Lucrarea de laborator nr. 8

I. Scopul lucrării

Metoda Newton

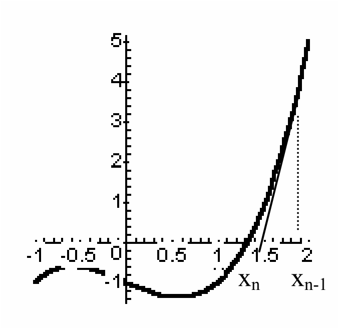
II. Conținutul lucrării

1. Metoda tangentei
2. Metoda Newton – cazul m-dimensional

III. Prezentarea lucrării

III.1. Metoda tangentei

Metoda tangentei este utilizată pentru determinarea unei rădăcini a ecuației f(x) = 0. Presupunem că f este derivabilă şi că derivata nu se anulează. Rădăcina ecuației este determinată ca limita unui şir. Se pleacă de la un punct x_0 dat. Presupunând că s-a construit termenul x_{n-1}, termenul x_n se determină ca fiind abscisa intersecției dintre tangenta la graficul funcției în x_{n-1} şi axa Ox.



Ecuația tangentei în x_{n-1} este:

$$y - f(x_{n-1}) = f'(x_{n-1})(x - x_{n-1})$$

Deci intersecția cu axa Ox se află rezolvând sistemul

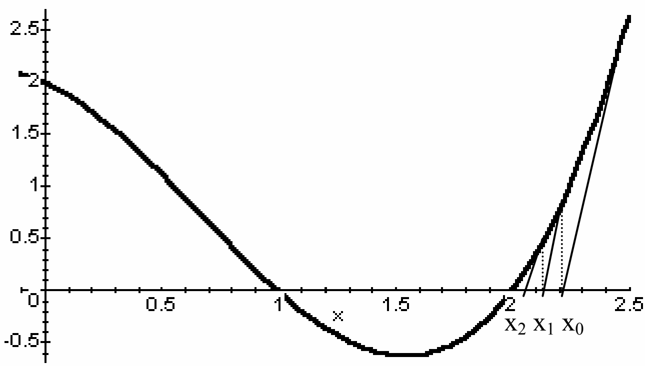
$$\begin{cases} y - f(x_{n-1}) = f'(x_{n-1})(x - x_{n-1}) \\ y = 0 \end{cases}$$

În consecință

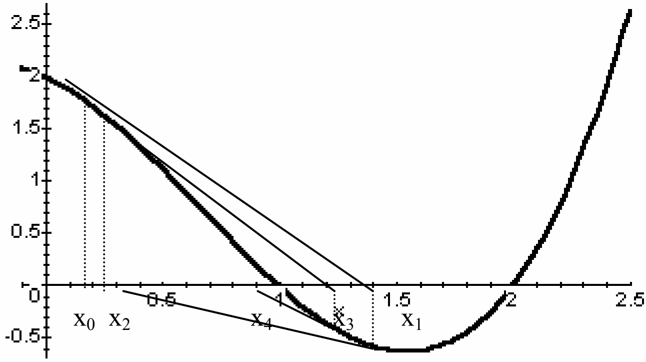
$$x_n = x_{n-1} - \frac{f(x_{n-1})}{f'(x_{n-1})}.$$

Convergența șirului este determinată de termenul inițial x_0 așa cum rezultă din următoarele două exemple:

Exemplul 1:



Exemplul 2.



Următoarea teoremă stabileşte condiţii suficiente pentru convergenţa metodei tangentei.

Teoremă (Metoda tangentei). *Fie $f : [a, b] \to R$ o aplicaţie de două ori derivabilă cu $f'(x) \neq 0$, $f''(x) \neq 0$ oricare ar fi $x \in [a, b]$ şi $f(a)f(b)<0$. Atunci ecuaţia $f(x) = 0$ are o unică soluţie x^*. x^* poate fi obţinută ca limită a şirului $(x_n)_n$ definit prin:*

$$x_n = x_{n-1} - \frac{f(x_{n-1})}{f'(x_{n-1})},\ n \geq 1$$

unde $x_0 \in [a, b]$ este ales astfel încât $f(x_0)f''(x_0) > 0$. În plus, oricare ar fi $n > 1$ au loc următoarele inegalităţi:

$$|x^* - x_n| \leq \frac{|f(x_n)|}{m_1}$$

$$|x^* - x_n| \leq \frac{M_2}{2m_1}(x_n - x_{n-1})^2$$

unde $m_1 = \inf_{x \in [a,b]} |f'(x)|$ *şi* $M_2 = \sup_{x \in [a,b]} |f''(x)|$.

Semnificaţie geometrică. Deoarece f' şi f'' nu se anulează pe [a, b], rezultă că sunt fie strict pozitive fie strict negative.

Cazul 1. f'' > 0 (f strict convexă)

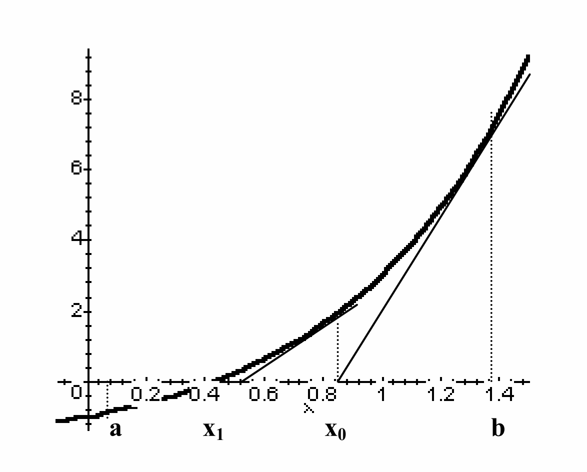
1.1 f' > 0 (f strict crescătoare)

1.2 f' < 0 (f strict descrescătoare)

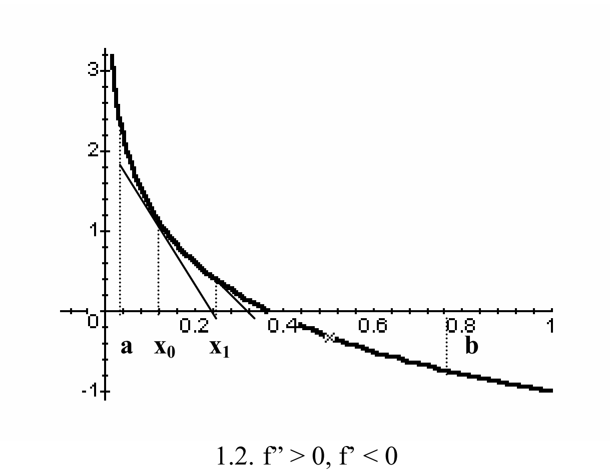
Cazul 2. f’ < 0 (f strict concavă)

2.1. f’ > 0 (f strict crescătoare)

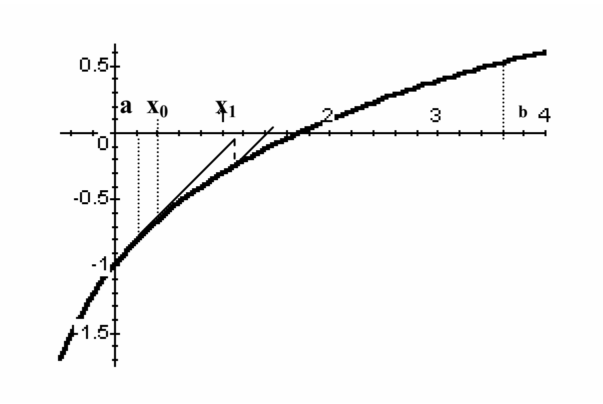
2.2. f’ < 0 (f strict descrescătoare)



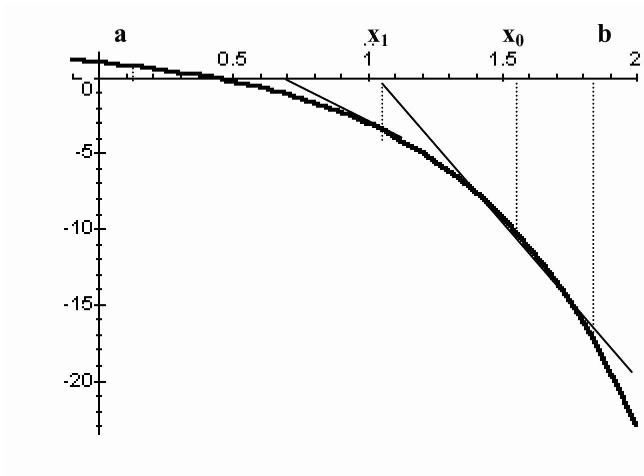
1.1. f’ > 0, f’ > 0



1.2. f’ > 0, f’ < 0



2.1. f’ < 0, f’ > 0



2.2. f’ < 0, f’ < 0

Deci pentru aplicarea metodei tangentei în rezolvarea ecuației f(x) = 0 trebuie stabilite intervalele de monotonie şi intervalele de convexitate/concavitate pentru funcția f. Dacă a şi b sunt capetele unui astfel de interval şi dacă f(a)f(b)<0, atunci se alege în intervalul [a, b] un punct x_0 astfel încât $f(x_0)f''(x_0)>0$. Şirul construit rin metoda tangentei, având termenul inițial x_0 converge la unica rădăcină a ecuației f(x) = 0, situată în intervalul [a, b].

Algoritm

Date de intrare:

f - în condițiile 1.1,1.2,2.1 sau 2.2

x_{00} - $f(x_{00})f''(x_{00})>0$

eps – eroarea (determină condiția de oprire a iterațiilor)

Date de ieşire: x_n care verifică $| x_n - x_{n-1} |^2 <$ eps.

(x_n este considerat o aproximație satisfăcătoare a unicei soluții a ecuației f(x)=0)

x0 := x00;

$x1 := x0 - \dfrac{f(x0)}{f'(x0)}$

cât timp $| x_1 - x_0 |^2 \geq$ eps execută

  x0 := x1;

  $x1 := x0 - \dfrac{f(x0)}{f'(x0)}$;

Prezentăm în continuare o variantă a acestui algoritm pentru cazul în care f nu verifică neapărat condițiile suficiente de convergență. Introducem ca dată suplimentară de intrare numărul maxim de termeni din şir ce urmează a fi calculați (Nmax). Condiția de oprire se transformă

$$| x_n - x_{n-1} |^2 < \text{eps sau } n > \text{Nmax}$$

x0 := x00;

$x1 := x0 - \dfrac{f(x0)}{f'(x0)}$

n : = 1;

cât timp ($| x_1 - x_0 |^2 \geq$ eps) şi (n ≤ Nmax) execută

  x0 := x1;

  $x1 := x0 - \dfrac{f(x0)}{f'(x0)}$;

  n : = n + 1;

Trebuie verificat la ieşirea din ciclu dacă f(x1) ≅ 0. Pentru a urmării convergența se poate afişa la fiecare pas diferența dintre termenii consecutivi curenți.

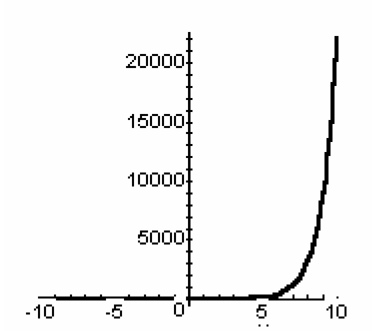
Proceduri MAPLE

```
>   mtangenta := proc(f, x00, eps)
      local x0, x1;
          x0 := x00;
          x1 := evalf(x0 - f(x0)/D(f)(x0));
          print(x1, abs(x1 - x0));
          while eps <= abs(x1 - x0) do
              x0 := x1;
              x1 := evalf(x0 - f(x0)/D(f)(x0));
              print(x1, abs(x1 - x0))
          od;
          RETURN(x1)
      end;
```

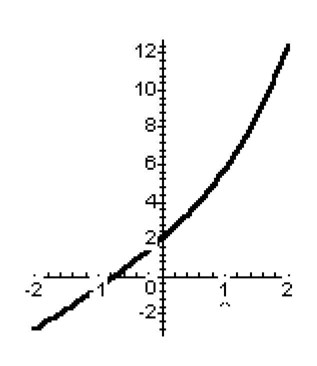
```
>   mtangentaN := proc(f, x00, eps, Nmax)
      local x0, x1, n;
          x0 := x00;
          x1 := x0 - f(x0)/D(f)(x0);
          n := 1;
          print(x1, abs(x1 - x0));
          while eps <= abs(x1 - x0) and n < Nmax do
              x0 := x1;
              x1 := x0 - f(x0)/D(f)(x0);
              print(x1, abs(x1 - x0));
              n := n + 1
          od;
          print(`Numar de termeni calculati`, n);
          RETURN(x1)
      end;
```

Exemple de utilizare a procedurilor

```
> with(plots);
> plot(exp(x)+2*x+1,x);
```




```
> plot(exp(x)+2*x+1,x=-2..2);
```




```
> f1:=(x->exp(x)+2*x+1);
```

$$f1 := x \to e^x + 2\ x + 1$$

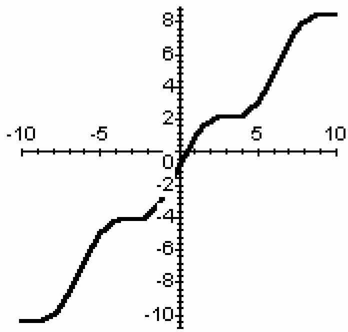
```
> mtangenta(f1,0.1,0.00001);
```

$$-.6423652285,\ .7423652285$$
$$-.7378964278,\ .0955311993$$
$$-.7388349460,\ .0009385182$$
$$-.7388350311,\ .851\ 10^{-7}$$
$$-.7388350311$$

```
> fsolve(f1(x),x);
```

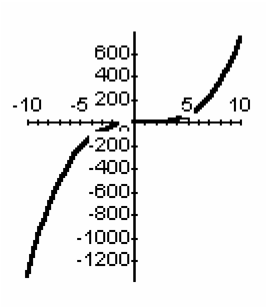
$$-.7388350311$$

```
> plot(sin(x)+x-1,x,color=black);
```

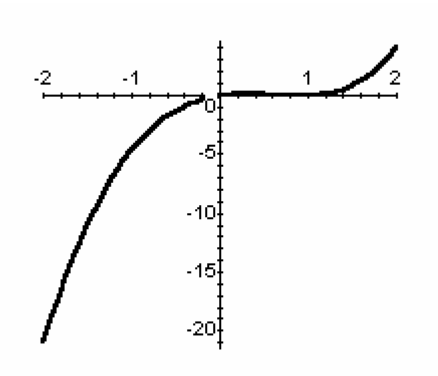



```
> f2:=(x->sin(x)+x-1);

                    f2 := x -> sin(x) + x - 1

> mtangenta(f2,1.1,0.000001);

              .4180998864, .6819001136
              .5099954153, .0918955289
              .5109733047, .0009778894
               .5109734294, .1247 10^(-6)
                     .5109734294

> fsolve(f2(x),x);
                     .5109734294
> mtangentaN(f2,-10.1,0.000001,10);

              37.63660155, 47.73660155
              19.33165959, 18.30494196
              9.366076811, 9.965582779
             -4881.865455, 4891.231532
             -2422.891650, 2458.973805
              7244.949448, 9667.841098
              3444.353274, 3800.596174
               970.454133, 2473.899141
             -21032.51790, 22002.97203
              195025.8435, 216058.3614
          Numar de termeni calculati, 10
                     195025.8435
```

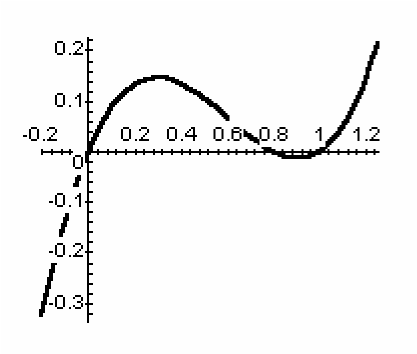
```
> f2(19025.8453);
                              19025.19775

> plot(x^3-3*x^2+3*x-1,x);
```




```
> plot(x^3-3*x^2+3^x-1,x=-2..2);
```




```
> plot(x^3-3*x^2+3^x-1,x=-0.2..1.25);
```




```
> f3:=(x->x^3-3*x^2+3^x-1);
```

$$f3 := x \rightarrow x^3 - 3\ x^2 + 3 - 1$$

```
> mtangenta(f3,-0.2,0.00001);
```

-.0522829463, .1477170537

-.00507552191, .04720742439

-.000055257405, .005020264505

$-.664299\ 10^{-8}$, .00005525076201

$.1756159\ 10^{-11}$, $.6644746159\ 10^{-8}$

$.1756159\ 10^{-11}$

```
> mtangenta(f3,0.6,0.00001);
```

.7746216808, .1746216808

.7989988007, .0243771199

.8009484461, .0019496454

.8009623627, .0000139166

.8009623627, 0

.8009623627

```
> mtangenta(f3,1.2,0.00001);
```

1.081545448, .118454552

1.022198510, .059346938

1.002452318, .019746192

1.000035892, .002416426

1.000000003, .000035889

.9999999996, $.34\ 10^{-8}$

.9999999996

```
> fsolve(f3(x),x);
                         1.000000000
```

III.2. Metoda Newton – cazul m-dimensional

Metoda Newton este o generalizare a metodei tangentei prezentată în secțiunea precedentă. Este o metodă iterativă de rezolvare a unor ecuații de forma f(x) = 0, unde $f : G \to \mathbf{R}^m$, $G \subset \mathbf{R}^m$. Metoda Newton este o metodă frecvent folosită deoarece este foarte rapid convergentă. Convenim să notăm cu $x^1, x^2,\ldots, x^n,\ldots$ un şir de elemente din $\mathbf{R}^m$. Rezervăm indicii inferiori pentru a desemna componentele unui element $x = (x_1, x_2,\ldots,x_m)$ din $\mathbf{R}^m$. Dacă $f : G \to \mathbf{R}^m$ este o funcție diferențiabilă pe G, vom identifica diferențiala de ordinul I a lui f în x, f'(x), cu matricea

$$\left(\frac{\partial f_i}{\partial x_j}(x)\right)_{1\le i,j\le m}$$

numită jacobianul lui f în x.

Metoda Newton constă în aproximarea soluției ecuației considerate cu x^n, unde

$$x^{n+1} = x^n - \left(f'\left(x^n\right)\right)^{-1} f\left(x^n\right) \qquad (*)$$

iar aproximația inițială $x^0 \in G$ este suficient de apropiată de soluția ecuației.

Observații. 1) Amplificând relația (*) cu $f'(x^n)$ rezultă

$$f'\left(x^n\right)\left(x^{n+1} - x^n\right) = -f\left(x^n\right) \qquad (**)$$

sau echivalent

$$\sum_{j=1}^{m} \frac{\partial f_i\left(x^n\right)}{\partial x_j}\left(x_j^{n+1} - x_j^n\right) = -f_i\left(x^n\right), \quad i = 1,2,\ldots, m$$

Dacă se foloseşte relația (*) pentru determinarea lui x^{n+1} este necesar să se calculeze inversa matricei $f'(x^n)$. Dacă se foloseşte relația (**), este necesar să se rezolve un sistem liniar cu m ecuații, şi necunoscutele

$\Delta x_k^n = x_k^{n+1} - x_k^n$, k = 1,…,m. În ambele cazuri, se înlocuieşte rezolvarea sistemului neliniar prin succesivă a unor sisteme liniare.

2) Una din dificultățile metodei este necesitatea determinării derivatelor parțiale $\frac{\partial f_i}{\partial x_j}(x)$, componentele matricei f'(x) (dacă se utilizează MAPLE aceasta nu e o dificultate majoră). O posibilitate de eliminare a acestei dificultăți este aproximarea derivatelor parțiale prin diferențe finite

$$\frac{\partial f_i}{\partial x_j}(x) \approx \frac{1}{h_{ij}}\left(f_i\left(x + h_{ij}e^j\right) - f_i(x)\right) = \Delta_{ij}(x)$$

unde h_{ij} sunt parametri specifici discretizării considerate, iar

$$e^j = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \begin{array}{l} \\ \\ \\ \longleftarrow \ j \\ \\ \\ \\ \end{array}$$

sunt vectorii bazei canonice. În acest fel f'(x) se înlocuieşte prin

$$J(x) = (\Delta_{ij}(x))_{1 \le i,j \le m}$$

Metoda obținută în acest caz se numeşte metoda iterativă discretă a lui Newton:

$$x^{n+1} = x^n - \left(J\left(x^n\right)\right)^{-1} f\left(x^n\right).$$

Ca şi în cazul metodei tangentei convergența metodei depinde de alegerea aproximației inițiale. Aproximația inițială tebuie luată cât mai aproape de soluția problemei, eventual utiliyând o altă metodă de găsire a soluției. În următoarea teoremă se presupune că s-a fixat o normă pe $\mathbf{R}^m$, notată $\| \ \|$, iar pe spațiile de operatori liniari L($\mathbf{R}^m$, $\mathbf{R}^m$), L($\mathbf{R}^m$, L($\mathbf{R}^m$, $\mathbf{R}^m$)) se consideră normele operatoriale induse. Pentru x $\in$ $\mathbf{R}^m$ şi r > 0, se notează B(x,r) mulțimea:

$$\left\{y \in R^m,\ \|y - x\| < r\right\}$$

şi cu $\overline{B\left(x^0, r\right)}$ închiderea acestei mulțimi, adică

$$\left\{y \in R^m,\ \|y - x\| \le r\right\}$$

Teoremă (Metoda Newton). *Fie $G \subset \mathbf{R}^m$ o mulțime deschisă, $f : G \to \mathbf{R}^m$ o funcție de clasă C^2, cu proprietatea că există $M > 0$ astfel ca $\|f''(x)\| \leq M$ pentru orice $x \in G$. Presupunem că ecuația $f(x) = 0$ admite o soluție z astfel încât $f'(z)$ să fie inversabil. Atunci pentru orice $q \in (0, 1)$ există r, $\mu > 0$ astfel încât $f'(x)$ este inversabil pentru orice $x \in B(z,r)$, şirul $(x^n)_n$, definit prin*

$$x^{n+1} = x^n - \left(f'\left(x^n\right)\right)^{-1} f\left(x^n\right)$$

rămâne în $B(z,r)$ oricare ar fi $x^0 \in B(z,r)$ şi converge la z. În plus, au loc următoarele relații

$$\left\|x^n - z\right\| \leq \frac{2\mu}{M} q^{2^n}$$

$$\left\|x^n - z\right\| \leq \frac{1}{\mu}\left\|f\left(x^n\right)\right\| \leq \frac{M}{2\mu}\left\|x^n - x^{n-1}\right\|^2.$$

Printre dezavantajele acestei metode se află necesitatea calculării la fiecare pas a inversei unei matrice, $f'(x^n)$, sau eventual a rezolvării unui sistem de ecuații liniare (aşa cum remarcam mai înainte). Un alt dezavantaj este localizarea teoretică a procesului iterativ într-o vecinătate a soluției căutate. Metoda Newton simplificată înlătură primul inconvenient. Această variantă a metodei constă în aproximarea soluției cu cu x^n, unde x^0, $c \in G$, şi

$$x^{n+1} = x^n - \left(f'(c)\right)^{-1} f\left(x^n\right) \qquad (***)$$

Următoarea teoremă stabileşte condiții suficiente de convergență a acestei metode.

Teoremă (Metoda Newton simplificată). *Fie $G \subset \mathbf{R}^m$ o mulțime deschisă, $f : G \to \mathbf{R}^m$ o funcție de clasă C^2, cu proprietatea că există $M > 0$ astfel ca $\|f''(x)\| \leq M$ pentru orice $x \in G$. Presupunem că ecuația $f(x) = 0$ admite o soluție z astfel încât $f'(z)$ să fie inversabil. Atunci pentru orice $q \in (0, 1)$ există r, μ, $L > 0$ astfel încât $f'(x)$ este inversabil pentru orice $x \in B(z,r)$, şirul $(x^n)_n$, definit prin*

$$x^{n+1} = x^n - \left(f'(c)\right)^{-1} f\left(x^n\right)$$

rămâne în $B(z,r)$ oricare ar fi $x^0 \in B(z,r)$ şi converge la z. În plus, au loc următoarele relații

$$\left\|x^n - z\right\| \leq q^n \left\|x^0 - z\right\|$$

$$\left\|x^n - z\right\| \leq \frac{1}{\mu}\left\|f\left(x^n\right)\right\| \leq \frac{L}{\mu}\left\|x^{n+1} - x^n\right\|^2.$$

Metoda Newton – Kantorovici nu localizează procesul iterativ într-o vecinătate a soluției problemei. Metoda Newton – Kantorovici constă în aproximarea rădăcinii ecuației considerate cu cu x^n, unde

$$x^{n+1} = x^n - \left(f'\left(x^n\right)\right)^{-1} f\left(x^n\right) \qquad (****)$$

iar aproximația inițială $x^0 \in G$ satisface condițiile din teorema următoare.

Teoremă (Metoda Newton-Kantorovici). *Fie $G \subset \boldsymbol{R}^m$ o mulțime deschisă, $f : G \to \boldsymbol{R}^m$ o funcție de clasă C^2, cu proprietatea că există $M > 0$ astfel ca $\left\|f''(x)\right\| \leq M$ pentru orice $x \in G$. Presupunem că există $x^0 \in G$ şi există a, b >0 astfel încât:*

$f'(x^0)$ este inversabilă şi $\left\|\left(f'\left(x^0\right)\right)^{-1}\right\| \leq a$

$\left\|\left(f'\left(x^0\right)\right)^{-1} f\left(x^0\right)\right\| \leq b$

$abM < \dfrac{1}{2}$

$\overline{B\left(x^0, r\right)} \subset G$

unde $r = \dfrac{1}{aM}\sqrt{1 - 2abM}$.

Atunci ecuația f(x) = 0 are o unică soluție z în $\overline{B\left(x^0, r\right)}$, *şirul $(x^n)_n$, este corect definit prin*

$$x^{n+1} = x^n - \left(f'\left(x^n\right)\right)^{-1} f\left(x^n\right), \; n \geq 0,$$

rămâne în $B(x^0, r)$ şi converge la z. În plus, are loc următoarea relație

$\left\|x^n - z\right\| \leq \dfrac{b}{2^{n-1}}$, $n \geq 1$.

Proceduri MAPLE

Parametrii procedurii `mnewton` (de mai jos) sunt

m = dimensiunea spațiului pe care se lucrează, i.e numărul de necunoscute (respectiv numărul de ecuații)

f = vectorul ce conține componenetele funcției f (se presupune că se rezolvă sistemul f(x) = 0)

v = vectorul necunoscutelor

x00 = termenul inițial din şirul definit de (**)

eps = eroarea

Nmax = numărul maxim de termeni din şir ce vor fi calulați

Se calculează n termeni, cu n verificând

$$(\left\|x_n - x_{n-1}\right\|_\infty < \text{eps}) \text{ sau } (n \geq \text{Nmax}).$$

În procedura mnewton apar câteva comenzi pe care nu le-am folosit în lucrările precedente. Comanda

```
>subs(expr1,expr2);
```

substituie subexpresia expr1 în expresia expr2. Comanda

```
>jacobian(f,v);
```

calculează jacobianul lui f. Este o comnadă ce apaţine pachetului linalg. Comanda

```
>norm(a,n);
```

calculează norma n (n =1,2,infinity) a vectorului (sau matricei) a. Este de asemenea o comanda ce aparţine pachetului linalg.

```
> mnewton := proc(m, f, v, x00, eps, Nmax)
   local x1, x0, dx, b, fx, fx1, n, i, j, ex, r;
     x0 := x00;
     x1 := vector(m);
     dx := vector(m);
     b := vector(m);
     fx := jacobian(f, v);
     fx1 := matrix(m, m);
     ex := seq(v[i] = x0[i], i = 1 .. m);
     for i to m do
         for j to m do fx1[i, j] := evalf(subs(ex,
 fx[i, j])) od
     od;
     for i to m do b[i] := evalf(-subs(ex, f[i]))
 od;
     dx := linsolve(fx1, b, 'r');
     if r <> m then print(`Metoda nu se aplica`);
 RETURN(NULL) fi;
     for i to m do x1[i] := x0[i] + dx[i] od;
     n := 1;
     print(x1, norm(dx, infinity)^2);
     while eps <= norm(dx, infinity)^2 and n < Nmax
 do
         x0 := x1;
         ex := seq(v[i] = x0[i], i = 1 .. m);
         for i to m do
             for j to m do fx1[i, j] :=
 evalf(subs(ex, fx[i, j]))
             od
         od;
         for i to m do b[i] := evalf(-subs(ex,
 f[i])) od;
         dx := linsolve(fx1, b, 'r');
```

```
        if   r   =   0   then   print(`Metoda   nu   se
aplica`); RETURN fi;
        for i to m do x1[i] := x0[i] + dx[i] od;
        n := n + 1;
        print(x1, norm(dx, infinity)^2)
    od;
    print(`Numar de pasi`, n);
    ex := seq(v[i] = x1[i], i = 1 .. m);
    for  i  to  m  do  b[i]  :=  evalf(subs(ex,  f[i]))
od;
    print(`Valoarea functiei`, b);
    print(`norma   valorii   functiei`,   norm(b,
infinity));
    RETURN(evalm(x1))
end;
```

Procedura `mnewtonsimplif` de mai jos poate fi folosită pentru rezolvarea unui sistem prin metoda Newton simplificată.

Parametrii procedurii `mnewtonsimplif` sunt

m = dimensiunea spaţiului pe care se lucrează, i.e numărul de necunoscute (respectiv numărul de ecuaţii)

f = vectorul ce conţine componenetele funcţiei f (se presupune că se rezolvă sistemul f(x) = 0)

v = vectorul necunoscutelor

x00 = termenul iniţial din şirul definit de (***)

c = punctul în care se evaluează inversa matricei jacobiene

eps = eroarea

Nmax = numărul maxim de termeni din şir ce vor fi calulaţi

Se calculează n termeni, cu n verificând

$$(\|x_n - x_{n+1}\|_\infty < \text{eps}) \text{ sau } (n \geq \text{Nmax}).$$

```
>mnewtonsimplif := proc(m, f, v, x00, c, eps, Nmax)
  local x1, x0, b, dx, fx, fx1, n, i, j, ex, r;
      x0 := x00;
      x1 := vector(m);
      b := vector(m);
      dx := vector(m);
      fx := jacobian(f, v);
      fx1 := matrix(m, m);
      ex := seq(v[i] = c[i], i = 1 .. m);
      for i to m do
```

```
          for   j   to   m   do   fx1[i,   j]   :=
   evalf(subs(ex, fx[i, j])) od
       od;
       if det(fx1) = 0 then
           print(`Metoda    nu    se    aplica`);
   RETURN(NULL)
       fi;
       fx1 := inverse(fx1);
       ex := seq(v[i] = x0[i], i = 1 .. m);
       for i to m do b[i] := evalf(subs(ex, f[i]))
   od;
       for i to m do
           dx[i] := 0;
           for j to m do dx[i] := dx[i] + fx1[i,
   j]*b[j] od
       od;
       n := 1;
       print(x0, norm(dx, infinity)^2);
       for i to m do x1[i] := x0[i] - dx[i] od;
       while eps <= norm(dx, infinity)^2 and n <
   Nmax do
           x0 := x1;
           ex := seq(v[i] = x0[i], i = 1 .. m);
           for  i  to  m  do  b[i]  :=  evalf(subs(ex,
   f[i])) od;
           for i to m do
               dx[i] := 0;
               for  j  to  m  do  dx[i]  :=  dx[i]  +
   fx1[i, j]*b[j] od
           od;
           for i to m do x1[i] := x0[i] - dx[i] od;
           n := n + 1;
           print(x0, norm(dx, infinity)^2)
       od;
       print(`Numar de pasi`, n);
       ex := seq(v[i] = x0[i], i = 1 .. m);
       for i to m do b[i] := evalf(subs(ex, f[i]))
   od;
       print(`Valoarea functiei`, b);
       print(`norma   valorii   functiei`,   norm(b,
   infinity));
       RETURN(evalm(x0))
end;
```

Exemple de utilizare a procedurii mnewton

```
>with(linalg);
> f:=vector(2,[x^2-y,x^3-5*y]);

                    f := [x^2  - y, x^3  - 5 y]

> mnewton(2,f,[x,y],[0.1,0.1],0.0001,9);

     [.04948453608, -.0001030928], .01002062919
  [.02461798707, -.00001229997147], .0006183452597
  [.01227846581, -.150306221 10^(-5) ], .0001522637849
 [.006131667004, -.185795422 10^(-6) ], .00003778313556
                   Numar de pasi, 4
Valoarea functiei,[.00003778313567,.1159511481 10^(-5)]
      norma valorii functiei, .00003778313567
        [.006131667004, -.185795422 10^(-6) ]

> f1:=vector(2,[x^2+y^2-1,x^3-y]);

                 f1 := [x^2  + y^2  - 1, x^3  - y]

> mnewton(2,f1,[x,y],[0.9,0.5],0.0000001,10);

      [.8316784870, .5629787234], .004667829140
     [.8260617824, .5636079088], .00003154737019
     [.8260313586, .5636241618], .9256069907 10^(-9)
                   Numar de pasi, 3
      Valoarea functiei, [.12 10^(-8)  , .23 10^(-8) ]
          norma valorii functiei, .23 10^(-8)
             [.8260313586, .5636241618]


> Digits:=25;

                      Digits := 25

>mnewton(2,f1,[x,y],[.8260313586, .5636241618],10^(-
24),10);

[.8260313576541869568589847,
.5636241621612585481734990],
```

```
                        .89456231257566806093490002 10^-18
[.82603135765418695596898 70,
.56362416216125854856849 80],
                        .79209591150628277874674 14 10^-36

                    Numar de pasi, 2
               Valoarea functiei, [0, 0]
               norma valorii functiei, 0
              [.82603135765418695596898 70,
              .56362416216125854856849 80]

> fsolve({f1[1],f1[2]},{x,y});
  {y = -.56362416216125854856849 80,
   x = -.82603135765418695596898 70}

> Digits:=10;
                              Digits := 10

> mnewton(2,f1,[x,y],[1,1],0.0000001,10);

       [.8750000000, .6250000000], .1406250000
      [.8290363483, .5643491124], .003678530164
     [.8260401082, .5636197735], .8977454941 10^-5
    [.8260313577, .5636241621], .7657110305 10^-10


                    Numar de pasi, 4
            Valoarea functiei, [0, .2 10^-9]
            norma valorii functiei, .2 10^-9
               [.8260313577, .5636241621]

> mnewton(2,f1,[x,y],[-1,1],0.0000001,10);

       [-.2500000000, 1.250000000], .5625000000
      [-81.50000000, -15.25000000], 6601.562500
       [-54.33007552, 64.91769867], 6426.859910
       [-36.21723951, 24.89070048], 1602.160584
       [-24.14389018, 3.68381158], 449.7321368
      [-16.10992940, -24.48752160], 793.6240131
      [-10.75342462, -10.48890788], 195.9611861
      [-7.178878909, -3.444508422], 49.62356372
       [-4.781922714, .617903494], 16.50319058
      [-3.270918985, -5.691806302], 39.81243771
```

```
Numar de pasi, 10
Valoarea functiei, [42.09556999, -29.30346483]
norma valorii functiei, 42.09556999
[-3.270918985, -5.691806302]

> mnewton(2,f1,[x,y],[1,-1],0.0000001,10);

[.2500000001, -1.250000000], .5624999999
[81.50000065, 15.25000014], 6601.562606
[54.33007595, -64.91769908], 6426.859998
[36.21723981, -24.89070064], 1602.160604
[24.14389039, -3.68381157], 449.7321440
[16.10992955, 24.48752227], 793.6240503
[10.75342471, 10.48890823], 195.9611950
[7.178878965, 3.444508634], 49.62356567
[4.781922754, -.617903286], 16.50319061
[3.270919047, 5.691808811], 39.81246675
Numar de pasi, 10
Valoarea functiei, [42.09559895, 29.30346431]
norma valorii functiei, 42.09559895
[3.270919047, 5.691808811]

> mnewton(2,f1,[x,y],[-1,-1],0.0000001,10);
```

$[-.8750000000, -.6250000000], .1406250000$
$[-.8290363483, -.5643491124], .003678530162$
$[-.8260401082, -.5636197735], .8977454941\ 10^{-5}$
$[-.8260313577, -.5636241621], .7657110310\ 10^{-10}$
Numar de pasi, 4
Valoarea functiei, $[0, -.2\ 10^{-9}]$
norma valorii functiei, $.2\ 10^{-9}$
$[-.8260313577, -.5636241621]$

```
>f2:=vector(3,[x+x^2-2*y*z-0.1,y-y^2+3*x*z+0.2,
z+z^2+2*x*y-0.3]);
```

$$f2 := [x + x^2 - 2\ y\ z - .1,\ y - y^2 + 3\ x\ z + .2,\ z + z^2 + 2\ x\ y - .3]$$

```
> mnewton(3,f2,[x,y,z],[0,0,0],0.000001,10);

[.1, -.2, .3], .09
```

[.02245322248, -.1743243243, .2461538462],
.006013502704

[.01287849238, -.1778109522, .2447473526],
.00009167545653

[.01282415093, -.1778006638, .2446880471],
$.3517140658\ 10^{-8}$

Numar de pasi, 4
Valoarea functiei, [$.418\ 10^{-8}$, $.9440\ 10^{-8}$, $.24\ 10^{-8}$]
norma valorii functiei, $.9440\ 10^{-8}$

[.01282415093, -.1778006638, .2446880471]

```
> fsolve({f2[1],f2[2],f2[3]},{x,y,z});
>f3:=vector(2,[x+3*log[10](x)-y^2,2*x^2-x*y-5*x+1]);
```

$$f3 := [x + 3\frac{\ln(x)}{\ln(10)} - y , 2 x - x y - 5 x + 1]$$

```
> mnewton(2,f3,[x,y],[3.4,2.2],0.000001,10);
```

[3.489912371, 2.263364462], .008084234387
[3.487444647, 2.261629933], $.6089659287\ 10^{-5}$
[3.487442787, 2.261628630], $.3459910861\ 10^{-11}$
Numar de pasi, 3
Valoarea functiei, [$.2\ 10^{-8}$, $-.13\ 10^{-7}$]
norma valorii functiei, $.13\ 10^{-7}$
[3.487442787, 2.261628630]

```
> f4:=vector(2,[sin(x)-y-1.32, cos(y)-x+0.5]);
```

f4 := [sin(x) - y - 1.32, cos(y) - x + .5]

```
> mnewton(2,f4,[x,y],[0,0],0.000001,10);
```

[1.5, .18], 2.25
[1.572883945, -.3173494071], .2473564327
[1.449318570, -.3197442221], .01526840192
[1.446824072, -.3276716350], .00006284387503
[1.446791999, -.3276786889], $.1028698590\ 10^{-8}$

```
Numar de pasi, 5
Valoarea functiei, [-.5 10^-9 , -.3 10^-9 ]
norma valorii functiei, .5 10^-9
[1.446791999, -.3276786889]

> mnewton(2,f4,[x,y],[10,10],0.000001,10);

[-1.53015112, 7.810600421], 132.9443849
[10.18790803, -1.843022290], 137.3129102
[4.221883353, 2.300316544], 35.59345045
[2.631494884, -1.452907372], 14.08668976
[1.883080256, -.178598396], 1.623863366
[1.472775858, -.2423066352], .1683496987
[1.450469607, -.3269831308], .007170108909
[1.446794479, -.3276716792], .00001350656460
[1.446791999, -.3276786895], .4914427445 10^-10
Numar de pasi, 9
Valoarea functiei, [.1 10^-9 , -.5 10^-9 ]
norma valorii functiei, .5 10^-9
[1.446791999, -.3276786895]

> fsolve({f4[1],f4[2]},{x,y});

{x = 1.446791998, y = -.3276786895}
```

Exemple de utilizarea a procedurii mnewtonsimplif

```
>with(linalg);
> f1:=vector(2,[x^2+y^2-1,x^3-y]);

f1 := [x^2 + y^2 - 1, x^3 - y]

>mnewtonsimplif(2,f1,[x,y],[0.9,0.5],[0.9,0.5],0.000
0001,10);

[.9, .5], .004667829139
[.8267332449, .5632460105], .00002445541970
[.8261303960, .5635972120], .3634268114 10^-6
[.8260445915, .5636184114], .7362406744 10^-8
Numar de pasi, 4
Valoarea functiei, [.0000153808, .0000328407]
norma valorii functiei, .0000328407
[.8260445915, .5636184114]
```

```
> Digits:=25;

                    Digits := 25


>mnewtonsimplif(2,f1,[x,y],[.8260445915,
.5636184114],[.8260445915, .5636184114],10^(-
24),10);

             [.8260445915, .5636184114],
                       .1751300130918763163665720 10^(-9)


            [.8260313576541916355603878,
            .5636241621612565751734863],
                     .3102263117306504342412713 10^(-19)


            [.8260313576541869560933791,
            .5636241621612585485162004],
                     .2189741148584550825078084 10^(-28)


                   Numar de pasi, 3
  Valoarea functiei, [.1465512 10^(-18), .3069261 10^(-18)]
        norma valorii functiei, .3069261 10^(-18)

            [.8260313576541869560933791,
             .5636241621612585485162004]


> fsolve({f1[1],f1[2]},{x,y});

          {x = -.8260313576541869559689870,
           y = -.5636241621612585485684980}

> Digits:=10;

                              Digits := 10

>mnewtonsimplif(2,f1,[x,y],[1,1],[1,1],0.0000001,10)
;
```

```
[1, 1], .1406250000
[.8442382813, .5776367188], .002243280411
[.8324169200, .5662568529], .0001397445832
[.8280863872, .5638050096], .00001875351433
[.8266269335, .5634628865], .2130005102 10^-5
[.8261812973, .5635072668], .1985915893 10^-6
[.8260608316, .5635697447], .1451198789 10^-7
Numar de pasi, 7
Valoarea functiei, [-.0000126454, .0001147523]
norma valorii functiei, .0001147523
[.8260608316, .5635697447]
> mnewtonsimplif(2,f1,[x,y],[-1,1],[-1,1],
0.0000001,10);

[-1, 1], .5625000000
[.2265625000, 1.414062500], .2271118164
[.6650531295, 1.327101470], .1922740322
[.8806553122, .9409566642], .1491078110
[.8443975610, .5742223016], .1344940927
[.8197933010, .5282487952], .002113563289
[.8206651612, .5535667324], .0006409979415
[.8261112430, .5690490972], .0002397036183
[.8271727754, .5669722992], .4313089933 10^-5
[.8262589002, .5632222297], .00001406302107
Numar de pasi, 10
Valoarea functiei, [-.0000769498, .0008678366]
norma valorii functiei, .0008678366
[.8262589002, .5632222297]


> mnewtonsimplif(2,f1,[x,y],[1,-1],[1.5,-
1],0.0000001,10);

[1, -1], .08163265311
[.5034707760, -1.058569346], .04444293817
[.3131537097, -1.157019004], .03622058573
[.1285157308, -1.215596861], .03409118325
[-.0563649270, -1.245821837], .03418085763
[-.2407493532, -1.244773949], .03399761663
[-.4173415130, -1.205950971], .03118479090
[-.5733443920, -1.125709448], .02433689826
[-.6951091023, -1.010383737], .01482664467
[-.7755847708, -.8790712597], .01724296669
```

```
                  Numar de pasi, 10
    Valoarea functiei, [.3742980163, .4125324056]
         norma valorii functiei, .4125324056
             [-.7755847708, -.8790712597]

>mnewtonsimplif(2,f1,[x,y],[-1,-1],[-1,-
1],0.0000001,10);

                 [-1, -1], .1406250000
      [-.8442382813, -.5776367188], .002243280411
     [-.8324169200, -.5662568529], .0001397445832
    [-.8280863872, -.5638050096], .00001875351433
     [-.8266269335, -.5634628865], .2130005102 10^-5
     [-.8261812973, -.5635072668], .1985915893 10^-6
     [-.8260608316, -.5635697447], .1451198789 10^-7
                  Numar de pasi, 7
   Valoarea functiei, [-.0000126454, -.0001147523]
         norma valorii functiei, .0001147523

             [-.8260608316, -.5635697447]

> f2:=vector(3,[x+x^2-2*y*z-0.1,y-y^2+3*x*z+0.2,
z+z^2+2*x*y-0.3]);

 f2 := [x + x^2  - 2 y z - .1, y - y^2  + 3 x z + .2, z
                    + z^2  + 2 x y - .3]


>mnewtonsimplif(3,f2,[x,y,z],[0,0,0],[0,0,0],0.00000
1,10);

                [0, 0, 0], .09000000000
      [-.0300000000, -.2500000000, .2500000000],
                                           .01690000000

     [-.02590000000, -.1150000000, .2225000000],
                                           .01822500000


      [.04815419000, -.1694867500, .2445367500],
                                          .005484023057
```

[.01478969595, -.2066006490, .2565247722],
.001377441496

[-.00621510391, -.1686979420, .2403061628],
.001436615202

[.01888306226, -.1670604211, .2401559976],
.0006299179451

[.01940230578, -.1856954577, .2486343215],
.0003472645876

[.00728302227, -.1799894344, .2453868143],
.0001468770328

[.01161288976, -.1729652764, .2424070455],
.00004933879561

Numar de pasi, 10

Valoarea functiei, [-.00439624778, .005562875689, -.0028490322]
norma valorii functiei, .005562875689
[.01161288976, -.1729652764, .2424070455]

```
> fsolve({f2[1],f2[2],f2[3]},{x,y,z});
>f3:=vector(2,[x+3*log[10](x)-y^2,2*x^2-x*y-5*x+1]);
```

$$f3 := [x + 3\frac{\ln(x)}{\ln(10)} - y\ ,\ 2\ x\ \ -\ x\ y - 5\ x + 1]$$

```
>mnewtonsimplif(2,f3,[x,y],[3.4,2.2],[3.4,2.2],0.000
001,10);
```

[3.4, 2.2], .008084234389
[3.487301362, 2.261529394], $.6817366478\ 10^{-5}$

```
[3.487450767, 2.261634225], .2232185277 10^-7
Numar de pasi, 3
Valoarea functiei, [-.000014345, .000033847]
norma valorii functiei, .000033847
[3.487450767, 2.261634225]

> f4:=vector(2,[sin(x)-y-1.32, cos(y)-x+0.5]);

f4 := [sin(x) - y - 1.32, cos(y) - x + .5]

>mnewtonsimplif(2,f4,[x,y],[0,0],[0,0],0.000001,10);
[0, 0], 2.250000000
[1.483843693, -.3386613206], .2690095655
[1.443200253, -.3644214390], .001651889215
[1.434330122, -.3369994719], .0007519642796
[1.443751058, -.3198761353], .0002932086563
[1.449274375, -.3225360850], .00003050702516
[1.448434599, -.3282144863], .00003224424132
[1.446619419, -.3292920409], .3294880248 10^-5
[1.446271516, -.3280479535], .1547753459 10^-5
[1.446673088, -.3273416290], .4988942993 10^-6
Numar de pasi, 10
Valoarea functiei, [-.0003517752, .0002273383]
norma valorii functiei, .0003517752
[1.446673088, -.3273416290]

>mnewtonsimplif(2,f4,[x,y],[10,10],[10,10],0.000001,
10);

[10, 10], 132.9443849
[-3.890162836, -.338955462], 66.41526109
[-.340619473, -3.617728895], 12.59925809
[.3597838733, -2.241759545], 1.893291652
[.5049748365, -1.089753650], 1.327117582
[.9139489099, -1.179373083], .1672597927
[1.134944023, -.7135095030], .2170288752
[1.330171045, -.5772988928], .03811359028
[1.420849814, -.4248968191], .02322639207
[1.449132610, -.3549522197], .004892246988
Numar de pasi, 10
Valoarea functiei, [.0275603144, -.0114695143]
norma valorii functiei, .0275603144
[1.449132610, -.3549522197]
```

```
> fsolve({f4[1],f4[2]},{x,y});
                        {x = 1.446791998, y = -.3276786895}
```

Probleme propuse

Dați exemple de probleme rău condiționate. Se cunoaşte că factorul de condiționare este $\left\| J^{-1}\left(x^{*}\right) \right\|$, unde x* este rădăcina ecuației f(x) = 0, iar J(x*) este jacobianul lui f în x.