*Pi*v)*cos(Pi*t):
> Mode2Z:=(u,v,t)->Z(u,v)+c*NZ(u,v)*sin(4*Pi*u)*sin(4*Pi*v)*cos(Pi*t):
> animate3d([Mode2X(u,v,t),Mode2Y(u,v,t),Mode2Z(u,v,t)],u=0..1,
>  v=0..1,t=0..2,style=patch,frames=16,scaling=constrained);
> Donut1:=(u,v,t)->(1-t)*cos(u)+t*(3*cos(u)+0.6*cos(u)*cos(v)):
> Donut2:=(u,v,t)->(1-t)*(v-Pi)+t*(3*sin(u)+0.6*sin(u)*cos(v)):
> Donut3:=(u,v,t)->(1-t)*sin(u) + t*0.6*sin(v):
> animate3d([Donut1(u,v,t),Donut2(u,v,t),Donut3(u,v,t)],u=0..2*Pi,
>  v=0..2*Pi,t=0..1,frames=16,scaling=constrained);
> Pipe1 := (u,v,t) -> (1-t)*u + t*cos(u): Pipe2 := (u,v,t) -> v:
> Pipe3 := (u,v,t) -> t*sin(u):
> animate3d([Pipe1(u,v,t),Pipe2(u,v,t),Pipe3(u,v,t)],u=0..2*Pi,
>  v=0..2*Pi,t=0..1,frames=16,scaling=constrained);
```

3.9 Algebră liniară

3.9.1 Vectori şi matrice

Maple deţine un pachet de bibliotecă pentru algebră liniară. Cu ajutorul acestuia, putem calcula valori proprii şi vectori proprii, determinanţi de matrice, putem aplica o serie operaţii asupra liniilor şi coloanelor unei matrice, sau putem determina anumite forme canonice de matrice. Majoritatea acestor funcţii pot fi utilizate asupra matricelor simbolice.

Pentru a putea utiliza facilităţile Maple-ului pentru algebră liniară, trebuie încărcat pachetul `linalg`. Ca rezultat se obţine o listă a funcţiilor disponibile.

```
> with(linalg);
Warning: new definition for   norm
Warning: new definition for   trace
```

[*BlockDiagonal*, *GramSchmidt*, *JordanBlock*, *Wronskian*, *add*, *addcol*, *addrow*, *adj*, *adjoint*, *angle*, *augment*, *backsub*, *band*, *basis*, *bezout*, *blockmatrix*, *charmat*, *charpoly*, *col*, *coldim*, *colspace*, *colspan*, *companion*, *concat*, *cond*, *copyinto*, *crossprod*, *curl*, *definite*, *delcols*, *delrows*, *det*, *diag*, *diverge*, *dotprod*, *eigenvals*, *eigenvects*, *entermatrix*, *equal*, *exponential*, *extend*, *ffgausselim*, *fibonacci*, *frobenius*, *gausselim*, *gaussjord*, *genmatrix*, *grad*, *hadamard*, *hermite*, *hessian*, *hilbert*, *htranspose*, *ihermite*, *indexfunc*, *innerprod*, *intbasis*, *inverse*, *ismith*, *iszero*, *jacobian*, *jordan*, *kernel*, *laplacian*, *leastsqrs*, *linsolve*, *matrix*, *minor*, *minpoly*, *mulcol*, *mulrow*, *multiply*, *norm*, *normalize*, *nullspace*, *orthog*, *permanent*, *pivot*, *potential*, *randmatrix*, *randvector*, *rank*, *ratform*, *row*, *rowdim*, *rowspace*, *rowspan*, *rref*, *scalarmul*, *singularvals*, *smith*, *stack*, *submatrix*, *subvector*, *sumbasis*, *swapcol*, *swaprow*, *sylvester*, *toeplitz*, *trace*, *transpose*, *vandermonde*, *vecpotent*, *vectdim*, *vector*]

Un **vector** în Maple este reprezentat printr-o matrice unidimensională indexată de la 1, iar o structură **array** este o matrice bidimensionlă cu indicii de linie şi coloană pornind de la 1. Considerăm de exemplu matricea Hilbert simetrică H cu intrările $H_{ij} = 1/(i+j-1)$:

```
> H:=array(1..4, 1..4):
> for i to 4 do for j to 4 do H[i,j]:=1/(i+j-1) od od;
> H;
```

$$H$$

```
> evalm(H);
```

$$\begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \frac{1}{4} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} \\ \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} \end{bmatrix}$$

Valoarea lui **H** este numele matricei H. Regulile de evaluare a matricelor sunt speciale. Ideea care stă la baza acestui fapt este aceea că o matrice poate să aibă intrări nedefinite. Pentru a afişa o matrice sau a returna o matrice dintr-o procedură trebuie utilizată funcţia **eval**.

Pachetul **linalg** conţine multe funcţii pentru calcul vectorial şi matriceal. O matrice poate fi creată şi cu comanda **matrix** din pachetul **linalg**:

```
> H1:=linalg[matrix](5,5,(i,j)->1/(i+j-1)):
> a:=vector([2,sin(x),4,5.3,beta]);
```

$$a := [2 \sin(x) 4 5.3 \beta]$$

```
> A:=matrix([[1,x,y],[0,1,z],[0,0,1]]);
```

$$A := \begin{bmatrix} 1 & x & y \\ 0 & 1 & z \\ 0 & 0 & 1 \end{bmatrix}$$

Utilizând funcţiile predefinite în `linalg` se poate calcula inversa, determinantul, sau soluţia unui sistem liniar şi multe altele.

3.9.2 Operaţii elementare

Maple utilizează operatorul &* pentru a desemna o multiplicare necomutativă. Pentru a evalua o expresie matriceală trebuie utilizată funcţia `evalm`.

```
> evalm(A&*A);
```

$$\begin{bmatrix} 1 & 2\,x & 2\,y + x\,z \\ 0 & 1 & 2\,z \\ 0 & 0 & 1 \end{bmatrix}$$

```
> evalm(A^2);
```

$$\begin{bmatrix} 1 & 2\,x & 2\,y + x\,z \\ 0 & 1 & 2\,z \\ 0 & 0 & 1 \end{bmatrix}$$

```
> det(A);
```

$$1$$

```
> B:=matrix([[a,a,c],[b,5,f],[g,c,d]]):
> det(B);
```

$$5\,a\,d - a\,f\,c - b\,a\,d + b\,c^2 + g\,a\,f - 5\,g\,c$$

```
> inverse(A);
```

$$\begin{bmatrix} 1 & -x & x\,z - y \\ 0 & 1 & -z \\ 0 & 0 & 1 \end{bmatrix}$$

Matricea exponenţială e^A este adesea utilizată în rezolvarea sistemelor liniare de ecuaţii diferenţiale.

```
> exponential(A);
```

$$\begin{bmatrix} \mathrm{e} & x\,\mathrm{e} & \frac{1}{2}\,x\,z\,\mathrm{e} + y\,\mathrm{e} \\ 0 & \mathrm{e} & z\,\mathrm{e} \\ 0 & 0 & \mathrm{e} \end{bmatrix}$$

3.9.3 Vectori şi valori proprii

Determinarea valorilor proprii şi a vectorilor proprii se poate face manual sau automat:

```
> C:=matrix([[2,3],[4,5]]);
```

$$C := \begin{bmatrix} 2 & 3 \\ 4 & 5 \end{bmatrix}$$

```
> F:=evalm(C-lambda*array(identity,1..2,1..2));
```

$$F := \begin{bmatrix} 2-\lambda & 3 \\ 4 & 5-\lambda \end{bmatrix}$$

```
> eq1:=det(F);
```

$$eq1 := -2 - 7\,\lambda + \lambda^2$$

Acesta este polinomul caracteristic. Rădăcinile sale sunt:

```
> egv:=solve(det(F),lambda);
```

$$egv := \frac{7}{2} + \frac{1}{2}\sqrt{57}, \frac{7}{2} - \frac{1}{2}\sqrt{57}$$

```
> evalm(subs(lambda=C,eq1));
```

$$\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

Vectorii proprii se pot obţine astfel:

```
> nullspace(subs(lambda=egv[1],evalm(F)));
```

$$\left\{ \left[-\frac{3}{8} + \frac{1}{8}\sqrt{57}\,1 \right] \right\}$$

Maple are comenzi pentru determinarea valorilor proprii şi a polinomului caracteristic:

```
> eigenvals(C);
```

$$\frac{7}{2} + \frac{1}{2}\sqrt{57}, \frac{7}{2} - \frac{1}{2}\sqrt{57}$$

```
> charpoly(C,lambda);
```

$$-2 - 7\,\lambda + \lambda^2$$

```
> eigenvects(C,'radical');
```

$$\left[\frac{7}{2} + \frac{1}{2}\sqrt{57}, 1, \left\{ \left[-\frac{3}{8} + \frac{1}{8}\sqrt{57}\,1 \right] \right\} \right], \left[\frac{7}{2} - \frac{1}{2}\sqrt{57}, 1, \left\{ \left[-\frac{3}{8} - \frac{1}{8}\sqrt{57}\,1 \right] \right\} \right]$$

Fiecare listă din această secvenţă conţine o valoare proprie, ordinul său de multiplicitate şi vectorul propriu asociat. Fără specificarea opţiunii `radical`:

```
> eigenvects(C);
```

$$\left[\mathrm{RootOf}(\_Z^2 - 7\,\_Z - 2), 1, \left\{ \left[1\,\frac{1}{3}\,\mathrm{RootOf}(\_Z^2 - 7\,\_Z - 2) - \frac{2}{3} \right] \right\} \right]$$

3.9.4 Forme canonice

Maple poate reduce matricele la diferite forme canonice. Se consideră o matrice generată aleator:

```
> G := array(1..4,1..4): r:=rand(100):
> for i to 4 do for j to 4 do G[i,j] := r() od od:
> evalm(G);
```

$$\begin{bmatrix} 81 & 70 & 97 & 63 \\ 76 & 38 & 85 & 68 \\ 21 & 9 & 55 & 63 \\ 57 & 60 & 74 & 85 \end{bmatrix}$$

```
> H:=extend(G,0,4,0): #evalm(H);
> for i to 4 do H[i,i+4] := 1 od: #evalm(H);
> J:=gaussjord(H);
```

$$J := \begin{bmatrix} 1 & 0 & 0 & 0 & \frac{-77657}{2815042} & \frac{127423}{2815042} & \frac{-45908}{1407521} & \frac{23671}{2815042} \\ 0 & 1 & 0 & 0 & \frac{23495}{1407521} & \frac{-38316}{1407521} & \frac{-11969}{1407521} & \frac{22110}{1407521} \\ 0 & 0 & 1 & 0 & \frac{54858}{1407521} & \frac{-35367}{1407521} & \frac{49454}{1407521} & \frac{-49020}{1407521} \\ 0 & 0 & 0 & 1 & \frac{-76611}{2815042} & \frac{30225}{2815042} & \frac{-3820}{1407521} & \frac{71383}{2815042} \end{bmatrix}$$

Submatricea 4×4 din dreapta este inversa lui G.

```
> H:=submatrix(J,1..4,5..8):
> evalm(G&*H);
```

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Maple poate calcula forma normală Jordan:

```
> J:=matrix([[34,25],[-16,-6]]):
> jordan(J,'T1');
```

$$\begin{bmatrix} 14 & 1 \\ 0 & 14 \end{bmatrix}$$

```
> evalm(T1&*J&*inverse(T1));
```

$$\begin{bmatrix} 14 & 1 \\ 0 & 14 \end{bmatrix}$$

Maple poate crea matrice Hilbert, matrice Vandermonde sau matrice Toeplitz:

```
> V := vandermonde( [x.(1..4)] );
```

$$V := \begin{bmatrix} 1 & x1 & x1^2 & x1^3 \\ 1 & x2 & x2^2 & x2^3 \\ 1 & x3 & x3^2 & x3^3 \\ 1 & x4 & x4^2 & x4^3 \end{bmatrix}$$

```
> factor(det(V));
```

$$(x2 - x3)(x4 - x3)(x4 - x2)(x1 - x3)(x1 - x2)(x1 - x4)$$

Exerciţii:

```
> T:=toeplitz([x.(1..4)]);
> C:=hilbert(5);
> r[0] := vector([2,4,6]); s[0] := vector([-3,7,-9]);
> y[0] := evalm(2*r[0]-s[0]/7); m[0] := dotprod(r[0],s[0]);
> angle(s[0],r[0]); evalf(convert(angle(s[0],r[0]),degrees));
> length_of_r := norm(r[0],2);
```

```
> r[t] := vector([sin(t),cos(t),t^2]);
> subs(t=2.0,eval(r[t]));
> dr[t] := map(diff,r[t],t);
> R[t] := convert(r[t],list);
> with(plots):setoptions3d(axes=NORMAL): spacecurve(R[t],t=0..5*Pi);
```

3.9.5 Factorizare

O matrice de eliminare este o matrice pătratică cu valori 1 pe diagonală şi 0 în rest, exceptând doar un element al diagonalei principale.

```
> elim := proc(size,row,col,entry) local A,i;
>   A := diag(seq(1,i = 1 .. size)); A[row,col] := entry; evalm(A)
>  end:
> elim(3,3,2,9);
```

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 9 & 1 \end{bmatrix}$$

Dacă A este o matrice pătratică care poate fi redusă la forma trinughiulară superioară fără interschimbarea liniilor astfel încât elementul pivot să nu fie zero, atunci A poate fi factorizată ca produs dintre o matrice inferior triunghiulară cu valori 1 pe diagonală şi o matrice superior triunghiulară cu valori nenule pe diagonala principală.

```
>  A := randmatrix(3,3);
```

$$A := \begin{bmatrix} 19 & -50 & 88 \\ -53 & 85 & 49 \\ 78 & 17 & 72 \end{bmatrix}$$

```
> e21 := elim(3,2,1,-A[2,1]/A[1,1]):
> U := evalm(e21 &* A);
```

$$U := \begin{bmatrix} 19 & -50 & 88 \\ 0 & \frac{-1035}{19} & \frac{5595}{19} \\ 78 & 17 & 72 \end{bmatrix}$$

```
> e31 := elim(3,3,1,-U[3,1]/U[1,1]):
> U := evalm(e31 &* U);
```

$$U := \begin{bmatrix} 19 & -50 & 88 \\ 0 & \frac{-1035}{19} & \frac{5595}{19} \\ 0 & \frac{4223}{19} & \frac{-5496}{19} \end{bmatrix}$$

```
> e32 := elim(3,3,2,-U[3,2]/U[2,2]):
> U := evalm(e32 &* U);
```

$$U := \begin{bmatrix} 19 & -50 & 88 \\ 0 & \frac{-1035}{19} & \frac{5595}{19} \\ 0 & 0 & \frac{62945}{69} \end{bmatrix}$$

Am obţinut $e_{32}e_{32}e_{21}A = U$.

```
> L := evalm(inverse(e21)&*inverse(e31)&*inverse(e32));
```

$$L := \begin{bmatrix} 1 & 0 & 0 \\ \frac{-53}{19} & 1 & 0 \\ \frac{78}{19} & \frac{-4223}{1035} & 1 \end{bmatrix}$$

Pe baza acestor observaţii putem construi o procedură de factorizare a matricei A:

```
> lufact := proc(A,`L`,`U`) local k,i,j,el;
>    k:=rowdim(A); U:=evalm(A); L:=diag(seq(1,i=1..k));
>    for i to k-1  do
>     if U[i,i] = 0 then ERROR(cat(`pivot 0 la pasul `,i))
>     else for j from i+1 to k do
>           el:=elim(k,j,i,-U[j,i]/U[i,i]); L[j,i]:=U[j,i]/U[i,i];
>           U := evalm(el &* U) od
>     fi od;
>    if U[k,k] = 0 then ERROR(cat(`Pivot zero la pas `,i)) fi;
>    NULL:
> end:
> A := randmatrix(4,4): lufact(A ,L,U):
> map(evalm,[A,L,U,L&*U]);
```

$$\left[\begin{bmatrix} 9 & 29 & -66 & -32 \\ 78 & 39 & 94 & 68 \\ -17 & -98 & -36 & 40 \\ 22 & 5 & -88 & -43 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 & 0 \\ \frac{26}{3} & 1 & 0 & 0 \\ \frac{-17}{9} & \frac{389}{1911} & 1 & 0 \\ \frac{22}{9} & \frac{593}{1911} & \frac{127399}{283054} & 1 \end{bmatrix}, \right.$$

$$\left. \begin{bmatrix} 9 & 29 & -66 & -32 \\ 0 & \frac{-637}{3} & 666 & \frac{1036}{3} \\ 0 & 0 & \frac{-566108}{1911} & \frac{-24772}{273} \\ 0 & 0 & 0 & \frac{-4401043}{141527} \end{bmatrix}, \begin{bmatrix} 9 & 29 & -66 & -32 \\ 78 & 39 & 94 & 68 \\ -17 & -98 & -36 & 40 \\ 22 & 5 & -88 & -43 \end{bmatrix} \right]$$

În cazul în care un element de pe diagonală este zero este necesară interschim-

barea unor linii. Matricele de permutare efectuează această operaţie.

O matrice de permutare este o matrice pătratică cu exact un element 1 în fiecare linie şi coloană şi 0 în rest. Sunt obţinute din matricea identitate prin permutarea liniilor.

```
> interchange := proc(size,row1,row2)
>  local A,i;
>      A := diag(seq(1,i = 1 .. size));
>      A := swaprow(A,row1,row2)
>  end:
```

Ştim că dacă o matrice pătratică A este inversabilă, atunci există o matrice de permutare P astfel încât PA poate fi factorizată $PA = LU$, unde L este inferior triunghiulară, iar U superior triunghiulară.

Exerciţiu: Modificaţi corespunzător definiţia procedurii `lufact` în `plufact('P',A,'L','U')`. Indicaţie: o observaţie cheie constă în faptul că o matrice de permutare P_{kl} comută cu o matrice de eliminare E_{ij} dacă $l > j$ şi $k > i$. O procedură pentru testarea comutativităţii este următoarea:

```
> checkit := proc(k,l,i,j) local eij,pkl;
>   eij := elim(5,i,j,m); pkl := interchange(5,k,l);
>   equal(evalm(eij &* pkl), evalm(pkl &* eij))
> end:
> checkit(2,1,4,3);
```

true

În pachetul `linalg` există implementări ale eliminării gaussiene cu interschimbarea liniilor. Acestea pot ajuta utilizatorul la rezolvarea sistemelor liniare:

```
>  A := randmatrix(3,4);
```

$$A := \begin{bmatrix} 62 & 11 & 88 & 1 \\ 30 & 81 & -5 & -28 \\ 4 & -11 & 10 & 57 \end{bmatrix}$$

```
> gausselim(A);
```

$$\begin{bmatrix} 4 & -11 & 10 & 57 \\ 0 & \frac{327}{2} & -80 & \frac{-911}{2} \\ 0 & 0 & \frac{2377}{109} & \frac{-41077}{109} \end{bmatrix}$$

```
> backsub('');
```

$$\left[\frac{63082}{2377} \; \frac{-26721}{2377} \; \frac{-41077}{2377}\right]$$

Exerciţii:

```
> A:= matrix( 2 , 2, [ -4, 5/2, -5/2, 1]);
> eigenvals(A); V:= eigenvects(A);
> v1:= op( op(3, V) );
> A:= matrix(2, 2, [4, 1, -2, 3]);   eigenvals(A);
> A:= matrix(2, 2, [-0.4, -4, 2, 0.4]);   eigenvals(A);
```

```
> A:=  matrix(2, 2, [2, 3, 4, 1]);   eigenvals(A);
> V:= eigenvects(A);
> v1:=op( op(3, op(1, [V]) ));   v2:=op( op(3, op(2, [V]) ));
> array(3..6,[a,b,c,d]);
> linalg[vector](3,[x1,y1,z1]);
> linalg[matrix](2,2,[1,2,3,4]); array(1..2,1..2,[[1,2],[3,4]]);
> A:=array([[-1,4,-2],[6,2,-10]]): B:=array([[2,-4,8],[7,4,2]]):
> evalm(3*A-9*B);
> A:=array([[-4,-2,-1],[5,-4,-3],[5,1,-2]]): linalg[det](A);
> alias(Id=&*()); evalm(A &* Id); linalg[inverse](A);
> A:=array([[4,-6],[3,-7]]);
> linalg[eigenvals](A); linalg[eigenvects](A);
> linalg[charpoly](A,lambda);
> linalg[eigenvects](array([[2,-2,-2],[4,-4,-2],[-2,1,-1]]));
> array(1..4,1..2,[[3,2],[5],[6,3]]);
> array(1..3,1..3,[(1,2)=f,(3,1)=h,(2,3)=g]);
> A:=array(1..3,1..3,[]);
> A[1,2]:=f: A[2,3]:=g: A[3,1]:=h: print(A);
> A[2,3];
> array(-3..0,[a,b,c,d]);
> entries(");
> A, eval(A), op(A), print(A), entries(A);
> B:=matrix(2,3,[a,b,c,d,e,f]); row(B,2);
> col(B,1..2);
> delrows(B,2..2); delcols(B,2..3);
> C:=extend(B,2,3,0);
> copyinto(matrix(2,3,[1$6]),C,3,4);
> array(1..2,1..2,identity);
> array(1..2,1..6,[(1,3)=p,(2,4)=r],sparse);
> array(1..3,1..3,[(1,3)=x^2,(2,1)=y/z,(2,3)=1],antisymmetric);
```

Aplicaţia 8: Se cere construirea unui program care efectuează reducerea unei matrice la o matrice superior triunghiulară prin eliminare Gauss. Parametrul secund, opţional, al funcţiei ne oferă rangul matricei.

```
> GaussianElimination:=proc(A,rang) local m,n,i,j,B,r,c,t;
>   if not type(A,`matrix(rational)`) then
>     ERROR(`Se asteapta o matrice de numere rationale`,A) fi;
>   m:=linalg[rowdim](A);n:=linalg[coldim](A);B:=array(1..m,1..n);
>   for i to m do for j to n do B[i,j]:=A[i,j] od od;
>   r:=1;
>   for c to m  while r<=n do
>    for i from r to n while B[i,c]=0 do od;
>     if i<=n then
>      if i<>r then
```

```
>      for j from c to m do t:=B[i,j];B[i,j]:=B[r,j];B[r,j]:=t od
>      fi;
>      for i from r+1 to n do
>       if B[i,c]<>0 then
>        t:=B[i,c]/B[r,c];
>        for j from c+1 to m do B[i,j]:=B[i,j]-t*B[r,j] od;
>        B[i,c]:=0
>       fi
>      od;
>      r:=r+1
>     fi
>   od;
>   if nargs>1 then rang:=r-1 fi;
>   eval(B)
> end:
```

Aplicaţia 9: Relaţii recursive via produs de matrice. În şirul numerelor lui Fibonacci,

$$1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, \ldots$$

fiecare termen este obţinut din adunarea celor doi termeni precedenţi. Aceste numere apar într-o varietate de probleme: de exemplu, în numărarea descendenţilor unei perechi de iepuri (problema iniţială a lui Fibonacci), în numărarea strămoşilor unei albine, înţelegerea aranjamentului spiral al frunzelor în jurul tulpinei unei plante etc. Dacă notăm cu $a(n)$ numărul n al lui Fibonacci, atunci definiţia recursivă este $a(n) = a(n-1) + a(n-2)$. Se pune problema determinării unei relaţii explicite pentru termenul general al şirului.

Deşi această problema nu pare să aibă nimic comun cu matricele, vom observa că soluţia sa implică matrice, valori proprii, vectori proprii şi diagonalizare, rezultatul fiind o formulă concretă pentru $a(n)$. Definiţia recursivă a numerelor Fibonacci poate fi exprimată în termenii unei anumite matrice 2×2, notată cu A:

```
> A := linalg[matrix]([[0,1], [1,1]]):
```

Punem primii doi termeni ai şirului Fibonacci intr-un vector coloană:

```
> V := linalg[matrix]([[1], [1]]):
```

Multiplicăm matricea A cu vectorul coloană V:

```
> evalm(A&*V);
```

$$\begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

Multiplicăm A cu noul vector coloană, ceea ce este echivalent cu a multiplica A^2 cu vectorul iniţial V ...Şi încă o dată...

```
> evalm(A^2&*V), evalm(A^3&*V);
```

$$\begin{bmatrix} 2 \\ 3 \end{bmatrix}, \begin{bmatrix} 3 \\ 5 \end{bmatrix}$$

Dacă aplicăm matricea A la vectorul coloană ale cărui intrări sunt $a(n-1)$

şi $a(n)$, obţinem vectorul colonă a cărui intrări sunt $a(n)$ şi $a(n+1)$. Dacă aplicăm matricea A^n la vectorul coloană V cu intrările $a(0) = 1$ şi $a(1) = 1$, obţinem vectorul coloană a cărui intrări sunt $a(n)$ şi $a(n+1)$. Astfel, problema găsirii unei formule explicite pentru al n-lea termen a şirului este transformată în problema determinării unei formule explicite pentru a n putere a matricei A. Dacă diagonalizăm A: $P^{-1}AP = D$, adică $A = PDP^{-1}$, atunci $A^n = P(D^n)P^{-1}$.

Primul pas în diagonalizarea matricei A este calcularea valorilor sale proprii. Acestea sunt rădăcinile polinomului caracteristic, $\det(A - xI) = 0$

Notaţia pentru matricea identitate în Maple este:

```
> &*():
```

Deoarece această notaţie este greoaie, îi putem asocia un ,,alias", de exemplu Id (I în Maple este rezervat numărului complex $\sqrt{-1}$).

```
> alias(Id = &*());
```

$$I, Id$$

Calculăm polinomul caracteristic al lui A:

```
> B := evalm(A - x*Id):
> linalg[det](B);
```

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

Determinăm rădăcinile sale (acestea vor fi valorile proprii ale lui A):

```
> lambda :=solve(linalg[det](B),x);
```

$$\lambda := \frac{1}{2} + \frac{1}{2}\sqrt{5}, \frac{1}{2} - \frac{1}{2}\sqrt{5}$$

Determinăm vectorii proprii corespunzători:

```
> S1 := linalg[nullspace](A - lambda[1]*Id);
```

$$S1 := \left\{ \left[-\frac{1}{2} + \frac{1}{2}\sqrt{5}\, 1 \right] \right\}$$

Continuăm cu a doua valoare:

```
> S2 := linalg[nullspace](A - lambda[2]*Id);
```

$$S2 := \left\{ \left[-\frac{1}{2} - \frac{1}{2}\sqrt{5}\, 1 \right] \right\}$$

Construim matricea P de 2×2 cu vectorii proprii de mai sus drept coloane.

```
> P := linalg[matrix](2,1,S1[1]):
> P := linalg[augment](P,S2[1]):
```

Avem nevoie de inversa lui P:

```
> DIAG := (linalg[inverse](P)&*A)&*P:
```

Utilizăm `DIAG` în loc de `D` deoarece Maple rezervă "D" pentru diferenţiere.

```
> evalm(DIAG);
```

$$\begin{bmatrix} \frac{1}{2} + \frac{1}{2}\sqrt{5} & 0 \\ 0 & \frac{1}{2} - \frac{1}{2}\sqrt{5} \end{bmatrix}$$

Puterea n-a a lui D, poate fi scrisă explicit uşor:

```
> Dn:=linalg[diag]((1/2+(1/2)*sqrt(5))^n,(1/2-(1/2)*sqrt(5))^n):
> An := (P&*Dn)&*linalg[inverse](P):
> evalm(An &* V);
```

$$\left[\begin{array}{l} \frac{1}{10}\,\%2\,\sqrt{5}+\frac{1}{2}\,\%2-\frac{1}{10}\,\%1\,\sqrt{5}+\frac{1}{2}\,\%1 \\ \frac{3}{10}\,\%2\,\sqrt{5}-\frac{3}{10}\,\%1\,\sqrt{5}+\frac{1}{2}\,\%2+\frac{1}{2}\,\%1 \end{array}\right]$$

$$\%1 := \left(\frac{1}{2}-\frac{1}{2}\sqrt{5}\right)^{n}$$

$$\%2 := \left(\frac{1}{2}+\frac{1}{2}\sqrt{5}\right)^{n}$$

Prima intrare a vectorului este răspunsul la problemă: este al n termen al şirului Fibonacci. Simplificat, acesta se poate scrie:

```
> a(n):=(1/sqrt(5))*((1/2+(1/2)*sqrt(5))^(n+1)-
>         (1/2-(1/2)*sqrt(5))^(n+1));
```

$$\mathrm{a}(n) := \frac{1}{5}\sqrt{5}\left(\left(\frac{1}{2}+\frac{1}{2}\sqrt{5}\right)^{(n+1)}-\left(\frac{1}{2}-\frac{1}{2}\sqrt{5}\right)^{(n+1)}\right)$$

Exerciţiu: Utilizaţi această formulă pentru a găsi limita la $n \to \infty$ a şirului $a(n)/a(n-1)$. Cât de repede se apropie acest raport de limita sa?

3.9.6 Sisteme liniare

Pachetul de algebră liniară conţine funcţia `linsolve` pentru rezolvarea sistemelor de ecuaţii liniare.

```
> A := matrix([[-85, -55, -37, -35], [97, 50, 79, 56],
>             [49, 63, 57, -59], [45, -8, -93, 92]]):
> b := matrix([[43], [-62], [77], [66]]):
```

Putem utiliza eliminarea gaussiană:

```
> C := gausselim( concat(A,b), 4 ):
> x:=linsolve(submatrix(C,1..4,1..4), col(C,5));
```

$$x := \left[\frac{13019547}{1456498}\ \frac{-7299476}{728249}\ \frac{-1873417}{1456498}\ \frac{-184492}{31663}\right]$$

```
> #evalm( A &* x - b );
```

Se pot considera cazuri în care rezultatele sunt exprimate în termenii unor parametrii liberi.

```
> A := matrix([[-68, -36, 0, 0, 90], [0, 0, 81, 0, 0],
> [-90, -79, 0, -92, 0], [0, 0, -24, 0, 0], [-58, 0, 0, 0, 0]]):
> b := vector(5,0):
> C := gausselim( concat(A,b), 5 ):
> x := linsolve( submatrix(C,1..5,1..5), col(C,6) );
```

$$x := \left[0\ \frac{5}{2}\,\_t_1\ 0\ -\ \frac{395}{184}\,\_t_1\ \_t_1\right]$$

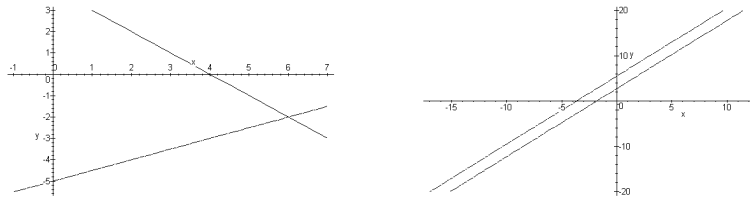


Figura 3.34:

3.10 Rezolvarea ecuaţiilor algebrice

Rezolvarea sistemelor de ecuaţii se poate realiza prin metoda grafică, metoda substituţiei, sau utilizând `solve`.

În metoda grafică se trasează graficul iar pe baza acestuia se aproximează soluţia. De exemplu, nu punem problema punctului de intersecţie a lui $x+y=4$ cu $x-2y=10$. Utilizăm Maple pentru vizualizarea celor două linii:

```
> with(plots):
> implicitplot({x+y=4, x-2*y=10}, x=-1..7, y=-6..3);
```

Grafic: vezi figura 3.34.a

Graficul este utilizat pentru estimarea punctului de intersecţie; în cazul curent, punctul de intersecţie este (6,-2). Comanda `subs` este utilizată pentru a verifica estimarea făcută:

```
> subs({x=6, y=-2}, {x+y=4, x-2*y=10});
```

$$\{\, 4=4, 10=10 \,\}$$

În metoda substituţiei, exprimăm y în termenii lui x:

```
> solve(x+y=1,{y});
```

$$\{\, y=-x+1 \,\}$$

Substituim rezultatul în ecuaţia a doua:

```
> subs(y=-x+1, 4*x-8*y=49);
```

$$12\,x-8=49$$

şi rezolvăm ecuaţia în x:

```
> solve( " , {x});
```

$$\left\{ x=\frac{19}{4} \right\}$$

Substituim valoarea lui x în ecuaţia lui y:

```
> subs(x=19/4, y=1-x);
```

$$y=\frac{-15}{4}$$

Putem utiliza direct funcţia `solve`. Rezolvăm, de exemplu:

$$(a)\ \begin{cases} 6x-7y+11z-4u=74 \\ x+3y-9z=u-22 \\ x-y-z-u=6 \\ x+y=z-u-10 \end{cases} \qquad (b)\ \begin{cases} 2y-3x-11=0 \\ 4y-6x-11=0 \end{cases}$$

```
> solve({6*x-7*y+11*z-4*u=74, x+3*y-9*z=u-22, x-y-z-u=6,
```

```
> x+y=z-u-10}, {x,y,z,u});
```

$$\{\, z = 3, u = -7, x = 1, y = -1 \,\}$$

```
> solve({2*y-3*x-11=0, 4*y-6*x-11=0}, {x,y});
```

În cazul al doilea Maple nu a oferit nici o soluţie pentru că nu există soluţie. Verificăm această concluzie trasând curbele asociate ecuaţiilor:

```
> implicitplot({2*y-3*x-11=0, 4*y-6*x-11=0}, x=-20..20, y=-20..20);
```

Grafic: vezi figura 3.34.b

Maple poate rezolva ecuaţii sau sisteme de ecuaţii, fie exact, fie numeric în cazul în care nu poate determina o soluţie simbolică.

```
> eqs1:={x+2*y+3*z+4*t+5*u=6,5*x+5*y+4*z+3*t+2*u=1,
> 3*y+4*z-8*t+2*u=1,x+y+z+t+u=9,8*x+4*z+3*t+2*u= 1}:
> a1:=solve(eqs1, {x,y,z,t,u});
```

$$a1 := \left\{ u = \frac{8589}{110}, x = 56, y = \frac{168}{5}, t = \frac{-1736}{55}, z = \frac{-13983}{110} \right\}$$

Răspunsul este returnat sub formă de mulţime de expresii.

```
> subs(a1,eqs1);
```

$$\{\, 6 = 6, 1 = 1, 9 = 9 \,\}$$

Răspunsul poate să nu fie numeric. De exemplu, căutăm soluţia unui sistem de 3 ecuaţii în 4 necunoscute:

```
> eqs2:={(3*a^2+9)*x+(1-3*a^2)*y+(a^2-2*a-2)*z = -7*a^2,
> (7+4*a)*x+(-8*a^2+3+5*a)*y+2*a*z = 9*a+4+2*a^2,
> -5*a^2*x+(-3*a-7*a^2)*y+(5-6*a)*z = 4-5*a-a^2}:
> a2:=solve(eqs2,{x,y,z});
```

$$a2 := \left\{ x = \frac{22\,a^6 + 524\,a^5 - 603\,a^4 - 181\,a^3 - a^2 - 45\,a - 4}{\%1}, \right.$$
$$y = -\frac{10\,a^6 + 61\,a^5 - 379\,a^4 - 292\,a^3 + 79\,a^2 + 99\,a + 124}{\%1},$$
$$\left. z = -2\,\frac{188\,a^6 + 228\,a^5 + 237\,a^4 + 518\,a^3 + 41\,a^2 + 86\,a + 40}{\%1} \right\}$$
$$\%1 := 40\,a^6 - 293\,a^5 + 224\,a^4 - 580\,a^3 + 268\,a^2 - 127\,a - 100$$

```
> map(normal,subs(a2,eqs2));
```

$$\{\, -7\,a^2 = -7\,a^2, 9\,a + 4 + 2\,a^2 = 9\,a + 4 + 2\,a^2, 4 - 5\,a - a^2 = 4 - 5\,a - a^2 \,\}$$

Maple poate rezolva şi ecuaţii neliniare:

```
> eqn3:=cos(x) + y = 9:
> solve(eqn3,x);
```

$$\pi - \arccos(\, y - 9 \,)$$

```
> eqn4 := 2^u + G = 0;
```

$$eqn4 := 2^u + G = 0$$

```
> solve(eqn4,u);
```

$$\frac{\ln(\, -G \,)}{\ln(\, 2 \,)}$$

Pentru rădăcinile unui polinom de grad trei, Maple returnează o secvenţă a soluţiilor exacte.

```
> eqn3:=x^3-54*x^2+972*x-5839:
> solve(eqn3,x);
```

$$7^{1/3}+18, -\frac{1}{2}\,7^{1/3}+18+\frac{1}{2}\,I\,\sqrt{3}\,7^{1/3}, -\frac{1}{2}\,7^{1/3}+18-\frac{1}{2}\,I\,\sqrt{3}\,7^{1/3}$$

Pentru polinoame de grad patru sau mai mare, Maple utilizează reprezentarea `RootOf` pentru rădăcini.

```
> eqn4:=x^5+3*x^2-2*x-1;
```

$$eqn4 := x^5+3\,x^2-2\,x-1$$

```
> a4:=solve(eqn4,x);
```

$$a4 := \mathrm{RootOf}(\_Z^5+3\,\_Z^2-2\,\_Z-1)$$

Putem determina valorile rădăcinilor utilizând funcţia `allvalues`. Dacă Maple nu găseşte o soluţie acceptabilă în forma `RootOf`, atunci returnează un răspuns numeric.

```
> s4:=[allvalues(a4)];
```

$$s4 := [-1.569778412, -.3343775377, .5194892704-1.390014884\,I, .5194892704+1.390014884\,I, .8651774091]$$

Maple poate de asemenea rezolva sisteme de ecuaţii neliniare.

```
> eqs5 := {2*x*y = 1, x + a*z = 0, 2*x - 3*z = a}:
> solve(eqs5,{x,y,z});
```

$$\left\{z=-\frac{a}{3+2\,a}, y=\frac{1}{2}\,\frac{3+2\,a}{a^2}, x=\frac{a^2}{3+2\,a}\right\}$$

În cazul unor soluţii multiple, Maple returnează mulţimea soluţiilor.

```
> eqs6:={x+y+z+w=6,x^2+y^2+z^2+w^2=14,
>        x^3+y^3+z^3+w^3=36,x^4+y^4+z^4+w^4=98}:
> solve(eqs6,{w,x,y,z});
```

$$\{x=0,z=3,w=2,y=1\},\{x=0,z=3,w=1,y=2\},$$
$$\{x=0,w=3,y=2,z=1\},\{y=3,x=0,w=2,z=1\},$$
$$\{x=0,w=3,y=1,z=2\},\{y=3,x=0,w=1,z=2\},$$
$$\{z=0,x=1,w=3,y=2\},\{y=3,z=0,x=1,w=2\},$$
$$\{x=1,w=3,y=0,z=2\},\{y=3,x=1,w=0,z=2\},$$
$$\{z=3,x=1,w=2,y=0\},\{z=3,x=1,w=0,y=2\},$$
$$\{z=0,w=3,y=1,x=2\},\{y=3,z=0,w=1,x=2\},$$
$$\{z=0,w=2,y=1,x=3\},\{z=0,w=1,y=2,x=3\},$$
$$\{w=2,y=0,z=1,x=3\},\{w=0,y=2,z=1,x=3\},$$
$$\{w=1,y=0,x=3,z=2\},\{w=0,y=1,x=3,z=2\},$$
$$\{w=3,y=0,z=1,x=2\},\{y=3,w=0,z=1,x=2\},$$
$$\{z=3,w=1,y=0,x=2\},\{z=3,w=0,y=1,x=2\}$$

Sunt situaţii în care Maple nu poate oferi o soluţie simbolică.

```
> eqn7:=exp(x*ln(18/10+x))+13/10*x-7=0:
> a7:=solve(eqn7,x);
```

$$a7 :=$$

Putem în asemenea caz să obţinem o soluţie numerică.

```
> a7:=fsolve(eqn7,x);
```

$$a7 := 1.408842218$$

Soluţia poate fi calculată cu un număr arbitrar de zecimale.

```
> Digits:=30:
> a7:= fsolve(eqn7,x);
```

$$a7 := 1.408842217802056795573666088 34$$

```
> Digits:=10:
> evalf(subs(x=a7,eqn7));
```

$$0 = 0$$

Funcţia **fsolve**, precum **solve**, poate fi utilizată pentru determinarea rădăcinilor unui polinom. În general, **fsolve** găseşte o singură rădăcină reală (exceptând cazul determinării rădăcinilor unui polinom). Dacă se adaugă opţiunea **complex**, se caută soluţiile în domeniul complex.

```
> eqn8 := 3*x^4 - 16*x^3 - 3*x^2 + 13*x + 16:
> a8:={fsolve(eqn8,x,complex)};
```

$$a8 := \{1.324717957, 5.333333333, -.6623589786 + .5622795121\, I,$$
$$-.6623589786 - .5622795121\, I\}$$

```
> for y in a8 do evalf(subs(x=y,eqn8)) od;
```

$$0$$
$$.1\, 10^{-7}$$
$$.2\, 10^{-8}\, I$$
$$-.2\, 10^{-8}\, I$$

Pentru a verifica soluţia unei ecuaţii putem utiliza comanda Maple **subs**.

Exerciţii:

```
> solve(4*(2*m+1)-11=2*(m+1)-3, {m});
> subs(m=1, 4*(2*m+1)-11=2*(m+1)-3);
> subs(y=3, 3*(y-2)=2*(y+5)+k);
> solve('', {k});
> eqns:={x^2*y^2=0,x+y=1};
> solve(eqns,{x,y});
> assign('');
> x,y;
> x:='x'; y:='y';
> isolve(2*x+3*y=7);
> poly:=45*x^6-2*x^5-45*x^4+10*x^2-12*x+1;
> solve(poly,x);
> fsolve(poly,x);
> fsolve(poly,x,1..2);
```

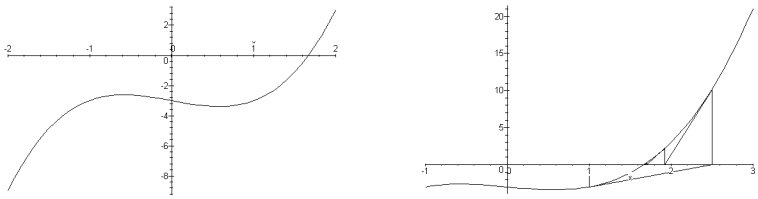


Figura 3.35:

```
> fsolve(poly,x,complex);
> readlib(realroot);
> realroot(x^3+37*x-21,1/100);
> realroot(x^8+5*x^7-4*x^6-20*x^5+4*x^4+20*x^3, 1/1000);
```

Maple are o varietate de subrutine de rezolvare a ecuaţiilor. De exemplu, pentru relaţiile recursive se poate utiliza `rsolve`:

```
> eq1 := {f(n+2) = f(n+1) + f(n) , f(0) = 1 , f(1) = 1}:
> a1:=rsolve(eq1,f);
```

$$a1 := -\frac{2}{5}\,\frac{\sqrt{5}\left(-2\,\dfrac{1}{-\sqrt{5}+1}\right)^{n}}{-\sqrt{5}+1} + \frac{2}{5}\,\frac{\sqrt{5}\left(-2\,\dfrac{1}{1+\sqrt{5}}\right)^{n}}{1+\sqrt{5}}$$

```
> normal(subs(n=7,a1),expanded);
```

21

Maple poate rezolva şi recurenţe tip `divide and conquer`:

```
> eq4 := s(n) = 3*s(n/2) + 5*n:
> rsolve(eq4,s);
```

$$\mathrm{s}(1)\,n^{\left(\frac{\ln(3)}{\ln(2)}\right)} + n^{\left(\frac{\ln(3)}{\ln(2)}\right)}\left(-15\left(\frac{2}{3}\right)^{\left(\frac{\ln(n)}{\ln(2)}+1\right)} + 10\right)$$

Exerciţii: :

```
> eq2 := f(n) = a*f(n-1) + f(n-2):
> rsolve(eq2,f);
> rsolve({f(n)= -3*f(n-1)-2*f(n-2), f(0)=1, f(1)=1}, f(n));
> rsolve(f(a*n)=b*f(n)+c,f(n));
```

Pentru rezolvarea ecuaţiilor neliniare se pot descrie procese iterative. De exemplu, considerăm metoda Newton de rezolvare a ecuaţiei neliniare $g(x) = 0$.

```
> Digits:=6: g := x -> x^3-x-3:
> plot(g(x),x=-2..2.0);
```

Grafic: vezi figura 3.35.a

Definim funcţia iterativă Newton:

```
> f := x -> x-g(x)/D(g)(x);
> x[0] := 1.0;
> iterates:=proc(g,p0,nmax)local i;seq((g@@i)(p0),i=0..nmax)end:
```

```
> iterates(f,x[0],5);
> points := NULL:
```

$$f := x \to x - \frac{\mathrm{g}(x)}{\mathrm{D}(g)(x)}$$

$$x_0 := 1.0$$

$$1.0, 2.50000, 1.92958, 1.70787, 1.67256, 1.67170$$

Generăm mulţimea de puncte de reprezentare a liniilor tangente:

```
> for n from 0 to 5 do  points := points,[x[n],0],[x[n],g(x[n])]:
>  x[n+1]:=f(x[n]): od: plot({[points],g(x)},x=-1.0..3.0,style=line);
```

Grafic: vezi figura 3.35.b

Aplicaţia 10: Fie f o funcţie şi fie a un punct din domeniul său. Dacă n este un întreg pozitiv pentru care $(f^n)(a) = a$, dar $(f^i)(a) \neq a$ pentru oricare $0 \leq i \leq n$, atunci a este numit punct periodic de perioadă n pentru f. Punctele periodice de perioadă unu sunt numite puncte fixe. Localizarea acetor puncte fixe se poate face intersectând graficul funcţiei $y = f(x)$ cu $y = x$. De exemplu, funcţia cosinus are un singur punct fix:

```
> fix := fsolve(cos(x)=x,x,0..Pi);
```

$$fix := .7390851332$$

Un punct fix de atracţie este un punct fix cu proprietatea că pentru punctele b apropiate de a,

```
> limit(b[n],n=infinity) = a;
```

$$\lim_{n \to \infty} b_n = a$$

unde $b_1 = b$ şi $b_n = f(b_{n-1})$ pentru $n = 2, 3, \ldots$. Următoarea procedură investighează punctele fixe şi punctele periodice ale unei funcţii:

```
> iterate := proc(f,n,x) local a,i,s;
>    a:=evalf(x); s:=a; for i to n do a:=f(a); s:=s,a od
> end:
```

De exemplu, pentru a investiga dacă un punct fix a funcţiei cosinus este atractiv sau nu, aplicăm această procedură:

```
> iterate(cos,10,fix-1),fix;
```

$$-.2609148668, .9661543793, .5684675409, .8427269503, .6654297419,$$
$$.7866515363, .7062199575, .7608204115, .7242705678, .7489829351,$$
$$.7323817612, .7390851332$$

În cazul dat punctul fix este atractiv.

Exerciţii:

a) Găsiţi punctele fixe ale funcţiei $y = cx(1-x)$, pentru $c = 2, 3, 4$. Care dintre ele sunt puncte de atracţie?

b) Determinaţi punctele de perioadă 2, 3 şi 4 ale funcţiei $y = cx(1-x)$, pentru $c = 2, 3, 4$. (Indicaţie: Punctele periodice de ordin doi ale lui f sunt puncte fixe ale lui $f \circ f$ care nu sunt puncte fixe ale lui f).

3.11 Optimizare

3.11.1 Puncte de extrem local

Există o legătură strânsă între prima şi a doua derivată ale unei funcţii şi graficul acesteia. Astfel punctele de extrem sunt puncte critice ale funcţiei, acele puncte

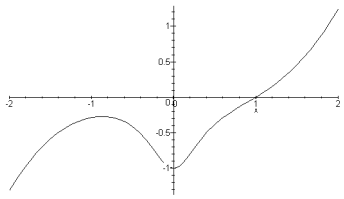


Figura 3.36:

în care prima derivată se anulează. Punctele de inflexiune se întâlnesc printre zerourile derivatei secunde.

De exemplu, se consideră funcţia:

```
> y := (x^5-1)/(6*x^2+1):
```

Se cere determinarea punctelor de extrem şi a celor de inflexiune. Din graficul funcţiei putem estima aceste valori:

```
> plot(y,x=-2..2);
```

Grafic: vezi figura 3.36

dar aproximarea poate fi inexactă.

```
> yp := simplify(diff(y,x));
```

$$yp := \frac{x\,(\,18\,x^5 + 5\,x^3 + 12\,)}{(\,6\,x^2 + 1\,)^2}$$

Punctele critice se obţin rezolvând ecuaţia $yp = 0$.

```
> cp := fsolve(numer(yp),x);
```

$$cp := 0, -.8657650194$$

Inspectând graficul aflăm că cp_1 este maxim local, iar cp_2 este minim local. Valorile de extrem local ale funcţiei sunt:

```
> subs(x = cp[1],y);
```

$$-1$$

```
> subs(x = cp[2],y);
```

$$-.2703889018$$

Determinăm punctele de inflexiune:

```
> ypp := simplify(diff(yp,x));
```

$$4\,\frac{54\,x^7 + 27\,x^5 + 5\,x^3 - 54\,x^2 + 3}{(\,6\,x^2 + 1\,)^3}$$

```
> pip := fsolve(numer(ypp),x);
```

$$pip := -.2324164053, .2392938071, .8746375912$$

```
> for i from 1 to 3 do print(pip[i], subs(x=pip[i],y)) od;
```

$$-.2324164053, -.7557396749$$
$$.2392938071, -.7437022342$$
$$.8746375912, -.08732685718$$

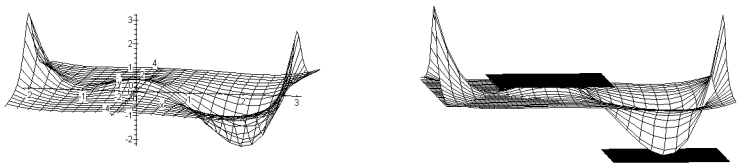


Figura 3.37:

Pentru determinarea punctelor de minim, maxim sau ale punctelor şa a funcţiilor de mai multe variabile se testează derivata secundă, dacă aceasta există. Se consideră funcţia:

```
> f:=(x,y) ->(3*x^4 -4*x^3 -12*x^2 +6)/(12*(1+y^2)):
> plot3d(f(x,y),x=-2..3,y=-4..4,style=patch,orientation=[-80,70],
>  axes=normal,labels=[x,y,z],color=[-y,-y,1]);
```

Grafic: vezi figura 3.37.a

Din acest grafic vedem că $f(x,y)$ are trei puncte critice, toate pe axa x: o şa aproape de (-1, 0), un maxim local aproape de (0, 0), şi un minim local aproape de (2,0). Încercăm să obţinem rezultatele exacte. În situaţii care presupun mai multe variabile, când vizualizarea este imposibilă, utilizarea formulelor este indicată.

Determinăm în primul rând punctele critice. Acestea sunt, prin definiţie, acolo unde derivatele de ordin întâi a lui $f(x,y)$ sunt zero.

```
> f1:= diff(f(x,y), x): f2:=diff(f(x,y), y);
> solve({f1=0, f2=0}, {x,y});
```

$$\{\, y = 0, x = 0 \,\}, \{\, y = 0, x = -1 \,\}, \{\, y = 0, x = 2 \,\}$$

Ca o alternativă la calculul derivatelor individuale, putem calcula vectorul gradient:

```
> grad(f(x,y), [x,y]);
```

$$\left[\frac{12\,x^3 - 12\,x^2 - 24\,x}{12 + 12\,y^2} \;-24\,\frac{(\,3\,x^4 - 4\,x^3 - 12\,x^2 + 6\,)\,y}{(\,12 + 12\,y^2\,)^2}\right]$$

```
> convert(",set);
```

$$\left\{\frac{12\,x^3 - 12\,x^2 - 24\,x}{12 + 12\,y^2}, -24\,\frac{(\,3\,x^4 - 4\,x^3 - 12\,x^2 + 6\,)\,y}{(\,12 + 12\,y^2\,)^2}\right\}$$

```
> solve(",{x,y});
```

$$\{\, y = 0, x = 0 \,\}, \{\, y = 0, x = -1 \,\}, \{\, y = 0, x = 2 \,\}$$

Verificăm grafic că (-1,0) este un punct şa trasând suprafaţa şi planul tangent (planul orizontal $z = f(-1,0)$ la acest punct.

```
> E1:=plot3d(f(x,y), x=-2..3, y=-4..4):
> E2:=plot3d(f(0,0), x=-1..1, y=-2..2):
> E3:=plot3d(f(-1,0), x=-2..-0, y=-4..4):
> E4:=plot3d(f(2,0), x=1..3, y=-2..2):
> display({E1, E2, E3,E4}, orientation=[-95, 80]);
```

Grafic: vezi figura 3.37.b

Calculăm apoi derivatele secunde.

```
> f11:=diff(f1,x): f12:=diff(f(x,y),x,y): f22:=diff(f(x,y),y,y):
> # sau D[2,2](f)(x,y): sau D[2,2](f);
> DDf:= matrix(2,2,(i,j)->D[i,j](f)):
> DDf(-1,0);
```

$$\begin{bmatrix} 3 & 0 \\ 0 & \dfrac{-1}{6} \end{bmatrix}$$

Deoarece aceasta este o matrice diagonală cu elemente de semn opus, concludem că (-1,0) este punct şa.

```
> DDf(0, 0);
```

$$\begin{bmatrix} -2 & 0 \\ 0 & -1 \end{bmatrix}$$

Deoarece matricea este negativ definită, (0,0) este maxim local.

```
> DDf(2,0);
```

$$\begin{bmatrix} 6 & 0 \\ 0 & \dfrac{13}{3} \end{bmatrix}$$

Deoarece matricea este pozitiv definită, (2,0) este punct de minim local.

3.11.2 Programare liniară

Maple are un pachet de rutine dedicat programării liniare.

```
> with(simplex);
```

$[basis, convexhull, cterm, define\_zero, display, dual, feasible, maximize,$
$minimize, pivot, pivoteqn, pivotvar, ratio, setup, standardize]$

De exemplu, considerăm un set de constrângeri:

```
> c1:={x<=-4,y<=3};
```

$$c1 := \{ x \leq -4, y \leq 3 \}$$

Testăm dacă setul de constrângeri are soluţie:

```
> feasible(c1);
```

true

Putem de asemenea verifica dacă soluţiile în fiecare parametru sunt pozitive.

```
> feasible(c1,NONNEGATIVE);
```

false

Maple utilizează metoda simplex pentru optimizare. Considerăm funcţia obiectiv:

```
> w := -x + y + 2*z;
```

$$w := -x + y + 2\,z$$

şi constrângerile

```
> eq1 := 3*x+4*y -3*z<= 23;
```

$$eq1 := 3\,x + 4\,y - 3\,z \leq 23$$

```
> eq2 := 5*x-4*y -3*z<= 10;
```

$$eq2 := 5\,x - 4\,y - 3\,z \leq 10$$

```
> eq3 := 7*x + 4*y + 11*z <= 30;
```
$$eq3 := 7\,x + 4\,y + 11\,z \leq 30$$

```
> c2 := { eq.(1..3)};
```
$$c2 := \{\,7\,x + 4\,y + 11\,z \leq 30, 5\,x - 4\,y - 3\,z \leq 10, 3\,x + 4\,y - 3\,z \leq 23\,\}$$

```
> maximize(w,c2);
```

Maple nu returnează nimic deoarece soluţia este nemărginită. Dacă adăugăm mai multe constrângeri, se poate obţine o soluţie finită:

```
> maximize(w,c2,NONNEGATIVE);
```
$$\left\{x = 0, z = \frac{1}{2}, y = \frac{49}{8}\right\}$$

sau

```
> maximize(w,c2 union {x>=0,y>=0,z>=0});
```
$$\left\{x = 0, z = \frac{1}{2}, y = \frac{49}{8}\right\}$$

În următorul exemplu, sistemul ce se rezolvă este degenerat, iar metoda simplex clasică necesită o serie de tehnici de pivotare pentru a ieşi din impas. Maple poate însă rezolva această problemă:

```
> z :=  2*x1 - x2 + 8*x3; obj:=z:
```
$$z := 2\,x1 - x2 + 8\,x3$$

```
> cnts1:=[2*x3<=1,2*x1-4*x2+6*x3<=3,-x1+3*x2+4*x3<=2];
```
$$cnts1 := [\,2\,x3 \leq 1, 2\,x1 - 4\,x2 + 6\,x3 \leq 3, -x1 + 3\,x2 + 4\,x3 \leq 2\,]$$

```
> sol1 := maximize(z,cnts1,NONNEGATIVE);
```
$$sol1 := \left\{x3 = 0, x2 = \frac{7}{2}, x1 = \frac{17}{2}\right\}$$

3.12 Rezolvarea ecuaţiilor diferenţiale

Considerăm pentru început o ecuaţie diferenţială ordinară cu ordin întâi: $y'(x)(3x^2 + 1) - 2x = 0$.

```
> de1:=diff(y(x),x)*(3*x^2+1)-2*x=0;
```
$$de1 := \left(\frac{\partial}{\partial x}\mathrm{y}(\,x\,)\right)\,(\,3\,x^2 + 1\,) - 2\,x = 0$$

```
> dsolve(de1,y(x));
```
$$\mathrm{y}(\,x\,) = \frac{1}{3}\ln(\,3\,x^2 + 1\,) + \_C1$$

Deoarece nu s-a dat nici o condiţie iniţială, Maple returnează răspunsul cu un parametru necunoscut. Dacă specificăm condiţia iniţială $y(0) = 0$.

```
> dsolve({de1,y(0)=0},y(x));
```
$$\mathrm{y}(\,x\,) = \frac{1}{3}\ln(\,3\,x^2 + 1\,)$$

Exerciţii:

```
> de2 := x^2 * diff(y(x),x) + y(x) = exp(x);
> dsolve(de2,y(x));
> de3 := diff(f(x),x$2) + 2*diff(f(x),x) + f(x) = x;
> dsolve(de3,f(x));
> de4 := diff(y(x),x) = (y(x)^2-3*x*y(x))/(y(x)^2-x^2);
> dsolve(de4,y(x));
> de5:=diff(y(t),t)+y(t)^2+(2*t+1)*y(t)+t^2+t+1=0;
> dsolve(de5,y(t));
```

Dacă Maple nu poate rezolva o ecuaţie, atunci putem cere o soluţie sub formă de serie. De exemplu, considerăm ecuaţia Van der Pol:

```
> de6:=diff(y(t),t$2)+mu*(y(t)^2-1)*diff(y(t),t)+y(t)=0;
```

$$de6 := \left(\frac{\partial^2}{\partial t^2}\,\mathrm{y}(t)\right) + \mu\,\left(\mathrm{y}(t)^2 - 1\right)\,\left(\frac{\partial}{\partial t}\,\mathrm{y}(t)\right) + \mathrm{y}(t) = 0$$

```
> dsolve(de6,y(t));
```

Maple este incapabil de a găsi o soluţie, astfel încât nu returnează nimic.

```
> dsolve({de6,y(0)=1,D(y)(0)=1},y(t),series);
```

$$\mathrm{y}(t) = 1 + t - \frac{1}{2}\,t^2 + \left(-\frac{1}{6} - \frac{1}{3}\,\mu\right)\,t^3 + \left(\frac{1}{6}\,\mu + \frac{1}{24}\right)\,t^4 + \left(+\frac{2}{15}\,\mu + \frac{2}{15}\,\mu^2 + \frac{1}{120}\right)\,t^5 + \mathrm{O}(t^6)$$

Maple poate rezolva sisteme de ecuaţii diferenţiale:

```
> de7:={ diff(y(t),t)=x(t)+1, diff(x(t),t)*a = y(t)-3 };
```

$$de7 := \left\{\frac{\partial}{\partial t}\,\mathrm{y}(t) = \mathrm{x}(t) + 1,\ \left(\frac{\partial}{\partial t}\,\mathrm{x}(t)\right)\,a = \mathrm{y}(t) - 3\right\}$$

```
> dsolve(de7,{x(t),y(t)});
```

$$\left\{\mathrm{y}(t) = 3 + \_C1\,\mathrm{e}^{\left(\frac{t}{\sqrt{a}}\right)}\,\sqrt{a} - \_C2\,\mathrm{e}^{\left(-\frac{t}{\sqrt{a}}\right)}\,\sqrt{a},\ \mathrm{x}(t) = -1 + \_C1\,\mathrm{e}^{\left(\frac{t}{\sqrt{a}}\right)} + \_C2\,\mathrm{e}^{\left(-\frac{t}{\sqrt{a}}\right)}\right\}$$

În anumite cazuri, putem specifica un algoritm care să fie utilizat în rezolvarea ecuaţiei. De exemplu, putem cere utilizarea transformării Laplace:

```
> de8:=diff(y(x),x$6) + 2*diff(y(x),x$4) + diff(y(x),x$2):
> dsolve(de8, y(x),laplace);
```

$$\begin{aligned}\mathrm{y}(x) = {} & \mathrm{y}(0) + D^{(4)}(y)(0) + 2\,D^{(2)}(y)(0) + x\,D^{(5)}(y)(0) + 2\,x\,D^{(3)}(y)(0) \\ & + x\,\mathrm{D}(y)(0) - \frac{3}{2}\,D^{(5)}(y)(0)\sin(x) - \frac{5}{2}\,D^{(3)}(y)(0)\sin(x) \\ & - D^{(4)}(y)(0)\cos(x) - 2\,D^{(2)}(y)(0)\cos(x) - \frac{1}{2}\,D^{(4)}(y)(0)\,x\sin(x) \\ & + \frac{1}{2}\,D^{(3)}(y)(0)\,x\cos(x) - \frac{1}{2}\,D^{(2)}(y)(0)\,x\sin(x) + \frac{1}{2}\,D^{(5)}(y)(0)\,x\cos(x)\end{aligned}$$

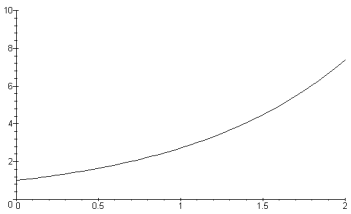


Figura 3.38:

Maple dispune de asemenea de o opţiune de determinare a unor soluţii aproximative prin metode numerice. Rezultatul este o procedură care evaluată într-un punct conduce la valoarea soluţiei (şi valorile derivatelor dacă ecuaţia are ordin mai mare decât 1). Algoritmul implicit utilizat este bazat pe metoda Runge-Kutta-Fehlberg 4-5.

```
> de9:={diff(y(t),t$3)-2*diff(y(t),t$2)+y(t)};
```

$$de9 := \left\{ \left(\frac{\partial^3}{\partial t^3} \mathrm{y}(t) \right) - 2 \left(\frac{\partial^2}{\partial t^2} \mathrm{y}(t) \right) + \mathrm{y}(t) \right\}$$

```
> ic9:={y(0)=1,D(y)(0)=1,(D@@2)(y)(0)=1};
```

$$ic9 := \{ \mathrm{y}(0) = 1, \mathrm{D}(y)(0) = 1, D^{(2)}(y)(0) = 1 \}$$

```
> s9:=dsolve(de9 union ic9,y(t),numeric);

s9 := proc(rkf45_x) ... end

> s9(1.);
```

$$\left[t = 1., \mathrm{y}(t) = 2.718281811394141, \frac{\partial}{\partial t} \mathrm{y}(t) = 2.718281811394141, \right.$$
$$\left. \frac{\partial^2}{\partial t^2} \mathrm{y}(t) = 2.718281811394141 \right]$$

Valorile oferite de procedura numerică pot fi vizualizate cu `odeplot`:

```
> with(plots): odeplot(s9,{t,y,yp},0..2,0..10);
```

Grafic: vezi figura 3.38

Soluţia particulară a unei ecuaţii diferenţiale neomogene se poate afla dacă se consideră o bază pentru spaţiul soluţiilor ecuaţiei omogene. Să găsim, de exemplu, funcţia $y(x)$ care safisface $y" + y = \tan(x)$ şi $y(0) = 1$, $y'(0) = -1$.

```
> restart;
> eq := (D@@2)(y)(x) + y(x) - tan(x):
```

Se consideră două soluţii linear independente (date de ecuaţia caracteristică, în cazul nostru $\sin(x)$ şi $\cos(x)$) şi se formează o soluţie particulară y_p ca o combinaţie lineară cu coeficienţi neconstanţi.

```
> yp := u1*cos + u2*sin:
```

Avem o multitudine de soluţii. Presupunem că $u_1' \cos + u_2' \sin = 0$.

```
> cond1 :=   'D'(u1) *cos = - 'D'(u2)*sin :
> ypp := subs({cond1},D(yp)):
```

O a doua condiţie asupra lui y_p este aceea de a satisface ecuaţia:

```
> cond2 := D(ypp) + yp - tan:
```

Se obţine un sistem liniar de ecuaţii în derivatele celor două funcţii necunoscute u_1 şi u_2.

```
> sol := solve({cond1,cond2},{D(u1),D(u2)});
```

$$sol := \left\{ \mathrm{D}(u1) = -\frac{\tan\sin}{\cos^2 + \sin^2}, \mathrm{D}(u2) = \frac{\cos\tan}{\cos^2 + \sin^2} \right\}$$

```
> sol2:=subs(cos^2+sin^2=1,sol);
```

$$sol2 := \{ \mathrm{D}(u1) = -\tan\sin, \mathrm{D}(u2) = \cos\tan \}$$

```
> u1p := rhs(sol2[1]);
```

$$u1p := -\tan\sin$$

```
> u2p := rhs(sol2[2]);
```

$$u2p := \cos\tan$$

```
>  u1 := unapply(int(u1p(x),x),x);
```

$$u1 := x \to \sin(x) - \ln(\sec(x) + \tan(x))$$

```
> u2 := unapply(int(u2p(x),x),x);
```

$$u2 := -\cos$$

```
> c1 := x->c1:  c2 := x->c2:
> yh :=c1*sin+c2*cos;
```

$$yh := c1\sin + c2\cos$$

Soluţia generală a ecuaţiei este:

```
> yg := (yp + yh);
> yg(x);
```

$$yg := u1\cos - \cos\sin + c1\sin + c2\cos$$

$$(\sin(x) - \ln(\sec(x) + \tan(x)))\cos(x) - \cos(x)\sin(x) + c1\sin(x) + c2\cos(x)$$

Verificare:

```
> simplify(eval(subs(y=yg,eq)),trig);
```

$$0$$

```
> sol := solve({yg(0)=2,D(yg)(0)=-3},{c1,c2});
```

$$sol := \{ c2 = 2, c1 = -2 \}$$

```
> subs(sol,yg(x));
```

$$(\sin(x) - \ln(\sec(x) + \tan(x)))\cos(x) - \cos(x)\sin(x) - 2\sin(x) + 2\cos(x)$$

Exerciţii:

```
> restart; deqn:=2*diff(y(x),x$2)+3*diff(y(x),x)-5*y(x)=0;
> dsolve(deqn,y(x));
> initconds:= y(0)=0, y(1)=1;
> dsolve({deqn,initconds},y(x));
> simplify(");
> assign(");
```

```
> deqn;
> simplify(");
> y(x):='y(x)';
> noiec:=4*diff(y(x),x$2)-diff(y(x),x)+3*y(x)=0:
> noicond:=y(0)=0, D(y)(0)=1:
> dsolve({noiec,noicond},y(x));
> dsolve({noiec,noicond},y(x),series);
> dsolve({noiec,noicond},y(x),laplace);
> F:=dsolve({noiec,noicond},y(x),numeric);
> F(0),F(1),F(Pi),F(10);
> restart; eq:=diff(x(t),t)=x(t)*(2-x(t));
> dsolve(eq,x(t));
> phi:=solve(",x(t));
> subs(_C1=-exp(-2*C)/2,phi);
> dsolve(eq,x(t),explicit);
> tt:=unapply(rhs("),t);
> eval(subs(x=tt,eq));
> simplify(");
> dsolve({eq,x(0)=3},x(t),explicit);
> sol:=rhs(");
> plot(sol,t=0..2);
> restart; eqs:=diff(x(t),t)=x(t)*y(t),diff(y(t),t)=2-x(t):
> dsolve({eqs,x(0)=1,y(0)=1},{x(t),y(t)});
> s:=dsolve({eqs, x(0)=1, y(0)=1},{x(t),y(t)},numeric);
> sc:=seq([rhs(s(t/10)[2]),rhs(s(t/10)[3])],t=0..100):
> plot([sc]);
> restart;
> eqs:= diff(x(t),t)=-x(t)+x(t)*y(t)/100, diff(y(t),t)=
>       2*y(t)-2*x(t)*y(t)/25: dsolve({eqs},{x(t),y(t)});
> sol:=dsolve({eqs, x(0)=25, y(0)=60},{x(t),y(t)},numeric);
> s:=seq([rhs(sol(t/10)[2]),rhs(sol(t/10)[3])],t=0..50):
> plot([s]);
> restart; with( DEtools );
> ODE := diff( y(t), t ) = sin(t) - y(t)/2;
> DEplot(ODE,[t,y],t=0..12,y=-3..3,arrows=THIN);
> IC := { seq( [0,i], i=-3..3 ) };
> DEplot(ODE,[t,y],t=0..12,IC,arrows=THIN);
> dsolve( ODE, y(t) );
> dsolve( { ODE, y(0)=1/2 }, y(t) );
> F1 := 2; F2 := 2 + Dirac(t-Pi);
> F3 := 2 + 2*sin(t-Pi)*(Heaviside(t-Pi) - Heaviside(t-3*Pi));
> ODE1 := diff(y(t),t$2) + 5*diff(y(t),t) + 6*y(t) = F1:
> ODE2 := diff(y(t),t$2) + 5*diff(y(t),t) + 6*y(t) = F2:
```

```
> ODE3 := diff(y(t),t$2) + 5*diff(y(t),t) + 6*y(t) = F3:
> IC := y(0)=1, D(y)(0)=2:
> y1 := rhs( dsolve( { ODE1, IC }, y(t) ) );
> y1L := rhs( dsolve( { ODE1, IC }, y(t), method=laplace) );
> y2 := rhs( dsolve( { ODE2, IC }, y(t) ) );
> y2L := rhs( dsolve( { ODE2, IC }, y(t), method=laplace) );
> y3 := rhs( dsolve( { ODE3, IC }, y(t) ) );
> y3L := rhs( dsolve( { ODE3, IC }, y(t), method=laplace) );
> map(evalb,[y1=y1L,y2=y2L,y3=y3L]);
> evalb(simplify(subs(y(t)=y2,ODE2)));
> evalb(simplify(subs(y(t)=y2L,ODE2)));
> evalb(simplify(subs(y(t)=y3,ODE3)));
> evalb(simplify(subs(y(t)=y3L,ODE3)));
> restart; with( DEtools ):
> equations:=[diff(h(t),t)=(0.1)*h-(0.005)*h*(1/60)*u,
>             diff(u(t),t) = (0.00004)*h*u-(0.04)*u ];
> vars:= [t, h, u];  # sau  vars:= [h(t) , u(t) ] ;
> init1:=[0,2000,600]:init2:=[0,2000,1200]:init3:=[0,2000,3000]:
> domain := 0 .. 320:
> DEplot(equations,vars,t=0..160,{init1,init2,init3},
>   stepsize= 0.5, scene= [t,h,u]);
```

3.13 Aplicaţii diverse: encriptare

Construim o regulă de convertire a literelor şi numere. De exemplu, se utilizează numai litere mici şi transformarea $a \to 1, \ldots, z \to 26$ iar spaţiul $\to 27$.

```
> to_number:=proc(st)
> local ll,nn,ss,ii,num;
> num:=table(['a'=1,'b'=2,'c'=3,'d'=4,'e'=5,'f'=6,'g'=7,'h'=8,'i'=9,
> 'j'=10,'k'=11,'l'=12,'m'=13,'n'=14,'o'=15,'p'=16,'q'=17,'r'=18,
> 's'=19,'t'=20,'u'=21,'v'=22,'w'=23,'x'=24,'y'=25,'z'=26,'` `'=27]):
> if not type(st,string)
>  then ERROR(`wrong number (or type) of arguments`) fi;
> ll:=length(st);
> if ll= 0 then RETURN(0) fi;
> nn:=1;
> for ii from 1 to ll do
>   ss:=num[substring(st,ii..ii)];
>   if(not type(ss,numeric))
>    then ERROR(`wrong number (or type) of arguments`) fi;
>   nn:=100*nn+ss;
> od;
> nn-10^(2*ll);
```

```
> end:
> from_number:=proc(nn)
>   local ss,mm,ll,ii,ans,a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,
>    u,v,w,x,y,z,` `,alpha;
> alpha:=table([1=a,2=b,3=c,4=d,5=e,6=f,7=g,8=h,9=i,10=j,11=k,
> 12=l,13=m,14=n,15=o,16=p,17=q,18=r,19=s,20=t,21=u,22=v,23=w,
> 24=x,25=y,26=z,27=` `]):  mm:=nn;
>   if(not type(nn,integer))
>    then ERROR(`wrong number (or type) of arguments`) fi;
>    ll:=floor(trunc(evalf(log10(mm)))/2)+1;
>    ans:=``;
>    for ii from 1 to ll do
>     mm:=mm/100;
>     ss:=alpha[frac(mm)*100];
>     if(not type(ss,string))
>      then ERROR(`wrong number (or type) of arguments`) fi;
>      ans:=cat(ss, ans);
>     mm:=trunc(mm)
>   od;
>   ans;
> end:
> to_number('maple');
```

$$1301161205$$

```
> from_number(805121215);
```

$$hello$$

Se selectează două numere prime mari.

```
> p:=nextprime(7347124781320478301247);
```

$$p := 73471247813204783013 77$$

```
> q:=nextprime(47857483902754238905578423553478 2043);
```

$$q := 47857483902754238905578423553478 2067$$

Funcţia ϕ a lui Euler este numărul de întregi mai mici sau egali cu argumentul n care sunt coprimi cu n. Dacă $n = pq$ atunci $\phi(n) = (p-1)(q-1)$.

```
> n:=p*q;
```

$$n := 351614905953571547965301353965020031859090376703404100625 9$$

```
> phi_n:=(p-1)*(q-1);
```

$$phi\_n := 351614905953571547965253496481117276885472320147802792 2816$$

Selectăm un întreg e, între 2 şi $\phi(n)$, care este coprim cu $\phi(n)$. Acest număr poate fi ales ca număr prim. Verificăm dacă nu este divizor a lui ϕ_n.

```
> e:=nextprime(432432);
```

$$e := 432433$$

```
> evalf(phi_n/e);
```

$$.8131084028\,10^{52}$$

Trebuie rezolvată ecuaţia liniară diofantică $ed + \phi_n k = 1$

```
> igcdex(e,phi_n,'d','k');
```

$$1$$

```
> d:=d mod phi_n;
```

$d := 21859037125338838607169895271523727582527739613372306937 13$

Numerele n şi e sunt furnizate drept chei publice, iar d este secret şi utilizat în decriptare.

Pentru codarea mesajului, utilizatorul transformă mesajul într-un întreg M şi îl împarte în bucăţi mai mici sau egale decât n. Utilizatorul calculează apoi restul împărţirii M^e/n.

```
> M:=to_number('hi');
```

$$M := 809$$

```
> C:=Power(M,e) mod n;
```

$C := 44530624788417878422706927513001432735217596804936374221 7$

Pentru decodare utilizăm d. Calculăm $C^d \bmod n$.

```
> Power(C,d) mod n;
```

$$809$$

```
> from_number(");
```

$$hi$$

Un alt exemplu:

```
> M:=to_number(`i think therefore i am`);
```

$$M := 927200809141127200805180506151805270927011 3$$

```
> C:=Power(M,e) mod n;
```

$C := 113076808796295564761111574599273133873595148813346030238 3$

```
> Power(C,d) mod n;
```

$$927200809141127200805180506151805270927011 3$$

```
> from_number(");
```

$$i\ think\ therefore\ i\ am$$

Capitolul 4

Probleme rezolvate

4.1 Enunţuri

1. (*Ecuaţia de grad trei*) Rezolvaţi ecuaţia în x: $x^3 - ax + 1 = 0$. Determinaţi soluţiile particulare pentru $a = 1, 2, \ldots$. Trasaţi polinomul de gradul trei care apare în ecuaţie într-un caz în care ecuaţia are o rădăcină reală şi într-un caz în care ecuaţia are toate rădăcinile reale.

2. (*Ecuaţie neliniară*) Găsiţi toate soluţiile reale ale ecuaţiei $7\cos x + x + x^2 = 15$.

3. (*Sistem neliniar 1*) Găsiţi toate soluţiile reale ale sistemului de ecuaţii

$$\begin{cases} x^2 + y^2 = 4 \\ \sin(x+y) + \cos(x) = 1 \end{cases}.$$

4. (*Sistem neliniar 2*) Evaluaţi $\sin(xy + z)$ cu 5 cifre exacte în cazul în care (x, y, z) este unica soluţie reală a sistemului de ecuaţii

$$\begin{cases} x + y + z = 1 \\ 3x + 2y - z = 5 \\ xy + 7z^3 = 0 \end{cases}.$$

5. (*Sistem neliniar 3*) Găsiţi toate soluţiile reale ale sistemului

$$\begin{cases} x^2 + y^2 + z^2 = 4 \\ x + y + z = 0 \\ x\sin(yz) = -1 \end{cases}.$$

6. (*Sistem neliniar 4*) Găsiţi toate soluţiile reale ale sistemului

$$\begin{cases} x^2 + y^2 = 9, \\ x^3 + y^3 - \sin(xy) = 7 \end{cases}.$$

7. (*Valori complexe*) Reprezentaţi în planul complex valorile proprii ale unei matrice.

8. (*Pătrat magic*) Creaţi o procedură care verifică dacă o matrice oarecare este un pătrat magic (adică sumele elementelor de pe fiecare linie, fiecare coloană şi diagonala principală sunt egale). Verificaţi procedura creată pe următoarele trei matrice:

$$\begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{bmatrix}, \quad \begin{bmatrix} 1 \end{bmatrix}, \quad \begin{bmatrix} 92 & 99 & 1 & 8 & 15 & 67 & 74 & 51 & 58 & 40 \\ 98 & 80 & 7 & 14 & 16 & 73 & 55 & 57 & 64 & 41 \\ 4 & 81 & 88 & 20 & 22 & 54 & 56 & 63 & 70 & 47 \\ 85 & 87 & 19 & 21 & 3 & 60 & 62 & 69 & 71 & 28 \\ 86 & 93 & 25 & 2 & 9 & 61 & 68 & 75 & 52 & 34 \\ 17 & 24 & 76 & 83 & 90 & 42 & 49 & 26 & 33 & 65 \\ 23 & 5 & 82 & 89 & 91 & 48 & 30 & 32 & 39 & 66 \\ 79 & 6 & 13 & 95 & 97 & 29 & 31 & 38 & 45 & 72 \\ 10 & 12 & 94 & 96 & 78 & 35 & 37 & 44 & 46 & 53 \\ 11 & 18 & 100 & 77 & 84 & 36 & 43 & 50 & 27 & 59 \end{bmatrix}$$

9. (*Metoda trapezelor*) Determinaţi valoarea aproximativă a integralei $\int_0^1 \sin(x^3)dx$ utilizând metoda trapezelor.

10. (*Integrare prin părţi*) Integraţi prin părţi $\int x^3 e^x dx$.

11. (*Dreaptă 3D*) Determinaţi punctul de intersecţie a liniei din spaţiu care trece prin $A = (2, 1, 9)$, $B = (-3, 0, 1)$ cu planul xy.

12. (*Paralelogram*) Demonstraţi utilizând vectori că patrulaterul format din mijloacele laturilor unui patrulater convex oarecare din plan este un paralelogram.

13. (*Distanţa*) Determinaţi distanţa de la punctul $X = (3, 2, 5)$ la dreapta ce trece prin $A = (2, 0, 0)$ şi $B = (0, 0, 4)$.

14. (*Polinom*) Se consideră expresia

$$(ax^2 + bx \sin y + c \sin y)^2 + (a \sin^2 y + bx)^3.$$

a) Rescrieţi expresia ca polinom în x şi determinaţi coeficientul lui x^2.

b) Rescrieţi expresia ca multipolinom în variabilele x şi $\sin y$ şi determinaţi coeficientul lui $x \sin^2 y$.

15. (*Şir în oglindă*) a) Construiţi o procedură de inversare a unei liste.

b) Construiţi o procedură de transformare a unui şir de caractere într-o listă de simboluri. Construiţi şi procedura se efectuează operaţia inversă.

c) Utilizând procedurile construite anterior, inversaţi un şir de caractere. Efectuaţi un test asupra şirului 'Acesta este un exemplu'. Verificaţi corectitudinea rezultatului după ce se execută $x := 2$ şi, eventual, stabiliţi modificările ce trebuie aduse secvenţei de comenzi.

16. (*Calculul valorii π*) În calculul unei aproximaţii a constantei π, poate fi utilizată seria Gregory

$$4 \sum_{n=1}^{N} \frac{(-1)^{n+1}}{2n-1}$$

dacă N este suficient de mare. Care este eroarea absolută şi cea relativă produse prin aceasta aproximare, dacă $N = 50000$?

17. (*Triunghi*) Rotiţi triunghiul cu vârfurile în (0,0), (2,0), (1,3) cu 90 grade în sensul ceasornicului în jurul centrului său de greutate. Care sunt noile coordonate?

18. (*Plan*) Care este ecuaţia planului care este paralel cu xOy şi care trece prin intersecţia dreptelor AB şi CD, unde $A = (0,0,0)$, $B = (2,2,2)$, $C = (3/5, 1/3, 7/8)$, $D = (9/5, 7/3, 5/4)$?

19. (*Primitive grafice 2D*) Construiţi proceduri pentru trasarea următoarelor obiecte bidimensionale: linie, poligon, cerc, elipsă, arc de cerc.

20. (*Serii Taylor*) Comparaţi graficul funcţiei $\sin x$ cu cele ale expansiunii Taylor a lui $\sin x$ în $x = 0$ cu ordinul 1,3,5,7 pe intervalul $-\pi < x < \pi$.

21. (*Tangenta*) Se consideră funcţiile $f(x) = x^2 - x + 1$ şi $g(x, y) = x^3 - x + 3y^2$. Trasaţi dreapta, respectiv planul tangent într-un punct oarecare al funcţiei f, respectiv g.

22. (*Cilindrii*) Desenaţi corpul obţinut prin intersecţia a doi cilindrii cu axe perpendiculare şi cu aceeaşi rază a bazei.

23. (*Corp compus*) Desenaţi un solid compus dintr-un cilindru circular vertical cu o pălărie conică şi, baza, o semisferă. Consideraţi razele şi înălţimile unitare.

24. (*Tor*) Scrieti ecuaţia parametrică a unui tor şi trasaţi-l în condiţiile în care dacă datele de intrare sunt: r raza cercului (de secţiune) generator; prin rotaţie centrul acestui cerc descrie un cerc în planul xOy cu centrul în (x_0, y_0, z_0) şi raza d.

25. (*Transformări liniare în R^2*) Ca obiect asupra căruia se vor face transformările se consideră faţada unei case.

a) Trasaţi simetricul obiectului faţă de axa verticală;

b) Concepeţi o procedură generală de aplicare a unei transformări liniare;

c) Construiţi o procedură pentru generarea unei secvenţe animate care prezintă o trecere lentă de la un obiect iniţial la cel obţinut printr-o transformare liniară. Verificaţi procedura pentru simetrica faţadei faţă de axa verticală.

26. (*Transformări liniare în R^3*) Daţi vectorii liniar independenţi a_1 la a_n din R^m, găsiţi matricea care proiectează R^m în subspaţiul generat de vectorii a_i.

a) În particular, găsiţi proiecţia P a lui R^3 în planul generat de $a_1 = (1,2,3)$ şi $a_2 = (3,0,1)$.

b) Dat un obiect din R^3 ce reprezintă o casă,

(i) concepeţi o procedură de translatare a obiectul cu un deplasament oarecare;

(ii) construiţi proiecţia paralelă a obiectului pe planul generat de $a_1 = (1,2,3)$ şi $a_2 = (3,0,1)$;

(iii) trasaţi obiectul obţinut prin simetrie faţă de planul generat de a_1 şi a_2 de la (b).

27. (*Puncte critice 1*) Găsiţi şi clasificaţi punctele critice ale funcţiei de patru variabile

$$h(x, y, z, w) = (x^2 - y^2 + 2z^2 + 3w^2)e^{-x^2-y^2-z^2-w^2}$$

28. (*Puncte critice 2*) Găsiţi şi clasificaţi punctele critice ale funcţiei de trei variabile $P(x, y, z) = 1 - 2x + 3x^2 - xy + xz - z^2 + 4z + y^2 + 2yz$.

29. (*Puncte critice 3*) Discutaţi punctele critice ale funcţiei $g(x, y) = x^2 + c(x - y)^4$ pentru diferite valori ale lui c.

30. (*Ceaşca*) O ceaşcă este realizată dintr-un cilindru cu fund şi fără capac. Volumul ceştii este de 192π cm^3. Dacă materialul pentru fundul ceştii costă 3 unităţi monetare pentru un cm^2, iar pentru restul ceştii 1 unitate monetară per cm^2, care sunt dimensiunile ceştii cu cost minim?

31. (*Maxmin*) Găsiţi valorile extreme ale funcţiei

$$6\cos 2x + \frac{5x\sin x}{x+1}$$

în intervalul [5,9] cu 6 zecimale exacte.

32. (*Suma minimă*) Se consideră m puncte date în plan $P_1, \ldots, P_m$. Să se determine linia $y = ax+b$ pentru care suma pătratelor distanţelor de la cele m puncte la dreaptă este minimă. Care sunt cerinţele şi soluţiile în cazul tridimensional?

33. (*Jacobian*) Fiind dat $T : R^m \to R^n$, construiţi matricea derivatelor parţiale. Caz paticular: $T(x, y) = (xy, x^2 + y^2, 0)$.

34. (*Continuitate*) Studiaţi continuitatea lui x^2 în $x = 2$ urmărind definiţia continuităţii unei funcţii într-un punct.

35. (*Împrumut*) Presupunem că împrumutăm o sumă de A unităţi monetare pentru T ani de la o bancă cu dobândă anuală r (procente). Dacă se va plăti datoria într-o rată anuală constantă k, care trebuie să fie valoarea lui k în termenii lui A, T şi r?

36. (*Haos*) Se consideră ecuaţia cu diferenţe $x_{n+1} = f(x_n)$, unde $f(x) = rx(1 - x)$, iar r este un număr fixat, numit factorul ratei de creştere. Pornind de la o valoare iniţială x_0, ecuaţia poate fi utilizată pentru a genera şirul $x_0, x_1, \ldots$ prin iteraţii.

Diagrama scării asociată cu şirul $x_0, x_1, \ldots$ este diagrama în care punctele $(x_0, 0), (x_0, x_1)$, (x_1, x_1), (x_1, x_2), (x_2, x_2), $\ldots$, (x_n, x_n), (x_n, x_{n+1}), $\ldots$ sunt unite prin linii.

a) Ecuaţia de mai sus este utilizată pentru modelarea schimbării populaţiei în generaţii succesive. Experimental s-a stabilit că rata de creştere a populaţiei umane este $r \approx 3.57$. Trasaţi diagrama pentru această valoare a ratei de creştere.

b) Studiaţi influenţa ratei de creştere asupra diagramei.

37. (*Bila de biliard*) Se consideră o masă pătratică de biliard. Două bile sunt aşezate pe suprafaţa mesei în coordonate bine precizate. Care sunt direcţiile în care prima bilă trebuie lovită astfel încât să lovească marginea mesei şi apoi cea de a doua bilă? Desenaţi traiectoriile corespunzătoare. Cum se modifică soluţia dacă masa este circulară?

38. (*Trei bile*) O bilă grea (masă 90 grame) este aruncată cu viteza de 2 m/s spre alte două bile aflate la 4 m. La impact, una din bile face un unghi de 45 de grade cu orizontala. Viteza sa este de 3 m/s si masa de 20 grame. O altă bilă, cu masa de 25 grame, se îndreaptă într-o direcţie care face cu orizontala un unghi θ. Viteza sa este de 5 m/s.

a) Calculaţi viteza bilei grele după impact în funcţie de θ.

b) Cum se va schimba traiectoria bilei grele când θ creşte de la 0 la 90 de grade. Cum se va schimba şi viteza?

c) Modelaţi mişcarea bilelor când $\theta = 30$ grade.

39. (*Transformarea conformă a unui cerc*) Fie cercul descris prin $z(t) = re^{it}$ şi

p soluţia ecuaţiei diferenţiale

$$z^2 p" + zp' + (\alpha z^2 + \beta z + \gamma)p = 0$$

cu condiţiile iniţiale $p'(0) = p_0'$ şi $p(0) = p_0$. Această funcţie mapează $z(t)$ în planul complex prin $P(t) = p(z(t)) = U(t) + iV(t)$.

Determinaţi funcţia p şi trasaţi-i graficul pentru următoarele valori numerice: $r = 1,\ U(0) = -0.563,\ U'(t) = 0,\ V(0) = 0,\ V'(t) = 0.869,\ 0 \leq t \leq 6\pi,\ \alpha = 1, \beta = 0.5, \gamma = -4/9$.

40. (*Compresiunea metalului*) Se consideră o bucată de metal presată între două suprafeţe plane. Feţele laterale se deformează. Se cere determinarea distorsiunii pentru a preciza spaţiul necesar elementului deformat.

Se consideră cazul particular al unui disc de raza r_0 şi înălţime h_0. Se notează cu R_{max} raza maximă a discului deformat, h_1 înălţimea sa şi r_1 cea mai mică rază a discului deformat (de la suprafaţa de contact cu planele presoare). Se consideră că materialul este incompresibil, adică volumul acestuia este constant, independent de forma materialului (se notează cu V volumul şi p presiunea, deci $dV/dp = 0$).

Din datele experimentale rezultă că verticala discului la marginea acestuia se transformă prin compresiune într-o parabolă (fie y axa verticală, x axa orizontală, $R = ay^2 + by + c$ parabola). Experimental s-a stabilit că $r_1 = r_0[1 + k(\sqrt{h_0/h_1} - 1)]$, unde $0 \leq k \leq 1$ este coeficientul de fricţiune. Din condiţiile $x(0) = c = r_1,\ x(h_1) = ah_1^2 + bh_1 + c = r_1,\ V = \pi \int_0^{h_1} R(x)dx,\ V_0 = \pi r_0^2 h_0, V = V_0$ rezultă a, b, c.

Simulaţi succesiunea de transformări suferite de disc (desen) pentru diferite valori ale lui k dacă $r_0 = 1,\ h_0 = 3,\ h_1 = 1$.

41. (*Mingea de tenis*) Se consideră o minge de tenis cu masa m şi diametrul d. Mingea este aruncată cu viteza v_0 de la o înălţime h.

Să se determine traiectoria mingii în cazurile:

(a) condiţii de vid (fără frecare cu aerul);

(b) în condiţiile frecării cu aerul.

Pentru a construi modelul matematic, se consideră că mingea se deplasează în planul $x0z$. În cazul (a) ecuaţiile mişcării sunt $x" = 0,\ z" = -g$, unde g este acceleraţia gravitaţională, iar în cazul (b) ecuaţiile mişcării sunt $x" = -C_d \alpha v x',\ z" = -g - C_d \alpha v z'$ unde $v = \sqrt{x'^2 + z'^2}$ este viteza la un moment dat, iar $\alpha = (\rho \pi d^2)/(8m)$ unde ρ este densitatea aerului. Conditiile iniţiale sunt: $x(0) = 0,\ z(0) = h,\ x'(0) = v_0 \cos V,\ z'(0) = v_0 \sin V$, unde V este unghiul dintre v_0 şi axa x. Se va lua în considerare cazul în care $h = 1,\ v = 25,\ V = 15,\ g = 9.81,\ d = 0.063,\ m = 0.05,\ \rho = 1.29,\ C_d = 0.508$.

42. (*Balistică 1*) Se presupune că un proiectil de lungime L şi densitate ρ_p, construit dintr-un material cu duritatea Y_p, atinge cu viteza v_0 suprafaţa unui solid cu densitatea ρ_t şi rezistenţa R_t. Viteza de penetrare a vârfului proiectilului, u, este dată de relaţia

$$\frac{1}{2}\rho_p (v - u)^2 + Y_p = \frac{1}{2}\rho_t u^2 + R_t$$

unde v este viteza capătului opus al proiectilului. Proiectilul este decelerat de forţa de frecare care depinde de lungimea acestuia $l = l(t)$:

$$\rho_p l \frac{dv}{dt} = -Y_p.$$

Schimbarea lungimii proiectilului este consecinţa procesului de eroziune:

$$\frac{dl}{dt} = -(v - u).$$

Ecuaţiile de mai sus constituie un sistem în necunoscutele $l = l(t),\ v = v(t),\ u = u(t)$. Se consideră condiţiile iniţiale $v(0) = v_0,\ l(0) = L$.

Să se traseze graficul dependenţei vitezei de penetrare funcţie de timp şi viteza iniţială pentru $\rho = \rho_t = \rho_p = 7810,\ L = 0.25,\ R_t = Y_p = 9 \cdot 10^8,\ t \in [0, 0.0004],\ v_0 \in [1000, 2500]$.

43. (*Balistică 2*) Un proiectil cilindric cu secţiunea A_p, diametru d_p, lungimea l_p, masa iniţială m_p, densitate ρ_p şi duritatea Y_p loveşte cu viteza v_0 o ţintă cu duritatea materialului Y_t şi densitatea ρ_t. Prin eroziune proiectilul pierde din masa sa. Fie $P(t)$ poziţia capului proiectilului (adâncimea de penetrare) funcţie de timp. Ecuaţia acesteia este

$$(m_p - \mu_p P)k\frac{d^2P'}{dt^2} = -\left[a + c\left(\frac{dP}{dt}\right)^2\right],$$

unde $c = c(\mu_p) = b - a_p/u_p^2,\ a_p = 1/(2\rho_p A_p),\ a = 3Y_t A_p,\ k = k(\mu_p) = 1 + \mu_p a_p$ iar μ_p este soluţia ecuaţiei $P_f(\mu) = p_f$, unde p_f este adâncimea finală de penetrare, iar

$$P_f(\mu_p) = m_p/\mu_p[1 - (1 + c(\mu_p)/av_p^2)^{-\epsilon(\mu_p)}]$$

cu $\epsilon(\mu_p) = \mu_p k/(2c)$, iar v_p este viteza din modelul anterior $v_p := v_0/2 + (Y_t - Y_p)/(\rho_t v_0)$.

Din ecuaţia diferenţială de mai sus, se obţine viteza de penetrare

$$P' = \sqrt{\frac{a}{c}\left[\left(1 + \frac{c}{a}v_p^2\right)\left(1 - \frac{\mu_p}{m_p}P\right)^{1/\epsilon} - 1\right]}$$

şi din $P' = 0$ la t_f, ecuaţia pentru $P_f(\mu_p)$.

Se consideră condiţii iniţiale $P(0) = 0,\ P'(0) = v_p$ şi valori numerice de test $m_p = 1.491,\ v_0 = 1457.9,\ \rho_t = \rho_p = 7810,\ d_p = 0.0242,\ Y_t = 9.7 \cdot 10^8,\ Y_p = 15.81 \cdot 10^8,\ p_f = 0.245$.

a) Trasaţi graficul adâncimii de penetrare P_f funcţie de μ pe intervalul [0,6].

b) Determinaţi μ_p.

c) Trasaţi graficul vitezei de penetrare P' funcţie de adâncimea P.

d) Trasaţi graficul adâncimii de penetrare P funcţie de timp pe intervalul [0,0.0006].

e) Trasaţi graficul adâncimii de penetrare P_f funcţie de viteza iniţială v_0 şi Y_t, pentru $v_0 \in [1400..1600],\ Y_t \in [900 \cdot 10^6, 1200 \cdot 10^6]$.

44. (*Poligoane ortogonale*) Polinoamele $p_0(x),\ p_1(x), \ldots$ se numesc polinoame ortogonale relativ la un anumit produs scalar $(\cdots, \cdot)$, dacă sunt îndeplinite următoarele condiţii:

(i) $p_k(x)$ este de grad k;

(ii) polinomul $p_k(x)$ satisface condiţia de ortogonalitate $(p_k, p) = 0$ pentru toate polinoamele $p(x)$ de grad mai mic decât k;

(iii) polinomul $p_k(x)$ satisface condiţia de normalitate $(p_k, p_k) = 1$.

Polinoamele ortogonale $p_0(x), p_1(x), \ldots$ satisfac relaţia de recurenţă

$$xp_{k-1} = \beta_{k-1}p_{k-2} + \alpha p_{k-1} + \beta_k p_k, \ k = 1, 2, \ldots$$

unde $\beta_0 = 0$, $p_{-1}(x) = 0$, $p_0(x) = \pm\sqrt{1/(1,1)}$. În plus, dacă notăm $q_k := \beta_k p_k$, atunci $\beta_k = \pm\sqrt{(q_k, q_k)}$ din condiţia de normalitate, iar $q_k = (x - \alpha_k)p_{k-1} - \beta_{k-1}p_{k-2}$.

Construiţi o procedură care construieşte aceste poligoane ortogonale. Comparaţi poligoanele obţinute cu poligoanele Lagrange din pachetul `'orthopoly'`.

Se va considera în calcule $k \leq 4$ şi produsul scalar

$$(f, g) = \int_0^\infty fg\omega$$

unde $\omega(x) = e^{-x}$.

45. (*Factor prim*) Găsiţi cel mai mare factor prim al unui număr întreg şi multiplicitatea sa.

46. (*Timp*) Cosnstituiţi o listă cu primele n numere din şirul 1,4,7,10,... prin următoarele procedee:

a) creând o listă vidă si utilizând un ciclu în care se adaugă câte un nou element;

b) utilizând comanda `seq`;

c) utilizând operatorul \$.

Care variantă este mai rapidă şi care este mai lentă?

47. (*Iluminare*) Se consideră un drum orizontal iluminat de două lămpi de putere P_1, P_2 şi înălţime h_1, h_2. Coordonatele lămpilor sunt $(0, h_1)$ şi (s, h_2) unde s este distanţa orizontală dintre cele două surse de lumină.

a) Fie $X = (x, 0)$ un punct oarecare de pe drum dintre cele două lămpi. Determinaţi punctul X care este minimal luminat, ţinând seama că intensitatea luminoasă în punctul X datorată primei lampi este $P_1 \cos\alpha_1 / r_1^2$ unde r_1 este distanţa de la X la lampă, iar α_1 este unghiul de impact al razei luminoase în punctul X faţă de drum. Se consideră următoarele valori numerice de test: $P_1 := 2000$, $P_2 := 3000$, $s := 20$, $h_1 := 5$, $h_2 := 6$. Verificati prin grafic corectitudinea rezultatului obţinut.

b) Determinaţi înălţimea celei de a doua lămpi care mazimizează iluminarea în punctul X de minim. Verificaţi prin grafic valoarea obţinută. Se consideră $P_1 := 2000$, $P_2 := 3000$, $s := 20$, $h_1 := 5$.

c) (Iluminare optimală) Se consideră puteriile lămpilor şi distanţa dintre ele cunoscute. Pentru o iluminare cât mai uniformă diferenţa dintre minimul şi maximul iluminării trebuie redusă. Ţinând seama că poziţiile cele mai luminate sunt sub lampă, uniformizarea este posibilă prin varierea înălţimii lămpilor. Care sunt înălţimile optime? Verificaţi grafic rezultatul. Se consideră $P_1 := 2000$, $P_2 := 3000$, $s := 20$.

48. (*Meciul*) Era ziua mare a meciului, Poli contra Steaua! Steaua conducea cu 24-21, în cel de-al 4-lea sfert, iar Poli avea mingea în terenul său. Un atacant de la Poli a început să alerge înspre terenul advers. Trecând de linia de mijloc, constată că în faţa lui sunt 75 m între el şi linia porţii şi se avântă spre ea. La 30 m la stânga atacantului se află un apărător al echipei adverse. Acesta ştie că este

mai rapid decât atacantul. De fapt, atacantul aleargă 8 m/sec, iar apărătorul 8.4 m/sec. Apărătorul porneşte imediat în urmărire. Consideră că cel mai bun mijloc de a-l ajunge este de a ajusta constant direcţia sa astfel încât să alerge întotdeauna înspre atacant (vectorul său de viteză este orientat permanent spre atacant).

a) Când apărătorul a pornit în urmărire s-a auzit un geamăt puternic în tribune. De ce?

b) Cât de rapid trebuie să fie apărătorul pentru a-l ajunge pe atacant la linia porţii?

49. (*Arie*) Fie R regiunea mărginită de graficele lui $f(x) = 3-x$ şi $g(x) = x^2-3x$. Determinaţi aria lui R.

50. (*Volum*) Fie R regiunea mărginită de $f(x) = 1$ şi $g(x) = (x-10)^2$.

a) Presupunem că R este rotit în jurul axei x. Determinaţi volumul solidului de rotaţie.

b) Care este volumul solidului de rotaţie obţinut prin rotirea lui R în jurul axei y?

c) Care volum este mai mare?

51. (*Regula trapezului*) În cazul în care trebuie să determinăm o integrală definită putem utiliza o aproximare numerică, de exemplu cu regula trapezului sau regula Simpson. Pachetul `student` conţine aceste două reguli. Construiţi o procedură similară care se bazează pe evaluări ale integralei utilizând o secvenţă de n, $2n$, $3n$, $4n$, $5n$ subdiviziuni şi formula de aproximare

$$\int_a^b f(x)dx = \frac{d}{2}\sum_{i=1}^{kn}\left[f(a+(i-1)d)+f(a+id)\right].$$

Aplicaţi procedura construită la $\int_0^1 e^{x^2}dx$.

52. (*Drum*) Fie a_1, a_2, $a_3, \ldots, a_n$ un şir de numere pozitive care converg la 0. Imaginaţi-vă că plecaţi din origine şi străbateţi a_1 kilometrii spre est, apoi a_2 spre nord, apoi a_3 spre vest şi aşa mai departe, în mod ciclic. Unde sfârşeşte drumul? Testaţi cazurile $a_n = 1/n$ şi $a_n = 1/2^n$.

53. (*Cele mai mici pătrate*) Se dau valorile (x_i, y_i), $i = 1 \ldots m$, şi o funcţie $f(x) = a_1u_1(x) + a_2u_2(x) + ... + a_nu_n(x)$ se presupune că satisface $y_i = f(x_i)$ pentru oricare i. Găsiţi valorile a_i, $i = 1 \ldots n$ pentru care $\sum_{i=1}^{m}[f(x_i)-y_i]^2$ este minimă. Determinaţi soluţia utilizând funcţiile corespunzătoare din pachetul `linalg`. Se consideră $m = 5$, $f(x) = a_1x + a_2$ şi o mulţime aleatoare de valori de intrare.

54. (*Factorizare QR*) Determinaţi factorizarea QR a unei matrice A pătratice urmărind următoarele etape:

1. se transformă A într-o matrice cu coloane ortogonale Q';
2. se normalizează coloanele lui Q', obţinuându-se o matrice Q;
3. se determină $R = Q^TA$.

55. (*Două bile*) O bilă se află la distanţa de 2 unităţi de la marginea mesei. Jucătorul vrea să lanseze această bilă înspre marginea mesei şi aceasta să lovească o bilă roşie aflată la 3 unităţi de marginea mesei. Dacă distanţa dintre proiecţiile verticale ale bilelor pe marginea mesei este de 5 unităţi, găsiţi unghiul sub care jucătorul trebuie să lanseze bila (faţă de marginea mesei).

56. (*Revizuire*) Presupunem că laturile opuse ale mesei din problema anterioară

se află la 4 unităţi distanţă şi că jucătorul doreşte ca bila să lovească de două ori laturile înainte de a atinge bila roşie. Sub ce unghi trebuie să lovească bila?

57. (*Turn de apă*) Să considerăm că ne aflăm în punctul (0,0,4) al axei z, cu privirea înspre axa x. Un turn de apă sferic cu 2 unităţi în diametru este centrat în origine (0,0,0). Punctul axei x apropiate de (0,0,0) nu sunt vizibile, fiind mascate de turnul de apă. Un punct îndepărtat pe axa x este vizibil în măsura în care linia ce conectează punctul cu (0,0,4) nu atinge turnul de apă (de exemplu, punctul (10,0,0)). Care este punctul vizibil cu cea mai mică coordonată pozitivă x?

58. (*Scara 1*) Piciorul unei scări este plasat în punctul $(3, 0)$. Scara este sprijinită pe un cupolă parabolică $y = 3 - x^2$. Unde intersectează scara axa y?

59. (*Scara 2*) O scară de 5 unităţi este sprijinită pe parabola $y = 3-x^2$. Piciorul scării se află pe axa x, iar partea de sus pe axa y. Descrieţi poziţia scării indicând coordonatele piciorului şi capătului scării.

60. (*Aria benzii Moebius*) Se consideră un segment de dreaptă de înălţime $2a$ care se mişcă în jurul unui cerc de rază r din planul xy . În plus se roteşte cu 180 de grade în jurul mijlocului său care este pe cerc. Determinaţi aria suprafeţei descrise. Caz particular: $a = 1, r = 2$. Comparaţi această arie cu cea a unui cilindru de înălţime 2 şi raza bazei 2.

61. (*Integrală dublă*) Estimaţi cu două zecimale integrala funcţiei $f(x, y) = 10e^{\sqrt{\sin y \cos x}}$ în regiunea S: $0 < x < 1,\ x^2 < y < x$.

62. (*Lemniscata*) Găsiţi aria mărginită de lemniscata $(x^2 + y^2)^2 = 2a(x^2 - y^2)$.

63. (*Convergenţa sumelor parţiale*) Scrieţi două proceduri, una pentru crearea unei secvenţe de sume parţiale ale unei serii dependente de un parametru real şi a doua pentru vizualizarea modului de convergenţă a secvenţei sumelor parţiale pentru un interval dat al parametrului real. Teste numerice pentru

$$\sum_{n=0}^{\infty} \frac{x^n}{n!},\ x \in [-2, 1], \qquad \sum_{n=1}^{\infty} \frac{(x+2)^n \ln n}{n3^n},\ x \in [-6, 4].$$

64. (*Bara de metal*) Temperatura la capătul unei bare de metal de 4 m este menţinută la 100, respectiv 150 grade. Iniţial, bara de metal are 0 grade peste tot. Trei termometre sunt plasate la distanţe egale de 1 m şi consultate la intervale de 1 minut, iar temperatura este înregistrată într-un tabel de tipul:

	x_1	x_2	x_3	
100	0	0	0	150

Persoana care se ocupă de notarea temperaturilor consemnează că temperatura la x_2 este 90% din media temperaturii la x_1, x_2 şi x_3 în minutul anterior, că temperatura la x_1 este 90% din media dintre 100 şi temperatura la x_1 şi x_2 în minutul anterior, iar temperatura la x_3 este 90% din media dintre 150 şi temperatura la x_2 şi x_3 în minutul anterior. Care vor fi temperaturile citite la minutul 100? Aceste temperaturi tind să devină constante?

65. (*Limita unui şir*) Construiţi o procedură `Sequence` care oferă următoarele informaţii:

a) valoarea numerică a primilor N termeni ai şirului recepţionat ca argument;

b) limita şirului dacă acesta converge;
c) graficul cu primii N termeni ai şirului.
Testaţi procedura pentru $a_n = 1 + 1/n$.

66. (*Aria torului*) Care este aria unui tor?

67. (*Numere aleatoare*) Se consideră 1000 numere aleatoare între 0 şi 1. Ce valoare medie a pătratelor acestor numere se aşteaptă?

68. (*Aria dintre parabole*) Găsiţi parabola $y = a - x^2$ astfel încât aria dintre aceasta şi parabola $y = x^2$ este 100.

69. (*Parabola şi funcţia modul*) Parabola $y = ax^2 + bx + c$ este tangentă la graficul funcţiei $y = 2 + |x - 3|$ în două puncte şi aria regiunii mărginite de graficele lor este 10. Găsiţi a, b şi c.

70. (*ODE 1*) Trasaţi un set de soluţii ale ecuaţiei diferenţiale $y' = e^{-x} - 2y$.

71. (*ODE 2*) Trasaţi graficul soluţiei problemei Cauchy $y' - \sin(2\pi x)y = 1$, $y(0) = 1$.

72. (*ODE 3*) Trasaţi soluţiile ecuaţiei diferenţiale $y' = (2y - 4x - 5)/(2y - 2x)$.

73. (*ODE 4*) Trasaţi soluţiile ecuaţiei diferenţiale $y' = (\cos y - y\cos x)/(x\sin y + \sin x - 1)$. Trasaţi câmpul direcţional asociat cu ecuaţia.

74. (*ODE 5*) Trasaţi soluţia ecuaţiei diferenţiale $xy^3dx - (x^4 + y^4)dy = 0$ care trece prin (1,1).

75. (*ODE 6*) Rezolvaţi problema $(1 + 5x - y)dx - (x + 2y)dy = 0$, $y(0) = 1$. Trasaţi câmpul direcţional şi soluţia.

76. (*ODE 7*) Găsiţi soluţia generală a ecuaţiei diferenţiale $y' + (1 + x^2)y = 0$ şi trasaţi soluţia care trece prin punctul (0,1).

77. (*Curbe ortogonale 1*) Două curbe sunt ortogonale într-un punct (x, y) dacă panta unei curbe este $dy/dx = f(x, y)$ şi panta curbei ortogonale este $dy/dx = -1/f(x, y)$. Verificaţi ortogonalitatea curbelor $y = x$ şi $y = \sqrt{1 - x^2}$ în punctul $(\sqrt{2}/2, \sqrt{2}/2)$.

78. (*Izoterme*) Fie $T(x, y)$ temperatura la punctul (x, y). Curbele date de $T(x, y) = c$ se numesc izoterme. Traiectoriile ortogonale sunt curbe de-a lungul cărora căldura se scurge. Determinaţi izotermele în cazul particular în care curbele de scurgere a căldurii sunt date de $y^2 + 2xy - x^2 = c$. Trasaţi izotermele şi curbele de scurgere a căldurii pe acelaşi grafic.

79. (*Curbe ortogonale 2*) Determinaţi familia de traiectorii ortogonale la familia de curbe $y = cx^2$. Trasaţi ambele familii de curbe.

80. (*Sistem ODE 1*) Găsiţi soluţia generală a sistemului

$$X' = \begin{pmatrix} 3 & -2 \\ 4 & -1 \end{pmatrix} X.$$

Trasaţi câmpul direcţional şi schiţaţi soluţiile pentru un set de condiţii iniţiale.

81. (*Sistem ODE 2*) Rezolvaţi problema

$$X' = \begin{pmatrix} 5 & 5 & 2 \\ -6 & -6 & -5 \\ 6 & 6 & 5 \end{pmatrix} X, \quad X(0) = \begin{pmatrix} 0 \\ 0 \\ 2 \end{pmatrix}.$$

Trasaţi soluţia şi câmpul direcţional.

82. (*Fibra nervoasă*) Trasaţi soluţiile sistemului de ecuaţii diferenţiale neliniare (provenit din studiul impulsurilor într-o fibră nervoasă):

$$\frac{dV}{d\xi} = W, \quad \frac{dW}{d\xi} = F(V) + R - uW, \quad \frac{dR}{d\xi} = \frac{\epsilon}{u}(bR - V - a),$$

unde $F(V) = V^3/3 - V$, $\epsilon = 0.08$, $a = 0.7$, $b = 0.8$, $u = 0.6$ şi $V(0) = 1$, $W(0) = 0.5$, $R(0) = 0.5$. Se vor trasa: curba parametrică $(V(t), W(t), R(t))$, graficele funcţiilor $V(t)$, $W(t)$, $R(t)$, funcţie de t, graficele V funcţie de W, respectiv V de R, şi W de R).

83. (*Ecuaţia Van der Pol*) Trasaţi soluţiile ecuaţiei Van der Pol's $x" + \mu(x^2 - 1)x' + x = 0$ pentru $\mu = 1$.

84. (*Metode cu diferenţe pentru ODE*) Construiţi proceduri pentru metodele numerice iterative de rezolvare a problemei

$$y'(x) = f(x, y(x)), \ y(0) = y_0, \ 0 \leq x \leq T$$

ce aproximează $y_n \approx y(x_n)$, unde $x_n - x_{n-1} = h$, $x_0 = 0 < x1 < \ldots < x_n = T$ este o diviziune a intervalului de integrare, prin:

a) metoda Euler explicită $y_n = y_{n-1} + hf(x_{n-1}, y_{n-1})$;

b) metoda Euler imbunătăţită $y_n^* = y_{n-1} + hf(x_{n-1}, y_{n-1})$, $y_n = y_{n-1} + \frac{h}{2}[f(x_{n-1}, y_{n-1}) + f(x_n, y_n^*)]$;

c) metoda Runge Kutta $y_n = y_{n-1} + h(k_1 + k_2)/2$, $k_1 = f(x_{n-1}, y_{n-1})$, $k_2 = f(x_n, y_{n-1} + hk_1)$

d) metoda Runge Kutta standard $y_n = y_{n-1} + h(k_1 + 2k_2 + 2k_3 + k_4)/6$, $k_1 = f(x_{n-1}, y_{n-1})$, $k_2 = f(x_{n-1} + h/2, y_{n-1} + hk_1/2)$, $k_3 = f(x_{n-1} + h/2, y_{n-1} + hk_2/2)$.

Se cer următoarele:

1. testarea procedurile pentru $f(x, y) = xy$, $n = 10$, $T = 1$.;
2. compararea rezultatele obţinute cu soluţia exactă;
3. care oferă o solutie mai apropiată de cea exactă, metoda Euler explicită cu pas $h/2$ sau cea îmbunătăţită cu pas h?
4. care metodă oferă cea mai bună aproximare la $T = 1$?

85. (*Coeficienţii metodei Runge-Kutta standard*) Se consideră problema Cauchy $y'(x) = f(x, y(x))$, $y(x_0) = y_0$ rezolvată numeric prin metoda Runge-Kutta explicită

$$y_{n+1} = y_n + h\sum_{i=1}^{s} b_i k_i$$

unde $k_1 = f(x, y)$, $k_2 = f(x + c_2 h, y + ha_{2,1}k_1) \ \ldots k_s = f(x + c_s h, y + h\sum_{j=1}^{s-1} a_{s,j}k_j)$. Dacă $y_n \approx y(t_0 + nh)$, atunci $y_{n+1} \approx y(t_0 + (n+1)h)$.

Construirea unei metode Runge Kutta explicite cu s-etape şi ordin m presupune rezolvarea sistemului de ecuaţii neliniare pentru parametrii $a_{i,j}$, b_i, c_i rezultat din condiţia de coincidenţă a formulei Runge Kutta cu seria Taylor până la ordinul m.

Se cere:

a) să se determine coeficienţii pentru $s = 3$, $m = 3$.

b) construiţi o procedură care determină coeficienţii pentru cazul general s, m oarecare. Se adaugă condiţia de simetrie $c_i = \sum_{j=1}^{i-1} a_{i,j}$. Verificare pentru $m = s = 2$.

86. (*Metoda Picard*) Metoda Picard este utilizată pentru rezolvarea sistemelor de ecuaţii diferenţiale de ordin întâi. Presupune construirea unui şir de funcţii y_i care converg uniform la soluţie pe un anumit interval. Soluţia problemei cu valori iniţiale $y'(x) = f(x, y(x)),\ y(0) = y_0$ satisface şi ecuaţia integrală $y(x) = y_0 + \int_0^x f(t, y(t))dt$. Ca primă aproximare a lui $y(x)$ se consideră funcţia constantă y_0. Funcţia y_1 este definită prin $y_1(x) = y_0 + \int_0^x f(t, y_0(t))dt$. Prin inducţie, y_n este definit prin $y_n = y_0 + \int_0^x f(t, y_{n-1})(t)dt$. Speranţa este ca, pentru n mare, secvenţa y_n să conveargă spre soluţia exactă.

Considerăm de exemplu $f(x, y) = y + 1,\ y(0) = 1$. Soluţia exactă este $y = 2e^x - 1$. Generaţi termenii şirului Picard şi verificaţi convergenţa spre soluţia exactă.

87. (*Metoda Euler explicită*) Se consideră ecuaţia diferenţială $dy/dx = g(x, y),\ y(x_0) = y_0$. Dacă $g(x, y) = y$ şi $y(0) = 1$, soluţia este $y(x) = e^x$. Pe baza acestei obesrvaţii, construiţi un tabel cu valorile aproximative date de metoda Euler explicită $y_n = y_{n-1} + hg(x + ih, y_{n-1})$ pentru $e^{i/10}$ unde i de la 1 la 10.

88. (*Ecuaţia Euler*) Fie a şi b două numere reale. Ecuaţia $x^2y''(x) + axy'(x) + by(x) = 0$ este numită ecuaţia Euler. Această ecuaţie poate fi transformată într-o ecuaţie cu coeficienţi constanţi prin schimbarea de variabilă $x = e^z$. Astfel, ecuaţia Euler devine:

$$\frac{dy^2}{dz^2} - \frac{dy}{dz} + a\frac{dy}{dz} + by = 0.$$

a) Demonstraţi cu ajutorul Maple-ului că această transformare este corectă.

b) Rezolvaţi ecuaţia în $y(z)$ şi, utilizând rezultatul, determinaţi soluţia $y(x)$ a ecuaţiei iniţiale.

89. (*Hemodializă*) În procesul de hemodializă, sângele unui pacient este pompat printr-un dializor şi separat de lichidul de curăţare printr-o membrană semipermeabilă care permite celulelor auxiliare (nu de sânge) să difuze în lichidul de curăţire. Dacă se notează $u(x)$ concentraţia celulelor auxiliare în sânge, $v(x)$ concentraţia celulelor auxiliare în lichidul de curăţire, Q_d rata de scurgere a lichidului de curăţire, Q_b rata de scurgere a sângelui, atunci modelul matematic este

$$Q_b u' = -k(u - v),\ -Q_d v' = k(u - v),$$

unde k este o constanta de proporţionalitate.

Trasaţi graficele funcţiilor u, v, pentru următoarele valori numerice: $k = 2.25,\ L = 1,\ Q_b = 2,\ Q_d = 4, u_0 = 35.8$.

90. (*Infecţie*) Propagarea infecţiei unei populaţii este descrisă prin ecuaţia $i' + (\gamma + \mu - \lambda) = -\lambda i^2$, unde $i(t)$ este procentul populaţiei infectate, λ este raportul de contact zilnic (o persoană infectată va răspândi boala la λ persoane per zi), μ este raportul dintre numărul indivizilor ce mor în urma bolii şi numărul indivizilor infectaţi, γ este raportul dintre numărul indivizilor ce îşi revin din boală, numărul indivizilor infectaţi, iar $u(i_0) = i_0$.

Trasaţi graficul solutiei pentru:

1. $\lambda = 0.5,\ \gamma = 0.75,\ \mu = 0.65,\ i_0 = 0.1 \ldots 0.9,\ t = 0 \ldots 5$

2. $\lambda = 1.5,\ \gamma = 0.75,\ \mu = 0.65,\ i_0 = 0.01 \ldots 0.09,\ t = 0 \ldots 20$
3. $\lambda = 3 - 2.5\sin(6t),\ \gamma = 2,\ \mu = 1,\ i_0 = 0.1 \ldots 0.9,\ t = 0 \ldots 10$

91. (*Evoluţia populaţiei*) Se presupune că raportul de creştere a unei populaţii $y(t)$ se schimbă proporţional cu cantitatea actuală. Matematic, acest fapt se transpune în ecuaţia $y'(t) = ky(t),\ y(0) = y_0$, unde y_0 este populaţia iniţială. Dacă $k > 0$, populaţia creşte, iar dacă $k < 0$, populaţia descreşte.

a) Populaţia unei ţări înregistrată în 1800 era de 5.3 milioane. Experimental, constanta k din modelul de mai sus s-a stabilit a fi 0.03. Comparaţi rezultatele înregistrate de-a lungul anilor şi înscrise în tabelul de mai jos cu cele teoretice propuse de model. Oferă acest model o aproximare bună?

b) Un alt model înlocuieşte constanta k cu $r - ay(t)$, unde r şi a sunt constante (ecuaţia logistică). Pentru estimarea creşterii populaţiei menţionate mai sus se consideră $r = 0.03,\ a = 0.0001,\ y_0 = 5.3$. Comparaţi rezultatele obţinute prin modelul anterior cu cele ale modelului logistic. Care este populatia estimată pentru anul 2000?

An	Diferenţa	Populaţie (mil.)	An	Diferenţa	Populaţie (mil.)
1800	(0)	5.30	1900	(100)	76.21
1810	(10)	7.24	1910	(110)	92.23
1820	(20)	9.64	1920	(120)	106.02
1830	(30)	12.68	1930	(130)	123.20
1840	(40)	17.06	1940	(140)	132.16
1850	(50)	23.19	1950	(150)	151.33
1860	(60)	31.44	1960	(160)	179.32
1870	(70)	38.56	1970	(170)	203.30
1880	(80)	50.19	1980	(180)	226.54
1890	(90)	62.98	1990	(190)	248.71

92. (*Timp de înjumătăţire*)

a) Timpul de înjumătăţire al unei substanţe radioctive este timpul în care aceasta descreşte la jumătate. Dacă timpul de înjumătăţire a polonium-ului 209 este de 100 ani, determinaţi procentul dintr-o cantitate de poloniu care rămâne după 50 de ani.

b) În lemnul unui sarcofag a fost găsit carbon-14 într-un procent de 63% faţă de o mostră de carbon-14 actual. Ţinând seama că timpul de înjumătăţire a carbonului-14 este de 5730, care este vârsta sarcofagului?

93. (*Prăjitura*) Rata în care temperatura $T(t)$ unui corp se schimbă este proporţională cu diferenţa dintre temperatura corpului şi temperatura constantă T_s a mediului înconjurător. Această situaţie este reprezentată prin problema Cauchy

$$\frac{dT}{dt} = k(T - T_s),\ T(0) = T_0$$

unde T_0 este temperatura iniţială a corpului, iar k este o constantă de proporţionalitate.

Aplicaţie: o prăjitură este scoasă de la un cuptor cu 350 grade K şi plasată într-o cameră cu temperatura de 75 grade K. În 15 minute, prăjitura are temperatura de 150 grade K. Determinaţi timpul necesar ca prăjitura să atingă temperatura de 80 grade K, astfel încât să fie mâncabilă.

94. (*Temperatura în clădire*) În modelul din problema anterioară temperatura mediului se consideră constantă. Daca $C(t)$ este temperatura mediului, temperatura la timpul t, $u(t)$ este solutia ecuaţiei

$$\frac{du}{dt} = k(C(t) - u(t))$$

Aplicaţie: se presupune că în luna Aprilie în Georgia temperatura mediului este dată de $C(t) = 70 - 10\cos(\pi t/12)$, $0 \leq t \leq 24$. Determinaţi temperatura maximă dintr-o clădire care are iniţial temperatura de 60 grade K, dacă $k = 1/4$. La ce oră se produce? Comparaţi această temperatura cu cea din iunie când temperatura mediului este de $C(t) = 80 - 10\cos(\pi t/12)$ şi temperatura iniţială este de 70 K. Comparaţi orele de maxim.

95. (*Cădere liberă sau forţată*) Se consideră legea mişcării unui obiect

$$m\frac{dv}{dt} = mg - F_r$$

unde m este masa obiectului, iar F_r este forţa de frecare, proporţională cu viteza corpului ($g = 32$).

Se cer următoarele:

a) Determinaţi viteza şi poziţia unui corp cu masa $m = 1$ care este aruncat cu viteza iniţială $v_0 = 2$ de la înălţimea de 1000 m. Se presupune că obiectul este subiectul forţei de rezistenţă a aerului echivalentă cu viteza momentană a obiectului. Determinaţi timpul la care obiectul loveşte pământul şi viteza în acel moment.

b) Un obiect de masa $m = 1$ cade de la înălţimea de 50 m de deasupra unui bazin. Cât timp obiectul este în aer, forţa de rezistenţă a aerului este echivalentă cu viteza instantanee v. Cât timp obiectul este în apă, forţa de rezistenţă a apei este echivalentă cu $6v$. Adâncimea bazinului este de 25 m. Determinaţi timpul necesar obiectului pentru a atinge fundul bazinului.

96. (*Aruncarea pe verticală*) Determinaţi soluţia generală (pentru viteză şi poziţie) a ecuaţiei diferenţiale de mai jos care modelează mişcarea unui obiect de masă m aruncat pe verticală, în sus, de la înălţimea h_0, cu viteza iniţială v_0, presupunând că forţa de rezistenţă este cv (c constant):

$$\frac{dv}{dt} = -g - \frac{c}{m}v, \ v(0) = v_0$$

Consideraţi pentru teste numerice valorile $m = 1/128$, $c = 1/160$, $g = 32$, $v_0 = 48$, $y_0 = 48$. Determinaţi înălţimea maximă atinsă de obiect. Studiaţi influenţa vitezei iniţiale şi a înălţimii iniţiale asupra înălţimii maxime.

97. (*Ecuaţia Lotka-Volterra*) Fie $x(t)$ populaţia unei specii de animale de pradă la momentul t şi $y(t)$ populaţia speciei cu care se hrănesc animalele din prima categorie, la momentul t. Dacă a este rata de naştere a speciei de pradă şi b este rata de interacţiune dintre populaţia $x(t)$ şi $y(t)$, atunci rata de schimbare a primei specii este

$$\frac{dx}{dt} = ax - bxy.$$

Dacă rata de creştere a celei de a doua specii este d, iar rata morţii este c, atunci

$$\frac{dy}{dt} = -cy + dxy.$$

Schiţaţi dependenţa y de x şi evoluţia celor două populaţii în timp pentru $a = 2,\ b = 1,\ c = 3,\ d = 1,\ x(0) = 1,\ y(0) = 1$.

98. (*Mişcare armonică simplă*) Se ataşează un corp de masă m la un resort suspendat. Resortul are lungimea naturală b. Când corpul este ataşat resortului, lungimea sa este, la poziţia de echilibru $x = 0$, de s ori lungimea naturală. Când sistemul este pus în mişcare, distanţa corpului faţă de poziţia de echilibru la timpul t este dată de funcţia $x(t)$.

Considerând că singurele forţe care acţionează sunt $F(t) = ma(t)$, unde $a(t)$ este acceleraţia corpului şi $F_1(t) = kx(t)$, unde k este constanta resortului (forţe opuse), se poate construi un model matematic care se bazează pe ecuaţiile

$$m\frac{d^2x}{dt^2} + kx = 0,\ x(0) = \alpha,\ \frac{dx}{dt}(0) = \beta$$

unde α este poziţia iniţială a resortului faţă de punctul de echilibru, iar β, viteza iniţială imprimată corpului suspendat.

Se cer următoarele:

1. Simulaţi mişcarea resortul pentru $m = 1,\ k = 64,\ \alpha = 1,\ \beta = 0$.

2. Un obiect de masă $m = 1$ este ataşat unui resort cu constanta $k = 4$. Determinaţi funcţia de pozitie a obiectului şi trasaţi soluţia pentru (a) $\alpha = 1, 4, -2,\ \beta = 0$ (b) $\alpha = 0,\ \beta = 1, 4, -2$

99. (*Mişcare încetinită*) Se reconsideră modelul resort-corp (vezi *mişcare armonică*). Un model realist introduce în sistem şi forţa de rezistenţă, care este proporţională cu viteza, $F_r(t) = cdx/dt$. În acest caz ecuaţiile au forma:

$$m\frac{d^2x}{dt^2} + c\frac{dx}{dt} + kx = 0\ x(0) = \alpha,\ \frac{dx}{dt}(0) = \beta$$

a) Un corp de masă $8/g$ este ataşat la un resort de lungime 4. La echilibru resortul are lungimea 6. Determinati funcţia de poziţie dacă $F_r = 2dx/dt$ şi resortul este eliberat din poziţia de echilibru cu viteza iniţială de comprimare de 1 unitate. Se consideră $g = 32$. Simulaţi mişcarea resortului.

b) Determinaţi funcţia de poziţie în cazul în care masa este deplasată iniţial în jos cu 0.5 unităţi de la punctul de echilibru şi viteza iniţială de comprimare este de 5 unităţi.

100. (*Studiul mişcării încetinite*) Construiţi o procedură pentru trasarea soluţiei generale a problemei Cauchy din enunţul anterior.

a) Testaţi procedura pentru cazul unei mase $16/g$ care modifică cu 2 unităţi lungimea unui resort, $F_r(t) = 0.5\,dx/dt$, masa este iniţial în poziţia de echilibru, iar viteza de compresie este unitară.

b) Studiaţi influenţa unei viteze iniţiale de întindere asupra funcţiei de poziţie pentru un sistem cu $m = 1, c = 5$, şi masa $32/g$ ce întinde resortul cu 8 unităţi, aflată iniţial la 1 unitate deasupra poziţiei de echilibru. Fie, de exemplu, $\beta = -1,\ \beta = -6$.

c) Studiaţi influenţa constantei c asupra soluţiei în cazul $k = 6,\ \alpha = 0,\ \beta = 1$. Consideraţi, de exemplu, $c = 2\sqrt{6},\ 4\sqrt{6},\ \sqrt{6}$.

101. (*Mişcare forţată*) Se reconsideră modelul resort-corp cu frecare. În cazul în care intervine o forţă exterioară $f(t)$, funcţia de poziţie este guvernată de legea

$$m\frac{d^2x}{dt^2} + c\frac{dx}{dt} + kx = f(t),\ x(0) = \alpha,\ \frac{dx}{dt}(0) = \beta.$$

a) Construiţi o procedură pentru soluţia generală. Testaţi-o pentru (a) $m = 1,\ c = 0,\ k = 4,\ \alpha = 0,\ \beta = 0,\ f = 1,\ \cos(t),\ \sin(t)$ şi (b) $m = 1,\ c = 4,\ k = 13,\ \alpha = 0,\ \beta = 1$.

b) Trasaţi soluţia pentru $m = 1, k = 13, c = 4$, şi

$$f(t) = \begin{cases} \sin t, & 0 \le t < \pi/2, \\ 0, & t \ge \pi/2 \end{cases}.$$

102. (*Circuite LRC*) Presupunem că $I(t)$ este intensitatea curentului dintr-un circuit de tip L-R-C cu inductanţa L, rezistenţa R şi capacitatea C. Dacă Q este sarcina pe condensator, iar E este tensiunea, atunci $I = dQ/dt$ şi

$$L\frac{d^2Q}{dt^2} + R\frac{dQ}{dt} + \frac{1}{C}Q = E(t),\ Q(0) = Q_0,\ dQ/dt(0) = I_0$$

Determinaţi intensitatea curentului în cazul în care:

a) $L = 1,\ R = 40,\ C = 1/4000,\ E(t) = 24,\ Q_0 = 0,\ I_0 = 0$.

b) $L = 0,\ R = 100,\ C = 10^{-2}$, iar

$$E(t) = \begin{cases} 100, & 0 \le t < 1 \\ 0, & t \ge 1 \end{cases}.$$

103. (*Pendulul*) Se consideră un pendul de lungime L şi cu masa ataşată m. Se consideră θ unghiul dintre fir şi poziţia de verticală. Legea mişcării este exprimată prin ecuaţiile

$$L\frac{d^2\theta}{dt^2} + b\frac{d\theta}{dt} + g\theta = 0,\ \theta(0) = \theta_0,\ \frac{d\theta}{dt}(0) = v_0,$$

unde b este coeficientul corespunzător forţei de frecare $F_r(t) = bd\theta/dt$, iar v_0 este viteza initială imprimată pendulului.

1. Dacă $L = g = 32,\ \theta_0 = 0,\ v_0 = 2,\ b = 0$, care este perioada pendulului?

2. Simulaţi mişcarea pendulului pentru $L = 8/5,\ b = 32/5,\ g = 32,\ \theta_0 = 1,\ v_0 = 2$.

3. Studiaţi influenţa condiţiilor iniţiale asupra graficului soluţiei. Teste numerice pentru:

(a) $v_0 = 0,\ \theta_0 = -1, -0.5, 0.5, 1$;

(b) $\theta_0 = 0,\ v_0 = -2, -1, 1, 2$;

(c) $\theta_0 = 1,\ v_0 = 1;\ \theta_0 = -1,\ v_0 = -1;\ \theta_0 = 1,\ v_0 = 5;\ \theta_0 = -1,\ v_0 = -5$.

104. (*Calibrare*) Se măsoară presiunea unui gaz într-un container prevăzut cu piston. Mişcarea acestui piston este dată de legea $x(t) = x_a(t) + x_b(t)$, unde

$$x"_a(t) + a^2x_a(t) = 0,\ x_a(0) = 0,\ x'_a(0) = v_0,$$

cu v_0 viteza iniţială a pistonului, iar

$$x"_b(t) + a^2x_b(t) + bx_b(t)^2 + 2bx_a(t) + b[x_a(t)]^2 = 0,\ x_b(0) = 0,\ x'_b(0) = 0,$$

În plus, integrarea are loc pe intervalul $[0, t_m)$ unde

$$t_m = \min\left(\frac{\sqrt{6}}{a}, \left(\frac{21}{bv_0}\right)^{1/3}, \left(\frac{12}{abv_0}\right)^{1/4}\right)$$

şi $a = \sqrt{\alpha\pi d^2/(4Ml)},\ b = \beta\pi d^2/(4Ml^2)$. Presiunea este definită prin

$$p(t) := \frac{x(t)}{l}\left(\alpha + \beta\frac{x(t)}{l}\right).$$

Rezolvaţi sistemul utilizând serii de ordin 6,...,12. Reprezentaţi grafic soluţiile. Valori numerice: $\alpha = 3.8\cdot 10^9,\ \beta = 30\cdot 10^9,\ d = 0.015,\ l = 0.02,\ M = 5,\ v_0 = 5$.

105. (*ODE cu funcţii discontinue*) Să se traseze graficul soluţiei problemei:

a) $y" + 9y = f,\ f(t) = \begin{cases} 1, & 0 \le t < \pi, \\ 0, & t \ge \pi \end{cases},\ y(0) = y'(0) = 0.$

b) $y" + y = \delta(t-\pi),\ y(0) = y'(0) = 0.$

106. (*Sfera*) Se consideră o sferă metalică cu interiorul gol, raza la exterior: r_2, la interior: r_1. Temperatura într-un punct al sferei este funcţie de raza r, unde $r_1 \le r \le r_2$.

a) Considerând problema unidimensională, ecuaţia modelului este

$$\frac{dT}{dr} = -\frac{a}{r^2},$$

unde a este necunoscut. Care este soluţia acestei ecuaţii dacă temperatura în exterior este T_2 şi în interior T_1.

b) Considerând problema în coordonatele polare r şi ϕ, ecuaţia este

$$\frac{d^2T}{dr^2} + \frac{2}{r}\frac{dT}{dr} = 0.$$

Se consideră condiţiile iniţiale: $T_1 = 400,\ T_2 = 300, 200, 100,\ r_1 = 0.1,\ r_2 = 0.4$. Trasaţi curbele temperaturii funcţie de raza $r_1 \le r \le r_2$.

107. (*Câmp*) Un câmp de cultură este încălzit de soare. Se cere determinarea temperaturii în sol, la o adâncime x, după o perioadă de timp Δt, dacă temperatura la suprafaţă se menţine constanta T_s, iar initial temperatura, indiferent de adâncime, era T_f. Modelul matematic se bazează pe ecuaţiile

$$a^2\frac{d^2T}{dx^2} = \frac{dT}{dt},\ T(0,t) = T_s,\ T(x,0) = T_f$$

Se caută soluţia sub forma unei funcţii $V(u) = T(x,t)$, unde $u = u(x,t) = x/(2a\sqrt{t})$. Trasaţi:

a) temperatura funcţie de timp pentru un set de valori de adâncime;

b) temperatura ca funcţie de timp şi adâncime.

în condiţiile în care $a = \sqrt{0.003}$, adâncimea maximă=0.5, Δt=1800, T_s=300, T_f=285.

108. (*Sistem ODE 1*) Trasaţi soluţia sistemului

$$x' = x - 4y + 1,\ y' = x + y,\ x(0) = 1,\ y(0) = 0.$$

109. (*Sistem ODE 2*) Determinaţi soluţia generală a sistemului

$$x' = x + 2y + z,\ y' = 6x - y,\ z' = -x - 2y - z$$

şi trasaţi soluţia ce trece prin $x(0) = 1,\ y(0) = 2,\ z(0) = 3/2$.

110. (*Sistem ODE 3*) Determinaţi soluţia generală a sistemului

$$x' = 2x + 5y + \cos(4t),\ y' = -4x - 2y + \sin(4t)$$

şi trasaţi soluţiile ce trec prin $x(0) = -1..1,\ y(0) = -1..1$.

111. (*Sistem ODE 4*) Trasaţi soluţia sistemului

$$x' = y,\ y' = x^3/3 - x + z - y,\ z' = -0.08(x + 0.7)$$

112. (*Metoda Runge-Kutta standard*) Construiţi o procedură care aplică unui sistem de ecuaţii diferenţiale de dimensiune *dim* şi funcţie f, metoda Runge-Kutta standard cu pas h, pornind de la valorile iniţiale date prin vectorul $rk0$.

a) Verificaţi procedura pentru sistemul:

$$\frac{dx}{dt} = x - y + 1,\ \frac{dy}{dt} = x + 3y + e^{-t}\ x(0) = 0,\ y(0) = 1,$$

şi pasul $h = 0.1$. Verificaţi corectitudinea rezultatelor numerice pe intervalul [0,1], comparându-le cu soluţia exactă.

b) Determinaţi solutia problemei $x" + \sin x = 0,\ x(0) = 0,\ x'(0) = 1$ aplicând metoda Runge-Kutta standard. Trasaţi soluţia aproximativă. Trasaţi curba parametrică $(x(t), x'(t))$.

113. (*Soluţii periodice*) Se consideră sistemul de ecuaţii diferenţiale

$$x' = 2x - 6xy/5,\ y' = -y + 9xy/10$$

a) Trasaţi curbele soluţiilor sistemului de mai sus din planul (x, y) care la $t = 0$ trec prin punctele (1,0.5) şi (1,1), pentru $0 \leq t \leq 10$;

b) Trasaţi cele două solutii în planul (t, x), pentru $0 \leq t \leq 14$;

c) Trasaţi cele două soluţii în planul (t, y), pentru $0 \leq t \leq 14$;

d) Determinaţi aproximativ perioadele acestor soluţii.

114. (*Ecuaţia căldurii*) Se consideră o bară de metal de lungime p. Se presupune că rata de scurgere a căldurii prin bucata de metal este c (relaţionată cu difuzia termală a materialului). Temperatura într-un punct x al barei la momentul t este $u(x, t)$, soluţia ecuaţiei diferenţiale $u_t = c^2 u_{xx}$. Diferite situaţii se pot întâlni în practică, situaţii care pot conduce la rezolvarea acestei ecuaţii: de exemplu, temperatura la capete poate fi menţinută constantă. În acest caz, modelul matematic se bazează pe sistemul

$$\begin{cases} u_t = c^2 u_{xx}, 0 < x < p, t > 0 \\ u(0,t) = T_0,\ u(p,t) = T_1,\ t > 0, . \\ u(x,0) = f(x),\ 0 < x < p \end{cases}$$

În acest caz, se consideră $S(x)$ temperatura în condiţii de echilibru, soluţie a problemei

$$\begin{cases} S" = 0,\ 0 < x < p \\ S(0) = T_0,\ S(p) = T_1 \end{cases}$$

şi temperatura tranzientă $v(t, x)$, soluţie a problemei omogene:

$$\begin{cases} v_t = c^2 v_{xx}, 0 < x < p, t > 0 \\ v(0,t) = 0,\ v(p,t) = 0,\ t > 0, \\ v(x,0) = f(x) - S(x),\ 0 < x < p \end{cases} .$$

Atunci $u(x,t) = v(x,t) + S(x)$. Pentru rezolvarea problemei omogene se consideră

$$v(x,t) = \sum_{n=1}^{\infty} c_n \sin \frac{n\pi x}{p} e^{-c^2 n^2 \pi^2 t},$$

unde c_n satisfac

$$v(x,0) = \sum_{n=1}^{\infty} c_n \sin \frac{n\pi x}{p} = f(x) - S(x),$$

adică c_n reprezintă coeficienţii seriei Fourier a funcţiei $f(x) - S(x)$, daţi de

$$c_n = \frac{1}{p} \int_0^p (f(x) - S(x)) \sin \frac{n\pi x}{p} dx,\ n = 1, 2, \ldots$$

Se cere trasarea curbele $u(x,T)$ la diferite momente T în cazul în care $c = 1,\ p = 1,\ T_0 = 10,\ T_1 = 60,\ f(x) = 10$.

115. (*Ecuaţia undei*) Se consideră problema:

$$\begin{cases} c^2 u_{xx} = u_{tt}, 0 < x < p, t > 0 \\ u(0,t) = 0,\ u(p,t) = 0,\ t > 0, \\ u(x,0) = f(x),\ u_t(x,0) = g(x),\ 0 < x < p \end{cases}.$$

Pentru rezolvarea problemei se consideră

$$u(x,t) = \sum_{n=1}^{\infty} \left(a_n \cos \frac{cn\pi t}{p} + b_n \sin \frac{cn\pi t}{p} \right) \sin \frac{n\pi x}{p},$$

unde $a_n,\ b_n$ se obţin din condiţiile iniţiale

$$a_n = \frac{2}{p} \int_0^p f(x) \sin \frac{n\pi x}{p} dx,\ b_n = \frac{p}{cn\pi} \frac{2}{p} \int_0^p g(x) \sin \frac{n\pi x}{p} dx,\ n = 1, 2, \ldots$$

Rezolvaţi ecuaţia undei pentru $c = 1,\ p = 1,\ f(x) = \sin(\pi x),\ g(x) = 3x + 1$. Simulaţi evoluţia undei prin animaţie.

116. (*Ecuaţia potenţialului*) Se consideră problema asociată ecuaţiei Laplace în spaţiul bidimensional:

$$\begin{cases} u_{xx} + u_{yy} = 0,\ 0 < x < a,\ 0 < y < b \\ u(x,0) = 0,\ u(x,b) = f(x),\ 0 < x < a, . \\ u(0,y) = 0,\ u(a,y) = 0,\ 0 < y < b \end{cases}$$

Pentru rezolvarea problemei se consideră

$$u(x,t) = \sum_{n=1}^{\infty} B_n \sinh \frac{n\pi b}{a} \sin \frac{n\pi x}{a}$$

unde coeficienţii B_n se obţin din condiţiile iniţiale

$$B_n = \frac{2}{a \sinh \frac{n\pi b}{a}} \int_0^a f(x) \sin \frac{n\pi x}{p} dx,\ n = 1, 2, \ldots$$

Rezolvaţi ecuaţia undei pentru $f(x) = x(1-x),\ a = 1,\ b = 2$.

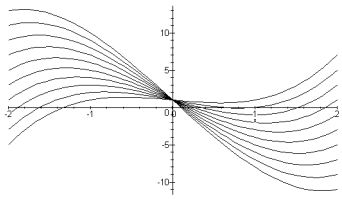


Figura 4.1:

4.2 Răspunsuri

1. (*Ecuaţia de grad trei*)

```
> solve(x^3-a*x+1=0,x);
```

$$\%1^{1/3}+\frac{1}{3}\,\frac{a}{\%1^{1/3}},-\frac{1}{2}\,\%1^{1/3}-\frac{1}{6}\,\frac{a}{\%1^{1/3}}+\frac{1}{2}\,I\,\sqrt{3}\,\left(\%1^{1/3}-\frac{1}{3}\,\frac{a}{\%1^{1/3}}\right),$$
$$-\frac{1}{2}\,\%1^{1/3}-\frac{1}{6}\,\frac{a}{\%1^{1/3}}-\frac{1}{2}\,I\,\sqrt{3}\,\left(\%1^{1/3}-\frac{1}{3}\,\frac{a}{\%1^{1/3}}\right)$$
$$\%1:=-\frac{1}{2}+\frac{1}{18}\,\sqrt{-12\,a^3+81}$$

```
> num:=proc(a) fsolve(x^3-a*x+1=0); end:
> seq([num(i)],i=1..4);
```

$$[-1.324717957],[-1.618033989,.6180339887,1.],$$
$$[-1.879385242,.3472963553,1.532088886],$$
$$[-2.114907541,.2541016884,1.860805853]$$

```
> plot({seq(x^3-a*x+1,a=1..10)},x=-2..2);
```

Grafic: vezi figura 4.1

2. (*Ecuaţie neliniară*)

```
> restart: solve(7*cos(x)+x+x^2=15,x);
> plot(7*cos(x)+x+x^2-15,x=-10..10);
```

Grafic: vezi figura 4.2

```
> fsolve(7*cos(x)+x+x^2=15,{x},x=-5..-4);
```

$$\{x=-4.548616097\}$$

```
> fsolve(7*cos(x)+x+x^2=15,{x},x=3..4);
```

$$\{x=3.970102935\}$$

```
> f1:=x^2+x-15-7: f2:=x^2+x-15+7:
> sol1:=solve(f1=0,x); sol2:=solve(f2=0,x);
```

$$sol1:=-\frac{1}{2}+\frac{1}{2}\,\sqrt{89},-\frac{1}{2}-\frac{1}{2}\,\sqrt{89}$$
$$sol2:=-\frac{1}{2}+\frac{1}{2}\,\sqrt{33},-\frac{1}{2}-\frac{1}{2}\,\sqrt{33}$$

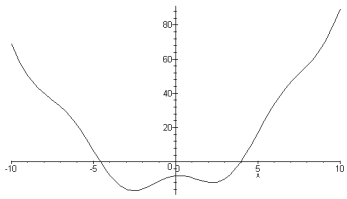


Figura 4.2:

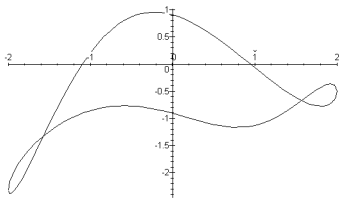


Figura 4.3:

```
> map(evalf,[sol1,sol2]);
```

$$[4.216990566, -5.216990566, 2.372281324, -3.372281324]$$

3. (*Sistem neliniar 1*)

```
> restart; solve({x^2+y^2=4, sin(x+y)+cos(x)=1},{x,y});
> x:='x': y:='y':
> f1:=x->sin(x+sqrt(4-x^2))+cos(x)-1;
> f2:=x->sin(x-sqrt(4-x^2))+cos(x)-1;
```

$$f1 := x \to \sin(\, x + \mathrm{sqrt}(\, 4 - x^2 \,)\,) + \cos(\, x \,) - 1$$
$$f2 := x \to \sin(\, x - \mathrm{sqrt}(\, 4 - x^2 \,)\,) + \cos(\, x \,) - 1$$

```
> plot({f1(x),f2(x)},x=-2..2);
```

Grafic: vezi figura 4.3

```
> solve(f1(x)=0,x);
> assign(fsolve(f1(x)=0,{x},x=-2..0)):
> x1:=x; x:='x': y1:=sqrt(4-x1^2);
```

$$x1 := -1.097155805$$
$$y1 := 1.672198894$$

```
> assign(fsolve(f1(x)=0,{x},x=0..2)):x2:=x;x:='x':y2:=sqrt(4-x2^2);
```

$$x2 := .9500217911$$
$$y2 := 1.759959828$$

```
> g1:=proc(x,y) x^2+y^2-4;end: g2:=proc(x,y) sin(x+y)+cos(x)-1;end:
> g1(x1,y1),g2(x1,y1); g1(x2,y2),g2(x2,y2);
```

$$.1\,10^{-8}, .1\,10^{-8}$$

$$0, 0$$

4. (*Sistem neliniar 2*)

```
> solve({x+y+z=1, 3*x+2*y-z=5, x*y+7*z^3=0},{x,y,z});
```

$$\left\{ y = \mathrm{RootOf}(\,90\,\_Z^2 - 12\,\_Z + 7\,\_Z^3 + 56\,),\right.$$
$$x = -\frac{3}{4}\,\mathrm{RootOf}(\,90\,\_Z^2 - 12\,\_Z + 7\,\_Z^3 + 56\,) + \frac{3}{2},$$
$$\left. z = -\frac{1}{4}\,\mathrm{RootOf}(\,90\,\_Z^2 - 12\,\_Z + 7\,\_Z^3 + 56\,) - \frac{1}{2} \right\}$$

```
> solutii:=allvalues(",'d'):
> solutii[1];
```

$$\left\{ z = \frac{1}{4}\,\%1^{1/3} + \frac{232}{49}\,\frac{1}{\%1^{1/3}} + \frac{4}{7}, y = -\%1^{1/3} - \frac{928}{49}\,\frac{1}{\%1^{1/3}} - \frac{30}{7},\right.$$
$$\left. x = \frac{3}{4}\,\%1^{1/3} + \frac{696}{49}\,\frac{1}{\%1^{1/3}} + \frac{33}{7} \right\}$$
$$\%1 := \frac{29632}{343} + \frac{64}{49}\,\sqrt{393}$$

```
> assign(");
> digits:=5: evalf(sin(x*y+z));
```

$$.2676512971$$

5. (*Sistem neliniar 3*)

```
> restart;
> solve({x^2+y^2+z^2=4, x+y+z=0, x*sin(y*z)=-1},{x,y,z});
> fsolve({x^2+y^2+z^2=4, x+y+z=0, x*sin(y*z)=-1},{x,y,z},x=-2..2,
> y=-2..2,z=-2..2);
```

$$\{\, x = -1.630958259, y = .8860579420, z = .7449003167\,\}$$

```
> fsolve({x^2+y^2+(x+y)^2=4,x*sin(y*(x+y))=1},{x,y},x=-2..2,y=-2..2);
```

$$\{\, x = -1.630958259, y = .7449003167\,\}$$

```
> solve(x^2+y^2+(x+y)^2=4);
```

$$\{\, x = \mathrm{RootOf}(\,\_Z^2 + y^2 + \_Z\,y - 2\,), y = y\,\}$$

```
> x1:=-1/2*y+1/2*sqrt(8-3*y^2): x2:=-1/2*y-1/2*sqrt(8-3*y^2):
> f1:=y->x1*sin(y*(x1+y))-1:   f2:=y->x2*sin(y*(x2+y))-1:
```

$$f2 := y \to x2\,\sin(\,y\,(\,x2 + y\,)\,) - 1$$

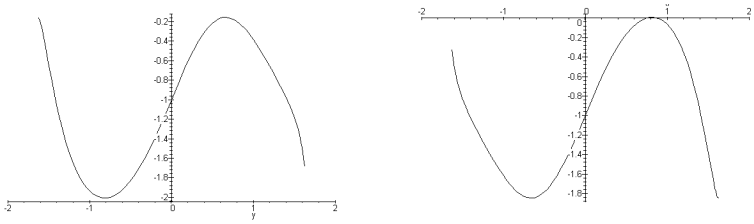


Figura 4.4:

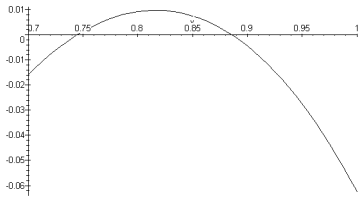


Figura 4.5:

```
> plot(f1(y),y=-2..2);
```
Grafic: vezi figura 4.4.a
```
> plot(f2(y),y=-2..2);
```
Grafic: vezi figura 4.4.b
```
> plot(f2(y),y=0.7..1);
```
Grafic: vezi figura 4.5
```
> assign(fsolve(f2(y)=0,{y},y=0.7..0.8)): y1:=y; y:='y':
```
$$y1 := .7449003167$$
```
> assign(fsolve(f2(y)=0,{y},y=0.8..0.9)): y2:=y; y:='y':
```
$$y2 := .8860579420$$
```
> x1:=-1/2*y1-1/2*sqrt(8-3*y1^2); z1:=-(x1+y1);
```
$$x1 := -1.630958259$$
$$z1 := .8860579423$$
```
> x1^2+y1^2+z1^2=4, x1+y1+z1=0, x1*sin(y1*z1)=-1;
```
$$4.000000002 = 4, 0 = 0, -1.000000000 = -1$$
```
> x2:=-1/2*y2-1/2*sqrt(8-3*y2); z2:=-(x2+y2);
```
$$x2 := -1.598648521$$
$$z2 := .7125905790$$
```
> x2^2+y2^2+z2^2=4, x2+y2+z2=0, x2*sin(y2*z2)=-1;
```
$$3.848561104 = 4, 0 = 0, -.9436384626 = -1$$

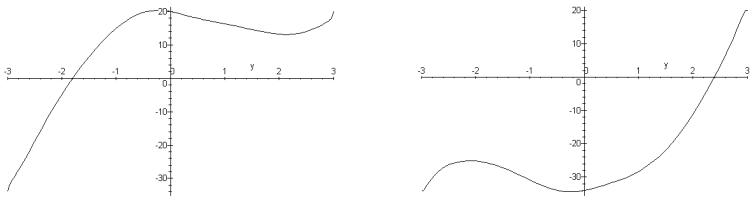


Figura 4.6:

6. (*Sistem neliniar 4*)

```
> x:='x': y:='y': solve({x^2+y^2=9, x^3+y^3-sin(x*y)=7});
> fsolve({x^2+y^2=9,x^3+y^3-sin(x*y)=7},{x,y},x=-3..3,y=-3..3);
```

$$\{\, y = -1.802943923, x = 2.397789234 \,\}$$

```
> xa:=sqrt(9-y^2): xb:=-sqrt(9-y^2):
> f1:=y->xa^3+y^3-sin(xa*y)-7: f2:=y->xb^3+y^3-sin(xb*y)-7:
> plot(f1(y),y=-3..3);
```

Grafic: vezi figura 4.6.a

```
> fsolve(f1(y)=0,{y}, y=-3..3);
```

$$\{\, y = -1.802943923 \,\}$$

```
> assign(''): y1:=y; x1:=xa; y:='y':
```

$$y1 := -1.802943923$$
$$x1 := 2.397789234$$

```
> plot(f2(y),y=-3..3);
```

Grafic: vezi figura 4.6.b

```
> fsolve(f2(y)=0,{y},y=-3..3);
```

$$\{\, y = 2.397789234 \,\}$$

```
> assign(''): y2:=y; x2:=xb; y:='y':
```

$$y2 := 2.397789234$$
$$x2 := -1.802943923$$

```
> x1^2+y1^2=9,x1^3+y1^3-sin(x1*y1)=7;
> x2^2+y2^2=9, x2^3+y2^3-sin(x2*y2)=7;
```

$$9.000000000 = 9, 7.000000006 = 7$$
$$9.000000000 = 9, 7.000000006 = 7$$

7. (*Valori complexe*)

```
> A:= array( [ [2 , -2,  3.1], [ 1, -5, -3], [ 0, 1.1, -2] ]):
> EV:= linalg[eigenvals](A);
```

$$EV := 1.855340452, -3.427670226 + 1.506674239\, I,$$
$$-3.427670226 - 1.506674239\, I$$

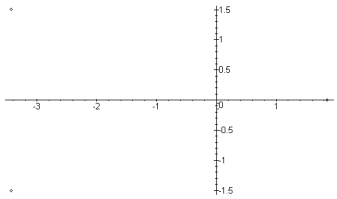


Figura 4.7:

```
> plot( [ seq( [ Re(EV[k]),  Im(EV[k]) ] , k = 1 .. 3 ) ] ,
> style = point , symbol = diamond , color = magenta);
```

Grafic: vezi figura 4.7

8. (*Pătratul magic*)

```
> sumcols:=proc(mat)
> local Vect,i,j;
> Vect:=linalg[matrix](1,linalg[coldim](mat));
> for i from 1 to linalg[coldim](mat) do
> Vect[1,i]:=convert([seq(mat[j,i],j=1..linalg[rowdim](mat))],`+`);
> od;
> linalg[transpose](Vect);
> end:
> sumdiag:=proc(mat)
> local Vect,i,j;
> Vect:=linalg[matrix](2,1):
> Vect[1,1]:=convert([seq(mat[i,i],i=1..linalg[coldim](mat))],`+`):
> Vect[2,1]:=convert([seq(mat[linalg[coldim](mat)-i+1,i],
>    i=1..linalg[coldim](mat))],`+`):
> eval(Vect);
> end:
> sumrows:=proc(mat)
> local Vect,i,j;
> Vect:=linalg[matrix](linalg[rowdim](mat),1);
> for i from 1 to linalg[rowdim](mat) do
> Vect[i,1]:=convert([seq(mat[i,j],j=1..linalg[coldim](mat))],`+`);
> od;
> eval(Vect);
> end:
> checkmagic:=proc(mat)
>    local Col,Lin,Dia,val,i,j,k;
>    Col:=sumcols(mat); Lin:=sumrows(mat); Dia:=sumdiag(mat);
>    for i from 1 to linalg[coldim](mat)-1 do
```

```
>        if Col[i,1]<>Col[i+1,1] then RETURN(false); fi;
>    od;
>    for i from 1 to linalg[rowdim](mat)-1 do
>        if Lin[i,1]<>Lin[i+1,1] then RETURN(false); fi;
>    od;
>    if Col[1,1]<>Lin[1,1] then RETURN(false); fi;
>    if Dia[1,1]<>Dia[2,1] then RETURN(false); fi;
>    if Col[1,1]<>Dia[1,1] then RETURN(false); fi;
>    for i from 1 to linalg[coldim](mat)*linalg[rowdim](mat) do
>      val:=false;
>      for j from 1 to linalg[rowdim](mat) do
>          for k from 1 to linalg[coldim](mat) do
>             if mat[j,k]=i then val:=true; fi;
>          od;
>      od;
>      if val=false then RETURN(false); fi;
>    od;
>    RETURN(true);
> end:
> A:=linalg[matrix](4,4,[$1..16]):
> checkmagic(A);
```

false

```
> B:=linalg[matrix](1,1,[1]):
> checkmagic(B);
```

true

```
> C:=linalg[matrix](10,10,[
> [92,99,1,8,15,67,74,51,58,40],[98,80,7,14,16,73,55,57,64,41],
> [4,81,88,20,22,54,56,63,70,47],[85,87,19,21,3,60,62,69,71,28],
> [86,93,25,2,9,61,68,75,52,34],[17,24,76,83,90,42,49,26,33,65],
> [23,5,82,89,91,48,30,32,39,66],[79,6,13,95,97,29,31,38,45,72],
> [10,12,94,96,78,35,37,44,46,53],[11,18,100,77,84,36,43,50,27,59]]):
> checkmagic(C);
```

true

9. (*Metoda trapezelor*)

```
> int(sin(x^3),x=0..1);
```

$$\int_0^1 \sin(\,x^3\,)\,dx$$

```
> student[trapezoid](sin(x^3),x=0..1,4);
```

$$\frac{1}{4}\left(\sum_{i=1}^{3}\sin\left(\frac{1}{64}\,i^3\right)\right)+\frac{1}{8}\sin(\,1\,)$$

```
> evalf(");
```

$$.2426265918$$

10. (*Integrare prin părţi*)

```
> k:=Int(x^3*exp(x),x);
```

$$k := \int x^3 \, \mathrm{e}^x \, dx$$

```
> student[intparts](k,x^3);
```

$$x^3 \, \mathrm{e}^x - \int 3 \, x^2 \, \mathrm{e}^x \, dx$$

```
> student[intparts](",x^2);
```

$$x^3 \, \mathrm{e}^x - 3 \, x^2 \, \mathrm{e}^x + \int 6 \, x \, \mathrm{e}^x \, dx$$

```
> student[intparts](",x);
```

$$x^3 \, \mathrm{e}^x - 3 \, x^2 \, \mathrm{e}^x + 6 \, x \, \mathrm{e}^x - \int 6 \, \mathrm{e}^x \, dx$$

```
> value(");
```

$$x^3 \, \mathrm{e}^x - 3 \, x^2 \, \mathrm{e}^x + 6 \, x \, \mathrm{e}^x - 6 \, \mathrm{e}^x$$

11. (*Dreaptă 3D*)

```
> restart; with(linalg):
> A := array([2,1,9]): B := array([-3,0,1]):
> X := evalm(A+t*(B-A));
```

$$X := [\, 2 - 5 \, t \; 1 - t \; 9 - 8 \, t \,]$$

```
> t:=solve(X[3]=0,t );
```

$$t := \frac{9}{8}$$

```
> [x,y] = [X[1],X[2]];
```

$$[\, x, y \,] = \left[\frac{-29}{8}, \frac{-1}{8} \right]$$

12. (*Paralelogram*)

```
> A := vector(2): B := vector(2): C := vector(2): D := vector(2):
> M1 := evalm(1/2*(A+B)): M2 := evalm(1/2*(B+C)):
> M3 := evalm(1/2*(C+D)): M4 := evalm(1/2*(D+A)):
> evalm((M2-M1)-(M3-M4));
```

$$0$$

13. (*Distanţa*)

```
> A:=vector([3,2,5]): B:=vector([2,0,0]): X:=vector([0,0,4]):
> proj := evalm(dotprod(X-A,B-A)/dotprod(B-A,B-A)*(B-A));
```

$$proj := \left[\frac{-2}{5} \; \frac{-4}{5} \; -2 \right]$$

```
> perp := evalm(X-A-proj);
```

$$perp := \left[\frac{-13}{5}\ \frac{-6}{5}\ 1\right]$$

```
> dist := sqrt(dotprod(perp,perp));
```

$$dist := \frac{1}{5}\sqrt{230}$$

```
> distpttoline := proc(X,A,B)
>     local proj,perp;
>         proj := evalm(dotprod(X-A,B-A)/dotprod(B-A,B-A)*(B-A));
>         perp := evalm(X-A-proj); sqrt(dotprod(perp,perp))
>     end:
> distpttoline(vector([3,4,1]),X,B);
```

$$\frac{1}{5}\sqrt{445}$$

14. (*Polinom*)

```
> expr:=(a*x^2+b*x*sin(y)+c*sin(y))^2+(a*(sin(y))^2+b*x)^3;
```

$$expr := (\,a\,x^2 + b\,x\sin(\,y\,) + c\sin(\,y\,)\,)^2 + \left(a\sin(\,y\,)^2 + b\,x\right)^3$$

```
> expand(expr);
```

$$a^2\,x^4 + 2\,b\,x^3\sin(\,y\,)\,a + 2\,c\sin(\,y\,)\,a\,x^2 + b^2\,x^2\sin(\,y\,)^2 + 2\,b\,x\sin(\,y\,)^2\,c + c^2\sin(\,y\,)^2 + a^3\sin(\,y\,)^6 + 3\,a^2\sin(\,y\,)^4\,b\,x + 3\,a\sin(\,y\,)^2\,b^2\,x^2 + b^3\,x^3$$

```
> collect(expr,x);
```

$$a^2\,x^4 + (\,b^3 + 2\,b\sin(\,y\,)\,a\,)\,x^3 + \left(3\,a\sin(\,y\,)^2\,b^2 + 2\,c\sin(\,y\,)\,a + b^2\sin(\,y\,)^2\right)\,x^2 + \left(3\,a^2\sin(\,y\,)^4\,b + 2\,c\sin(\,y\,)^2\,b\right)\,x + c^2\sin(\,y\,)^2 + a^3\sin(\,y\,)^6$$

```
> coeff(",x^2);
```

$$3\,a\sin(\,y\,)^2\,b^2 + 2\,c\sin(\,y\,)\,a + b^2\sin(\,y\,)^2$$

```
> factor(");
```

$$\sin(\,y\,)\,(\,3\sin(\,y\,)\,a\,b^2 + b^2\sin(\,y\,) + 2\,c\,a\,)$$

```
> collect(",b);
```

$$\sin(\,y\,)\,(\,3\sin(\,y\,)\,a + \sin(\,y\,)\,)\,b^2 + 2\,c\sin(\,y\,)\,a$$

```
> collect(expr,[x,sin(y)],distributed);
```

$$a^2\,x^4 + b^3\,x^3 + 2\,b\,x^3\sin(\,y\,)\,a + 2\,c\sin(\,y\,)\,a\,x^2 + (\,3\,a\,b^2 + b^2\,)\sin(\,y\,)^2\,x^2 + 2\,b\,x\sin(\,y\,)^2\,c + 3\,a^2\sin(\,y\,)^4\,b\,x + c^2\sin(\,y\,)^2 + a^3\sin(\,y\,)^6$$

```
> s1:=[coeffs(",[x,sin(y)],'t')];
```

$$s1 := [\,c^2, a^2, b^3, a^3, 2\,b\,a, 2\,c\,a, 3\,a\,b^2 + b^2, 2\,c\,b, 3\,a^2\,b\,]$$

```
> t;
```
$$\sin(y)^2, x^4, x^3, \sin(y)^6, \sin(y)\,x^3, x^2\sin(y), x^2\sin(y)^2, \sin(y)^2\,x, x\sin(y)^4$$

```
> s1[5];
```
$$2\,b\,a$$

15. (*Şir în oglindă*)

```
> reverselist:=proc(l)
>   local i;
>   RETURN([seq(l[nops(l)+1-i],i=1..nops(l))]);
> end:
> stringtolist:=proc(s)
>   local i;
>   RETURN([seq(substring(s,i..i),i=1..length(s))]);
> end:
> listtostring:=proc(l)
>    local i;
>    RETURN(cat(seq(substring(convert(l[i],string),1..1),
>    i=1..nops(l))));
> end:
> stringtolist(‘Acesta este un exemplu‘);
```
$$[A, c, e, s, t, a, \ , e, s, t, e, \ , u, n, \ , e, x, e, m, p, l, u]$$

```
> reverselist(");
```
$$[u, l, p, m, e, x, e, \ , n, u, \ , e, t, s, e, \ , a, t, s, e, c, A]$$

```
> listtostring(");
```
ulpmexe nu etse atsecA

```
> x:=2;
```
$$x := 2$$

```
> stringtolist(‘Acesta este un exemplu‘);
```
$$[A, c, e, s, t, a, \ , e, s, t, e, \ , u, n, \ , e, x, e, m, p, l, u]$$

```
> reverselist(");
```
$$[u, l, p, m, e, 2, e, \ , n, u, \ , e, t, s, e, \ , a, t, s, e, c, A]$$

```
> listtostring(");
```
ulpme2e nu etse atsecA

```
> reversestring:=proc(s)
>    listtostring(reverselist(stringtolist(s)));
> end:
> Sir:=‘Acesta este un exemplu‘;
```
Sir := *Acesta este un exemplu*

```
> reversestring(Sir);
```
$$ulpmexe\ nu\ etse\ atsecA$$

```
> x:=2;
```
$$x := 2$$

```
> reversestring(Sir);
```
$$ulpmexe\ nu\ etse\ atsecA$$

16. (*Calculul valorii* π)

```
> Digits:=50:
> Pi2:=0:
> for i from 1 to 50000 do
>    Pi2:=Pi2+(-1.)^(i+1)/(2.*i-1);
> od:
> Pi2:=4*Pi2;
```
$$\pi 2 := 3.1415726535897952384626423832795041041971666293794$$

```
> evalf(Pi-Pi2);
```
$$.00001999999999800000000099999999878000000027699957$$

```
> evalf((Pi-Pi2)/Pi);
```
$$.63661977230391936587060790776204920391635883049460\,10^{-5}$$

17. (*Triunghi*)

```
> with(geometry):
> point(A,[0,0]), point(B,[2,0]), point(C,[1,3]):
> triangle(ABC,[A,B,C]):
> centroid(ABC,cent_ABC):
> rotate(ABC[1],Pi/4,clockwise,v_rot_1,cent_ABC):
> rotate(ABC[2],Pi/4,clockwise,v_rot_2,cent_ABC):
> rotate(ABC[3],Pi/4,clockwise,v_rot_3,cent_ABC):
> coordinates(v_rot_1),coordinates(v_rot_2),coordinates(v_rot_3);
```
$$\left[-\sqrt{2}+1, 1\right], \left[1, -\sqrt{2}+1\right], \left[\sqrt{2}+1, \sqrt{2}+1\right]$$

```
> triangle(ABC_rot,[v_rot_1,v_rot_2,v_rot_3]):
> op(ABC_rot);
```
$$\begin{array}{l} \text{table}([\\ \quad 1 = v\_rot\_1 \\ \quad 2 = v\_rot\_2 \\ \quad 3 = v\_rot\_3 \\ \quad form = triangle \\ \quad given = [\, v\_rot\_1\, , v\_rot\_2, v\_rot\_3\,] \\ \quad]) \end{array}$$

18. (*Plan*)

```
> with(geom3d):
> point3d(AA,[0,0,0]), point3d(BB,[2,2,2]):
> point3d(CC,[3/5,1/3,7/8]),point3d(DD,[9/5,7/3,5/4]):
> line3d(AB,[AA,BB]), line3d(CD,[CC,DD]):
> inter(AB,CD,AB_CD):
> op(AB_CD);
```

$$\begin{aligned}&\text{table}([\\&\quad x = 1\\&\quad y = 1\\&\quad z = 1\\&\quad form = point3d\\&\quad])\end{aligned}$$

```
> plane(P,[z=0]):
> parallel(AB_CD,P,para_plan):
> op(para_plan);
```

$$\begin{aligned}&\text{table}([\\&\quad equation = (\, z - 1 = 0\,)\\&\quad normal\_vector = [\,0, 0, 1\,]\\&\quad form = plane\\&\quad given = [\,AB\_CD, [\,0, 0, 1\,]\,]\\&\quad])\end{aligned}$$

19. (*Primitive grafice 2D*)

```
> line:=proc(pt1,pt2) local l1,p1,p2,i,rangex,rangey,tplot;
>   p1:=[seq(convert(pt1[i],float),i=1..2)];
>   p2:=[seq(convert(pt2[i],float),i=1..2)];
>   rangex:=abs(p2[1]-p1[1]); rangey:=abs(p2[2]-p1[2]);
>   if rangex=0 then rangex:=1;fi; if rangey=0 then rangey:=1;fi;
>   l1:=[min(p1[1],p2[1])-.2*rangex..max(p1[1],p2[1])+2.*rangex,
>   min(p1[2],p2[2])-.2*rangey..max(p1[2],p2[2])+.2*rangey];
>   tplot:=plot([p1[1],p1[2],p2[1],p2[2]],scaling=CONSTRAINED);
> end:
> linie1:=line([3,-2],[-1,-2]): linie2:=line([-1,1],[-1,3]):
> linie3:=line([2,3],[-1,0]):
> plots[display]({linie1,linie2,linie3});
```

Grafic: vezi figura 4.8.a

```
> poligon:=proc(lpt) local cnt,tmp1,tmp2,xs,xd,ys,yd,range,tplot,i;
>   cnt:=nops(lpt);
>   tmp1:=min(seq(op(1,op(i,lpt)),i=1..cnt));
>   tmp2:=max(seq(op(1,op(i,lpt)),i=1..cnt));
>   range:=abs(tmp1-tmp2)*.2;
```

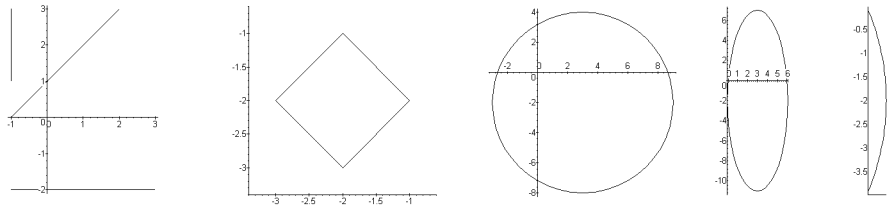


Figura 4.8:

```
>   xs:=tmp1-range; xd:=tmp2+range;
>   tmp1:=min(seq(op(2,op(i,lpt)),i=1..cnt));
>   tmp2:=max(seq(op(2,op(i,lpt)),i=1..cnt));
>   range:=abs(tmp1-tmp2)*.2;
>   ys:=tmp1-range; yd:=tmp2+range;
>   tplot:=plot(lpt,xs..xd,ys..yd,scaling=CONSTRAINED);
> end:
> poligon([[-1,-2],[-2,-3],[-3,-2],[-2,-1],[-1,-2]]);
```

Grafic: vezi figura 4.8.b

```
> cerc:=proc(centru,raza) local tplot,c,r,t,i;
>    c:=[seq(convert(centru[i],float),i=1..2)];
>    r:=convert(raza,float);
>    tplot:=plot([r*sin(t)+c[1],r*cos(t)+c[2],t=-Pi..Pi],
>    scaling=CONSTRAINED);
> end:
> cerc([3,-2],-6);
```

Grafic: vezi figura 4.8.c

```
> elipsa:=proc(centru,raza) local tplot,c,r,t,i;
>    c:=[seq(convert(centru[i],float),i=1..2)];
>    r:=[seq(convert(raza[i],float),i=1..2)];
>    tplot:=plot([r[1]*sin(t)+c[1],r[2]*cos(t)+c[2],t=-Pi..Pi],
>    scaling=CONSTRAINED);
> end:
> e:=elipsa([3,-2],[3,9]): plots[display](e);
```

Grafic: vezi figura 4.8.d

```
> arccerc:=proc(centru,domeniu,raza) local tplot,i,c,r,unghi;
>  c:=[seq(convert(centru[i],float),i=1..2)];
>  unghi:=convert(op(1,domeniu),float)..convert(op(2,domeniu),float);
>  if evalf(abs(op(1,unghi)-op(2,unghi)))> evalf(2*Pi) then
>   unghi:=0..2*Pi; fi;
>  r:=convert(raza,float);
>  tplot:=plot([r*sin(t)+c[1],r*cos(t)+c[2],t=unghi],
>   scaling=CONSTRAINED);
> end:
> arccerc([3,-2],3*Pi/8..5*Pi/8,5);
```

Grafic: vezi figura 4.8.e

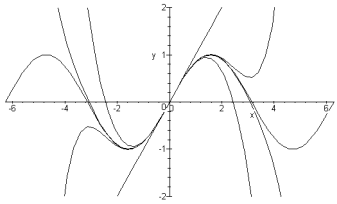


Figura 4.9:

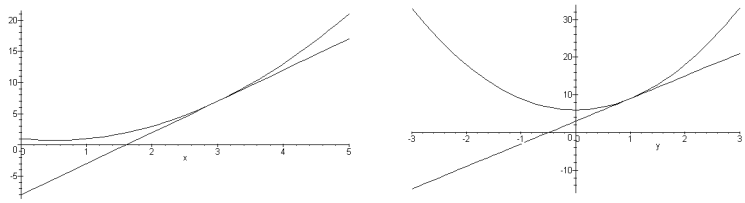


Figura 4.10:

20. (*Serii Taylor*)

```
> readlib(mtaylor):
> tay:=proc(i) if type(i,numeric)
>      then mtaylor(sin(x),[x=0],i+1) else 'tay'(i) fi
> end:
> polylist:=seq(tay(i),i={1,3,5,7});
```

$$polylist := x - \frac{1}{6}\,x^3, x - \frac{1}{6}\,x^3 + \frac{1}{120}\,x^5, x, x - \frac{1}{6}\,x^3 + \frac{1}{120}\,x^5 - \frac{1}{5040}\,x^7$$

```
> plot({sin(x),polylist},x=-2*Pi..2*Pi,y=-2..2);
```

Grafic: vezi figura 4.9

21. (*Tangenta*)

```
> f:=x->x^2-x+1:
> Tf := (x,x0)->f(x0)+D(f)(x0)*(x-x0):
> plot({f(x),Tf(x,3)},x=0..5);
```

Grafic: vezi figura 4.10.a

```
> g:=(x,y)->x^3-x+3*y^2:
> Tg:=(x,y,x0,y0)->g(x0,y0)+D[1](g)(x0,y0)*(x-x0)
> +D[2](g)(x0,y0)*(y-y0):
> plot({g(2,y),Tg(2,y,2,1)},y=-3..3);
```

Grafic: vezi figura 4.10.b

```
> plot({g(x,1),Tg(x,1,2,1)},x=-3..3);
> plot3d( {g(x,y),Tg(x,y,2,1)},x=-3..1,y=-3..2,axes=NORMAL);
```

Grafic: vezi figura 4.11

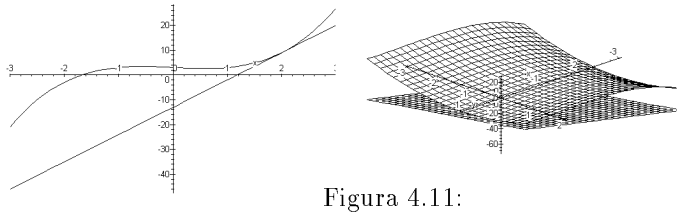


Figura 4.11:

Figura 4.12:

```
> # plots[animate]({g(x,y),Tg(x,y,2,1)}, x=-3..3,y=-3..3);
> # plots[animate]({g(x,y),Tg(x,y,2,1)}, y=-3..3,x=-3..3);
```

22. (*Cilindrii*)

```
> with(plots):
> cyl1:=[r,sin(phi),cos(phi)]:cyl2:=[sin(phi),r,cos(phi)]:
> sp:=grid=[29,9], scaling=CONSTRAINED:
> plot3d({cyl1,cyl2},phi=0..2*Pi,r=-2..1,sp);
> side1:=[r*sin(phi),sin(phi),cos(phi)]:
> side2:=[sin(phi),r*sin(phi),cos(phi)]:
> sdes1:=[r*sin(phi),-sin(phi),cos(phi)]:
> sdes2:=[-sin(phi),r*sin(phi),cos(phi)]:
> plot3d({side1,side2, sdes1,sdes2},phi=0..Pi,r=-1..1,sp);
```

Grafic: vezi figura 4.12

23. (*Corp compus*)

```
> f1:=plot3d([cos(theta),sin(theta),t],theta=0..2*Pi,t=-1/2..1/2):
> f2:=plot3d([cos(theta)*cos(phi),sin(theta)*cos(phi),
> -1/2+sin(phi)],theta=0..2*Pi,phi=Pi..3*Pi/2):
> f3:=plot3d([(t-3/2)*cos(theta),(t-3/2)*sin(theta),t],
> theta=0..2*Pi,t=1/2..3/2):
> plots[display3d]({f1,f2,f3});
```

Grafic: vezi figura 4.13

24. (*Tor*)

```
> d:=2:r:=1:x0:=1:y0:=2:z0:=3: plot3d([x0+r*cos(t)*(d/r+cos(u)),
> y0+r*sin(t)*(d/r+cos(u)),z0+r*sin(u)],t=0..2*Pi,u=0..2*Pi);
```

Grafic: vezi figura 4.14

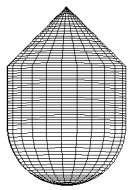


Figura 4.13:

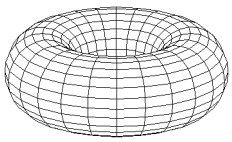


Figura 4.14:

25. (*Transformări liniare în* R^2)

```
> with(linalg):
> perete := [[-1,1],[-1,0],[1,0],[1,1]] :
> acoperis := [[-1.2,.8],[0,2],[1.2,.8]]:
> usa:=[[-.8,0],[-.8,.5],[-.5,.5],[-.5,0]]:
> house := {perete,acoperis,usa}:
> plot(house,scaling=constrained,axes=none);
```

Grafic: vezi figura 4.15.a

```
>  reflect := matrix([[-1,0],[0,1]]):
> newhouse := NULL:
> for   part in house do
>         newpart := NULL:
>        for   point in part do
```

Figura 4.15:

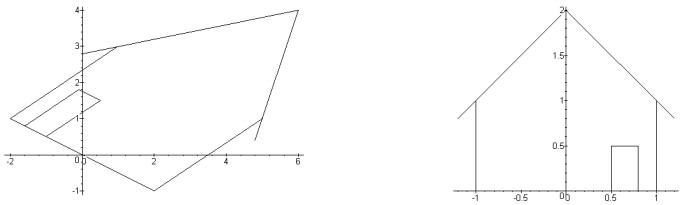


Figura 4.16:

```
>            npt := evalm(reflect &* point) ;
>            newpart := newpart,[npt[1],npt[2]];
>         od:
>         newhouse := newhouse,[newpart];
>   od:
>   newhouse := {newhouse};
```

$newhouse := \{[[.8,0],[.8,.5],[.5,.5],[.5,0]],[[1.2,.8],[0,2],[-1.2,.8]],$
$[[1,1],[1,0],[-1,0],[-1,1]]\}$

```
> plot(newhouse,scaling=constrained);
```

Grafic: vezi figura 4.15.b

O altă posibilitate de generare a noii case este:

```
>   {seq(map(convert,
>     [seq(evalm(reflect &* house[j][i]),i = 1 .. nops(house[j]))],
>     list),j = 1 .. nops(house))} ;
```

$\{[[.8,0],[.8,.5],[.5,.5],[.5,0]],[[1.2,.8],[0,2],[-1.2,.8]],$
$[[1,1],[1,0],[-1,0],[-1,1]]\}$

```
> showit :=proc(tr, house) local j,i;
>     plot({seq(map(convert,
>     [seq(evalm(tr &* house[j][i]),i = 1 .. nops(house[j]))],
>     list),j = 1 .. nops(house))},scaling = constrained); end:
> showit(matrix( [[2,3],[-1,2]]),house);
```

Grafic: vezi figura 4.16.a

```
> showit (reflect,house );
```

Grafic: vezi figura 4.16.b

```
> moveit := proc(tr,house) local id,movie,t,inter;
>               id := matrix([[1,0],[0,1]]); movie := NULL:
>               for t from 0 by 1/8 to 1 do
>                    inter := evalm(t*tr + (1-t)*id);
>                    movie := movie,showit(inter,house)
>               od;
>             plots[display]([movie],insequence = true )
>          end:
>  moveit( reflect,house);
```

Grafic: vezi figura 4.17

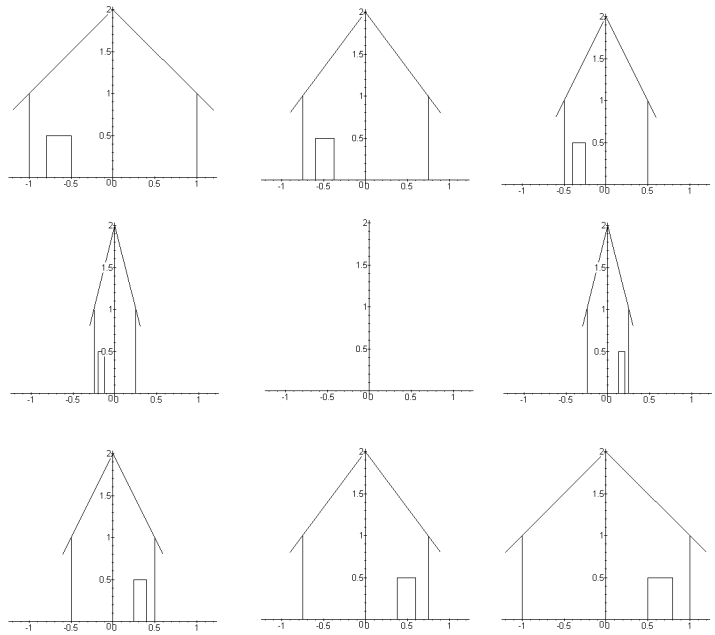


Figura 4.17:

26. (*Transformări liniare în* R^3) Se formează o matrice A cu vectorii coloană a_i. Atunci $A(A^TA)^{-1}A^T$ este matricea de proiecţie dorită.

```
> a := matrix(3,2,[1,3,2,0,3,1]):
> p:=evalm(evalm(a&*inverse(transpose(a)&*a))&*transpose(a)):
> fata := [[-1,0,0],[1,0,0],[1,0,4],[0,0,6],[-1,0,4]]:
> spate := [[-1,3,0],[1,3,0],[1,3,4],[0,3,6],[-1,3,4]]:
> acop1:=[[0,0,6],[0,3,6],[1,3,4],[1,0,4]]:
> acop2:=[[0,0,6],[0,3,6],[-1,3,4],[-1,0,4]]:
> lat1 :=[[1,0,0],[1,3,0],[1,3,4],[1,0,4]]:
> lat2 :=[[-1,0,0],[-1,3,0],[-1,3,4],[-1,0,4]]:
> usa := [[1,1,0],[1,1.5,0],[1,1.5,3],[1,1,3]]:
> casa := [fata,spate,acop1,acop2,lat1,lat2,usa]:
>  polygonplot3d(casa,scaling=constrained,style=patch);
```

Grafic: vezi figura 4.18.a

```
> pl1 :=polygonplot3d(casa,scaling=constrained ):
> trans := proc(pt,part)  local i;
> [seq(convert(evalm(pt+part[i]),list),i=1..nops(part))] end:
> transcasa := proc(pt,casa)
>   local tcasa,part;
>   tcasa := NULL:
```

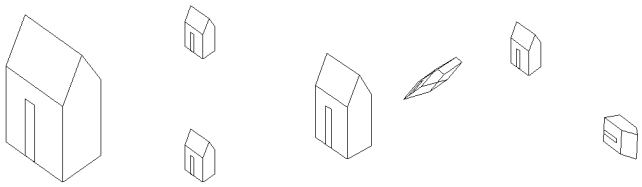


Figura 4.18:

```
>   for part in casa do
>    tcasa := tcasa,trans(pt,part) od;
> [tcasa] end:
> tcasa := transcasa([-8,-8,9],casa):
> pl2 := polygonplot3d(tcasa,scaling=constrained):
> display([pl1,pl2]);
```

Grafic: vezi figura 4.18.b

```
> proj := proc(p,part) local i;
> [seq(convert(evalm(p &* part[i]),list),i=1..nops(part))] end:
> projcasa := proc(p,casa)
>   local pcasa,part;
>   pcasa := NULL:
>   for part in casa do
>     pcasa := pcasa,proj(p,part) od:
>   [pcasa] end:
> pcasa := projcasa(p,tcasa):
> pl3 := polygonplot3d(pcasa,scaling=constrained):
> display([pl1,pl3]);
```

Grafic: vezi figura 4.18.c

Dacă P este matricea de proiecţie a lui R^3 pe planul generat de a_1 şi a_2, iar R matricea de reflecţie, pentru orice vector $b \in R^3$, $Rb = Pb + (Pb - b) = (2P - I)b$.

```
> reflect := proc(p)
>   local i;
>   evalm(2*p-diag(seq(1,i=1..rowdim(p)))) end:
> rcasa := projcasa(r,tcasa):
> pl4 := polygonplot3d(rcasa):
> display([pl1,pl4]);
```

Grafic: vezi figura 4.18.d

27. (*Puncte critice 1*)

```
> h:=(x,y,z,w) -> (x^2 - y^2 + 2*z^2 +3*w^2)*
> exp(-x^2 - y^2 -z^2-w^2):
> grad(h(x,y,z,w), [x,y,z,w]):
> convert(", set):
> solve(", {x,y,z,w});
```

$\{ z = 0, y = 0, x = 0, w = 0 \}, \{ y = 0, x = 0, z = 1, w = 0 \},$

$$\{\, z=-1, y=0, x=0, w=0 \,\}, \{\, z=0, x=0, y=1, w=0 \,\},$$
$$\{\, z=0, x=0, y=-1, w=0 \,\}, \{\, z=0, y=0, x=1, w=0 \,\},$$
$$\{\, z=0, y=0, x=-1, w=0 \,\}, \{\, z=0, y=0, x=0, w=1 \,\},$$
$$\{\, z=0, y=0, x=0, w=-1 \,\}$$

Deoarece funcţia h este simetrică, din cele nouă puncte se studiază doar cele cu coordonate pozitive.

```
> DDh:= matrix(4,4, (i, j) -> D[i, j](h)):
> DDh(0,0,0,0), DDh(1,0,0,0);
```

$$\begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & -2 & 0 & 0 \\ 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 6 \end{bmatrix}, \begin{bmatrix} -4\,e^{(-1)} & 0 & 0 & 0 \\ 0 & -4\,e^{(-1)} & 0 & 0 \\ 0 & 0 & 2\,e^{(-1)} & 0 \\ 0 & 0 & 0 & 4\,e^{(-1)} \end{bmatrix}$$

Deci (0,0,0,0), (1,0,0,0) şi (-1,0,0,0) sunt puncte şa.

```
> # Analog:
> # DDh(0,1,0,0), DDh(0,0,1,0), DDh(0,0,0,1);
```

Astfl (0,1,0,0) şi (0,-1,0,0) sunt puncte de minim local, (0,0,1,0) şi (0,0,-1,0), puncte şa, iar (0,0,0,1) şi (0,0,0,-1) sunt puncte de maxim local.

28. (*Puncte critice 2*)

```
> P:=(x,y,z) -> 1 -2*x +3*x^2 - x*y + x*z - z^2 +4*z + y^2 +2*y*z:
> solve({diff(P(x,y,z), x)=0,  diff(P(x,y,z), y)=0,
> diff(P(x,y,z), z)=0}, {x,y,z});
```

$$\{\, x=0, y=-1, z=1 \,\}$$

```
> DDP:= matrix(3,3, (i, j) -> D[i, j](P)) :
> DDP(0, -1, 1);
```

$$\begin{bmatrix} 6 & -1 & 1 \\ -1 & 2 & 2 \\ 1 & 2 & -2 \end{bmatrix}$$

```
> EV:=eigenvals(DDP(xy,z)):
> evalf(");
```

$$6.272612628, -3.018768109 + .1\,10^{-9}\,I, 2.746155481 + .1\,10^{-9}\,I$$

```
> evalf(EV[2]);
```

$$-3.018768109 + .1\,10^{-9}\,I$$

```
> evalf(EV[2], 50);
```

$$-3.0187681086892283920167954181492242982455820849420 - .11\,10^{-48}\,I$$

29. (*Puncte critice 3*)

```
> g:= (x,y) -> x^2 + c*(x - y)^4:
> g1:=diff(g(x,y), x):  g2:=diff(g(x,y), y):
> solve( { diff(g(x,y), x)=0, diff(g(x,y),y)=0}, {x,y});
```

$$\{\, y=0, x=0 \,\}$$

Această soluţie nu este completă deoarece Maple presupune că c este nenul.

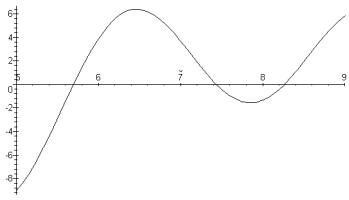


Figura 4.19:

Dacă $c = 0$ toate punctele $(0, y)$ de pe axa y sunt puncte critice.

```
> DDg:= matrix(2,2, (i, j) -> D[i, j](g)):
> DDg(0, 0);
```

$$\begin{bmatrix} 2 & 0 \\ 0 & 0 \end{bmatrix}$$

Deoarece această matrice este neinversabilă, testul derivatei secunde nu oferă informaţii suficiente. Ne spune însă că originea poate fi punct de minim local, punct şa, dar nu poate fi maxim local. Continuarea studiului se poate face doar pe cazuri. Metoda grafică ne poate ajuta în această situaţie. Trasarea funcţiei pentru o valoare $c > 0$ ne indică originea ca punct de minim local, iar pentru $c < 0$, punct şa.

```
> #plot3d(x^2 + (x-y)^4, x=-3..3, y=-3..3, view=-2..20);
> #plot3d(x^2  - (x-y)^4, x=-2..2, y=-2..2, view=-2..20);
> #plot3d(x^2 , x=-2..2, y=-2..2);
```

30. (*Ceaşca*) Fie r raza ceştii şi h înălţimea. Atunci volumul este $v = \pi r^2 h$. Deci

```
> h :=196/r^2:
```

Funcţia de cost care trebuie minimizată este

```
> c := 3*Pi*r^ 2 + 1*2*Pi*r*h:
```

Determinăm punctele critice şi trasăm graficul costului funcţie de rază:

```
> cp := diff(c,r):
> fsolve(cp,r);
```

4.027587079

```
> # plot(c,r=3..5);
```

Deci raza optimă este 4.027 cm, iar înălţimea:

```
> subs(r=4,h);
```

$$\frac{49}{4}$$

31. (*Maxmin*)

```
>  x:='x': f(x):='f(x)': f:=x->6*cos(2*x)+(5*x*sin(x)/(x+1)):
> plot(f(x),x=5..9);
```

Grafic: vezi figura 4.19

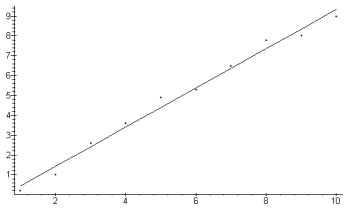


Figura 4.20:

```
> dfx:=diff(f(x),x):
> assign(fsolve(dfx=0,{x},x=7..9)): x1:=x; x:='x':
```

$$x1 := 7.850719243$$

```
> assign(fsolve(dfx=0,{x},x=5..7)): x2:=x; x:='x':
```

$$x2 := 6.465305279$$

```
> if evalf(f(5))< f(x1) then minim:=evalf(f(5),7)
> else minim:=evalf(f(x1),7) fi;
```

$$minim := -9.029947$$

```
> if f(x2) > evalf(f(9)) then maxim:=evalf(f(x2),7)
> else maxim:=evalf(f(9),7) fi;
```

$$maxim := 6.390640$$

32. (*Suma minimă*)

```
> restart: m:=10:
> G:=sum((y[i]-a*x[i]-b)^2,i=1..m):
> sol:=solve({diff(G,a)=0,diff(G,b)=0},{a,b}):
> x:=array(1..m,[seq(i,i=1..10)]):
> y:=array(1..m,[0.2,1.0,2.6,3.6,4.9,5.3,6.5,7.8,8.0,9.0]):
> assign(sol);
> a, b;
```

$$.9896969696,\ -.5533333333$$

```
> date:=seq([x[i],y[i]],i=1..m):
> gr1:=plot([date],style=POINT):
> gr2:=plot([seq([x[i],a*x[i]+b],i=1..m)]):
> plots[display]({gr1,gr2});
```

Grafic: vezi figura 4.20

```
> restart: m:=10:
> H:=sum((z[i]-a*x[i]-b*y[i]-c)^2,i=1..m):
> x:=array(1..m,[seq(rand() mod 10,i=1..m)]):
> y:=array(1..m,[seq(rand() mod 10, i=1..m)]):
> z:=array(1..m,[seq(rand() mod 10, i=1..m)]):
```

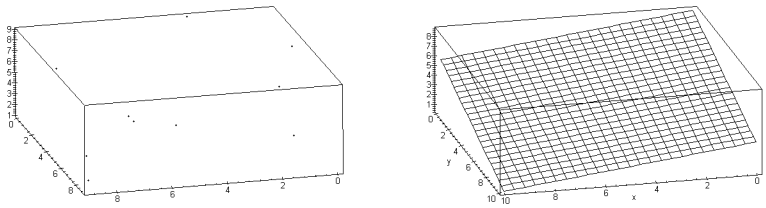


Figura 4.21:

```
> plots[pointplot]({seq([x[i],y[i],z[i]],i=1..m)},axes=BOXED);
```

Grafic: vezi figura 4.21.a

```
> solve({diff(H,a)=0, diff(H,b)=0, diff(H,c)=0},{a,b,c});
```

$$\left\{ b = \frac{-17996}{33217}, c = \frac{292738}{33217}, a = \frac{-10149}{33217} \right\}$$

```
> assign(''); f:=(x,y)->a*x+b*y+c:
> x:='x': y:='y': plot3d(f(x,y),x=0..10,y=0..10,axes=BOXED);
```

Grafic: vezi figura 4.21.b

33. (*Jacobian*)

```
> T := (x,y) ->  [x*y,x^2+y^2,0] :
> with(linalg):
> dT := stack(op(map(grad,T(x,y),[x,y])));
```

$$dT := \begin{bmatrix} y & x \\ 2\,x & 2\,y \\ 0 & 0 \end{bmatrix}$$

34. (*Continuitate*)

```
> restart; abs(x^2-4)<epsilon;
```

$$\left|x^2 - 4\right| < \varepsilon$$

```
> subs(epsilon=0.1,'');solve('');
```

$$\left|x^2 - 4\right| < .1$$

$$\{ -2.024845673 < x, x < -1.974841766 \}, \{ x < 2.024845673, 1.974841766 < x \}$$

```
> subs(epsilon=0.001,'');solve('');
```

$$\left|x^2 - 4\right| < .1$$

$$\{ -2.024845673 < x, x < -1.974841766 \}, \{ x < 2.024845673, 1.974841766 < x \}$$

35. (*Împrumut*)

```
> A := 'A':  # suma initiala imprumutata.
> a := 'a':  # suma totala datorata dupa t ani
> k := 'k':  # rata anuala de plata
> t := 't':  # timpul masurat in ani
> r := 'r':  # rata anuala de dobanda
```

Ecuaţia diferenţială care guvernează rata de schimb a banilor datoraţi la timpul t este $da/dt = ra - k$.

```
> diffeq := diff(a(t),t) = r*a(t)-k:
> sol := dsolve({diffeq,a(0)=A},a(t));
```

$$sol := \mathrm{a}(t) = \frac{k}{r} + \mathrm{e}^{(r\,t)}\left(-\frac{k}{r} + A\right)$$

```
> deplatit:=unapply(simplify(solve(subs(t=T,rhs(sol)),k)),(T,A,r));
```

$$deplatit := (T, A, r) \to \frac{\mathrm{e}^{(r\,T)}\,A\,r}{-1 + \mathrm{e}^{(r\,T)}}$$

```
> deplatit(3.,8000.,.1); #per an
```

3086.636727

Acest rezultat se obţine în cazul unei plăţi continue. O asemenea situaţie nu este reală. Plata se face de obicei lunar. În acest caz, soluţia trebuie recalculată:

```
> A := 8000.:  T := 3.: r := .1: k := 'k':
> for n from 1 to 12*T do
> A := exp(r*1/12.)*A - k/12. od:
```

Se consideră cazul a 36 luni de plată. k este divizat în 12 rate egale.

```
> solve(A,k);
```

3099.533514

Se observă că diferenţa dintre cele două soluţii este minoră.

36. (*Haos*)

```
> stairs := proc(r,m,s)
>  local f,u0,x,u1,stairstep,j;
>  f := unapply(r*x*(1-x),x);
>  u0 := s; u1 := f(u0):
>  stairstep := u0,0,u0,u1,u1,u1;
>  for j from 1 to m do
>   u0 := u1; u1 := f(u0);
>   stairstep := stairstep,u0,u1,u1,u1 od:
>  [stairstep]
> end:
> k := .5: m:=10: st:=.75:
> plot({x->k*x*(1-x),x->x,stairs(k,m,st)},0..1,style=line,
> scaling= constrained, color=black);
```

Grafic: vezi figura 4.22

```
> for i from 1 to 8 do k := .8 + i*.2;
> f := x -> x*k*(1-x):
> pl.i := plot({x->x,f,stairs(k,10+3*i,.3)},0..1, style=line) od:
> #plots[display]([seq(pl.i,i=1..8)],insequence=true);
```

37. (*Bila de biliard*) Punctul de contact cu masa este cel pentru care se realizează minimul sumei distanţelor de la acel punct la cele două bile (simetricul faţă de

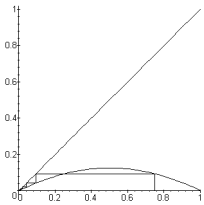


Figura 4.22:

margine a celei de a doua bile trebuie să fie în linie dreaptă cu prima bilă şi punctul de contact).

```
> restart;
> LP:=sqrt((X(t)-Px)^2+(Y(t)-Py)^2):
> LQ:=sqrt((X(t)-Qx)^2+(Y(t)-Qy)^2):
> lP:=diff(LP,t): lQ:=diff(LQ,t):
> dll:=numer(lP)*denom(lQ)+numer(lQ)*denom(lP):
> X1(t):=t:Y1(t):=-1:X2(t):=1:Y2(t):=t:X3(t):=t:Y3(t):=1:
> X4(t):=-1:Y4(t):=t:
> Px:=1/2: Py:=1/2: Qx:=-3/5: Qy:=0:
> X(t):=X1(t): Y(t):=Y1(t): dl:=simplify(dll):
> sol[1]:={solve(dl)}; solutii:=[[op(sol[1]),-1]]:
```

$$sol_1 := \left\{ \frac{-4}{25} \right\}$$

```
> X(t):=X2(t): Y(t):=Y2(t): dl:=simplify(dll):
> sol[2]:={solve(dl)}; solutii:=[op(solutii),[1,op(sol[2])]]:
```

$$sol_2 := \left\{ \frac{8}{21} \right\}$$

```
> X(t):=X3(t): Y(t):=Y3(t): dl:=simplify(dll):
> sol[3]:={solve(dl)};solutii:=[op(solutii),[op(sol[3]),1]]:
```

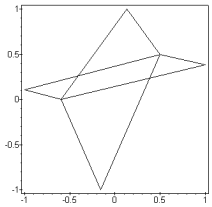
$$sol_3 := \left\{ \frac{2}{15} \right\}$$

```
> X(t):=X4(t): Y(t):=Y4(t): dl:=simplify(dll):
> sol[4]:={solve(dl)};solutii:=[op(solutii),[-1,op(sol[4])]]:
```

$$sol_4 := \left\{ \frac{2}{19} \right\}$$

```
> P:=[Px,Py]: Q:=[Qx,Qy]:
> plot({[P,solutii[1],Q],[P,solutii[2],Q],[P,solutii[3],Q],
> [P,solutii[4],Q]},axes=BOXED);
```

Grafic: vezi figura 4.23.a



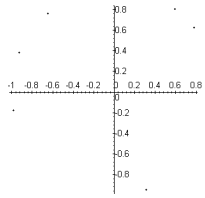


Figura 4.23:

```
> X(t):=sin(t): Y(t):=cos(t): dl:=simplify(dll):
> expr:=solve(dl,t);
```

$$expr := 2\arctan(\,\mathrm{RootOf}(\,2\_Z^4 - 11\,\_Z^3 - 9\,\_Z^2 + \_Z + 1\,)\,),\quad 2\arctan(\,\mathrm{RootOf}(\,7\,\_Z^2 + 5\,\_Z - 4\,)\,)$$

```
> soll:={allvalues(expr[1]),allvalues(expr[2])};
```

$$soll := \left\{2\arctan\left(\frac{11}{8} - \frac{3}{8}\sqrt{17} - \frac{1}{8}\sqrt{354 - 82\sqrt{17}}\right),\right.$$
$$2\arctan\left(\frac{11}{8} - \frac{3}{8}\sqrt{17} + \frac{1}{8}\sqrt{354 - 82\sqrt{17}}\right),$$
$$2\arctan\left(\frac{11}{8} + \frac{3}{8}\sqrt{17} - \frac{1}{8}\sqrt{354 + 82\sqrt{17}}\right),$$
$$2\arctan\left(\frac{11}{8} + \frac{3}{8}\sqrt{17} + \frac{1}{8}\sqrt{354 + 82\sqrt{17}}\right), 2\arctan\left(-\frac{5}{14} + \frac{1}{14}\sqrt{137}\right),$$
$$\left.2\arctan\left(-\frac{5}{14} - \frac{1}{14}\sqrt{137}\right)\right\}$$

```
> pct:=seq([sin(soll[i]),cos(soll[i])],i=1..6):
> plot([pct],style=POINT);
```

Grafic: vezi figura 4.23.b

38. (*Trei bile*)

```
> with (linalg): with (plots):
```

Fie $p = mv$ impulsul bilei grele înainte de impact. Presupunând că impulsul se conservă, suma impulsurilor celor trei bile după impact trebuie să fie egală cu p. Fie p_2 impulsul bilei grele, `mar1` impulsul primei bile lovite şi `mar2` impulsul celei de a doua din enunţ.

```
>  v := vector([2,0]):  p := evalm(90*v);
```

$$p := [\,180\;0\,]$$

```
> mar1 := evalm(20*3*vector([cos(Pi/4), sin(Pi/4)]));
```

$$mar1 := \left[30\sqrt{2}\;30\sqrt{2}\right]$$

```
> mar2:= evalm(25*5*vector([cos(-theta), sin(-theta)]));
```

$$mar2 := [\,125\cos(\,\theta\,)\;-125\sin(\,\theta\,)\,]$$

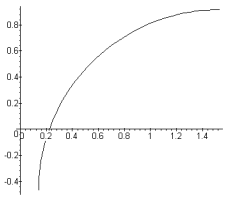
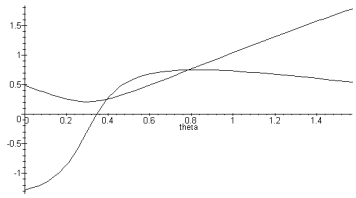



Figura 4.24:

```
> p2:=evalf(evalm(p -mar2-mar1));
```

$$p2 := [\,137.5735931 - 125.\cos(\theta)\ 125.\sin(\theta) - 42.42640686\,]$$

Viteza bilei grele după impact este 1/90 din $p2$.

```
> vel := evalm(p2/90);
```

$$vel := [\,1.528595479 - 1.388888889\cos(\theta)\ 1.388888889\sin(\theta) - .4714045206\,]$$

Dacă θ variază de la 0 la $\pi/2$, componenta x a vitezei vel creşte de la un minim de 0.14 la maxim 1.52, iar componenta pe y de la minim 0.47 la maxim 0.91. Orientarea vectorului indică pentru un θ un punct de pe cercul de centru (1.52,-0.47) şi rază 1.3(8).

```
> plot({[vel[1],vel[2],theta=0..Pi/2],[0,0]},scaling=constrained);
```

Grafic: vezi figura 4.24.a

```
> ang:=evalf(arctan(vel[2]/vel[1])):
> spd:=evalf(sqrt(dotprod(vel,vel))):
> plot({spd,ang},theta=0..Pi/2);
```

Grafic: vezi figura 4.24.b

```
> tminspd := fsolve(diff(spd,theta),theta,.2..(.4));
```

$$tminspd := .2991367319$$

```
> evalf(subs(theta=tminspd,ang)), ` radiani`;
> evalf(subs(theta=tminspd,spd)), ` metrii pe seconda`;
```

$-.2991367313$, *radiani*

$.2107443070$, *metrii pe seconda*

```
> tmaxang := fsolve(diff(ang,theta),theta,.7..(.9));
> evalf(subs(theta=tmaxang,ang)), ` radiani`;
> evalf(subs(theta=tmaxang,spd)), ` metrii pe seconda`;
```

$$tmaxang := .8182597410$$

$.7525365864$, *radiani*

$.7936082254$, *metrii pe seconda*

```
> mar := (x,n) -> plot( [x[1],x[2]],style=point,
> symbol=op(n,[box,cross,diamond])):
> s := (a,V,t)-> evalm(a+t*V):
```

Pentru a simula mişcarea, prima bilă va fi plasată în (0,0), iar celelalte două la (4,0). Utilizând informaţia de viteză, se crează 9 imagini reprezentând un

interval de 4 secunde: primele două înainte de impact şi două după.

```
> movie := NULL:
> for t from 0 by 1/2 to 2 do
> movie := movie,plots[display]([mar( s([0,0],v,t),1 ),
>  mar( s([4,0],mar1/20,0),2 ),mar( s([4,0 ],mar2/25,0),3 )],
>  scaling=constrained) od:
> theta := Pi/6:
> for t from 0 by 1/2 to 2 do
> movie :=
> movie,plots[display]([mar( s([4,0],vel,t),1 ),
>  mar( s([4,0],mar1/20,t),2 ),mar( s([4,0 ],mar2/25,t),3 )],
>  scaling=constrained) od:
> theta := 'theta':
> # plots[display]([movie],insequence=true,axes=none);
```

Pentru un θ general:

```
> mkmovie := proc(th)
> local movie,t;
>     movie := NULL;
>     for t from 0 by 1/4 to 2 do
>         movie := movie,plots[display]([mar(s([0,0],v,t),1),
>          mar(s([4,0],1/20*mar1,0),2),
>          mar(s([4,0],1/25*mar2,0),3)],scaling = constrained)
>     od;
>     for t from 0 by 1/4 to 2 do
>         movie := movie,plots[display]([
>             mar(s([4,0],subs(theta = th,op(vel)),t),1),
>             mar(s([4,0],1/20*mar1,t),2),
>             mar(s([4,0],1/25*subs(theta = th,op(mar2)),t),3)],
>             scaling = constrained)
>     od;
>     plots[display]([movie],insequence = true,axes = none)
> end;
> #mkmovie(.8*Pi/2);
```

39. (*Transformarea conformă a cercului*)

```
> p(z(t))=P(t):
> pprime:=solve(diff(",t),D(p)(z(t))):
> p2prime:=solve(diff("",t,t),D(D(p))(z(t))):
> D(p)(z(t)):=pprime: p2prime:
> ode:=z(t)^2*p2prime+z(t)*pprime+(a*z(t)^2+b*z(t)+c)*P(t):
> z(t):=r*exp(I*t): P(t):=U(t)+I*V(t):
> re:=combine(evalc(Re(ode)),trig):
> im:=combine(evalc(Im(ode)),trig):
> re:=collect(collect(re,U(t)),V(t))=0:
```

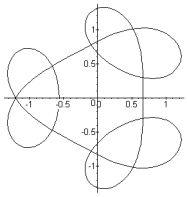


Figura 4.25:

```
> im:=collect(collect(im,U(t)),V(t))=0:
> odes:=re,im:
> init:=V(0)=0, U(0)=-0.563, D(V)(0)=0.869, D(U)(0)=0:
> a:=1: b:=0.5: c:=-0.4444444443: r:=1:
> F:=dsolve({odes, init}, {U(t), V(t)}, numeric):
> Tend:=6*Pi: numpts:=200: step:=Tend/numpts: L:=[]: t:=t:
> for j from 0 to numpts do
>    T:=j*step: x:=F(T): L:=[L[],[rhs(x[2]),rhs(x[4])]]:
> od:
> plot(L);
```

Grafic: vezi figura 4.25

40. (*Compresiunea metalului*)

```
> a:='a': b:='b': c:='c': V0:=Pi*r0^2*h0:
> R:=y->a*y^2+b*y+c:
> V1:=Pi*int(R(y)^2,y=0..h1):
> eqns:={V1=V0, R(0)=r1, R(h1)=r1}:
> Sol:=solve(eqns, {a,b,c});
```

$$Sol := \{c = r1, a = \mathrm{RootOf}(\,\_Z^2\, h1^5 + 30\, r1^2\, h1 - 10\, r1\, \_Z\, h1^3 - 30\, r0^2\, h0\,),$$
$$b = -\mathrm{RootOf}(\,\_Z^2\, h1^5 + 30\, r1^2\, h1 - 10\, r1\, \_Z\, h1^3 - 30\, r0^2\, h0\,)\, h1\,\}$$

```
> ax:=subs(Sol,a): bx:=subs(Sol,b):
> a12:=allvalues(ax): b12:=allvalues(bx):
> Sol1:=subs(ax=a12[1], bx=b12[1], Sol):
> Sol2:=subs(ax=a12[2], bx=b12[2], Sol):
> subs(Sol1, r1=2*r0, h1=h0/4, R(h1/2));
```

$$-8\,\frac{\frac{5}{16}\, r0\, h0^3 + \frac{1}{512}\,\sqrt{25}\,\sqrt{1024}\,\sqrt{r0^2\, h0^6}}{h0^3} + 2\, r0$$

```
> Rmax:=simplify(",power,symbolic);
```

$$Rmax := -3\, r0$$

```
> subs(Sol2, r1=2*r0, h1=h0/4, R(h1/2));
```

$$-8\,\frac{\frac{5}{16}\,r0\,h0^3-\frac{1}{512}\,\sqrt{25}\,\sqrt{1024}\,\sqrt{r0^2\,h0^6}}{h0^3}+2\,r0$$

```
> Rmax:=simplify(",power,symbolic);
```

$$Rmax := 2\,r0$$

```
> assign(Sol2);
> r1:=r0*(1+k*(sqrt(h0/h1)-1)):
> b1:=subs({1+k*(sqrt(h0/h1)-1)=K,h0=N*h1},b):
> b2:=simplify(b1,power,symbolic);
```

$$b2 := -\frac{r0\,\left(5\,K-\sqrt{5}\,\sqrt{-K^2+6\,N}\right)}{h1}$$

```
> b3:=factor(simplify(subs(-5=-q^2, 5=q^2, b2), symbolic));
```

$$b3 := -\frac{r0\,q\,\left(K\,q-\sqrt{-K^2+6\,N}\right)}{h1}$$

```
> b4:=subs(K=1+k*(sqrt(h0/h1)-1), N=h0/h1, q=sqrt(5), b3):
> b:=b4:
> a:=-b/h1:
> Z:=unapply(R(y),y,r0,h0,h1,k):
> BodyPlot:= proc(Shape,Rin,Hin,Hfin,Nsteps,Kvec)
>   local Nst,Nfr,Rf,j,Kfr,i,n,Hi,F,Ls,Rs,Top,Bot,LeftSides,
>         RightSides, TopSides, BottomSides;
>   Nfr:= nops(Kvec); Rf:=0;
>   for j from 1 to Nfr do
>      Kfr:=Kvec[j];
>      Rf:=Rf+2*round(Shape(Hfin/2,Rin,Hin,Hfin,Kfr)+1);
>      for i from 0 to Nsteps do
>         n:=(j-1)*(Nsteps+1)+i;
>         Hi:=Hfin+(Hin-Hfin)*i/Nsteps;
>         F(y):=Shape(y,Rin,Hin,Hi,Kfr);
>         F(0):=Shape(0,Rin,Hin,Hi,Kfr);
>         Ls[n]:=[-F(y)+Rf,y,y=0..Hi];
>         Rs[n]:=[F(y)+Rf,y,y=0..Hi];
>         Top[n]:=[-F(0)+Rf,Hi,F(0)+Rf,Hi];
>         Bot[n]:=[-F(0)+Rf,0,F(0)+Rf,0];
>       od; od;
>       LeftSides:=seq(Ls[i],i=0..n):
>       RightSides:=seq(Rs[i],i=0..n):
>       TopSides:=seq(Top[i],i=0..n):
>       BottomSides:=seq(Bot[i],i=0..n):
>     plot({LeftSides,RightSides,TopSides,BottomSides});
> end:
> BodyPlot(Z,1,3,1,5,[0,1/5]);
```

Grafic: vezi figura 4.26.a

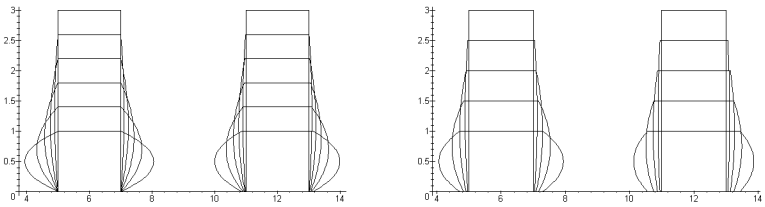


Figura 4.26:

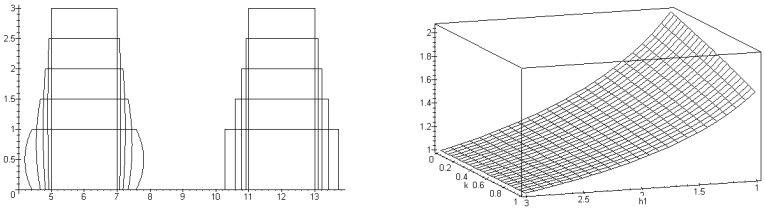


Figura 4.27:

```
> BodyPlot(Z,1,3,1,4,[2/5,3/5]);
```
Grafic: vezi figura 4.26.b
```
> BodyPlot(Z,1,3,1,4,[4/5,1]);
```
Grafic: vezi figura 4.27.a
```
> plot3d(Z(h1/2,1,3,h1,k),h1=1..3,k=0..1,axes=BOXED);
```
Grafic: vezi figura 4.27.b

41. (*Mingea de tenis*)
```
> restart; g:=9.81: d:=0.063: m:=0.05:
>  ro:=1.29: alpha:=evalf(Pi*d^2/(8*m)*ro):
> v:=(dx^2+dz^2)^(1/2):
> var:={x(t),z(t),dx(t),dz(t)}:
> initc:=x(0)=0, z(0)=h, dx(0)=v0*cos(theta), dz(0)=v0*sin(theta):
> eqnt:=diff(x(t),t)=dx(t),diff(dx(t),t)=-0.508*alpha*dx(t)*v(t),
> diff(z(t),t)=dz(t), diff(dz(t),t)=-g-0.508*dz(t)*alpha*v(t):
> h:=1: v0:=25: theta:=Pi/180*15:
> res:=dsolve({eqnt,initc},var,numeric):
> rezvid:=dsolve({diff(x(t),t$2)=0, diff(z(t),t$2)=-g, x(0)=0, z(0)
> =h, D(x)(0)=v0*cos(theta), D(z)(0)=v0*sin(theta)},{x(t),z(t)}):
> tmaxvid:=fsolve(subs(rezvid,z(t))=0,t,t=0..5);
```
$$tmaxvid := 1.458903640$$
```
> t0:=tmaxvid: tn:=t0: ts:=0.0: res(0); map(evalf,{initc});
```
$$[t = 0, \mathrm{x}(t) = 0, \mathrm{z}(t) = 1., \mathrm{dx}(t) = 24.14814566000000, \mathrm{dz}(t) = 6.470476125000000]$$

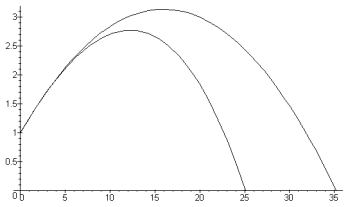


Figura 4.28:

$$\{\,\mathrm{x}(\,0\,)=0,\mathrm{z}(\,0\,)=1.,\mathrm{dx}(\,0\,)=24.14814566,\mathrm{dz}(\,0\,)=6.470476125\,\}$$

```
> while abs((tn-ts)/tn) > 10^(-4) do
>    ts:=tn; up:=res(ts); tn:=ts-rhs(up[3])/rhs(up[5]);
> od:
> tmaxaer:=tn;
```

$$tmaxaer := 1.362013705$$

```
> n:=20: pas:=tmaxaer/n: lista:=[]:
> for i from 0 to n do
>    val:=res(i*pas): lista:=[op(lista),[rhs(val[2]),rhs(val[3])]]:
> od:
> for i from 0 to n do
>    val:=res(tmaxaer+i*(tmaxvid-tmaxaer)/n):
>    lista:=[op(lista),[rhs(val[2]),0]]:
> od:
> plot({[subs(rezvid,x(t)),subs(rezvid,z(t)),t=0..tmaxvid],lista});
```

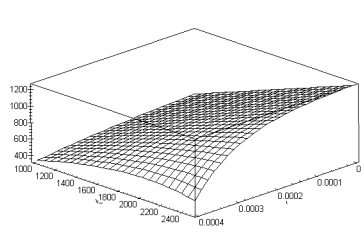
Grafic: vezi figura 4.28

42. (*Balistică 1*)

```
> restart;
> eqe:=(1/2)*rho*(v(t)-u)^2+Yp=(1/2)*rho*u^2+Rt:
> u:=solve(eqe,u);
```

$$u := \frac{\frac{1}{2}\,\rho\,\mathrm{v}(\,t\,)^2 + Yp - Rt}{\rho\,\mathrm{v}(\,t\,)}$$

```
> deqe1:=rho*l(t)*diff(v(t),t)=-Yp:
> deqe2:=diff(l(t),t)=-(v(t)-u):
> incond:=l(0)=L, v(0)=v0:
> sol:=dsolve({deqe1,deqe2,incond},{l(t),v(t)},series):
> v:=op(2,op(select(x->op(1,x)=(t)sol))):
> l:=op(2,op(select(x->op(1,x)=(t)sol))):
> uappr:=convert(series(v/2+(Yp-Rt)/(rho*v),t=0,6),polynom):
```



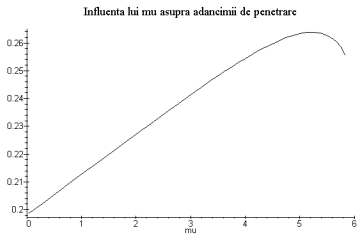


Figura 4.29:

```
> series(uappr,t=0,2);
```

$$\left(\frac{1}{2}\,v0+\frac{Yp-Rt}{\rho\,v0}\right)+\left(-\frac{1}{2}\,\frac{Yp}{\rho\,L}+\frac{(\,Yp-Rt\,)\,Yp}{\rho^2\,v0^2\,L}\right)\,t+\mathrm{O}(\,t^2\,)$$

```
> rho:=7810: L:=0.25:
> u:=subs(Rt=900*10^6, Yp=900*10^6, uappr):
> plot3d(u,t=0..0.0004,v0=1000..2500);
```

Grafic: vezi figura 4.29.a

43. (*Balistică 2*)

```
> rhot:=7810: rhop:=rhot: mp:=1.491: dp:=0.0242: Yt:=9.70*10^8:
> Yp:=15.81*10^8: Ap:=Pi*dp^2/4: v0:=1457.9: pf:=0.245:
> ap:=1/(2*rhop*Ap): a:=3*Yt*Ap: b:=rhot*Ap/2:
> vp:=v0/2+(Yt-Yp)/(rhot*v0): k:=mu->1+ap*mu: c:=mu->b-ap*mu^2:
> eps:=mu->mu*k(mu)/(2*c(mu)):
> Pf:=mu->mp*(1-(1+c(mu)*vp^2/a)^(-eps(mu)))/mu:
> plot(Pf(mu),mu=0..6,
>  title=`Influenta lui mu asupra adancimii de penetrare`);
```

Grafic: vezi figura 4.29.b

```
> mup:=fsolve(Pf(mu)=pf,mu);
```

$$mup := 3.268624353$$

```
> dP:=P->
>  sqrt(a*((1+c(mup)*vp^2/a)*(1-mup*P/mp)^(1/eps(mup))-1)/c(mup)):
> plot(dP(P),P=0..pf,
>  title=`Dependenta vitezei de penetratie de adancime`);
```

Grafic: vezi figura 4.30.a

```
> tf:=0.0006: p:='p':
> deq:=(mp-mup*p(t))*evalf(k(mup))*diff(p(t),t,t)=
> -(evalf(a)+evalf(c(mup))*diff(p(t),t)^2):
> incond:=p(0)=0, D(p)(0)=vp:
> F:=dsolve({deq,incond},p(t),numeric):
> plot([seq([i*tf/10,rhs(F(i*tf/10)[2])],i=0..10)],title=`Dependenta
>  adancimii de timp`);
```

Grafic: vezi figura 4.30.b

```
> vp1:=(vi,y)->vi/2+(y-Yp)/(rhot*vi):
```

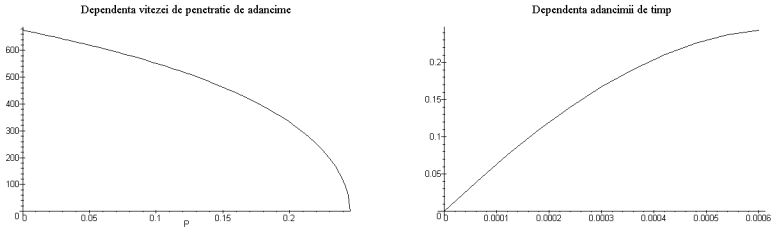


Figura 4.30:

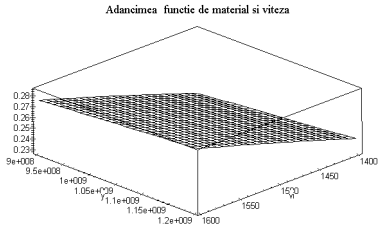


Figura 4.31:

```
> depth:=(vi,y)->mp*(1-(1+c(mup)*vp1(vi,y)^2/a)^(-eps(mup)))/mup:
> opt:=title=‘Adancimea  functie de material si viteza‘,axes=BOXED:
> plot3d(evalf(depth(vi,y)),vi=1400..1600,y=900*10^6..1200*10^6,opt);
```

Grafic: vezi figura 4.31

44. (*Polinoame ortogonale*)

```
> Lanczos:=proc(iprod,alpha,beta,n)
>   local k,q,x,p;
>   alpha:=array(1..n);
>   if n>1 then beta:=array(1..n) else beta:='beta' fi;
>   p[0]:=collect(1/sqrt(iprod(1,1,x)),x,simplify);
>   alpha[1]:=normal(iprod(x*p[0],p[0],x));
>   q:=collect((x-alpha[1])*p[0],x,simplify);
>   for k from 2 to n do
>     beta[k-1]:=normal(sqrt(iprod(q,q,x)));
>   p[k-1]:=collect(q/beta[k-1],x,simplify);
>   alpha[k]:=normal(iprod(x*p[k-1],p[k-1],x));
>   q:=collect((x-alpha[k])*p[k-1]-beta[k-1]*p[k-2],x,simplify);
>   od;
>   beta[n]:=normal(sqrt(iprod(q,q,x))):
>   p[n]:=collect(q/beta[n],x,simplify):
> end:
> a:=0: b:=infinity:
> omega:=t->exp(-t):
```

```
> iprod:=(f,g,x)->int(f*g*omega(x),x=a..b):
> for n from 1 to 4 do Lanczos(iprod,'alpha','beta',n); od;
```

$$x - 1$$

$$\frac{1}{2}\,x^2 - 2\,x + 1$$

$$\frac{1}{6}\,x^3 - \frac{3}{2}\,x^2 + 3\,x - 1$$

$$\frac{1}{24}\,x^4 - \frac{2}{3}\,x^3 + 3\,x^2 - 4\,x + 1$$

```
> [seq(alpha[i],i=1..4)], [seq(beta[i],i=1..3)];
```

$$[1, 3, 5, 7],\ [1, 2, 3]$$

```
> with(orthopoly);
```

$$[G, H, L, P, T, U]$$

```
> L(0,x), L(1,x), L(3,x), L(4,x);
```

$$1, 1 - x, \frac{3}{2}\,x^2 - 3\,x + 1 - \frac{1}{6}\,x^3, 3\,x^2 - 4\,x + 1 - \frac{2}{3}\,x^3 + \frac{1}{24}\,x^4$$

45. (*Factor prim*)

```
> largestfactor:=proc(n)
>   op(nops(ifactor(n)), ifactor(n));
> end:
> largestfactor(118277523);
```

$$(23)^2$$

```
> largestfactor(2387);
```

$$(31)$$

46. (*Timp*)

```
> var1:=proc(n) local lista,i;
>   lista:=[]:
>   for i from 1 to n do
>      lista:=[op(lista),1+3*(i-1)]:
>   od:
> end:
> var2:=proc(n) local lista,i;
>    lista:=[seq(1+3*(i-1),i=1..n)]:
> end:
> var3:=proc(n) local lista;
>    lista:=[1+3*(k-1)$k=1..n]:
> end:
> t1:=time(): var1(4000): t1:=time()-t1:
> t2:=time(): var2(40000): t2:=time()-t2:
> t3:=time(): var3(40000): t3:=time()-t3:
> t1; t2; t3;
```

$$8.000$$

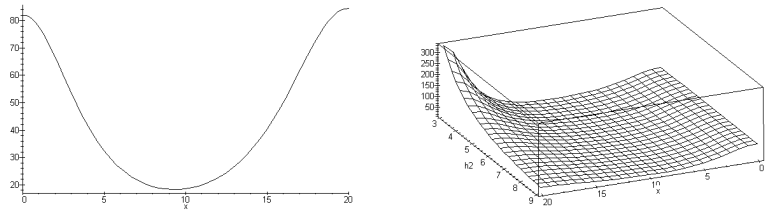


Figura 4.32:

1.000
2.000

47. (*Iluminare*)

```
> S1:=P1*h1/(h1^2+x^2)^(3/2): S2:=P2*h2/(h2^2+(s-x)^2)^(3/2):
> C:=S1+S2: dC:=diff(C,x):
> solve(dC=0,x);
> P1:=2000: P2:=3000: s:=20: h1:=5: h2:=6:
> plot(C,x=0..s);
```

Grafic: vezi figura 4.32.a

```
> xmin:=fsolve(dC=0,x,0..s); x:=xmin: C;
```

$$xmin := 9.338299136$$
$$18.24392572$$

```
> x:='x': h2:='h2': f:=unapply(C,x,h2):
> plot3d(C,x=0..s,h2=3..9, style=WIREFRAME,axes=BOXED);
```

Grafic: vezi figura 4.32.b

```
> g:=linalg[grad](C,[x,h2]):
> sol:=fsolve({g[1]=0,g[2]=0},{x,h2},{x=0..s,h2=3..9});
```

$$sol := \{\, h2 = 7.422392890, x = 9.503151310 \,\}$$

```
> assign('');
> P1:='P1': P2:='P2': h1:='h1': h2:='h2': x:='x': s:='s':
> C:=P1*h1/(h1^2+x^2)^(3/2)+P2*h2/(h2^2+(s-x)^2)^(3/2):
> g:=linalg[grad](C,[x,h1,h2]):
> sh1:={solve(g[2]=0,h1)};
```

$$sh1 := \left\{ \frac{1}{2}\sqrt{2}\,x, -\frac{1}{2}\sqrt{2}\,x \right\}$$

```
> sh2:={solve(g[3]=0,h2)};
```

$$sh2 := \left\{ \frac{1}{2}\sqrt{2}\,(\,s-x\,), -\frac{1}{2}\sqrt{2}\,(\,s-x\,) \right\}$$

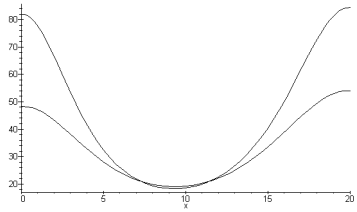


Figura 4.33:

```
> h10:=sh1[1]: h20:=sh2[1]: G:=subs(h1=h10, h2=h20,g[1]):
> P1:=2000: P2:=3000: s:=20:
> ss:=solve(G=0,x);
```

$$ss := \text{RootOf}(\_Z^3 - 24\,\_Z^2 + 480\,\_Z - 3200, 7.337374182 + 17.00932741\,I),$$
$$\text{RootOf}(\_Z^3 - 24\,\_Z^2 + 480\,\_Z - 3200, 9.325251636),$$
$$\text{RootOf}(\_Z^3 - 24\,\_Z^2 + 480\,\_Z - 3200, 7.337374182 - 17.00932741\,I),$$
$$-\text{RootOf}(-16000 - 2400\,\_Z - 120\,\_Z^2 + \_Z^3, 138.2033976)$$

```
> evalf(ss[1]), evalf(ss[2]), evalf(ss[3]), evalf(ss[4]);
```

$$7.337374182 + 17.00932741\,I, 9.325251636, 7.337374182 - 17.00932741\,I,$$
$$-138.2033976$$

```
> x:=evalf(ss[2]);
```

$$x := 9.325251636$$

```
> h1:=evalf(h10); h2:=evalf(h20);
```

$$h1 := 6.593948666$$
$$h2 := 7.548186951$$

```
> x:='x': Copt:=unapply(C,x):
> h1:=5: h2:=6: Cvar:=unapply(C,x):
> plot({Copt(x),Cvar(x)},x=0..s);
```

Grafic: vezi figura 4.33

48. (*Meciul*) Pentru a construi modelul matematic cosnsiderăm a distanta iniţială dintre atacant şi apărător, b, distanţa iniţială pe care trebuie să o parcurgă atacantul, $c = b/a$, v_1, viteza apărătorului, v_2, viteza atacantului, $v = v_2/v_1$ (< 1 pentru ca problema să aibă soluţie), $(0,0)$, poziţia iniţială a apărătorului, $(x(t), y(t))$, poziţia apărătoului la momentul t, $(a, v_2 t)$, poziţia atacantului la momentul t. Vectorul viteză a apărătorului este $(dx/dt, dy/dt)$. Atunci

$$\frac{dy/dt}{dx/dt} = \frac{v_2 t - y(t)}{a - x(t)}, \quad y(0) = 0, \quad \frac{dy}{dx}(0) = 0.$$

Fie $p = dy/dx$. Atunci

$$\frac{dp/dx}{\sqrt{1+p^2}} = \frac{v}{a-x}, \; p(0) = 0$$

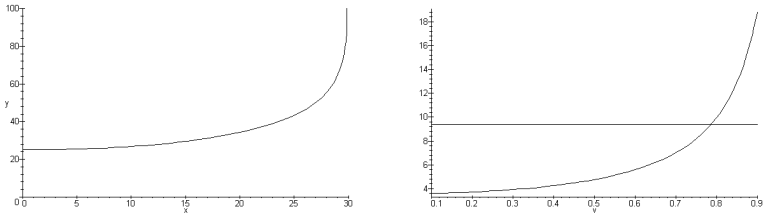


Figura 4.34:

```
> PARAM := { a=30, b=75, v1=8.4, v2=8 }:
> PARAM2 := PARAM union { v=8/8.4 }:
> dsolve({diff(p(x),x)/(sqrt(1+p(x)^2)) = v/(a-x), p(0)=0}, p(x) );
```

$$\mathrm{p}(x) = \sinh(-\ln(a-x)\,v + \ln(a)\,v)$$

```
> convert(",exp);
```

$$\mathrm{p}(x) = \frac{1}{2}\,\mathrm{e}^{(-\ln(a-x)\,v+\ln(a)\,v)} - \frac{1}{2}\,\frac{1}{\mathrm{e}^{(-\ln(a-x)\,v+\ln(a)\,v)}}$$

```
> ivp2:=combine(",ln,anything);
```

$$ivp2 := \mathrm{p}(x) = \frac{1}{2}\,(a-x)^{(-v)}\,a^{v} - \frac{1}{2}\,\frac{1}{(a-x)^{(-v)}\,a^{v}}$$

```
> explicit_soln := y = int( rhs(ivp2), x) + c2;
```

$$explicit\_soln := y = -\frac{1}{2}\,\frac{(a-x)^{(1-v)}\,a^{v}}{1-v} + \frac{1}{2}\,\frac{(a-x)^{(v+1)}}{(v+1)\,a^{v}} + c2$$

```
> const2 := simplify(solve( subs( y=0, x= 0, explicit_soln), c2));
```

$$const2 := -\frac{v\,a}{(v+1)\,(-1+v)}$$

```
> y := subs( c2  = const2, rhs(explicit_soln));
```

$$y := -\frac{1}{2}\,\frac{(a-x)^{(1-v)}\,a^{v}}{1-v} + \frac{1}{2}\,\frac{(a-x)^{(v+1)}}{(v+1)\,a^{v}} - \frac{v\,a}{(v+1)\,(-1+v)}$$

```
> plot(subs( PARAM2, y+25), x = 0..30, = 0..100);
```

Grafic: vezi figura 4.34.a

Este necesară calcularea timpului de-a lungul curbei trasare până când apărătorul şi atacantul se întâlnesc.

```
> diff_elt_arc := simplify( sqrt(1 + rhs(ivp2)^2) );
```

$$diff\_elt\_arc := \frac{1}{2}\,\sqrt{2 + \left(\frac{a^2}{a^2 - 2\,a\,x + x^2}\right)^{v} + \left(\frac{a^2}{a^2 - 2\,a\,x + x^2}\right)^{(-v)}}$$

```
> arc_length := 1/2*(-(a-x)^(v+1)*a^(-v)*v+(a-x)^(v+1)*a^(-v)
> +(a-x)^(1-v)*a^v*v+(a-x)^(1-v)*a^v-2*a)/(-v-1)/(1-v):
> simplify(subs(x = 0, arc_length));
```

$$0$$

```
> defense_time := simplify(subs(x = a, arc_length)/v1);
```

$$defense\_time := -\frac{a}{(v+1)(-1+v)\,v1}$$

```
> offense_time := b/v2;
```

$$offense\_time := \frac{b}{v2}$$

```
> evalf( subs( PARAM2, [ defense_time, offense_time ] ) );
```

$$[38.41463417, 9.375000000]$$

Concluzie: Atacantul ajunge la timp pentru a mări scorul.

```
> plot( subs( PARAM, { defense_time, offense_time } ), v=0.1..0.9 );
```

Grafic: vezi figura 4.34.b

```
> new_capture_eqn := defense_time = offense_time:
> subs( {v2=v1*v }, new_capture_eqn*v2 ):
> solve( '', {v} );
```

$$\left\{v = -\frac{1}{2}\,\frac{a+\sqrt{a^2+4\,b^2}}{b}\right\}, \left\{v = -\frac{1}{2}\,\frac{a-\sqrt{a^2+4\,b^2}}{b}\right\}$$

```
> critical_velocity := "[2]:
> subs( PARAM, op(critical_velocity) );
```

$$v = -\frac{1}{5} + \frac{1}{150}\sqrt{23400}$$

```
> evalf( '' );
```

$$v = .8198039030$$

```
> subs( '', v1=8/v );
```

$$v1 = 9.758431216$$

Aceasta este viteza pe care trebuie să o atingă apărătorul!

49. (*Arie*)

```
> line := 3-x: parabola := x^2-3*x:
> solve(line=parabola,x);
```

$$-1, 3$$

```
> Int(line-parabola,x=-1..3);
```

$$\int_{-1}^{3} 3 + 2\,x - x^2\,dx$$

```
> value('');
```

$$\frac{32}{3}$$

50. (*Volum*)

```
> volx := Int(Pi*(1-(x-10)^2),x=9..11);
```

$$volx := \int_{9}^{11} \pi\,\left(1-(x-10)^2\right)\,dx$$

```
> volx := value(volx);
```

$$volx := \frac{4}{3}\,\pi$$

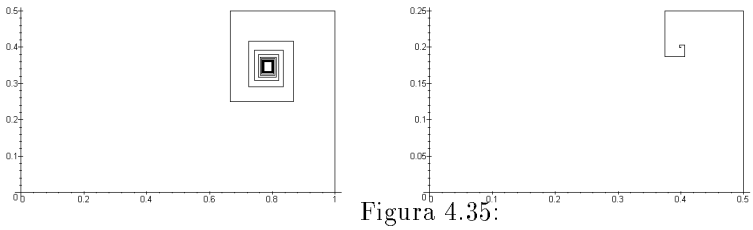


Figura 4.35:

```
> voly := Int(2*Pi*x*(1-(x-10)^2),x=9..11);
```

$$voly := \int_{9}^{11} 2\,\pi\,x\,\left(1-\left(x-10\right)^{2}\right)\,dx$$

```
> voly := value(voly);
```

$$voly := \frac{80}{3}\,\pi$$

```
> voly/volx;
```

20

51. (*Regula trapezului*)

```
> restart;
> trap := proc(f,a,b,n) local i,k,d;
>  for k to 5 do d := (b-a)/k/n;
>   print(k*n,evalf(1/2*d*sum(f(a+(i-1)*d)+f(a+i*d),i=1..k*n)))
>  od
> end:
> trap(x->exp(x^2),0,1,10);
```

$10, 1.467174693$
$20, 1.463783892$
$30, 1.463155038$
$40, 1.462934872$
$50, 1.462832952$

52. (*Drum*)

```
> cycle := proc(a,m) local path,i, dir;
>   path := 0,0; dir := 1,0;
>   for i from 1 to m do
>    path := path,path[2*i-1]+dir[1]*a(i), path[2*i]+dir[2]*a(i);
>    dir := -dir[2],dir[1];
>   od;
>   plot([path],style=LINE);
> end:
> cycle(n->1/n,40); cycle(n->1/2^n,40);
```

Grafic: vezi figura 4.35

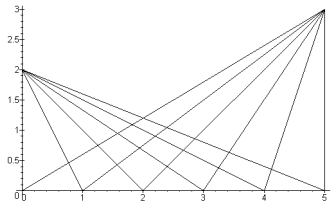


Figura 4.36:

53. (*Cele mai mici pătrate*)

```
> data := [[.3,1],[2,4],[4,2],[7,9],[10,3]]:
> with(linalg):
> A := transpose(  matrix(2,nops(data),[seq(data[i][1],
>      i=1..nops(data)),seq(1,i=1..nops(data))])):
>  A := evalf(evalm(A)):
> b := [seq(data[i][2],i=1..nops(data))]:
> xbar := evalm(inverse( transpose(A)&*A)&*transpose(A)&*b );
```

$$xbar := [\, .3430724493\ 2.201282391 \,]$$

```
> leastsqrs(A,b);
```

$$[\, .3430724479\ 2.201282393 \,]$$

54. (*Factorizare QR*)

```
> proj := proc(b,a)
>           evalm(dotprod(b,a)/dotprod(a,a)*a) end:
> A := randmatrix(4,3):
> q1 := col(A,1):
> q2 := evalm(col(A,2)-proj(col(A,2), q1 )):
> q3 := evalm(col(A,3)-proj(col(A,3), q1)-proj(col(A,3),q2)):
> for i from 1 to 3 do
>   q.i := evalm(1/sqrt(dotprod(q.i,q.i))*q.i) od:
> #Q := transpose(stack (q1,q2,q3));
> #R :=evalm(transpose(Q)&*A);
```

55. (*Două bile*) Putem considera drept axa x marginea mesei. Atunci bilele sunt plasate în punctele de coordonate (0,2) şi (5,3). Fie $(x, 0)$ punctul de pe axa x unde bila loveşte marginea mesei. Unghiul de incidenţă (0,2),(x,0),(0,0) este acelaşi cu unghiul (5,0),(x,0),(5,3) de reflexie.

```
> restart; bila := [0,2]: red := [5,3]:
>  path := x -> plot([bila,[x,0],red]) :
>  plots[display]([seq(path(i),i=0..5)],scaling=constrained);
```

Grafic: vezi figura 4.36

```
> eq1 := tan(a)=2/x:
> eq2 := tan(a)=3/(5-x):
> sol := solve({eq1,eq2},{a,x});
```

$$sol := \left\{ x = 2, a = \frac{1}{4}\,\pi \right\}$$

Deci jucătorul va trebui să lanseze bila sub un unghi de 45 de grade.

56. (*Revizuire*) Cazul 1. Presupunem că bila loveşte prima dată marginea $y = 0$. Considerăm că al doilea punct de contact cu marginea mesei este $(x2, 4)$.

Cazul 2. Bila loveşte prima dată marginea $y = 4$ (netratat).

```
> eq1 := tan(a)=2/x:
> eq2 := tan(a)=1/(5-x2):
> eq3 := tan(a)=4/(x2-x):
> solve({eq1,eq2,eq3},{x,x2,a});
```

$$\left\{ a = \arctan\left(\frac{7}{5}\right), x = \frac{10}{7}, x2 = \frac{30}{7} \right\}$$

```
> evalf(convert(arctan(7/5),degrees));
```

$$54.46232221\ degrees$$

Jucătorul trebuie să lovească mingea cu un unghi de 54.5 grade.

57. (*Turn de apă*) Fie $(c, 0, d)$ punctul de pe turn pentru care linia de la (0,0,4) la $(c, 0, d)$ este tangentă la turn. Astfel triunghiul cu vârfurile (0,0,0), $(c, 0, d)$, şi (0,0,4) este dreptunghic.

```
> eq1 := c^2 + d^2 + c^2 + (d-4)^2 = 4^2:
> eq2 := c^2+d^2=1:
> eq3 := (d-0)/(c-a)=4/(-a):
> solve({eq1,eq2,eq3},{c,d,a});
```

$$\left\{ d = \frac{1}{4}, a = \mathrm{RootOf}(\,15\_Z^2 - 16\,), c = \frac{15}{16}\,\mathrm{RootOf}(\,15\_Z^2 - 16\,) \right\}$$

Astfel $a = 4/\sqrt{(}15)$ este cea mai mică valoare pozitivă reală cerută.

58. (*Scara 1*) Fie punctul de intersecţie notat cu $(0, a)$, şi cu $(b, 3 - b^2)$ punctul în care scara atinge parabola. Scara este tangentă la parabolă în acest al doilea punct.

```
> f := x -> 3 - x^2:
> aproxpanta := (f(b+h) - f(b)) / ((b+h)  - b):
> aproxpanta := simplify(aproxpanta):
> panta1 := limit(aproxpanta,h=0);
```

$$panta1 := -2\,b$$

```
> panta2 := (3-b^2)/(b-3):
> sol := solve(panta1=panta2,b) ;
```

$$sol := 3 - \sqrt{6}, 3 + \sqrt{6}$$

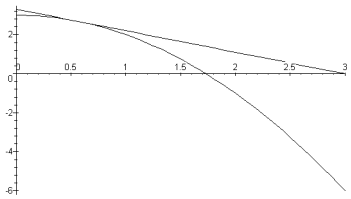


Figura 4.37:

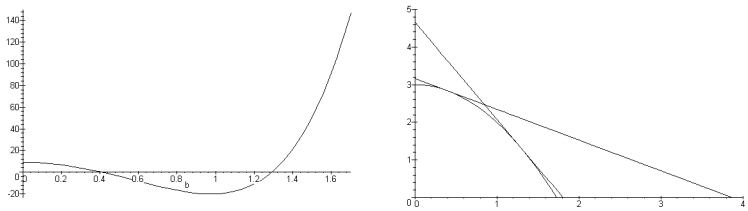


Figura 4.38:

```
> b := sol[1]:
> scara :=  x -> panta2*(x-b) + f(b):
> a := scara(0):
> evalf(a,4);
```

3.303

```
> plot({f,scara},0..3);
```

Grafic: vezi figura 4.37

59. (*Scara 2*) Considerăm punctul $(b, f(b))$ punctul de contact cu parabola (scara este tangentă la parabolă în acest punct). Utilizăm informaţia din problema anterioară conform căreia panta scării este $-2b$.

```
> f := x->3-x^2:
> panta := -2*b:
> tang := x -> -2*b*(x-b)+ f(b):
> xb := solve(tang(x)=0,x);
```

$$xb := \frac{1}{2}\,\frac{b^2+3}{b}$$

```
> yb := tang(0):
> eq1 := xb^2+yb^2=5^2:
> eq2 := expand(4*b^2*simplify(eq1)):
> eq3 := lhs(eq2)-rhs(eq2):
> plot(eq3,b=0..1.7);
```

Grafic: vezi figura 4.38.a

```
> sol := fsolve(eq3,b,0..1.5);
```

$sol := .4093966141, 1.289029267$

```
> plot({f,subs(b=sol[1],op(tang)),
>             subs(b=sol[2],op(tang))},0..4,0..5,color=black);
```

Grafic: vezi figura 4.38.b

60. (*Aria benzii Moebius*)

```
> restart; with(linalg):
> i := [1,0,0]: j := [0,1,0]: k := [0,0,1]:
> band := r ->evalm((cos(theta)*i+sin(theta)*j)*r  +
> t*(cos(theta/2)*(cos(theta)*i+sin(theta)*j)+sin(theta/2)*k)):
> #plot3d(band(2),theta=0..2*Pi,t=-1..1);
> T := convert(band(r),list):
> dT := jacobian(T,[theta,t]):
> Tu := evalm(dT*[1,0]):
> Tv := evalm(dT*[0,1]):
> cr := crossprod(Tu,Tv):
> da := simplify(sqrt(dotprod(cr,cr))):
> # int1 := int( da,t=-a..a) ;
> F:=  Int(int1,theta=0..2*Pi) :
> area := (a1,r1) -> evalf( subs({a=a1,r=r1},F)):
> area(1,2);
```

$$25.41308559$$

```
> evalf(2*(2*2*Pi));
```

$$25.13274123$$

61. (*Integrala dublă*)

```
> f := (x,y)->10*exp(sqrt(sin(y)*cos(x))):
> a := 0:  b := 1:
> bot := x -> x^2:  top := x -> x:
> g := proc(x,y)
> if not type(evalf(x*y),float) =true  then RETURN((x,y)
> elif evalf(bot(x) )<= evalf(y) and evalf(y) <= evalf(top(x))
> then 1 else 0 fi end:
> #plot3d(g(x,y)*f(x,y),x=0..1,y=0..1);
> with(student):
> Doubleint(f(x,y),y=bot(x)..top(x),x=0..1);
```

$$\int_0^1 \int_{x^2}^{x} 10\, e^{\left(\sqrt{\sin(y)\cos(x)}\right)}\, dy\, dx$$

```
> value(");
> for n from 16 by 4 to 24 do
>  evalf(value(trapezoid(trapezoid(f(x,y),y=bot(x)..top(x),n),
>  x=a..b,n))) od;
```

$$2.852622244$$

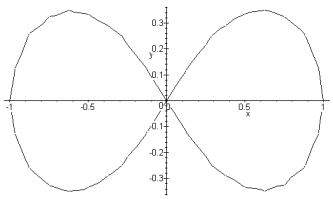
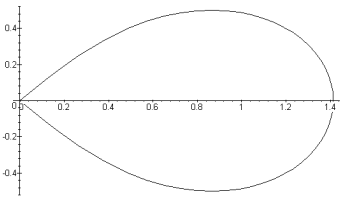



Figura 4.39:

2.856186401
2.858118400

62. (*Lemniscata*)

```
> lem := a->(x^2+y^2)^2 - 2*a*(x^2-y^2):
> plots[implicitplot](lem(.5),x=-2..2,y=-2..2,numpoints=1000);
```

Grafic: vezi figura 4.39.a

```
> polem := subs({x=r*cos(theta),y=r*sin(theta)},lem(a)):
> simplify(polem);
```

$$r^4 - 4\,a\,r^2\cos(\,\theta\,)^2 + 2\,a\,r^2$$

```
> polem :=  simplify(polem/r^2) ;
```

$$polem := r^2 - 4\,a\cos(\,\theta\,)^2 + 2\,a$$

```
> sol := solve(polem,r);
```

$$sol := \sqrt{4\,a\cos(\,\theta\,)^2 - 2\,a}, -\sqrt{4\,a\cos(\,\theta\,)^2 - 2\,a}$$

```
> f:= unapply(sol[1],a):
> plot([f(1),theta,theta=-Pi/4..Pi/4],coords=polar);
```

Grafic: vezi figura 4.39.b

```
> int(int(r,r=0..f(a)),theta=-Pi/4..Pi/4);
```

$$a$$

63. (*Convergenţa sumelor parţiale*)

```
> restart; with(plots):
> par_sum := proc( TERM, INDEX:name=range )
>   local k, lo, hi, n;
>   k := lhs(INDEX);
>   lo := op(1,rhs(INDEX));
>   hi := op(2,rhs(INDEX));
>   seq( sum(TERM, k=lo..n), n=lo..hi )
> end:
> par_sum_plot:=proc(TERM,INDEX:name=range,DOMAIN:name=range)
>   local opts;
```

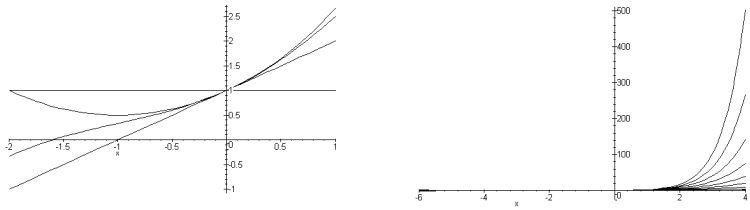


Figura 4.40:

```
>   opts := op( subsop(1=NULL,2=NULL,3=NULL, [args] ));
>   display(map(plot,[par_sum(TERM,INDEX)],DOMAIN),opts)
> end:
> par_sum( x^n/n!, n=0..3 );
```

$$1, 1+x, 1+x+\frac{1}{2}\,x^2, 1+x+\frac{1}{2}\,x^2+\frac{1}{6}\,x^3$$

```
> par_sum_plot( x^n/n!, n=0..3, x=-2..1 );
```

Grafic: vezi figura 4.40.a

```
> #par_sum_plot( x^n/n!, n=0..3, x=-2..1,  insequence=true );
> TERM := ln(n)/n/3^n*(x+2)^n:
> par_sum( TERM, n=1..3 );
```

$$0, \frac{1}{18}\ln(2)\,(x+2)^2, \frac{1}{18}\ln(2)\,(x+2)^2+\frac{1}{81}\ln(3)\,(x+2)^3$$

```
> #par_sum_plot( TERM, n=1..10, x=-6...4, insequence=true );
> par_sum_plot( TERM, n=1..10, x=-6..4 );
```

Grafic: vezi figura 4.40.b

64. (*Bara de metal*) Modificarea temperaturii de la un minut la altul se face conform legii $t_{n+1} = 0.9/3(v + At_n)$, unde $v = (100, 0, 150)$ şi

$$A = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{pmatrix}.$$

```
> restart;with(linalg):
> tridiag := proc(a,b,c,n)
>   local B, i, j;
>   B := matrix(n,n);
>   for i from 1 to n do for j from 1 to n do
>      if j = i -1 then B[i,j] := a
>      elif   j  = 1+i then B[i,j] := c
>      elif  j = i  then B[i,j] :=b else B[i,j] := 0 fi od od;
>   evalm(B);
>   end:
> n := 3: A :=tridiag(1.,1.,1.,n):
```

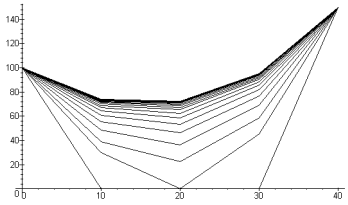


Figura 4.41:

```
> v := vector([ 100,seq( 0,i=1..n-2),150]):
> step := vector([0,0,0]):
> step := evalm(.9/3*(v+A&*step));
```

$$step := [30.00000000\ 0\ 45.00000000]$$

```
> for i from 1 to 10 do step := evalm(.9/3*(v+A&*step)) ;
>   print(evalm(step));
>    od :
```

$$[39.00000000\ 22.50000000\ 58.50000000]$$
$$[48.45000000\ 36.00000000\ 69.30000000]$$
$$[55.33500000\ 46.12500000\ 76.59000000]$$
$$[60.43800000\ 53.41500000\ 81.81450000]$$
$$[64.15590000\ 58.70025000\ 85.56885000]$$
$$[66.85684500\ 62.52750000\ 88.28073000]$$
$$[68.81530350\ 65.29952250\ 90.24246900]$$
$$[70.23444780\ 67.30718850\ 91.66259745]$$
$$[71.26249089\ 68.76127014\ 92.69093580]$$
$$[72.00712830\ 69.81440904\ 93.43566177]$$

```
> plotstep := proc(step,n) local i;
>  plot([[0,0],[0,100],seq([10*i,step[i]],i = 1 .. n),[40,150]])
> end:
> n := 3: A :=tridiag(1.,1.,1.,n):
> v := vector([ 100,seq( 0,i=1..n-2),150]):
> step := vector([0,0,0]):
> movie := plotstep(step,n):
> step := evalm(.9/3*(v+A&*step)):
> movie := movie,plotstep(step,n):
> for i from 1 to 30 do step := evalm(.9/3*(v+A&*step)) ;
>   movie := movie, plotstep(step,n): od:
> plots[display]([movie]);
```

Grafic: vezi figura 4.41

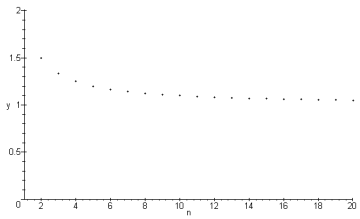


Figura 4.42:

```
> # plots[display]([movie],insequence=true);
```

65. (*Limita unui şir*)

```
> restart: with(plots):
> Sequence := proc( expr, arg:name=range )
>    local f, i, ii, s, S, var, lo, hi, m, M;
>    var:=lhs(arg); lo:=op(1,rhs(arg)); hi:=op(2,rhs(arg));
>    f := unapply( expr, var );
>    S := [seq( [i, evalf(f(i))], i=lo..hi )];
>    m := min( 0, seq( op(2,s), s=S ) );
>    M := max( 0, seq( op(2,s), s=S ) );
>    print( S );
>    print(Limit(f(var),var=infinity)=limit(f(ii),ii=infinity));
>    plot( S, var=lo..hi, y=m..M, style=POINT );
> end:
> Sequence( 1+1/n, n=1..20 );
```

$$[[1, 2.], [2, 1.500000000], [3, 1.333333333], [4, 1.250000000],$$
$$[5, 1.200000000], [6, 1.166666667], [7, 1.142857143],$$
$$[8, 1.125000000], [9, 1.111111111], [10, 1.100000000],$$
$$[11, 1.090909091], [12, 1.083333333], [13, 1.076923077],$$
$$[14, 1.071428571], [15, 1.066666667], [16, 1.062500000],$$
$$[17, 1.058823529], [18, 1.055555556], [19, 1.052631579],$$
$$[20, 1.050000000]]$$

$$\lim_{n\to\infty} 1 + \frac{1}{n} = 1$$

Grafic: vezi figura 4.42

66. (*Aria torului*)

```
> restart; with (linalg):
> i := [1,0,0]: j := [0,1,0]: k := [0,0,1]:
> tor :=convert(evalm(r1*(cos(theta)*i + sin(theta)*j) +
>  r2*(cos(phi)*(cos(theta)*i+sin(theta)*j)+sin(phi)*k)),list):
> T := (theta,phi) -> tor:
```

```
> dT := jacobian(T(theta,phi),[theta,phi]):
> Tu := evalm(dT*[1,0]): Tv := evalm(dT*[0,1]):
> da := sqrt(dotprod(Tu,Tu)*dotprod(Tv,Tv)-dotprod(Tu,Tv)):
> da := simplify(da,trig,radical);
```

$$da := \sqrt{r2^2\,(\,r1 + r2\cos(\,\phi\,)\,)^2}$$

```
> assume( r1>0,r2>0 );
> additionally(theta>=0,phi>=0,theta<=2*Pi,phi<=2*Pi);
> area2 := int(int(da,theta=0..2*Pi),phi=0..2*Pi);
```

$$area2 := 4\,r2\_\pi^2\,r1\_$$

67. (*Numere aleatoare*)

```
> s := 0 :
> for i from 1 to 1000 do s := s+(rand()/999999999999.)^2 od:
> s/1000;
```

.3395842263

```
> # sau
>    av := proc(f,a,b,n)
>     local i,dx;
>     dx := (b-a)/n;
>     evalf(sum(f(a+i*dx),i=1..n)/n);
>     end:
> for i by 10 from 10 to 40 do print(i, av(x->x^2,0,1,i)) od;
```

$10, .3850000000$
$20, .3587500000$
$30, .3501851852$
$40, .3459375000$

68. (*Aria dintre parabole*)

```
> restart;
> f := x -> a-x^2: g := x-> x^2:
> sol := solve(f(x)=g(x),x);
```

$$sol := -\frac{1}{2}\sqrt{2}\sqrt{a}, \frac{1}{2}\sqrt{2}\sqrt{a}$$

```
> area := int(a-2*x^2,x=sol[1]..sol[2]);
```

$$area := \frac{2}{3}\,a^{3/2}\sqrt{2}$$

```
> solve(area=100,{a});
```

$$\{\,a = 75^{2/3}\,2^{1/3}\,\}$$

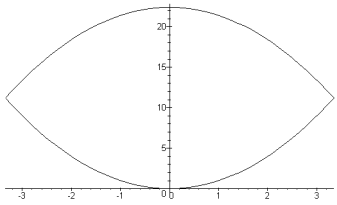


Figura 4.43:

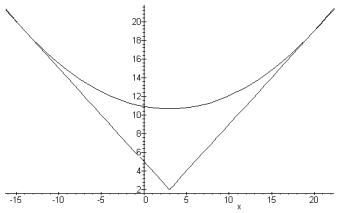


Figura 4.44:

```
> assign(");
> plot({f,g},sol[1]..sol[2]);
```

Grafic: vezi figura 4.43

69. (*Parabola şi funcţia modul*) Axa parabolei este $x = -b/(2a)$. Aceasta este şi axa pentru $y = 2 + |x - 3|$, astfel încât $-b/(2a) = 3$, sau $b = -6a$. Punctul în care panta parabolei este 1 se află pe ambele grafice. Fie (x_0, y_0) este acest punct. Atunci $y_0 = 2 + x_0 - 3 = x_0 - 1 = ax_0^2 - 6ax_0 + c$ şi $1 = 2ax_0 - 6a$.

```
> eq1 := x0-1 = a*x0^2 -6*a*x0 + c:
> eq2 := 1 = 2*a*x0 - 6*a:
> ac := solve({eq1,eq2},{a,c});
```

$$ac := \left\{ a = \frac{1}{2}\,\frac{1}{x0-3},\ c = \frac{1}{2}\,\frac{x0^2 - 2\,x0 + 6}{x0-3} \right\}$$

```
> eq3 := int (a*x^2-6*a*x+c-(2+x-3),x=3..x0)=50:
> sol :=solve(subs(ac,eq3),x0);
```

$$sol := 3 + 10\,\sqrt{3}, 3 - 10\,\sqrt{3}$$

```
> assign({x0=sol[1]}); assign(ac);
> plot({2+abs(x-3),a*x^2-6*a*x+c},x=sol[2]-2..sol[1]+2);
```

Grafic: vezi figura 4.44

70. (*ODE 1*)

```
> y:='y':
> diffeq:=diff(y(x),x)=exp(-x)-2*y(x): Gensol:=dsolve(diffeq,y(x));
```

$$Gensol := \mathrm{y}(x) = \mathrm{e}^{(-x)} + \mathrm{e}^{(-2\,x)}\_C1$$

Figura 4.45:

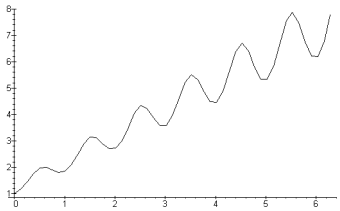


Figura 4.46:

```
> assign(Gensol): toplot:={seq(subs(_C1=i,y(x)),i=-3..3)}:
> plot(toplot,x=-1/2..2,-1..1);
```

Grafic: vezi figura 4.45

71. (*ODE 2*)

```
> x:='x': y:='y':
> dsolve({diff(y(x),x)-sin(2*Pi*x)*y(x)=1,y(0)=1},y(x));
```

$$y(x) = e^{\left(-\frac{\cos(\pi x)^2}{\pi}\right)} \int_0^x e^{\left(\frac{\cos(\pi u)^2}{\pi}\right)}\, du + \frac{e^{\left(-\frac{\cos(\pi x)^2}{\pi}\right)}}{e^{\left(-\frac{1}{\pi}\right)}}$$

```
> assign('');
> y:=unapply(y(x),x):
> lista:=[seq([2*Pi*i/50,evalf(y(2*Pi*i/50))],i=0..50)]:
> plot(lista);
```

Grafic: vezi figura 4.46

72. (*ODE 3*)

```
> restart;
> dsolve(diff(y(x),x)=(2*y(x)-4*x-5)/(2*y(x)-2*x),y(x));
```

$$-2\,x\,y(x) + 2\,x^2 + 5\,x + y(x)^2 = \_C1$$

```
> eq:=-2*x*y+2*x^2+5*x+y^2:
> plots[implicitplot]({eq=0,eq=-2,eq=2},x=-7..2,y=-7..2);
```

Grafic: vezi figura 4.47

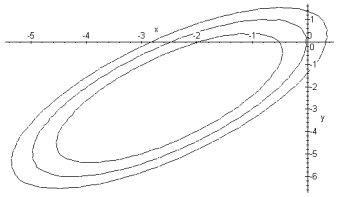


Figura 4.47:

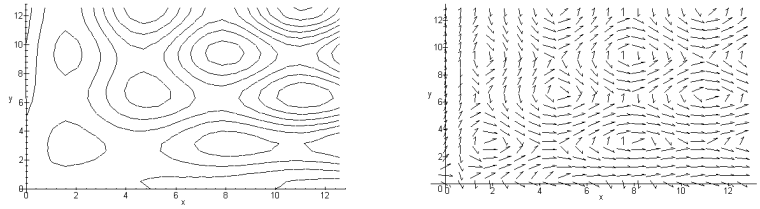


Figura 4.48:

73. (*ODE 4*)

```
> restart;
> Solgen:=dsolve(diff(y(x),x)=(cos(y(x))-y(x)*cos(x))/
> (x*sin(y(x))+sin(x)-1),y(x));
```

$$Solgen := -x\cos(\,\mathrm{y}(\,x\,)\,) + \mathrm{y}(\,x\,)\sin(\,x\,) - \mathrm{y}(\,x\,) = \_C1$$

```
> ptplot:=subs(y(x)=y,lhs(Solgen)):
> plots[contourplot](ptplot,x=0..4*Pi,y=0..4*Pi,axes=NORMAL);
```

Grafic: vezi figura 4.48.a

```
> DEtools[DEplot1](diff(y(x),x)=(cos(y(x))-y(x)*cos(x))/
>  (x*sin(y(x))+sin(x)-1),y(x),x=0..4*Pi,y=0..4*Pi);
```

Grafic: vezi figura 4.48.b

74. (*ODE 5*)

```
> restart;
> sol1:=dsolve(x*y(x)^3-(x^4+y(x)^4)*diff(y(x),x)=0,y(x));
```

$$sol1 := x = \frac{\_C1\ x\,\mathrm{e}^{\left(1/3\sqrt{3}\arctan\left(1/3\,\frac{\left(-2\,\mathrm{y}(\,x\,)^2+x^2\right)\sqrt{3}}{x^2}\right)\right)}}{\mathrm{y}(\,x\,)}$$

```
> pas1:=subs({x=1,y(x)=1},sol1);
```

$$pas1 := 1 = \_C1\ \mathrm{e}^{\left(1/3\sqrt{3}\arctan\left(-1/3\sqrt{3}\right)\right)}$$

```
> pas2:=eval(pas1);
```

$$pas2 := 1 = \_C1\ \mathrm{e}^{\left(-1/18\sqrt{3}\,\pi\right)}$$

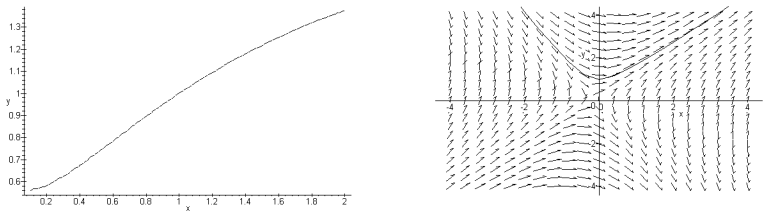


Figura 4.49:

```
> pas3:=simplify(solve(pas2,_C1));
```

$$pas3 := \mathrm{e}^{\left(1/18\,\sqrt{3}\,\pi\right)}$$

```
> grafic:=subs({_C1=pas3,y(x)=y},sol1);
```

$$grafic := x = \frac{\mathrm{e}^{\left(1/18\,\sqrt{3}\,\pi\right)}\ x\,\mathrm{e}^{\left(1/3\,\sqrt{3}\arctan\left(1/3\,\frac{\left(-2\,y^2+x^2\right)\sqrt{3}}{x^2}\right)\right)}}{y}$$

```
> plots[implicitplot](grafic,x=0.1..2,y=0.1..2,grid=[40,40]);
```

Grafic: vezi figura 4.49.a

75. (*ODE 6*)

```
> eq:=diff(y(x),x)=(1+5*x-y)/(x+2*y):
> DEtools[DEplot1](eq,[x,y],x=-4..4,{[0,1]},y=-4..4);
```

Grafic: vezi figura 4.49.b

76. (*ODE 7*)

```
> x:='x': y(x):='y(x)': eq:=diff(y(x),x)+(1+x^2)*y(x)=0:
> dsolve(eq,y(x));
```

$$\mathrm{y}(x) = \mathrm{e}^{\left(-1/3\,x\,\left(3+x^2\right)\right)}\_C1$$

```
> dsolve({eq,y(0)=1},y(x));
```

$$\mathrm{y}(x) = \mathrm{e}^{\left(-1/3\,x\,\left(3+x^2\right)\right)}$$

```
> assign(");
> plot(y(x),x=0..10);
```

Grafic: vezi figura 4.50

77. (*Curbe ortogonale 1*)

```
> y[1]:=x->x: y[2]:=x->sqrt(1-x^2):
> v1:=abs(D(y[1])(1/sqrt(2))):v2:=abs(D(y[2])(1/sqrt(2))):
> if v1=v2 and y[1](1/sqrt(2))=y[2](1/sqrt(2))
>   then true else false fi;
```

true

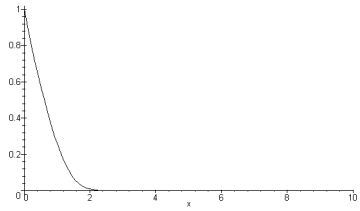


Figura 4.50:

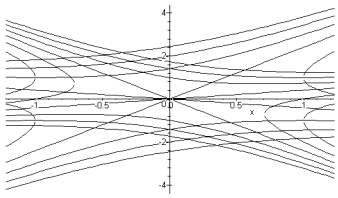


Figura 4.51:

78. (*Izoterme*)

```
> restart; eq1:=y^2+2*x*y-x^2=c: step1:=D(eq1):
> step2:=subs({D(x)=1,D(c)=0},step1):
> deriv:=solve(step2,D(y));
```

$$deriv := -\frac{2\,y - 2\,x}{2\,y + 2\,x}$$

```
> step3:=simplify(-1/deriv);
```

$$step3 := \frac{y + x}{y - x}$$

```
> diffeq:=diff(y(x),x)=subs(y=y(x),step3):
> Sol:=dsolve(diffeq,y(x));
```

$$Sol := -x\,\mathrm{y}(\,x\,) - \frac{1}{2}\,x^2 + \frac{1}{2}\,\mathrm{y}(\,x\,)^2 = \_C1$$

```
> eq2:=subs({y(x)=y,_C1=k}, Sol);
```

$$eq2 := -x\,y - \frac{1}{2}\,x^2 + \frac{1}{2}\,y^2 = k$$

```
> l1:=seq(solve(eq1,y),c=-3..3): l2:=seq(solve(eq2,y),k=-3..3):
> plot({l1,l2},x=-sqrt(3/2)..sqrt(3/2));
```

Grafic: vezi figura 4.51

79. (*Curbe ortogonale 2*)

```
> restart;
> dsolve(diff(y(x),x)=-x/(2*y(x)),y(x));
```

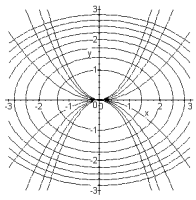


Figura 4.52:

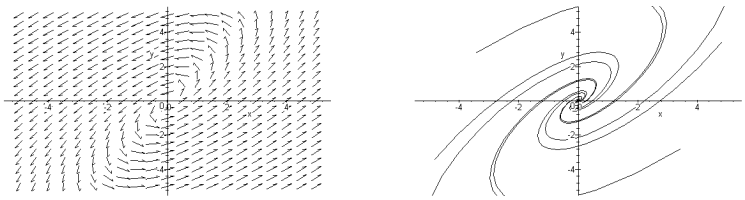


Figura 4.53:

$$\mathrm{y}(x)^2 = -\frac{1}{2}x^2 + \_C1$$

```
> parab:=seq(y=(-1+i/4)*x^2,i=0..8):
> elipse:=seq(y^2+x^2/2=k,k=1..8):
> plots[implicitplot]({parab,elipse},x=-3..3,y=-3..3);
```

Grafic: vezi figura 4.52

80. (*Sistem ODE 1*)

```
> A:=array([[3,-2],[4,-1]]):
> eqs:=student[equate]([[diff(x(t),t)],[diff(y(t),t)]],
>  A &* [[x(t)],[y(t)]]);
```

$$eqs := \left\{ \frac{\partial}{\partial t}\mathrm{y}(t) = 4\,\mathrm{x}(t) - \mathrm{y}(t),\ \frac{\partial}{\partial t}\mathrm{x}(t) = 3\,\mathrm{x}(t) - 2\,\mathrm{y}(t) \right\}$$

```
> dsolve(eqs,{x(t),y(t)});
```

$$\{\mathrm{y}(t) = \_C1\,\mathrm{e}^t \sin(2t) + (-2\,\_C2 + \_C1)\,\mathrm{e}^t \cos(2t),$$
$$\mathrm{x}(t) = \_C2\,\mathrm{e}^t \sin(2t) + (-\_C2 + \_C1)\,\mathrm{e}^t \cos(2t)\}$$

```
> coninit:={seq(seq([0,.25*i,.25*j],i=-1..1),j=-1..1)}:
> DEtools[DEplot2](A,[x,y],t=-3..3,x=-5..5,y=-5..5);
```

Grafic: vezi figura 4.53.a

```
> DEtools[DEplot2](A,[x,y],t=-3..3,coninit,x=-5..5,y=-5..5,
>  stepsize=0.1,arrows=NONE);
```

Grafic: vezi figura 4.53.b

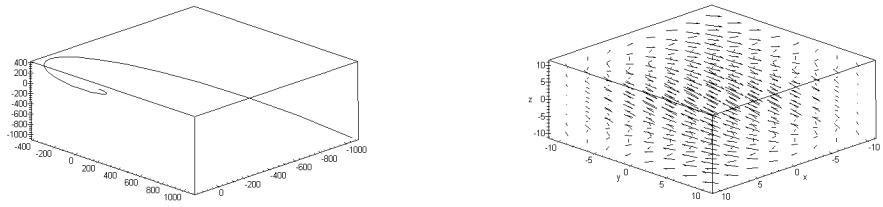


Figura 4.54:

81. (*Sistem ODE 2*)

```
> A:=array([[5,5,2],[-6,-6,-5],[6,6,5]]):
> eqs:=student[equate]([[diff(x(t),t)],[diff(y(t),t)],[diff(z(t),t)]],
>  A &* [[x(t)],[y(t)],[z(t)]]):
> problema:=eqs union {x(0)=0, y(0)=0, z(0)=2}:
> sol:=dsolve(problema,{x(t),y(t),z(t)});
```

$$sol := \{y(t) = 2 - 2e^{(2t)}\sin(3t) - 2e^{(2t)}\cos(3t),$$
$$z(t) = 2e^{(2t)}\sin(3t) + 2e^{(2t)}\cos(3t), x(t) = -2 + 2e^{(2t)}\cos(3t)\}$$

```
> assign(sol): plots[spacecurve]([x(t),y(t),z(t)],t=0..Pi,axes=BOXED);
```

Grafic: vezi figura 4.54.a

```
> plots[fieldplot3d]([5*x+5*y-2*z,-6*x-6*y-5*z,6*x+6*y+5*z],
>  x=-10..10,y=-10..10,z=-10..10,axes=BOXED);
```

Grafic: vezi figura 4.54.b

82. (*Fibra nervoasă*)

```
> eqs:=[diff(v(xi),xi)=w, diff(w(xi),xi)=1/3*v^3-v+r-u*w,
>   diff(r(xi),xi)=epsilon/u*(b*r-v-a)]:
> epsilon:=0.08: b:=0.8: a:=0.7: u:=0.6:
> DEtools[DEplot](eqs,[v,w,r],0..15,{[0,1,.5,.5]},stepsize=0.1);
> DEtools[DEplot](eqs,[v,w,r],0..15,{[0,1,.5,.5]},stepsize=0.1,
>   scene=[xi,v]);
> DEtools[DEplot](eqs,[v,w,r],0..15,{[0,1,.5,.5]},stepsize=0.1,
>   scene=[xi,w]);
> DEtools[DEplot](eqs,[v,w,r],0..15,{[0,1,.5,.5]},stepsize=0.1,
>   scene=[xi,r]);
> DEtools[DEplot](eqs,[v,w,r],0..15,{[0,1,.5,.5]},stepsize=0.1,
>   scene=[v,w]);
> DEtools[DEplot](eqs,[v,w,r],0..15,{[0,1,.5,.5]},stepsize=0.1,
>   scene=[v,r]);
> DEtools[DEplot](eqs,[v,w,r],0..15,{[0,1,.5,.5]},stepsize=0.1,
>   scene=[w,r]);
```

Grafic: vezi figura 4.55

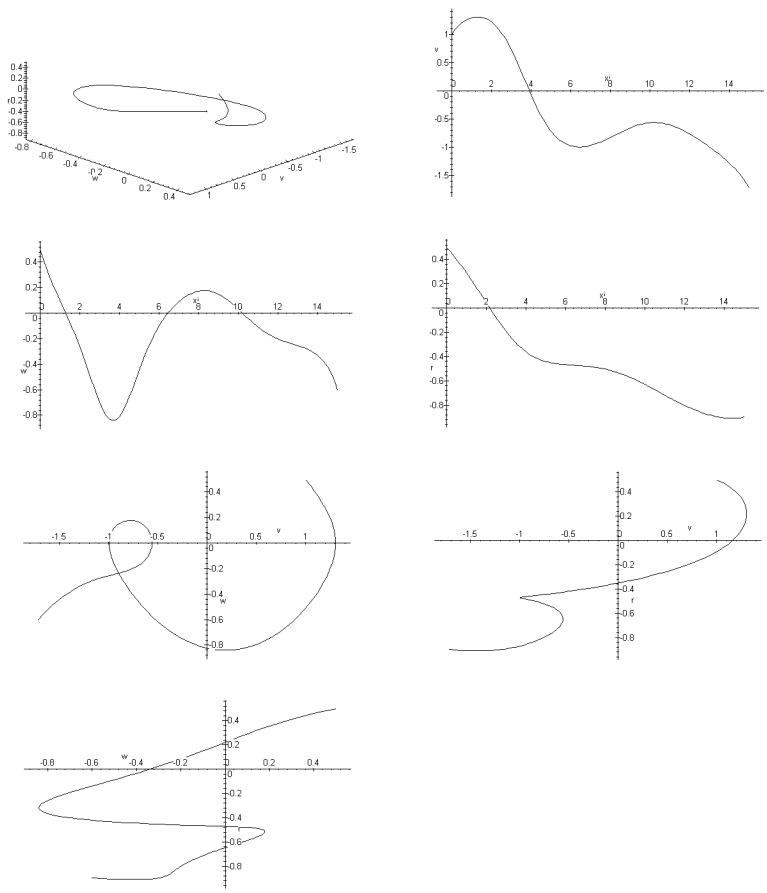


Figura 4.55:

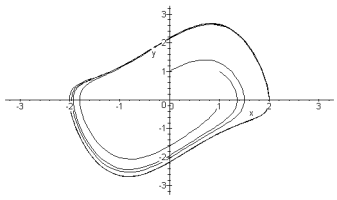


Figura 4.56:

83. (*Ecuaţia Van der Pol*)

```
> DEtools[DEplot2]([y,(1-x^2)*y-x],[x,y],t=0..10,{[0,1,1],[0,1,0],
>   [0,0,1]},x=-3..3,y=-3..3,stepsize=0.05,
> arrows=NONE);
```

Grafic: vezi figura 4.56

84. (*Metode cu diferenţe pentru ODE*)

```
> h:='h': f:=(x,y)->x*y:
> x:=(n,h)->n*h:
> y:=proc(n,h) option remember;
>   if n=0 then y(0) else y(n-1,h)+h*f(x(n-1,h),y(n-1,h)) fi
> end:
> z:=proc(n,h) option remember;
> if n=0 then y(0) else z(n-1,h)+h/2*(f(x(n-1,h),z(n-1,h))+
> f(x(n-1,h),z(n-1,h)+h*f(x(n-1,h),z(n-1,h)))) fi
> end:
> yrk:=proc(n,h) local k1,k2; option remember;
>   if n=0 then y(0) else
>    k1:=h*f(x(n-1,h),yrk(n-1,h));
>    k2:=h*f(x(n-1,h)+h,yrk(n-1,h)+k1);
>    yrk(n-1,h)+1/2*(k1+k2)
>   fi
> end:
> yrk4:=proc(n,h) local k1,k2,k3,k4; option remember;
>   if n=0 then y(0) else
>    k1:=f(x(n-1,h),yrk4(n-1,h));
>    k2:=f(x(n-1,h)+h/2,yrk4(n-1,h)+h*k1/2);
>    k3:=f(x(n-1,h)+h/2,yrk4(n-1,h)+h*k2/2);
>    k4:=f(x(n,h),yrk4(n-1,h)+h*k3);
>    yrk4(n-1,h)+h/6*(k1+2*k2+2*k3+k4)
>   fi
> end:
> y(0):=1:
> l1:=[seq(y(i,0.1),i=0..10)];
```

$l1 := [1, 1, 1.01, 1.0302, 1.061106, 1.10355024, 1.158727752, 1.228251417,$

$1.314229016, 1.419367337, 1.547110397]$

```
> l2:=[seq(y(i,0.05),i=0..20)];
```

$l2 := [1, 1, 1.0025, 1.00751250, 1.015068844, 1.025219532, 1.038034776,$
$1.053605298, 1.072043391, 1.093484259, 1.118087655, 1.146039846,$
$1.177555942, 1.212882620, 1.252301305, 1.296131851, 1.344736795,$
$1.398526267, 1.457963633, 1.523571997, 1.595941667]$

```
> l3:=[seq(z(i,0.1),i=0..10)];
```

$l3 := [1, 1, 1.010050000, 1.030453010, 1.061830304, 1.105152980,$
$1.161792070, 1.233590820, 1.322964475, 1.433035119, 1.567812072]$

```
> l4:=[seq(yrk(i,0.1),i=0..10)];
```

$l4 := [1, 1.005000000, 1.020175500, 1.045985940, 1.083223039, 1.133051299,$
$1.197068697, 1.277392007, 1.376773105, 1.498755202, 1.647881345]$

```
> l5:=[seq(yrk4(i,0.1),i=0..10)];
```

$l5 := [1, 1.005012521, 1.020201340, 1.046027859, 1.083287065, 1.133148446,$
$1.197217347, 1.277621279, 1.377127694, 1.499302362, 1.648721007]$

```
> exact:=dsolve({diff(Y(X),X)=X*Y(X),Y(0)=1},Y(X)): assign(exact):
> A1:=array([seq([evalf(subs(X=n/10,Y(X))),l1[n+1]],n=0..10)]):
> A2:=array([seq([evalf(subs(X=n/20,Y(X))),l2[n+1]],n=0..20)]):
> A2p:=array([seq([A2[2*n-1,1],A2[2*n-1,2]],n=1..11)]):
> A3:=array([seq([evalf(subs(X=n/10,Y(X))),l3[n+1]],n=0..10)]):
> A4:=array([seq([evalf(subs(X=n/10,Y(X))),l4[n+1]],n=0..10)]):
> A5:=array([seq([evalf(subs(X=n/10,Y(X))),l5[n+1]],n=0..10)]):
> # pentru compararea rezultatelor se omite semnul de comentariu
> #op(A1), op(A2p), op(A3), op(A4), op(A5);
> cine:=abs(A1[11,1]-A1[11,2]),abs(A2p[11,1]-A2p[11,2]),abs(A3[11
> ,1]-A3[11,2]), abs(A4[11,1]-A4[11,2]), abs(A5[11,1]-A5[11,2]);
```

$cine := .101610874, .052779604, .080909199, .000839926, .264\,10^{-6}$

```
> min(cine);
```

$$.264\,10^{-6}$$

85. (*Coeficienţii metodei Runge-Kutta standard*)

```
> restart;
> `diff/y`:=(a,x)->f(a,y(a))*diff(a,x): D(y):=x->f(x,y(x)):
> alias(F=f(x,y(x)), Fx=D[1](f)(x,y(x)), Fy=D[2](f)(x,y(x)), Fxx=
> D[1,1](f)(x,y(x)), Fxy=D[1,2](f)(x,y(x)), Fyy=D[2,2](f)(x,y(x))):
> diff(F,x);
```

$$Fx + Fy\,F$$

```
> m:=3:
> taylor(y(x+h),h=0,m+1);
```

$$y(x) + F\,h + \left(\frac{1}{2}\,Fx + \frac{1}{2}\,Fy\,F\right)\,h^2 + \left(\frac{1}{6}\,Fxx + \frac{1}{3}\,Fxy\,F + \frac{1}{6}\,Fyy\,F^2 + \frac{1}{6}\,Fy\,Fx + \frac{1}{6}\,Fy^2\,F\right)\,h^3 + O(h^4)$$

```
> TaylorP:=normal((convert('',polynom)-y(x))/h);
```

$$TaylorP := F + \frac{1}{2}\,h\,Fx + \frac{1}{2}\,h\,Fy\,F + \frac{1}{6}\,h^2\,Fxx + \frac{1}{3}\,h^2\,Fxy\,F + \frac{1}{6}\,h^2\,Fyy\,F^2 + \frac{1}{6}\,h^2\,Fy\,Fx + \frac{1}{6}\,h^2\,Fy^2\,F$$

```
> k1:=taylor(f(x,y(x)),h=0,m):
> k2:=taylor(f(x+c2*h,y(x)+h*(a21*k1)),h=0,m):
> k3:=taylor(f(x+c3*h,y(x)+h*(a31*k1+a32*k2)),h=0,m):
> RungeKuttaP:=convert(series(b1*k1+b2*k2+b3*k3,h,m),polynom);
```

$$RungeKuttaP := b1\,F + b2\,F + b3\,F + (\,b2\,(\,Fx\,c2 + Fy\,a21\,F\,) + b3\,(\,Fx\,c3 + Fy\,a31\,F + Fy\,a32\,F\,)\,)\,h + \left(b2\,\left(\frac{1}{2}\,Fxx\,c2^2 + c2\,Fxy\,a21\,F + \frac{1}{2}\,a21^2\,F^2\,Fyy\right) + b3\,\left(\frac{1}{2}\,Fxx\,c3^2 + c3\,Fxy\,a31\,F + c3\,Fxy\,a32\,F + \frac{1}{2}\,a31^2\,F^2\,Fyy + a31\,F^2\,Fyy\,a32 + \frac{1}{2}\,a32^2\,F^2\,Fyy + Fy\,a32\,Fx\,c2 + Fy^2\,a32\,a21\,F\right)\right)\,h^2$$

```
> d:=expand(TaylorP-RungeKuttaP):
> eqns:={coeffs(d,[h,F,Fx,Fy,Fxx,Fxy,Fyy])}:
> vars:=indets(eqns):
> sols:=solve(eqns,vars);
```

$$sols := \left\{a21 = \frac{2}{3}\,\frac{3\,b3\,c3^2 - 1}{-1 + 2\,b3\,c3}, b2 = -\frac{3}{4}\,\frac{1 - 4\,b3\,c3 + 4\,b3^2\,c3^2}{3\,b3\,c3^2 - 1}, c3 = c3, b3 = b3, b1 = \frac{1}{4}\,\frac{4\,b3 - 1 - 12\,b3\,c3 + 12\,b3\,c3^2}{3\,b3\,c3^2 - 1}, c2 = \frac{2}{3}\,\frac{3\,b3\,c3^2 - 1}{-1 + 2\,b3\,c3}, a32 = \frac{1}{4}\,\frac{-1 + 2\,b3\,c3}{b3\,(\,3\,b3\,c3^2 - 1\,)}, a31 = \frac{1}{4}\,\frac{1 - 6\,b3\,c3 + 12\,b3^2\,c3^3}{b3\,(\,3\,b3\,c3^2 - 1\,)}\right\}, \left\{b1 = \frac{1}{4}, c3 = \frac{2}{3}, a32 = a32, a21 = \frac{2}{9}\,\frac{1}{a32}, a31 = -a32 + \frac{2}{3}, b2 = 0, b3 = \frac{3}{4}, c2 = \frac{2}{9}\,\frac{1}{a32}\right\}$$

```
> RungeKutta:= proc(s,m)
>   local TaylorPhi, RungeKuttaPhi, d, vars, eqns, k,i,j,c,val;
>   D(y):=x->f(x,y(x));
```

```
>   TaylorPhi:=convert(taylor(y(x+h),h=0,m+1),polynom);
>   TaylorPhi:=normal((TaylorPhi-y(x))/h);
>   c[1]:=0;
>   for i from 1 to s do
>   k[i]:=taylor(f(x+c[i]*h,y(x)+sum(a[i,j]*k[j],j=1..i-1)*h),h=0,m);
>   od;
>   RungeKuttaPhi:=0;
>   for i from 1 to s do
>      RungeKuttaPhi:=RungeKuttaPhi+b[i]*k[i];
>   od;
>   RungeKuttaPhi:=series(RungeKuttaPhi,h,m);
>   RungeKuttaPhi:=convert(RungeKuttaPhi,polynom);
>   d:=expand(TaylorPhi-RungeKuttaPhi);
>   vars:={seq(c[i],i=2..s),seq(b[i],i=1..s),seq((seq(a[i,j],
>      j=1..i-1)),i=2..s)};
>   eqns:={coeffs(d, indets(d) minus vars)};
>   eqns:=eqns union {seq(sum(a[i,'j'],'j'=1..i-1)-c[i],i=2..s)};
>   val:=solve(eqns,vars);
> end:
> RungeKutta(2,2);
```

$$\left\{a_{2,1}=\frac{1}{2}\,\frac{1}{b_2},b_2=b_2,c_2=\frac{1}{2}\,\frac{1}{b_2},b_1=1-b_2\right\}$$

```
> subs(b[2]=1,"); #metoda Euler imbunatatita
```

$$\left\{c_2=\frac{1}{2},b_1=0,a_{2,1}=\frac{1}{2},1=1\right\}$$

```
> subs(b[2]=1/2,""); #metoda Heun
```

$$\left\{a_{2,1}=1,\frac{1}{2}=\frac{1}{2},c_2=1,b_1=\frac{1}{2}\right\}$$

86. (*Metoda Picard*)

```
> f := (x,y) -> y + 1:
> diffeq := diff(y(x),x)=f(x,y(x)):
> sol := dsolve({diffeq,y(0)=1},y(x)):
> p := unapply(rhs(sol),x);   # the true solution
```

$$p := x \to -1 + 2\,\mathrm{e}^{x}$$

```
> pl:=plot(p,0..1):
> p.0 := x -> 1; pl0:=plot(p0,0..1,linestyle=2):
> for i from 1 to 5 do
> p.i := unapply(p.(i-1)(0) + int(f(t,p.(i-1)(t)),t=0..x),x):
>  print(p.i); pl.i:=plot(p.i,0..1,linestyle=i+2): od:
```

$$p0 := 1$$

$$x \to 1 + 2\,x$$

$$x \to 1 + 2\,x + x^2$$

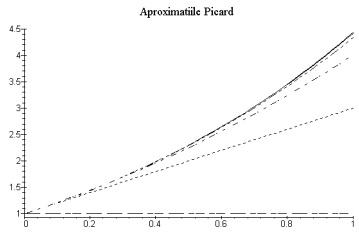


Figura 4.57:

$$x \to 1 + 2\,x + \frac{1}{3}\,x^3 + x^2$$

$$x \to 1 + 2\,x + \frac{1}{12}\,x^4 + x^2 + \frac{1}{3}\,x^3$$

$$x \to 1 + 2\,x + \frac{1}{60}\,x^5 + x^2 + \frac{1}{3}\,x^3 + \frac{1}{12}\,x^4$$

```
> plots[display]({pl,seq(pl.i,i=0..5)},title=`Aproximatiile Picard`);
```

Grafic: vezi figura 4.57

87. (*Metoda Euler explicită*)

```
> g := (x,y) -> y: x.0 := 0: y.0 := 1: h := .1:
> tab := x.0,y.0:
> for i from 1 to 10 do
>  x.i := x.(i-1) + h:
>  y.i := y.(i-1)+h*g(x.(i-1),y.(i-1)):
>  tab := tab, x.i,y.i:
> od:
> tab;
```

$0, 1, .1, 1.1, .2, 1.21, .3, 1.331, .4, 1.4641, .5, 1.61051, .6, 1.771561, .7,$
$1.9487171, .8, 2.14358881, .9, 2.357947691, 1.0, 2.593742460$

88. (*Ecuaţia Euler*)

```
> diffeq :=  x^2 *diff(y(x),x,x) + a*x* diff(y(x),x) + b* y(x) = 0:
> ypp := diff(y(x),x,x)=diff(y(z),z,z)*1/x^2-diff(y(z),z)*1/x^2:
> yp := diff(y(x),x) = diff(y(z),z)*1/x:
> diffeq1 := subs({ypp, yp,y(x)=y(z)},diffeq):
> diffeq1 :=simplify(diffeq1):
> diffeq1 := collect(diffeq1,diff(y(z),z)):
> sol := dsolve({diffeq1},y(z));
```

$$sol := \mathrm{y}(z) = \_C1\, \mathrm{e}^{\left(1/2\left(1-a+\sqrt{1-2\,a+a^2-4\,b}\right)z\right)} + \_C2\, \mathrm{e}^{\left(-1/2\left(-1+a+\sqrt{1-2\,a+a^2-4\,b}\right)z\right)}$$

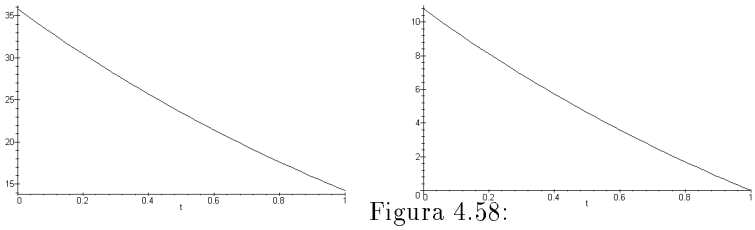


Figura 4.58:

```
> y := unapply(subs(z=ln(x),rhs(sol)),x):
> yg := unapply(simplify(y(x)),(a,b,_C1,_C2,x));
```

$$yg := (a, b, \_C1, \_C2, x) \rightarrow \_C1\, x^{\left(1/2-1/2\,a+1/2\,\sqrt{1-2\,a+a^2-4\,b}\right)} + \_C2\, x^{\left(1/2-1/2\,a-1/2\,\sqrt{1-2\,a+a^2-4\,b}\right)}$$

89. (*Hemodializă*)

```
> restart;
> dsolve({Qb*diff(u(t),t)=-k*(u(t)-v(t)),-Qd*diff(v(t),t)=
> k*(u(t)-v(t)), u(0)=u0, v(L)=0},{u(t),v(t)});
```

$$\left\{ \mathrm{v}(t) = \frac{u0\ Qb\, \mathrm{e}^{\left(\frac{k\,(-Qd+Qb)\,L}{Qd\ Qb}\right)}}{\%1} - \frac{Qb\ u0\, \mathrm{e}^{\left(\frac{k\,(-Qd+Qb)\,t}{Qd\ Qb}\right)}}{\%1}, \right.$$

$$\left. \mathrm{u}(t) = \frac{u0\ Qb\, \mathrm{e}^{\left(\frac{k\,(-Qd+Qb)\,L}{Qd\ Qb}\right)}}{\%1} - \frac{u0\ Qd\, \mathrm{e}^{\left(\frac{k\,(-Qd+Qb)\,t}{Qd\ Qb}\right)}}{\%1} \right\}$$

$$\%1 := -Qd + Qb\, \mathrm{e}^{\left(\frac{k\,(-Qd+Qb)\,L}{Qd\ Qb}\right)}$$

```
> assign('');sol:=subs({k=2.25,L=1,Qb=2,Qd=4,u0=35.8},{u(t),v(t)});
```

$$sol := \left\{ 71.6\, \frac{\mathrm{e}^{(-.5625000000)}}{\%1} - 143.2\, \frac{\mathrm{e}^{(-.5625000000\,t)}}{\%1}, \right.$$

$$\left. 71.6\, \frac{\mathrm{e}^{(-.5625000000)}}{\%1} - 71.6\, \frac{\mathrm{e}^{(-.5625000000\,t)}}{\%1} \right\}$$

$$\%1 := -4 + 2\, \mathrm{e}^{(-.5625000000)}$$

```
> u:=unapply(sol[1],t): v:=unapply(sol[2],t);
> plot(u(t),t=0..1); plot(v(t),t=0..1);
```

Grafic: vezi figura 4.58

90. (*Infecţie*)

```
> restart;
> eq:=diff(i(t),t)+(Gamma+Mu-Lambda)*i(t)=-Lambda*i(t)^2:
> sol:=dsolve({eq,i(0)=i0},i(t)):
> assign(sol);
> val1:=subs({Lambda=0.5,Gamma=0.75,Mu=0.65},i(t)):
```

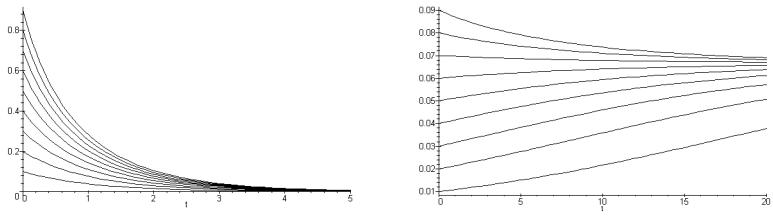


Figura 4.59:

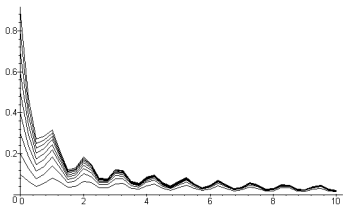


Figura 4.60:

```
> toplot1:={seq(subs(i0=.1*j,eval(val1)),j=1..9)}:
> plot(toplot1,t=0..5);
```

Grafic: vezi figura 4.59.a

```
> val2:=subs({Lambda=1.5,Gamma=0.75,Mu=0.65},i(t)):
> toplot2:={seq(subs(i0=.01*j,eval(val2)),j=1..9)}:
> plot(toplot2,t=0..20);
```

Grafic: vezi figura 4.59.b

```
> Lambda:='Lambda': Gamma:='Gamma': Mu:='Mu': i:='i':
> eq:=diff(i(t),t)=-(Gamma+Mu-Lambda(t))*i(t)-Lambda(t)*i(t)^2:
> assign(dsolve({eq,i(0)=i0},i(t)));
> Lambda:=t->3-2.5*sin(6*t):
> val3:=subs({Gamma=2,Mu=1},i(t)):
> i:='i': eq:=diff(i(t),t)=(Lambda(t)-3)*i(t)-Lambda(t)*i(t)^2:
> i0:=[seq(.1*j,j=1..9)]:
> sol:=[seq(dsolve({eq,i(0)=i0[j]},i(t),numeric),j=1..9)]:
> lista:={seq([seq([k/4,rhs(sol[j](k/4)[2])],k=0..40)],j=1..9)}:
> plot(lista);
```

Grafic: vezi figura 4.60

91. (*Evoluţia populaţiei*)

```
> k:='k': y:='y': peq:=dsolve({diff(y(t),t)=k*y(t),y(0)=y0},y(t));
```

$$peq := \mathrm{y}(t) = \mathrm{e}^{(k\,t)}\ y0$$

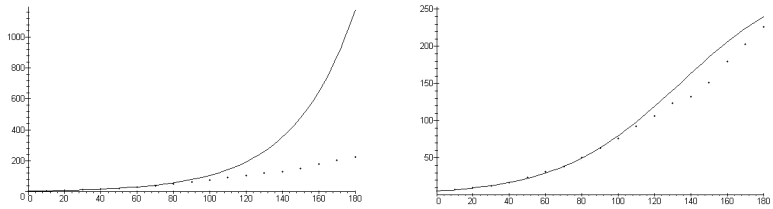


Figura 4.61:

```
> pop:=proc(t0,k0,y00) subs({t=t0,k=k0,y0=y00},op(2,peq)) end:
> popplot:=plot(pop(t,0.03,5.3),t=0..180):
> pdata:=[0,5.3,10,7.24,20,9.64,30,12.68,40,17.06,50,23.19,60,31.44,70
> 70,38.56,80,50.19,90,62.98,100,76.21,110,92.23,120,106.02,130,123.20
> ,140,132.16,150,151.33,160,179.32,170,203.30,180,226.54,190,248.71]:
> dataplot:=plot(pdata,style=POINT):
> plots[display]({popplot,dataplot});
```

Grafic: vezi figura 4.61.a

```
> y:='y': Sol:=dsolve({diff(y(t),t)=(r-a*y(t))*y(t),y(0)=y0},y(t));
```

$$Sol := \mathrm{y}(t) = -\frac{r}{-a - \frac{\mathrm{e}^{(-r\,t)}\,(r - y0\,a)}{y0}}$$

```
> pop2:=proc(t0) subs({t=t0,r=0.03,a=0.0001,y0=5.3},op(2,Sol)) end:
> pp:=plot(pop2(t),t=0..180):
> plots[display]({pp,dataplot});
```

Grafic: vezi figura 4.61.b

```
> evalf(pop2(200));
```

$$263.6602427$$

92. (*Timp de înjumătăţire*)

```
> k:=solve(y0*exp(100*k)=1/2*y0,k);
```

$$k := -\frac{1}{100}\ln(2)$$

```
> y:=t->y0*exp(k*t): simplify(y(t));
```

$$y0\,2^{(-1/100\,t)}$$

```
> evalf(y(50));
```

$$.7071067810\,y0$$

```
> k:='k':
> k:=solve(exp(5730*k)=1/2,k);
```

$$k := -\frac{1}{5730}\ln(2)$$

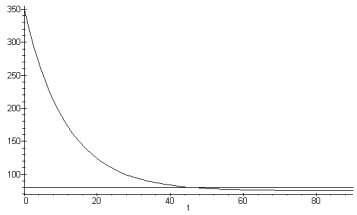


Figura 4.62:

```
> y:=t->exp(k*t): simplify(y(t));
```

$$2^{(-1/5730\,t)}$$

```
> solve(y(t)=.63);
```

$$3819.482007$$

93. (*Prăjitura*)

```
> DE1:=dsolve({diff(tp(t),t)=k*(tp(t)-temps),tp(0)=temp0},tp(t)):
> step1:=subs({temp0=350,temps=75},rhs(DE1));
```

$$step1 := 75 + 275\,\mathrm{e}^{(k\,t)}$$

```
> k:=solve(subs(t=15,step1)=150);
```

$$k := \frac{1}{15}\ln\left(\frac{3}{11}\right)$$

```
> simplify(step1,exp);
```

$$75 + 275\left(\frac{3}{11}\right)^{(1/15\,t)}$$

```
> t00:=solve(step1=80): evalf(t00);
```

$$46.26397676$$

```
> plot({80,step1},t=0..90);
```

Grafic: vezi figura 4.62

```
> fsolve(step1=80,t,40..50);
```

$$46.26397676$$

94. (*Temperatura în clădire*)

```
> u:='u': sol:=dsolve({diff(u(t),t)=1/4*(70-10*cos(Pi*t/12)-u(t)),
> u(0)=60},u(t)):
> assign(sol): simplify(u(t)):
> plot(u(t),t=0..24);
```

Grafic: vezi figura 4.63.a

```
> ora1:=fsolve(diff(u(t),t)=0,t,14..16); evalf(subs(t=ora1,u(t)));
```

$$ora1 := 15.15061632$$

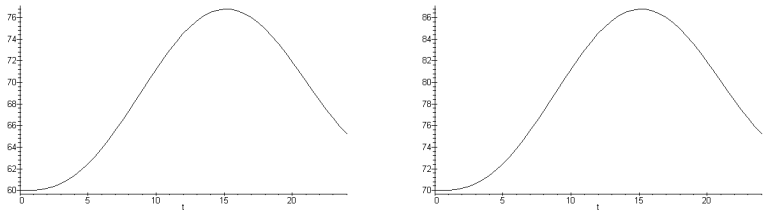


Figura 4.63:

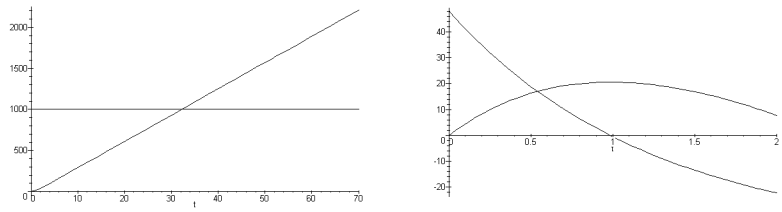


Figura 4.64:

76.78682255

```
> u:='u': Sol:=dsolve({diff(u(t),t)=1/4*(80-10*cos(Pi*t/12)-u(t)),
> u(0)=70},u(t)):
> assign(Sol): simplify(u(t)):
> plot(u(t),t=0..24);
```

Grafic: vezi figura 4.63.b

```
> ora2:=fsolve(diff(u(t),t)=0,t,14..16); evalf(subs(t=ora2,u(t)));
```

$ora2 := 15.15061632$

86.78682255

95. (*Cădere liberă sau forţată*)

```
> v:='v': s:='s':
> step1:=dsolve({diff(v(t),t)=32-v(t),v(0)=2},v(t)):
> assign(step1):
> step2:=dsolve({diff(s(t),t)=v(t),s(0)=0},s(t)):
> assign(step2):
> plot({1000,s(t)},t=0..70);
```

Grafic: vezi figura 4.64.a

```
> t00:=fsolve(s(t)=1000,t,30..40);
```

$t00 := 32.18750000$

```
> evalf(subs(t=t00,v(t)));
```

32.00000000

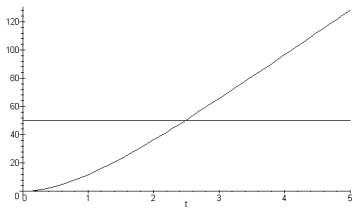
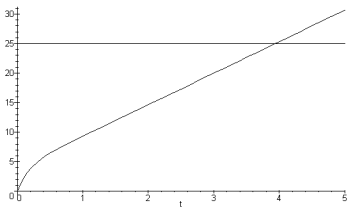



Figura 4.65:

```
> s:='s': v:='v':
> step1:=dsolve({diff(v(t),t)=32-v(t),v(0)=0},v(t)):
> assign(step1):
> step2:=dsolve({diff(s(t),t)=v(t),s(0)=0},s(t)):
> assign(step2):
> plot({50,s(t)},t=0..5);
```

Grafic: vezi figura 4.65.a

```
> t1:=fsolve(s(t)=50,t,2..3);
```

$$t1 := 2.478643063$$

```
> v1:=evalf(subs(t=t1,v(t)));
```

$$v1 := 29.31657802$$

```
> s:='s': v:='v':
> step3:=dsolve({diff(v(t),t)=32-6*v(t),v(0)=v1},v(t)):
> assign(step3):
> step4:=dsolve({diff(s(t),t)=v(t),s(0)=0},s(t)):
> assign(step4):
> plot({25,s(t)},t=0..5);
```

Grafic: vezi figura 4.65.b

```
> t2:=fsolve(s(t)=25,t,3..5);
```

$$t2 := 3.938023603$$

```
> t1+t2;
```

$$6.416666666$$

96. (*Aruncare pe verticală*)

```
> v:='v':
> Sol:=dsolve({diff(v(t),t)=-g-c/m*v(t),v(0)=v0},v(t)):
> viteza:=proc(m0,c0,g0,v00,t0)
>    subs({m=m0, c=c0, g=g0, v0=v00, t=t0}, rhs(Sol))
> end:
> y:='y':
> Pos:=dsolve({diff(y(t),t)=viteza(m,c,g,v0,t),y(0)=y0},y(t));
```

$$Pos := \mathrm{y}(t) = -\frac{g\,m\,t}{c} - \frac{m^2\,\mathrm{e}^{\left(-\frac{c\,t}{m}\right)}\,g}{c^2} - \frac{m\,\mathrm{e}^{\left(-\frac{c\,t}{m}\right)}\,v0}{c} + \frac{m^2\,g}{c^2} + \frac{m\,v0}{c} + y0$$

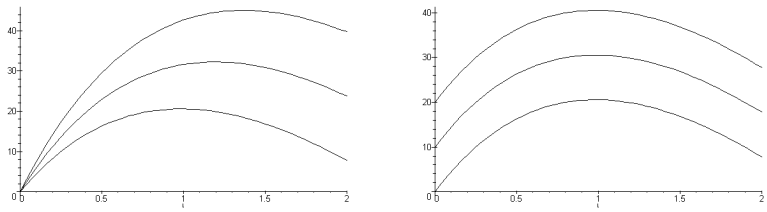


Figura 4.66:

```
> pozitia:=proc(m0,c0,g0,v00,y00,t0)
>    subs({m=m0,c=c0,g=g0,v0=v00,y0=y00,t=t0},rhs(Pos))
> end:
> plot({viteza(1/128,1/160,32,48,t),pozitia(1/128,1/160,32,48,0,t)}
>  ,t=0..2);
```

Grafic: vezi figura 4.64.b

```
> root:=solve(diff(pozitia(1/128,1/160,32,48,0,t),t)=0);
```

$$root := -\frac{5}{4}\ln\left(\frac{5}{11}\right)$$

```
> evalf(root);
```

.9855717006

```
> plot({pozitia(1/128,1/160,32,48,0,t), pozitia(1/128,1/160,
>  32,64,0,t), pozitia(1/128,1/160,32,80,0,t)},t=0..2);
```

Grafic: vezi figura 4.66.a

```
> plot({pozitia(1/128,1/160,32,48,0,t), pozitia(1/128,1/160,
>  32,48,10,t), pozitia(1/128,1/160,32,48,20,t)},t=0..2);
```

Grafic: vezi figura 4.66.b

97. (*Ecuaţia Lotka-Volterra*)

```
> restart;
> sol:=dsolve({diff(x(t),t)=2*x(t)-x(t)*y(t), diff(y(t),t)=-3*y(t)
>         +x(t)*y(t),x(0)=1,y(0)=1},{x(t),y(t)},numeric):
> plots[odeplot](sol,[x(t),y(t)],0..8,view=[0..8,0..6]);
> plots[odeplot](sol,[t,x(t),y(t)],0..6,axes=BOXED);
> plots[odeplot](sol,[t,x(t)],0..6);
> plots[odeplot](sol,[t,y(t)],0..6);
```

Grafic: vezi figura 4.67

98. (*Mişcare armonică simplă*)

```
> x:='x':
> DE1:=dsolve({diff(x(t),t$2)+64*x(t)=0,x(0)=1,D(x)(0)=0},x(t));
```

$$DE1 := \mathrm{x}(t) = \cos(8\,t)$$

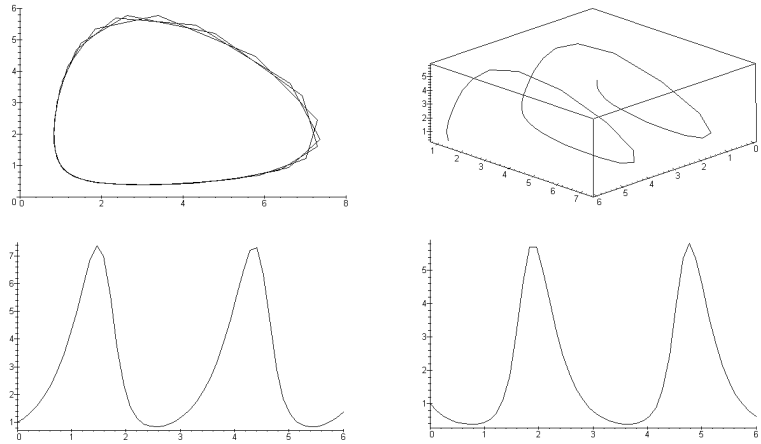


Figura 4.67:

```
> assign(DE1):
> n:=15: eps:=0.1:
> arcul:=proc(t0) local xt0, pts, m;
>   xt0:=evalf(subs(t=t0,x(t))):
>   pts:=[[0,xt0],seq([eps*(-1)^m,xt0+m*(1-xt0)/n],m=1..n-1),[0,1]]:
>   plot(pts,xtickmarks=2,ytickmarks=2);
> end:
> k_vals:=seq(k*2/9,k=0..9): to_animate:=[seq(arcul(k),k=k_vals)]:
> for i from 1 to 10 do plots[display]({to_animate[i]},
> view=[-1..1,-1.2..1.2]) od;
> #plots[display]({seq(to_animate[i],i=0..10)},insequence=true);
```

Grafic: vezi figura 4.68

```
> x:='x':
> d1:=proc(alpha,beta)
>   dsolve({diff(x(t),t$2)+4*x(t)=0,x(0)=alpha,D(x)(0)=beta},x(t)):
> end:
> plot(map(rhs,{d1(1,0),d1(4,0),d1(-2,0)}),t=0..Pi);
```

Grafic: vezi figura 4.69.a

```
> plot(map(rhs,{d1(0,1),d1(0,4),d1(0,-2)}),t=0..Pi);
```

Grafic: vezi figura 4.69.b

99. (*Mişcare încetinită*) Pentru a determina constanta resortului, aflăm $s = 6-4$ (vezi *Mişcare armonică*) şi în poziţia de echilibru $mg = ks$, deci $k = 4$.

```
> x:='x':
> Eq:=diff(x(t),t$2)+8*diff(x(t),t)+16*x(t)=0:
> DE:=dsolve({Eq,x(0)=0,D(x)(0)=1},x(t));
```

$$DE := \mathrm{x}(t) = \mathrm{e}^{(-4\,t)}\,t$$

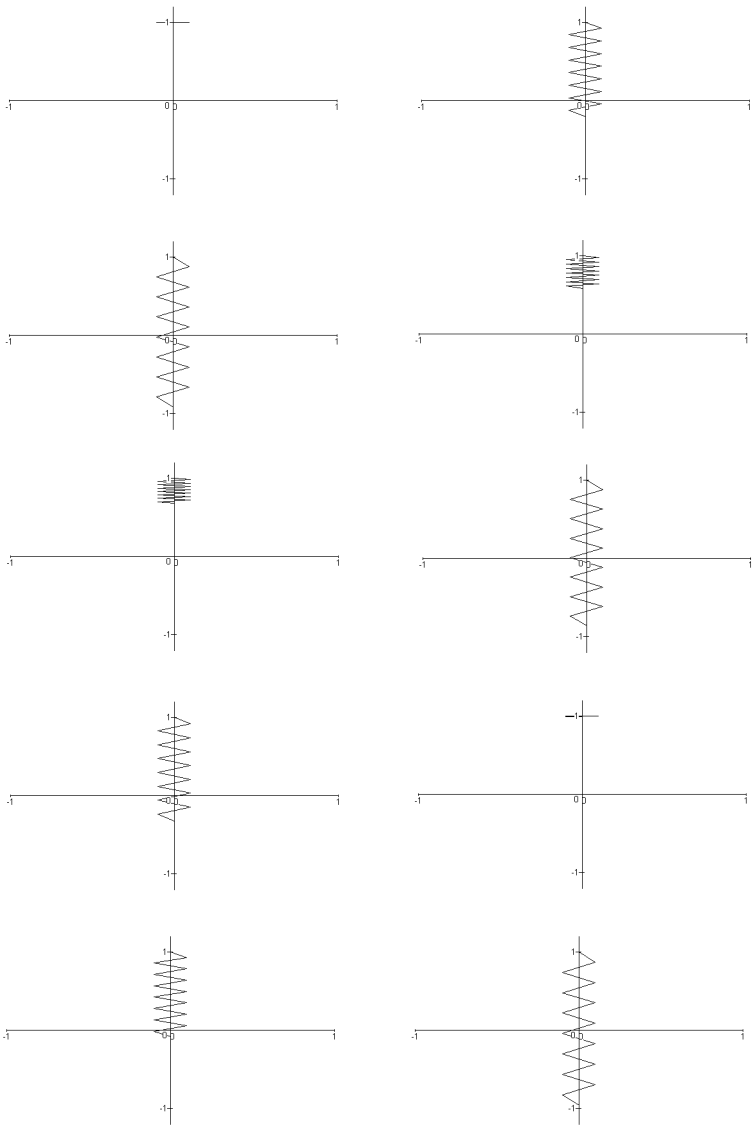


Figura 4.68:

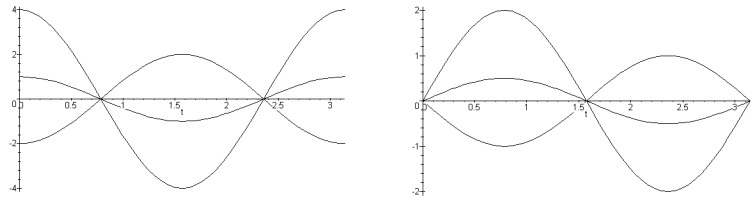


Figura 4.69:

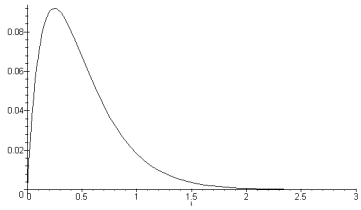


Figura 4.70:

```
> assign(DE): plot(x(t),t=0..3);
```

Grafic: vezi figura 4.70

```
> eps:=0.1: n:=15:
> resort:=proc(t0) local xt0,pts,m;
>    xt0:=evalf(subs(t=t0,x(t))):
>    pts:=[[0,xt0],seq([eps*(-1)^m,xt0+m*(0.1-xt0)/n],m=1..n-1),
>    [0,0.1]]; plot(pts,xtickmarks=2,ytickmarks=2);
> end:
> for k from 0  by 1.75/8 to 1.75 do
> #   plots[display]({resort(k)},view=[-1..1,0..0.1])
> od;
> #plots[display]({seq(resort{k*1.75/8),k=0..8)},insequence=true);
> x:='x':
> DE:=dsolve({Eq,x(0)=-0.5,D(x)(0)=5},x(t)):
> assign(DE): plot(x(t),t=0..2);
```

Grafic: vezi figura 4.71.a

100. (*Studiul mişcării încetinite*)

```
> x:='x': m:='m': k:='k':
> sol:=proc(m0,c0,k0,alpha0,beta0,t0)
>  dsolve({m0*diff(x(t),t$2)+c0*diff(x(t),t)+k0*x(t)=0,x(0)=alpha0,
>  D(x)(0)=beta0},x(t));
>  subs(t=t0,rhs(''));
> end:
> plot(sol(1,1,16,0,1,t),t=0..7);
```

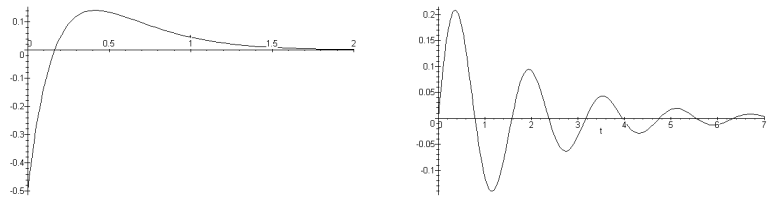


Figura 4.71:

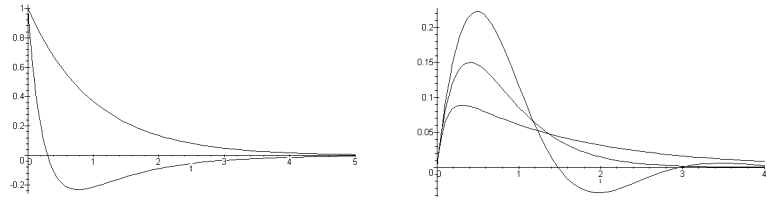


Figura 4.72:

Grafic: vezi figura 4.71.b

```
> plot({sol(1,5,4,1,-1,t),sol(1,5,4,1,-6,t)},t=0..5);
```

Grafic: vezi figura 4.72.a

```
> plot({sol(1,2*sqrt(6),6,0,1,t),sol(1,4*sqrt(6),6,0,1,t),
```

Grafic: vezi figura 4.72.b

```
> sol(1,sqrt(6),6,0,1,t)},t=0..4);
```

101. (*Mişcare forţată*)

```
> restart;
> sol:=proc(f0,m0,c0,k0,alpha0,beta0,t0)
>  eq:={m0*diff(x(t),t$2)+c0*diff(x(t),t)+k0*x(t)=f0,x(0)=alpha0,
>   D(x)(0)=beta0};  dsolve(eq,x(t));
>  evalf(subs(t=t0,rhs(")));
> end:
> sol0(f,t):=sol(f,1,0,4,0,0,t);
```

$$\mathrm{sol0}(f,t) := .2500000000\,f - .2500000000\,f\cos(2.t)$$

```
> plot({subs(f=1,sol0(f,t)),subs(f=cos(t),sol0(f,t)),subs(f=sin(t),
> sol0(f,t)),subs(f=cos(2*t),sol0(f,t)),subs(f=sin(2*t),sol0(f,t))}
> ,t=0..6);
```

Grafic: vezi figura 4.73.a

```
> sol1(t):=sol(cos(t),1,4,13,0,1,t):
> plot(sol1(t),t=0..14);
```

Grafic: vezi figura 4.73.b

```
> x:='x':
> Sol:=dsolve({diff(x(t),t$2)+4*diff(x(t),t)+13*x(t)=sin(t)*
```

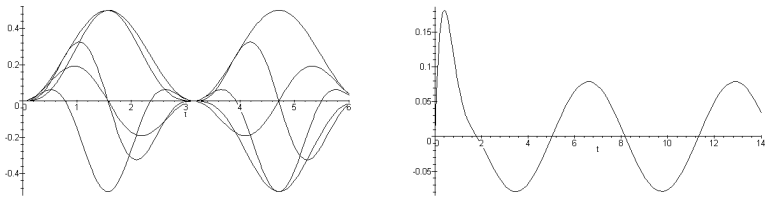


Figura 4.73:

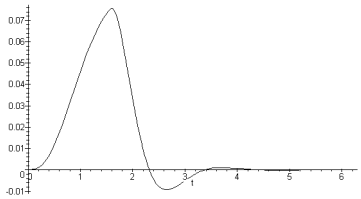


Figura 4.74:

```
> Heaviside(Pi/2-t),x(0)=0,D(x)(0)=0},x(t),laplace);
```

$$Sol := \mathrm{x}(t) = -\mathrm{Heaviside}\left(t-\frac{1}{2}\pi\right)\left(-\frac{7}{120}\,\mathrm{e}^{(-2t+\pi)}\sin\left(3t-\frac{3}{2}\pi\right)\right.$$
$$-\frac{3}{40}\,\mathrm{e}^{(-2t+\pi)}\cos\left(3t-\frac{3}{2}\pi\right)+\frac{1}{40}\sin\left(t-\frac{1}{2}\pi\right)+\frac{3}{40}\cos\left(t-\frac{1}{2}\pi\right)\Big)$$
$$-\frac{1}{120}\,\mathrm{e}^{(-2t)}\sin(3t)+\frac{1}{40}\,\mathrm{e}^{(-2t)}\cos(3t)+\frac{3}{40}\sin(t)-\frac{1}{40}\cos(t)$$

```
> assign(Sol): plot(x(t),t=0..2*Pi);
```

Grafic: vezi figura 4.74

102. (*Circuit LRC*)

```
> dsolve({diff(q(t),t$2)+40*diff(q(t),t)+4000*q(t)=24,q(0)=0,
>   D(q)(0)=0},q(t)):
> assign(''); plot(q(t),t=0..0.35);
```

Grafic: vezi figura 4.75.a

```
> dq:=diff(q(t),t):
> plot(dq,t=0..0.35);
```

Grafic: vezi figura 4.75.b

```
> q(t):='q(t)':
> dsolve({100*diff(q(t),t)+100*q(t)=100*Heaviside(1-t),q(0)=0},q(t)
> ,laplace);
```

$$\mathrm{q}(t) = -\mathrm{Heaviside}(-1+t)\,(1-\mathrm{e}^{(1-t)})+1-\mathrm{e}^{(-t)}$$

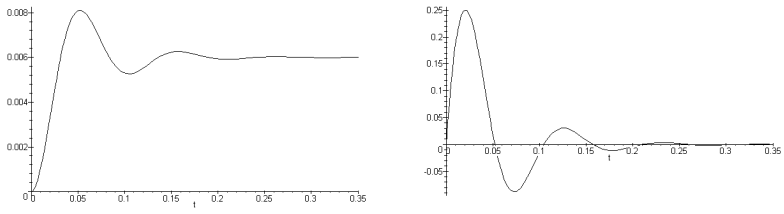


Figura 4.75:

Figura 4.76:

```
> assign(");
> i:=diff(q(t),t);
```

$$i := -\mathrm{Dirac}(-1+t)\,(1-\mathrm{e}^{(1-t)}) - \mathrm{Heaviside}(-1+t)\,\mathrm{e}^{(1-t)} + \mathrm{e}^{(-t)}$$

```
> plot(i,t=0..2);
```

Grafic: vezi figura 4.76

103. (*Pendul*)

```
> restart;
> Eq:=dsolve({diff(x(t),t$2)+x(t)=0,x(0)=a,D(x)(0)=b},x(t));
```

$$Eq := \mathrm{x}(t) = a\cos(t) + b\sin(t)$$

```
> assign(Eq): pen:=subs({a=0,b=2},x(t));
```

$$pen := 2\sin(t)$$

```
> sol:=dsolve({diff(theta(t),t$2)+4*diff(theta(t),t)+20*theta(t)=0,
>  theta(0)=1, D(theta)(0)=2}, theta(t));
```

$$sol := \theta(t) = \mathrm{e}^{(-2t)}\cos(4t) + \mathrm{e}^{(-2t)}\sin(4t)$$

```
> assign(sol): plot(theta(t),t=0..2);
```

Grafic: vezi figura 4.77

```
> pendul:=proc(t0) local pt1, xt0;
>   xt0:=evalf(subs(t=t0,theta(t)));
>   pt1:=[8/5*cos(3*Pi/2+xt0),8/5*sin(3*Pi/2+xt0)];
>   plot([[0,0],pt1],xtickmarks=2,ytickmarks=2);
> end:
```

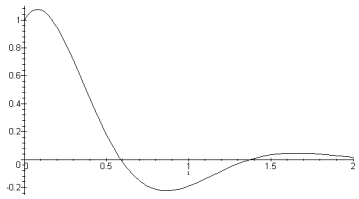


Figura 4.77:

```
> k_vals:=seq(k*2/9,k=0..9):
> for k in k_vals do
>  plots[display]({pendul(k)},view=[-2..2,-2..0]) od;
> #plots[display]({seq(pendul(k*2/9),k=0..9)},insequence=true);
```

Grafic: vezi figura 4.78

```
> theta:='theta':
> Eq:=diff(theta(t),t$2)+.5*diff(theta(t),t)+sin(theta(t))=0:
> s:=proc(pair,t0) local d1;
>   d1:=dsolve({Eq,theta(0)=pair[1],D(theta)(0)=pair[2]},
>   theta(t),numeric); d1(t0);
> end:
> t1:=[[-1,0],[-.5,0],[.5,0],[1,0]]: s([-1,0],1);
```

$$\left[t = 1, \theta(t) = -.6583967582216471, \frac{\partial}{\partial t}\theta(t) = .5953006449551431\right]$$

```
> i:=1: for pp in t1 do l1[i]:=[seq([t0/2,rhs(s(pp,t0/2)[2])],
>  t0=0..20)]; i:=i+1; od:
> plot({seq(l1[i],i=1..nops(t1))});
```

Grafic: vezi figura 4.79.a

```
> t2:=[[0,-2],[0,-1],[0,1],[0,2]]:
> i:=1: for pp in t2 do l2[i]:=[seq([t0/2,rhs(s(pp,t0/2)[2])],
>  t0=0..20)];i:=i+1; od:
> plot({seq(l2[i],i=1..nops(t2))});
```

Grafic: vezi figura 4.79.b

```
> t3:=[[1,1],[-1,-1],[1,5],[-1,-5]]:
> i:=1: for pp in t3 do l3[i]:=[seq([t0/2,rhs(s(pp,t0/2)[2])],
>  t0=0..20)];i:=i+1; od:
> plot({seq(l3[i],i=1..nops(t3))});
```

Grafic: vezi figura 4.80

104. (*Calibrare*)

```
> deq:={diff(xa(t),t,t)+a^2*xa(t)=0, xa(0)=0, D(xa)(0)=v0}:
> dsolve(deq,xa(t));
```

$$\mathrm{xa}(t) = \frac{v0\sin(a\,t)}{a}$$

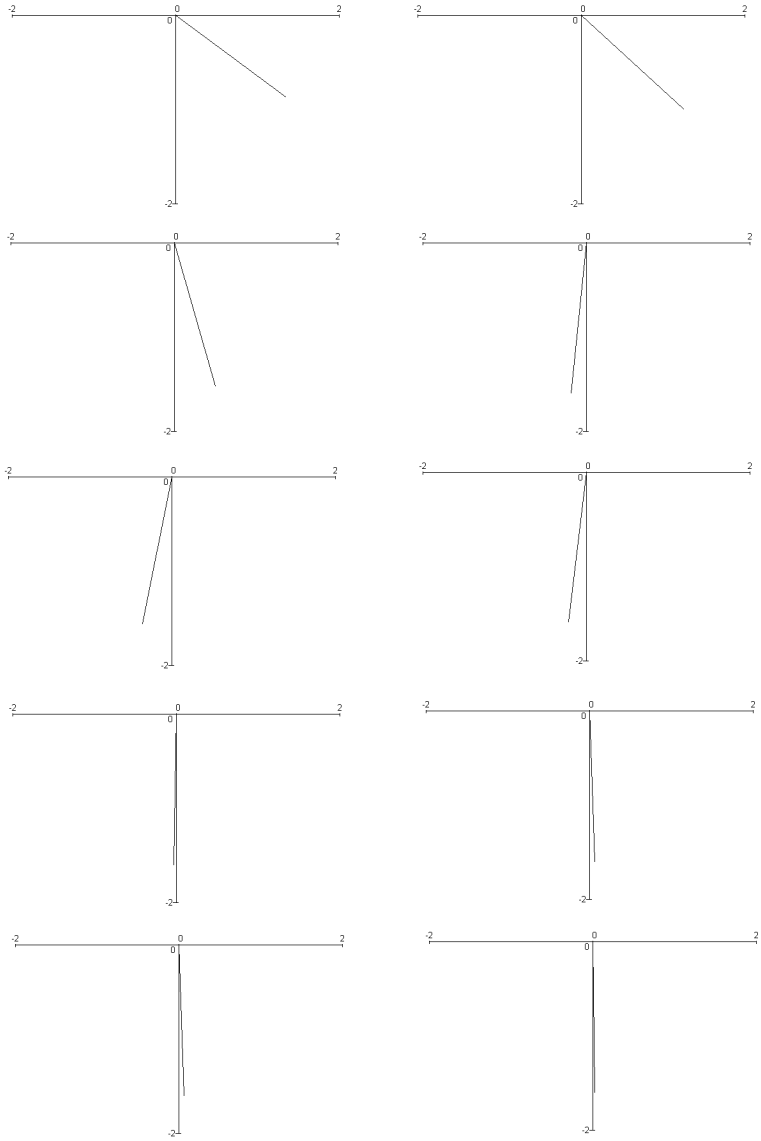


Figura 4.78:

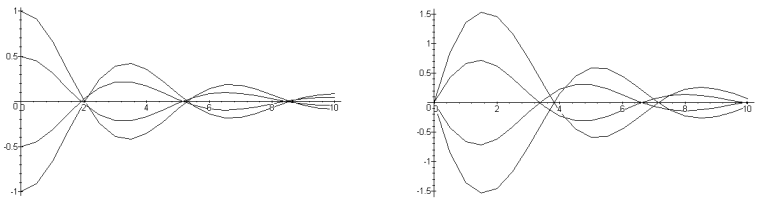


Figura 4.79:

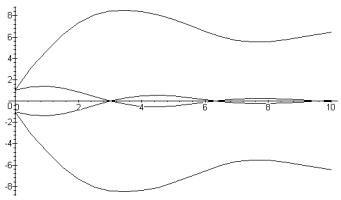


Figura 4.80:

```
> Xa(t):=rhs('):
> eqb:=diff(xb(t),t,t)+a^2*xb(t)+b*xb(t)^2+2*b*Xa(t)*xb(t)
> +b*(Xa(t))^2=0:
> incondb:= xb(0)=0, D(xb)(0)=0:
> alpha:=3.8*10^9: beta:=30*10^9: d:=0.015: l:=0.02: M:=5: v0:=5:
> a:=evalf(sqrt(alpha*Pi*d^2/(4*M*l))): b:=beta*Pi*d^2/(4*M*l^2):
> tm:=min(evalf(sqrt(6)/a),evalf(21/(v0*b))^(1/3),
> evalf(21/(a*b*v0))^(1/4)):
> ORD:=[seq(i,i=6..10)]:
> for ord in ORD do
>    Order:=ord:
>    dsolve({eqb,incondb},xb(t),series);
>    if type(',set) then pol:=convert(rhs('[1]),polynom); else
>             pol:=convert(rhs('),polynom); fi;
>    sol:=pol+Xa(t);
>    solp:=(sol/l)*(alpha+beta*sol/l)/10^9;
>    X[ord]:=plot(sol,t=0..tm);
>    P[ord]:=plot(solp,t=0..tm);
> od:
> plots[display]({seq(X[i],i=ORD)});
```

Grafic: vezi figura 4.81.a

```
> plots[display]({seq(P[i],i=ORD)});
```

Grafic: vezi figura 4.81.b

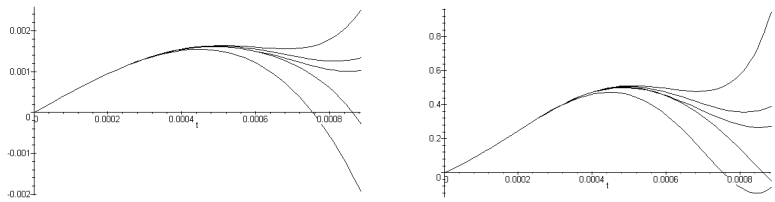


Figura 4.81:

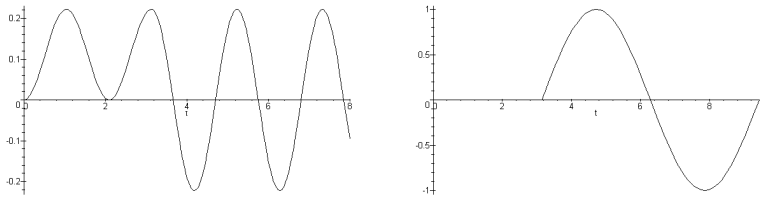


Figura 4.82:

105. (*ODE cu funcţii discontinue*)

```
> f:=t->Heaviside(t)-Heaviside(t-Pi): y(t):='y(t)':
> Eq:=diff(y(t),t$2)+9*y(t)=f(t):
> # rezultat lung la urmatoarea instructiune
> # dsolve({Eq,y(0)=0,D(y)(0)=0},y(t));
> dsolve({Eq,y(0)=0,D(y)(0)=0},y(t),laplace);
```

$$\mathrm{y}(t) = \frac{1}{9} - \frac{1}{9}\cos(3\,t) - \mathrm{Heaviside}(t-\pi)\left(\frac{1}{9} - \frac{1}{9}\cos(3\,t-3\,\pi)\right)$$

```
> assign(");
> plot(y(t),t=0..8);
```

Grafic: vezi figura 4.82.a

```
> y(t):='y(t)':
> dsolve({diff(y(t),t$2)+y(t)=Dirac(t-Pi),y(0)=0, D(y)(0)=0},y(t));
```

$$\mathrm{y}(t) = -\mathrm{Heaviside}(t-\pi)\sin(\pi)\cos(t) + \mathrm{Heaviside}(t-\pi)\cos(\pi)\sin(t)$$

```
> dsolve({diff(y(t),t$2)+y(t)=Dirac(t-Pi),y(0)=0, D(y)(0)=0},y(t),
>   laplace);
```

$$\mathrm{y}(t) = \mathrm{Heaviside}(t-\pi)\sin(t-\pi)$$

```
> assign(");
> plot(y(t),t=0..3*Pi);
```

Grafic: vezi figura 4.82.b

106. (*Sfera metalică*)

```
> a:='a': sol:=dsolve(diff(T(r),r)=-a/(r^2),T(r));
```

$$sol := \mathrm{T}(r) = \frac{a + \_C1\,r}{r}$$

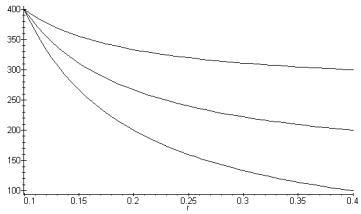


Figura 4.83:

```
> T:=unapply(rhs(sol),r):
> sol:=solve({T(r1)=T1, T(r2)=T2},{a,_C1});
```

$$sol := \left\{ \_C1 = \frac{T1\ r1 - T2\ r2}{r1 - r2}, a = -\frac{r1\ r2\ (-T2 + T1)}{r1 - r2} \right\}$$

```
> assign(sol);
> normal(T(r));
```

$$\frac{r2\ r1\ T2 - r2\ r1\ T1 + r\ T1\ r1 - r\ T2\ r2}{(r1 - r2)\ r}$$

```
> T:='T':
> inicon:=T(r1)=T1, T(r2)=T2:
> deq:=diff(T(r),r,r)+diff(T(r),r)*2/r=0:
> sol:=simplify(dsolve({deq,inicon},T(r))):
> T:=unapply(rhs(sol),r);
```

$$T := r \to \frac{r2\ r1\ T2 - r2\ r1\ T1 + r\ T1\ r1 - r\ T2\ r2}{(r1 - r2)\ r}$$

```
> r1:=0.1: T1:=400: r2:=0.4: k:=12:
> TTs:=[100,200,300]:
> for T2 in TTs do
>   Tpl[T2]:=T(r);
> od:
> plot({seq(Tpl[T2],T2=TTs)},r=r1..r2);
```

Grafic: vezi figura 4.83

107. (*Câmp*)

```
> T:='T': a:='a': eq:='eq':
> Subs:=u=x/(2*a*sqrt(t)):
> T(x,t):=V(op(Subs)[2]):
> eq:=a^2*diff(T(x,t),x,x)=diff(T(x,t),t):
> eq:=subs(x=solve(Subs,x),eq)*t*4:
> Sol:=dsolve(eq,V(u)):
> T:=unapply(rhs(subs(u=rhs(Subs),Sol)),x,t);
```

$$T := (x, t) \to \_C2 + \_C3\ \mathrm{erf}\left(\frac{1}{2}\frac{x}{a\sqrt{t}}\right)$$

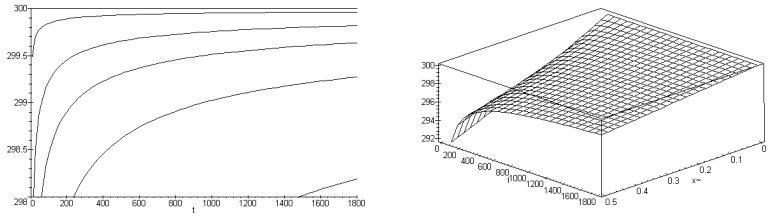


Figura 4.84:

```
> assume(a>=0): assume(x>=0):
> BounCon:=Ts=limit(T(x,t),x=0,right);
```

$$BounCon := Ts = \_C2$$

```
> Inicon:=Tf=limit(T(x,t),t=0,right);
```

$$Inicon := Tf = \_C3 + \_C2$$

```
> Sol:=solve({Inicon,BounCon},{_C2,_C3});
```

$$Sol := \{\, \_C2 = Ts, \_C3 = Tf - Ts \,\}$$

```
> assign(Sol);
> T(x,t);
```

$$Ts + (\, Tf - Ts \,) \operatorname{erf}\left(\frac{1}{2}\,\frac{x\_}{a\_\sqrt{t}}\right)$$

```
> Ts:=300: Tf:=285: a:=sqrt(0.003):
> Dvec:=[0,0.01,0.05,0.1,0.2,0.5]: for d in Dvec do Td[d]:=T(d,t) od:
> plot({seq(Td[d],d=Dvec)}, t=0..1800,298..300);
```

Grafic: vezi figura 4.84.a

```
> plot3d(T(x,t),x=0..0.5,t=0..1800);
```

Grafic: vezi figura 4.84.b

108. (*Sistem ODE 1*)

```
> dsolve({diff(x(t),t)=x(t)-4*y(t)+1, diff(y(t),t)=x(t)+y(t),
>   x(0)=1, y(0)=0}, {x(t),y(t)}):
> assign('');
> plot([x(t),y(t),t=0..10]);
```

Grafic: vezi figura 4.85.a

109. (*Sistem ODE 2*)

```
> dsolve({diff(x(t),t)=x(t)+2*y(t)+z(t), diff(y(t),t)=6*x(t)-y(t),
>  diff(z(t),t)=-x(t)-2*y(t)-z(t)},{x(t),y(t),z(t)});
```

$$\left\{ \mathrm{z}(t) = -13\,\_C1 + \_C2\,\mathrm{e}^{(3t)} + \_C3\,\mathrm{e}^{(-4t)}, \mathrm{x}(t) = \_C1 - \_C2\,\mathrm{e}^{(3t)} - \_C3\,\mathrm{e}^{(-4t)}, \right.$$

$$\left. \mathrm{y}(t) = 6\,\_C1 - \frac{3}{2}\,\_C2\,\mathrm{e}^{(3t)} + 2\,\_C3\,\mathrm{e}^{(-4t)} \right\}$$

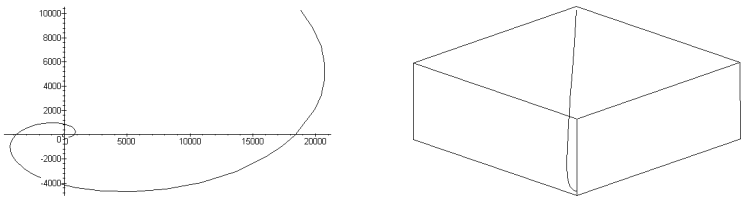


Figura 4.85:

```
> Sol:=subs(_C1=1,_C2=2,_C3=1.5,");
```

$$Sol := \{\mathrm{z}(t) = -13 + 2\,\mathrm{e}^{(3t)} + 1.5\,\mathrm{e}^{(-4t)}, \mathrm{y}(t) = 6 - 3\,\mathrm{e}^{(3t)} + 3.0\,\mathrm{e}^{(-4t)},$$
$$\mathrm{x}(t) = 1 - 2\,\mathrm{e}^{(3t)} - 1.5\,\mathrm{e}^{(-4t)}\}$$

```
> assign(");
> plots[spacecurve]([x(t),y(t),z(t)],t=0..1);
```

Grafic: vezi figura 4.85.b

110. (*Sistem ODE 3*)

```
> dsolve({diff(x(t),t)=2*x(t)+5*y(t)+cos(4*t), diff(y(t),t)=-4*x(t)-
> 2*y(t)+sin(4*t), x(0)=x0, y(0)=y0},{x(t),y(t)},laplace);
```

$$\left\{\mathrm{x}(t) = -\frac{1}{8}\,t\cos(4\,t) + \frac{1}{4}\,t\sin(4\,t) + \frac{9}{32}\sin(4\,t) + x0\cos(4\,t) + \frac{1}{2}\,x0\sin(4\,t)\right.$$
$$+ \frac{5}{4}\,y0\sin(4\,t),$$
$$\left.\mathrm{y}(t) = \frac{1}{4}\,t\cos(4\,t) - \frac{1}{16}\sin(4\,t) + y0\cos(4\,t) - \frac{1}{2}\,y0\sin(4\,t) - x0\sin(4\,t)\right\}$$

```
> assign(");
> vals:=[seq(seq([i,j],i=-1..1),j=-1..1)]:
> for k from 1 to nops(vals) do
> # plot(subs({x0=vals[k][1],y0=vals[k][2]},[x(t),y(t),t=0..10])):
> od;
```

111. (*Sistem ODE 4*)

```
> epsilon:=2/25: b:=0: a:=7/10: u:=1:
> syseq:={diff(v(xi),xi)=w(xi), diff(w(xi),xi)=1/3*v(xi)^3-v(xi)+
> r(xi)-u*w(xi), diff(r(xi),xi)=epsilon/u*(b*r(xi)-v(xi)-a),
> v(0)=1, w(0)=0, r(0)=1}:
> num_sol:=dsolve(syseq,{v(xi),w(xi),r(xi)},numeric):
> f1:=t->rhs(num_sol(t)[2]): f2:=t->rhs(num_sol(t)[3]):
> f3:=t->rhs(num_sol(t)[4]):
> l1:=seq([t/3,f1(t/3)],t=0..90): l2:=seq([t/3,f2(t/3)],t=0..90):
> l3:=seq([t/3,f3(t/3)],t=0..90):
> l4:=seq([f1(t/3),f2(t/3),f3(t/3)],t=0..90):
> plot([l1]); plot([l2]); plot([l3]); plots[pointplot]([l4]);
```

Grafic: vezi figura 4.86

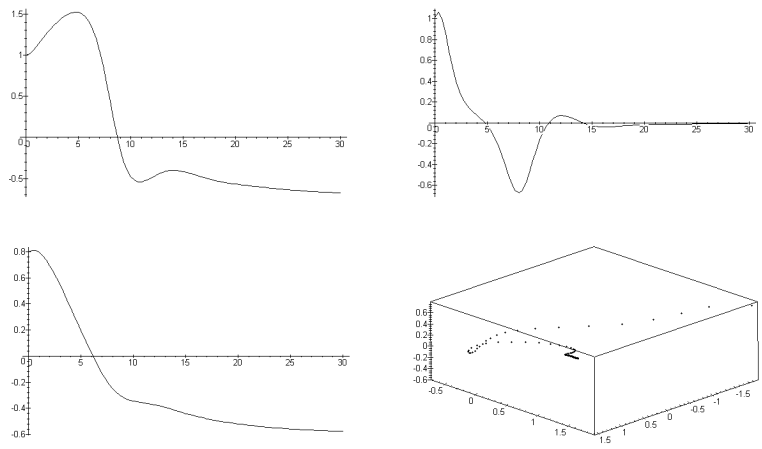


Figura 4.86:

112. (*Metoda Runge-Kutta standard*)

```
> t:=(n,h)->n*h:
> rk:=proc(f,rk0,n,h,dim) local k1,k2,k3,k4,i,j, vechi;
>  option remember;
>  k1:=array(1..dim): k2:=array(1..dim): k3:=array(1..dim):
>  k4:=array(1..dim): vechi:=array(1..dim):
>  if n=0 then seq(rk0[j],j=1..dim) else
>   for i from 1 to dim do
>    vechi[i]:=rk(f,rk0,n-1,h,dim)[i]; od;
>  for i from 1 to dim do
>    k1[i]:=f(t(n-1,h),seq(vechi[j],j=1..dim))[i]; od;
>  for i from 1 to dim do
>    k2[i]:=f(t(n-1,h)+h/2,seq(vechi[j]+h/2*k1[j],j=1..dim))[i]; od;
>  for i from 1 to dim do
>    k3[i]:=f(t(n-1,h)+h/2,seq(vechi[j]+h/2*k2[j],j=1..dim))[i]; od;
>  for i from 1 to dim do
>    k4[i]:=f(t(n-1,h)+h,seq(vechi[j]+h*k3[j],j=1..dim))[i]; od;
>  seq(vechi[j]+h/6*(k1[j]+2*k2[j]+2*k3[j]+k4[j]),j=1..dim);
> fi;
> end:
> f:=(t,x,y)->[x-y+1,x+3*y+exp(-t)]: rk0:=array([0,1]):
> rk(f,rk0,10,0.1,2);
```

$$-7.973791094, 20.89746891$$

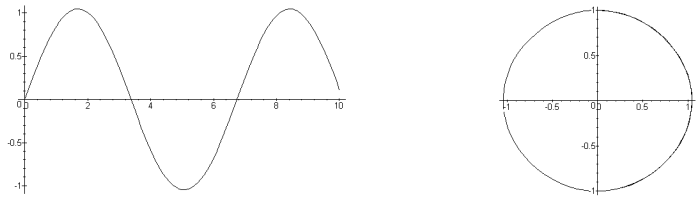


Figura 4.87:

```
> sol:=dsolve({diff(x(t),t)=x(t)-y(t)+1, diff(y(t),t)=x(t)+3*y(t)
> +exp(-t), x(0)=0, y(0)=1}, {x(t), y(t)}):
> assign(sol):
> #Pentru comparare valori se elimina caracterul de comentariu
> #array([seq([t(i,0.1),rk(f,rk0,i,0.1,2)[1],evalf(subs(t=t(i,0.1)
>  ,x(t) ))],i=0..10)]);
> #array([seq([t(i,0.1),rk(f,rk0,i,0.1,2)[2],evalf(subs(t=t(i,0.1)
>  ,y(t) ))],i=0..10)]);
> restart;
> f:=(t,x,y)->[y,-sin(x)]: rk0:=array([0,1]):
> rk(f,rk0,10,0.10,2);
```

$.8477981672, .5685692634$

```
> grafic:=seq([t(i,0.1),rk(f,rk0,i,0.10,2)[1]],i=0..100):
> plot([grafic]);
```

Grafic: vezi figura 4.87.a

```
> toplot:=seq([rk(f,rk0,i,0.10,2)],i=0..100):
> plot([toplot]);
```

Grafic: vezi figura 4.87.b

113. (*Soluţii periodice*)

```
> eqs:=diff(x(t),t)=2*x(t)-6*x(t)*y(t)/5,
>  diff(y(t),t)=-y(t)+9*x(t)*y(t)/10:
> sol1:=dsolve({eqs,x(0)=1,y(0)=0.5},{x(t),y(t)},numeric):
> sol2:=dsolve({eqs,x(0)=1,y(0)=1},{x(t),y(t)},numeric):
> sol1(0.), sol2(0.);
```

$[t = 0, \mathrm{x}(t) = 1., \mathrm{y}(t) = .5000000000000000], [t = 0, \mathrm{x}(t) = 1., \mathrm{y}(t) = 1.]$

```
> L:=seq([i/30,rhs(sol1(i/30)[2]),rhs(sol1(i/30)[3])],i=0..300):
> Lp:=seq([i/30,rhs(sol2(i/30)[2]),rhs(sol2(i/30)[3])],i=0..300):
> L1:=seq([L[i][2],L[i][3]],i=1..301):
> Lp1:=seq([Lp[i][2],Lp[i][3]],i=1..301): plot({[L1],[Lp1]});
```

Grafic: vezi figura 4.88

```
> L2:=seq([L[i][1],L[i][2]],i=1..301):
> Lp2:=seq([Lp[i][1],Lp[i][2]],i=1..301): plot({[L2],[Lp2]});
```

Grafic: vezi figura 4.89.a

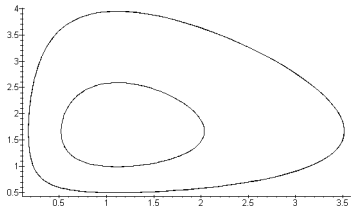


Figura 4.88:

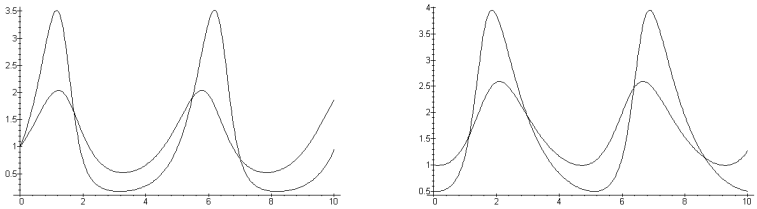


Figura 4.89:

```
> L3:=seq([L[i][1],L[i][3]],i=1..301):
> Lp3:=seq([Lp[i][1],Lp[i][3]],i=1..301): plot({[L3],[Lp3]});
```

Grafic: vezi figura 4.89.b

```
> j:=0: for i from 120 to 180 do if abs(L[i][2]-1)<abs(L[i-1][2]-1)
>  then j:=i; fi: od; evalf(j/30);
```

5.066666667

```
> j:=0: for i from 120 to 180 do if abs(Lp[i][3]-1)<abs(Lp[i-1][3]-1)
>  then j:=i; fi; od; evalf(j/30);
```

4.866666667

```
> j:=0: for i from 120 to 180 do if abs(L[i][3]-0.5)<abs(L[i-1][3]-0.5)
>   then j:=i; fi; od; evalf(j/30);
```

5.200000000

```
> j:=0: for i from 120 to 180 do if abs(Lp[i][3]-1)<
>  abs(Lp[i-1][3]-1) then j:=i; fi; od; evalf(j/30);
```

4.866666667

114. (*Ecuaţia căldurii*)

```
> sols:=dsolve({diff(S(x),x$2)=0,S(0)=10,S(1)=60},S(x));
```

$$sols := \mathrm{S}(x) = 10 + 50\,x$$

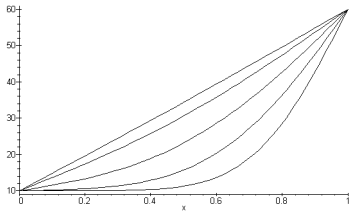


Figura 4.90:

```
> assign(sols);
> for i from 1 to 20 do
>   c[i]:=2*int((10-S(x))*sin(i*Pi*x),x=0..1); od:
> approx:=proc(k) option remember;
>     if k=0 then S(x);
>     else approx(k-1)+c[k]*sin(k*Pi*x)*exp(-k^2*Pi^2*t);
>    fi;
> end:
> tval:=0.025,0.05,0.1,0.2,0.4:
> plot({seq(subs(t=i,approx(15)),i=tval)},x=0..1);
```

Grafic: vezi figura 4.90

115. (*Ecuaţia undei*)

```
> restart;
> f:=x->sin(Pi*x): g:=x->3*x+1:
> readlib(`evalf/int`):
> a:=proc(n) option remember;
>     evalf(2*`evalf/int`(f(x)*sin(n*Pi*x),x=0..1))
> end:
> b:=proc(n) option remember;
>     evalf(2*`evalf/int`(g(x)*sin(n*Pi*x),x=0..1))
> end:
> u:=n->(a(n)*cos(n*Pi*t)+b(n)*sin(n*Pi*t))*sin(n*Pi*x):
> approx:=proc(k) option remember; approx(k-1)+u(k) end:
> approx(1):=u(1):
> plots[animate](approx(5),x=0..1,t=0..2,frames=10);
```

Grafic: vezi figura 4.91

116. (*Ecuaţia potenţialului*)

```
> restart;
> readlib(`evalf/int`):
> B:=proc(n) option remember;
> evalf(2/sinh(2*n*Pi)*`evalf/int`(x*(1-sin(x))*sin(n*Pi*x),x=0..1))
> end:
```

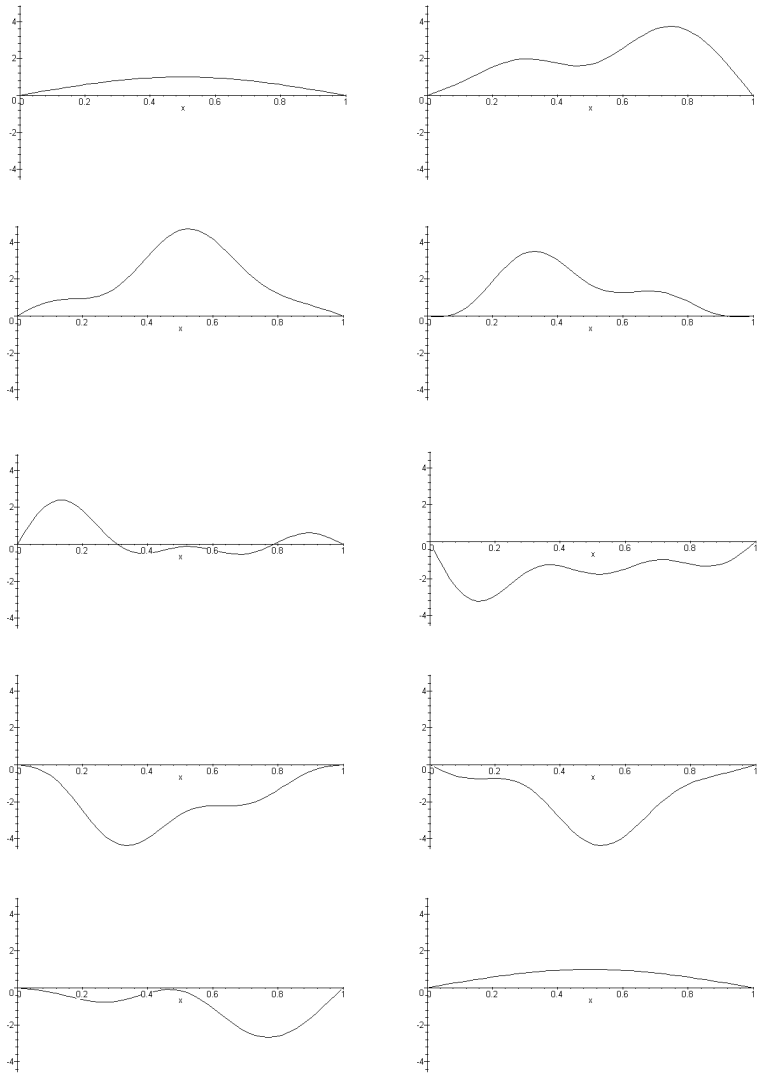


Figura 4.91:

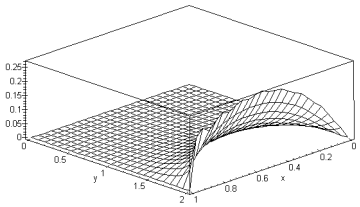


Figura 4.92:

```
> approx:=proc(k) option remember;
>    approx(k-1)+B(k)*sinh(k*Pi*y)*sin(k*Pi*x)
> end:
> approx(1):=B(1)*sinh(Pi*y)*sin(Pi*x):
> plot3d(approx(15),x=0..1,y=0..2,axes=BOXED);
```

Grafic: vezi figura 4.92

Capitolul 5

Probleme propuse

1. Pentru punctele $A = (3, 2, 1)$, $B = (2, 5, 3)$ şi $C = (7, 9, 5)$:

a) Găsiţi lungimea vectorilor AB, BC, CA. Verificaţi că suma a oricare două asemenea lungimi este mai mare decât a treia.

b) Determinaţi măsura unghiurilor ABC, BCA, BAC. Verificaţi că suma lor este π.

c) Determinaţi proiecţia vectorului AB pe dreapta AC.

d) Determinaţi aria triunghiului ABC.

2. Construiţi o secvenţă animată care prezintă o trecere lentă de la un obiect tridimensional la cel obţinut printr-o transformare liniară.

3. Construţi o procedură pentru obţinerea simetricului unui obiect tridimensional faţă de planul ce trece prin trei puncte necoliniare date.

4. Se presupune că un individ crează un cont la bancă în care introduce 1000 unităţi monetare cu 10% dobândă anuală. Ce sumă se va afla în cont după 10 ani?

5. Reconsideraţi problema *Meciului*. Utilizând valorile iniţiale ale lui v_1 şi v_2, cât de aproape trebuie să fie apărătorul de atacant la momentul iniţial astfel încât să-l ajungă la linia porţii?

6. Determinaţi aria regiunii cuprinse între axa x şi curba $y = \sin(x)$ pe intervalul $[0, \pi]$. Găsiţi linia verticală $x = a$ care divide regiunea în două jumătăţi de arii egale. Figuraţi rezultatul pe grafic.

7. Găsiţi lungimea graficului parabolei $y = x^2$ de la $O(0,0)$ la $P(10,100)$. Găsiţi punctul $Q(a, a^2)$ de pe grafic care se află la 10 unităţi depărtare de O de-a lungul graficului. Indicaţi printr-un grafic aceste puncte.

8. Determinaţi volumul solidului de rotaţie obţinut prin rotaţia în jurul axei x a regiunii cuprinsă între graficul funcţiei $y = e^x \sin(x)$ pe $[0, n\pi]$ şi axa x. Figuraţi acest solid. Se apropie acest volum de o limită finită când n devine mare?

9. Rezolvaţi problema celor mai mici pătrate pentru $f(x) = a_1x^2 + a_2x + a_3$ sau $f(x) = a_1e^x + a_2x + a_3$. Comparaţi rezultatele.

10. Definiţi o procedură `getqr(A,'Q','R')` care primeşte o matrice $m \times n$, A, de rang $n \leq m$, şi returnează o matrice Q ortogonală şi o matrice triunghiular superioară R astfel încât $A = QR$.

11. a) Construiţi o procedură pentru aproximarea unei integrale definite, utilizând o secvenţă de subdiviziuni ale intervalului de integrare, prin regula Simpson:

$$\int_a^b f(x)dx = \frac{d}{3}\left(f(a)+f(b)+4\sum_{i=1}^{kn/2} f(a+(2i-1)d)+2\sum_{i=1}^{kn/2-1} f(a+2id)\right).$$

Verificaţi procedura pentru $\int_0^1 e^{x^2}dx$.

b) Comparaţi eroarea cu cea produsă de regula trapezului (pe un caz concret).

c) Introduceţi condiţia de stopare a aproximaţiilor când diferenţa dintre două aproximaţii succesive este sub un nivel ϵ.

12. Reconsideraţi problema *turnului de apă*.

a) Modificaţi rezolvarea în condiţiile în care poziţia observatorului este $(0,0,h)$.

b) Care este cel mai mic h pozitiv care indică o poziţie din care se poate observa punctul (10,0,0)?

c) Cum se modifică soluţia dacă se înlocuieşte sfera cu un cilindru parabolic $z = 1 - x^2$?

13. Care este cea mai scurtă scară care se sprijină de parabola $y = 3 - x^2$, are piciorul pe axa x şi capătul opus pe axa y?

14. Fie elipsa $x^2/25 + y^2 = 1$. Estimaţi aria elipsei.

15. Se consideră funcţia: $f(x) = 10x^2 - x^4 + 10$.

a) Trasaţi regiunea R unde graficul lui f se află deasupra axei x.

b) Determinaţi aria regiunii R.

16. Calculaţi integralele definite pentru funcţiile:

a) $f(x) = (1 - x + x^2)^2$, pentru $x = -1..2$

b) $f(x) = \cos^2 x + 3x$, pentru $x = 0..1$

c) $f(x) = \begin{cases} \sqrt{(}x) & x < 1 \\ x^2 \end{cases}$, pentru $x = 0..2$.

17. Reconsideraţi problema *barei de metal*.

a) Depinde distribuţia temperaturii de distribuţia iniţială? Experimentaţi schimbând pasul iniţial de la [0,0,0] la alte valori, de exemplu [10,20,10].

b) Se presupune că nu există pierdere de căldură, astfel încât procentul de 90% se transformă în 100%. Ce se va întâmpla?

c) Dacă bara este lungimea de 6 m şi se utilizează 5 termometre, care este minimul dintre temperaturile stabile înregistrate?

18. Cum putem împărţi o sferă în trei părţi de suprafeţe egale?

19. Printre mulţimea de tor-uri cu un volum fixat care este cel cu suprafaţa minimală?

20. Reluaţi problema *metodei Picard*. Ce se întâmplă dacă se schimbă termenul iniţial al şirului lui Picard? De exemplu, înlocuiţi y_0 cu funcţia cosinus.

21. Studiaţi convergenţa şirului Picard pentru problemele:

a) $y' = 2x(1+y), y(0) = 1$.

b) $y' = \sin(x+y), y(0) = 0$.

22. Rezolvaţi ecuaţia Euler $x^2y'' + axy' + 5y = 0$.

a) Pentru ce valori ale lui a ambele soluţii fundamentale sunt funcţii reale?

b) Fie $a = 1$. Găsiţi o soluţie reală pentru ecuaţia Euler (Găsiţi o combinaţie liniară a soluţiilor complexe fundamentale).

23. Construiţi o funcţie de conversie din :
a) coordonate cilindrice în coordonate carteziene şi invers.
b) coordonate sferice în coordonate carteziene şi invers.

24. Să se scrie:
a) `swap`(A, r, s) – o procedură care interschimbă liniile r şi s ale lui A.
b) `scale`(A, r, c) – o procedură care scalează linia r a lui A cu un număr nenul c.
c) `replace`(A, r, m, p) – o procedură care înlocuieşte în A linia r cu suma dintre linia r şi linia p înmulţită cu factorul m.
d) `proj`(x, V) – coordonatele vectorului x în spaţiul generat de coloanele matricei inversabile V.
e) `Gauss`(r, A) – utilizând linia r ca linie pivot, crează zerouri pe coloana r a lui A, sub diagonala principală.

25. Scrieţi o procedură apelată `remove1`(x, L) care elimină prima apariţie a lui x din lista L, iar dacă x nu apare în listă, returnează 'Eroare'.

26. Scrieţi o procedură apelată `variance`(x) care operează asupra listei x şi calculează $\sum_{i=1}^{n}(x_i - \mu)^2$ unde n este este numărul elementelor din lista x iar μ este media numerelor din x.

27. Scrieţi o procedură care calculează norma Frobenius a unei matrice cu elemente reale. Norma Frobenius a matricei A este $\sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}|A_{ij}|^2}$.

28. Scrieţi o procedură care sortează o listă utilizând algoritmul `Quicksort`.

29. Scrieţi o procedură care calculează polinoamele Fibonacci $F_n(x)$ ce satisfac recurenţa liniară $F_0(x) = 1$, $F_1(x) = x$ şi $F_n(x) = xF_{n-1}(x) + F_{n-2}(x)$. Calculaţi şi factorizaţi primele 10 polinoame Fibonacci. Puteţi calcula $F_{50}(x)$?

30. Scrieţi o procedură `structure` care, dată o expresie Maple, returnează strutura expresiei ca arbore: a. pentru obiectele atomice (întregi, şiruri) exact obiectul; b. în caz contrar returnează o listă cu tipul obictului ca prim element, restul elementelor depinzând de structura operanzilor. De exemplu:

```
> structure(x^3*sin(x)/cos(x))
```

trebuie să returneze

$$[*, [\hat{}\,, x, 3], [function, \sin, x], [\hat{}\,, [function, \cos, x] - 1]]$$

31. a) Scrieţi o procedură care, dată o mulţime de valori şi un număr natural n returnează o listă a tuturor combinaţiilor de valori de lungime n. Astfel

```
> comb({a,b,c,d},3);
```

trebuie să producă ca ieşire

$$\{\{a, b, c\},\ \{a, b, d\},\ \{a, c, d\},\ \{b, c, d\}\}$$

b) Rescrieţi apoi procedura pentru a lucra asupra lisrelor care conţin duplicate. De exemplu:

```
> comb([a,b,b,c],2);
```

trebuie să producă ca ieşire

$$[[a, b],\ [a, c],\ [b, b],\ [b, c]]$$

32. 1. Scrieţi o procedură care adună şi multiplică polinoamele univariate $a(x) = \sum_{i=1}^{m} a_i x^i$ şi $b(x) = \sum_{i=0}^{m} b_i x^i$ care sunt reprezentate

a) ca vectori de coeficienţi, de exemplu, $[a_n, a_{n-1}, \ldots, a_1, a_0]$;

b) ca liste înlănţuite de coeficienţi $[a_n, [a_{n-1}, \ldots, [a_1, [a_0, NIL]] \ldots]]$.

2. În cazul în care polinoamele sunt rare (au puţini termeni relativ la gradul polinomului; de exemplu $x^{10} + x + 1$), o reprezentare mai condensată este aceea în care sunt stocate numai termenii nenuli. Modificaţi procedurile de adunare şi înmulţire de mai sus pentru cazul în care argumentele acestora sunt:

a) vectori de termeni nenuli $[[a_n, n], \ldots, [a_i, i] \ldots]$

b) liste înlănţuite cu termenii nenuli de tipul
$[[a_n, n], [\ldots, [a_i, i], [\ldots, NIL]] \ldots]$.

33. Scrieţi o procedură monomial(v, n) care calculează o listă de monomiale de ordin total n în variabilele $[v_1, v_2, \ldots, v_m]$. De exemplu,

```
> monomial([u,v],3);
```

trebuie să returneze lista

$$[u^3,\ u^2 v,\ uv^2,\ v^3]$$

Indicaţie: se consideră expansiunea în serii Taylor a polinomului

$$\sum_{i=1}^{n} \frac{1}{1 - v_i t}$$

în t pentru a obţine t^n.

34. Următoarea iteraţie este cunoscută drept iteraţia Halley:

$$x_{k+1} = x_k - \frac{\dfrac{f(x_k)}{f'(x_k)}}{1 - \dfrac{f(x_k) f"(x_k)}{2[f'(x_k)]^2}}$$

Programaţi această iteraţie şi verificaţi convergenţa sa cubică.

35. Dată o secvenţă de puncte $(x_0, y_0),\ (x_1, y_1), \ldots, (x_n, y_n)$ unde valorile x_i sunt distincte, funcţia Maple `interp` calculează un polinom de grad mai mic sau egal cu n care interpolează punctele utilizând algoritmul Newton de interpolare.

a) Scrieţi o procedură care calculează o funcţie cubică C-spline de interpolare a punctelor date. Un funcţie cubică C-spline este un polinom care pe fiecare interval $[x_i, x_{i+1}]$ este definit prin polinomul cubic $f_i(x) = a_i x^3 + b_i x^2 + c_i x + d_i$, $1 \leq i \leq n$ unde cei $4n$ coeficienţi necunoscuţi sunt unic determinaţi din următoarele $4n$ condiţii: $f_i(x_{i-1}) = y_{i-1}$, $f_i(x_i) = y_i$, $f'_i(x_i) = f'_{i+1}(x_{i-1})$, $f"_i(x_i) = f"_{i+1}(x_{i-1})$, $f"_1(x_0) = 0$, $f"_n(x_n) = 0$ (condiţii impuse de continuitatea de clasă C^2 a funcţiei cubice C-spline).

b) Modificaţi procedura astfel încât rezultatul să fie lista polinoamelor pe intervale. De exemplu:

```
> spline([0,1,2,3],[0,1,1,2],x);
```

$$[-1/3x^3 + 4/3x,\ 2/3x^3 - 3x^2 + 13/3x - 1,\ -1/3x^3 + 3x^2 - 23/3x + 7]$$

36. Deşi nu pentru toate sistemele de ecuaţii diferenţiale se poate găsi o soluţie simbolică, se poate găsi întotdeauna o aproximare cu ajutorul seriilor de puteri.

Considerând problema $y'(x) = f(x, y(x))$, $y(x_0) = y_0$ se caută determinarea soluţiei sub forma $y(x) = \sum_{k=0}^{\infty} y_k x^k$. Construiţi o procedură pentru calculul coeficienţilor seriei până la un ordin dat.

37. Creaţi o listă de numere naturale aleatoare şi determinaţi numerele prime din această listă.

38. Construiţi o listă de funcţii. Selectaţi din această listă pe cele continue.

39. Construiţi următoarele proceduri de operaţii asupra listelor:
a) `deletelist(lista,domeniu_indici)`: şterge elementele din listă cu indicii cuprinşi în domeniu;
b) `extractlist(lista,domeniu_indici)`: extrage din listă sublista cu indicii cuprinşi în domeniu;
c) `insertlist(lista1, lista2, pozitie)`: inserează lista a doua în prima listă începând de la poziţia indicată;
d) `replacelist(lista1, lista2,domeniu)`: înlocuieşte elementele din prima listă cu indicii cuprinşi în domeniu cu elementele celei de a doua liste;
e) `prependlist(expr,lista)`: adaugă expresia ca prim element al listei;
f) `appendlist(expr,lista)`: adaugă expresia ca ultim element al listei.
Construiţi proceduri similare pentru şiruri de caractere.

40. Construiţi următoarele proceduri:
a) `ziplist(lista1,lista2, primsalt,catesalt)` crează o listă unificată: lista a doua va fi inserată în prima listă treptat; `primsalt` reprezintă numărul elementelor primei liste introduse în lista finală înainte de primul element al listei a doua, iar `cateslat` indică câte elemente din lista a doua sunt introduse în lista finală înainte de a fi introdus următorul element al primei liste; procedeul se repetă până la epuizarea listelor.
b) `unziplist(lista,nume,primsalt, catesalt)` desface o listă în două subliste, a doua fiind regăsită prin nume.

41. Construiţi următoarele proceduri pentru structuri tip listă:
a) `reverse(lista)`: inversează elementele unei liste;
b) `rotatelist(lista,n)`: roteşte lista la stânga cu n poziţii;
c) `comparelist(lista1,lista2)`: compară elementele din poziţii corespunzătoare, oferind o listă cu indicii a căror elemente diferă;
d) `intersectlist(lista1,lista2)`: determină elementele comune celor două liste;
e) `unionlist(lista1,lista2)`: determină elementele reunite ale listelor, fără dubluri.
Construiţi proceduri similare pentru şiruri de caractere.

42. Construiţi următoarele proceduri:
a) `flatten(lista)`: elimină parantezele drepte interioare ale unei liste;
b) `uniquelist(lista)`: eliminină reapariţiile termenilor listei;
c) `frequency(expr,lista)`: calculează numărul de apariţii ale termenului `expr` în listă;

43. Construiţi proceduri de conversie de tip:
a) de la şir de caractere la listă de caractere şi invers;
b) de la polinom univariat la vector de coeficienţi şi invers;
c) de la polinom bivariat la matrice de coeficienţi şi invers.

44. Construiţi proceduri pentru:
a) `diagonal(V)`: crează o matrice cu elementele unui vector pe diagonala

principală;

b) `getdiagonal(M)`: extrage elementele de pe diagonala principală a matricei M într-un vector;

c) `sumdiagonal(M)/prodddiagonal(M)`: adună/înmulţeşte toate elementele de pe diagonala principală a matricei M;

d) `banded(V,a..b,n,m):` crează o matrice bandă de dimensiune $n \times m$, vectorul V pe prima linie, cu a poziţia primului element al lui V pe prima linie (probabil $a < 0$) şi b poziţia ultimului element al lui V pe prima linie;

e) `uppertri(A):` converteşte o matrice A la forma superior triunghiulară;

f) `lowertri(A):` converteşte o matrice A la forma inferior triunghiulară;

g) `sumrows(A)/prodrows(A):` adună/înmulţeşte elementele fiecărei linii într-un vector;

h) `sumcols(A)/prodcols(A):` adună/înmulţeşte elementele fiecărei coloane într-un vector;

i) `maprows(fnc,M)/mapcols(fnc,M):` pasează fiecare linie/coloană a matricei M ca parametrii ai funcţiei fnc;

j) `fliprows(M)/flipcols(M):` inversează ordinea liniilor/coloanelor matricei M;

k) `sortbyrow(M,i)/sortbycol(M,i):` interschimbă coloanele/liniile potrivit ordinii crescătoare a elementelor de pe linia/coloana i.

45. O staţie de radio doreşte să transmită pe o arie care include 20 de oraşe. Staţia are 6 turnuri în 6 din aceste oraşe care transmit semnalul radio fiecare la o anumită distanţă. Se dau coordonatele x, y ale oraşelor şi o a treia informaţie privind distanţa efectivă a transmiţătorului din fiecare oraş (valoare nulă dacă oraşul nu are turn de transmitere):

x	y	d
0.534	4.325	2
-2.196	-3.736	0
1.234	0.006	0
4.123	-1.855	0
2.134	-2.321	0
-0.123	-4.486	3/2
3.021	-3.012	0
-4.986	-4.123	0
-3.977	3.920	3/2
-1.808	2.231	0
0.924	-0.756	2
1.738	-3.111	0
2.009	1.845	0
3.997	2.068	0
4.845	-0.848	2
-0.777	4.777	0
-1.069	3.111	0
-4.067	-2.844	0
-3.238	1.825	0
-4.968	-4.111	1

a) Sunt suficiente cele 6 turnuri?

b) Care este factorul de creştere sau descreştere a puterii de transmitere (la toate turnurile acelaşi) pentru a include toate oraşele?

46. Construiţi proceduri pentru:

a) `line3d([x1,y1,z1],[x2,y2,z2])`: trasează linia din spaţiul tridimensional cu capetele în punctele de coordonate careteziene indicate;

b) `polygon3d([[x1,y1,z1],...,[xn,yn,zn]])`: verifică apartenenţa punctelor la un plan din spaţiul tridimesional şi trasează poligonul din planul respectiv a cărui coordonate carteziene sunt date în lista de intrare;

c) `circle3d([x1,y1,z1],[x2,y2,z2])`: trasează cercul circumscris triunghiului din spaţiu descris de punctele cu coordonatele carteziene date ca intrare;

d) `sphere([x,y,z],raza)`: trasează sfera cu centrul în (x, y, x) şi cu raza dată;

e) `cylinder({[x1,y1,z1],[x2,y2,z2],[x3,y3,z3]},inalt)`: trasează cilindrul drept cu baza dată de cercul circumscris celor trei puncte şi înălţimea dată;

f) `cone({[x1,y1,z1],[x2,y2,z2],[x3,y3,z3]},inalt)`: trasează conul drept cu baza dată de cercul circumscris celor trei puncte şi înălţimea dată.

47. Construiţi funcţii pentru translaţie, scalare, rotaţie, reflecţie şi transformare liniară oarecare ce se poate aplica unei liste sau mulţimi de coordonate bidimensionale sau tridimensionale.

Cuprins

Bibliografie

[1] Martha L.Abell, James P.Braselton, Differential Equations with Maple V, Academic Press, London, 1994.

[2] Carl Eberhart, Problem Solving with Maple. A Handbook for Calculus Students, Kentucky, 1995, document pe Internet.

[3] Walter Gandler, Jiri Hrebicek, Solving Problems in Scientific Computing Using Maple and MatLab, Second Edition, Springer Verlag, Berlin, 1995.

[4] G. Keady, Using Maple's linalg package, Monash, 1994, document pe Internet.

[5] Michael Monagan, Programming in Maple: The Basics, Zürich, 1996, document pe Internet.

[6] Darren Redfern, The Maple Handbook, Maple V Release 3, Springer Verlag, New York, 1994.

[7] Darren Redfern, The Practical Approach Utilities for Maple, Maple V Release 3, Springer-Verlag, New York, 1995.

[8] Michael Wester, A Review of CAS Mathematical Capabilities, New Mexico, 1994, document pe Internet.