Lucrarea de laborator nr. 11

I. Scopul lucrării

Derivarea aproximativă

II. Conținutul lucrării

1. Formule de derivare aproximativă folosind dezvoltări în serie Taylor
2. Metode de derivare numerică folosind interpolarea

III. Prezentarea lucrării

III.1. Formule de derivare aproximativă folosind dezvoltări în serie Taylor

Fie $f : [a, b] \to \mathbf{R}$, derivabilă. Se recurge la aproximarea derivatei funcției f sau a derivatelor de ordin superior atunci când expresiile acestora sunt prea complicate sau când nu se cunoaşte expresia analitică a funcției f (f este dată prin intermediul unui tabel de valori). Presupunem că se dau n+1 puncte distincte în intervalul [a, b], $x_0, x_1, \ldots, x_n$, în care se cunosc valorile funcției f. Prezentăm o tehnică de găsire a unor formule de aproximare pentru valorile derivatelor de orice ordin ale funcției f. Această tehnică are la bază formula lui Taylor. Reamintim această formulă.

Fie I un interval de numere reale, $a \in I$ şi $f : I \to \mathbf{R}$ o funcție de n ori derivabilă în a. ***Polinomul Taylor de ordin n*** ataşat lui f în punctul a este funcția polinomială $T_{a,n} : I \to \mathbf{R}$, definită prin:

$$T_{a,n}(x) = f(a) + \frac{f'(a)}{1!}(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \frac{f^{(n)}(a)}{n!}(x-a)^n.$$

Restul formulei Taylor de ordin n ataşat funcției f în punctul a este funcția $R_{a,n} : I \to \mathbf{R}$ definită prin $R_{a,n}(x) = f(x) - T_{a,n}(x)$. Egalitatea

$$f(x) = T_{a,n}(x) + R_{a,n}(x)$$

valabilă pentru orice $x \in I$ se numeşte ***formulă Taylor de ordin n*** ataşată funcţiei f în punctul a. Se demonstrează că $\lim\limits_{x \to a} \frac{R_{a,x}}{(x-a)^n} = 0$. Dacă f este n+1 ori derivabilă, atunci există u strict cuprins între a şi x (sau echivalent există $\theta \in (0, 1)$ astfel încât $u = a + \theta(x - a)$) cu proprietatea că

$$R_{a,n}(x) = \frac{f^{(n+1)}(u)}{(n+1)!}(x - a)^{n+1}.$$

Ne propunem să găsim valorile derivatelor de ordinul întâi şi doi ale funcţiei f în punctul $x_i \in [a, b]$ când sunt cunoscute valorile funcţiei în punctele x_{i-1}, x_i, x_{i+1} care nu sunt neapărat echidistante, adică:

$$x_{i-1} = x_i - h,\ x_{i+1} = x_i + \alpha h.$$

Presupunem că f este de 3 ori derivabilă şi scriem formula lui Taylor de ordinul 2 ataşată lui f în x_i:

$$f(x) = f(x_i) + \frac{f'(x_i)}{1!}(x - x_i) + \frac{f''(x_i)}{2!}(x - x_i)^2 + \frac{f'''(w_x)}{3!}(x - x_i)^3.$$

cu w_x stric cuprins între x şi x_i. Înlocuim în această formulă $x = x_{i-1}$, respectiv $x = x_{i+1}$ şi obţinem relaţiile:

$$(1)\qquad f(x_{i-1}) = f(x_i) + \frac{f'(x_i)}{1!}(-h) + \frac{f''(x_i)h^2}{2!} + \frac{f'''(u)}{3!}(-h)^3$$

$$(2)\qquad f(x_{i+1}) = f(x_i) + \frac{f'(x_i)}{1!}\alpha h + \frac{f''(x_i)}{2!}h^2\alpha^2 + \frac{f'''(v)}{3!}\alpha^3 h^3$$

cu u strict cuprins între x_{i-1} şi x_i şi v strict cuprins între x_i şi x_{i+1}. Scăzând din a doua relaţie prima relaţie înmulţită cu α^2 obţinem:

$$f'(x_i) = \frac{f(x_{i+1}) + f(x_i)(\alpha^2 - 1) - \alpha^2 f(x_{i-1}) + \frac{f'''(v)}{3!}\alpha^3 h^3 + \frac{f'''(u)}{3!}h^3\alpha^2}{h\alpha(\alpha+1)}$$

Putem aproxima valoarea derivatei funcţiei f în x_i prin:

$$f'(x_i) \approx \frac{f(x_{i+1}) + f(x_i)(\alpha^2 - 1) - \alpha^2 f(x_{i-1})}{h\alpha(\alpha+1)}$$

cu eroarea:

$$\varepsilon = \left(\frac{\alpha^2}{\alpha+1}\frac{f'''(v)}{3!} + \frac{\alpha}{\alpha+1}\frac{f'''(u)}{3!}\right)h^2$$

care tinde la zero odată cu h^2.

Adunând la a doua relaţie prima înmulţită cu α obţinem:

$$f''(x_i) = \frac{2\left(f(x_{i+1}) + (1+\alpha)f(x_i) + \alpha f(x_{i-1}) + \frac{f'''(v)}{3!}\alpha^3 h^3 - \frac{f'''(u)}{3!}\alpha h^3 \right)}{h^2\alpha(\alpha+1)}.$$

Putem astfel aproxima valoarea derivatei de ordinul doi al lui f în x_i prin

$$f''(x_i) \approx \frac{2(f(x_{i+1}) - (1+\alpha)f(x_i) + \alpha f(x_{i-1}))}{h^2\alpha(\alpha+1)}$$

cu eroarea

$$\varepsilon = \frac{h}{(\alpha+1)}\left(\frac{f'''(v)}{3}\alpha^2 - \frac{f'''(u)}{3} \right)$$

care tinde la zero odată cu h.

Dacă nodurile sunt echidistante (adică dacă α = 1) se obțin următoarele formule de aproximare:

$$f'(x_i) \approx \frac{f(x_{i+1}) - f(x_{i-1})}{2h}$$

$$f''(x_i) \approx \frac{f(x_{i+1}) - 2f(x_i) + f(x_{i-1})}{h^2}.$$

Algoritm pentru determinarea valorilor aproximative ale derivatelor de ordinul 1 şi 2

Date de intrare:

n – numărul de puncte distincte din [a, b] este n +1

xy – tablou ce conține pe prima linie cele n+1 puncte distincte din intervalul [a, b], iar pe a doua linie valorile corespunzătoare ale funcției.

x_0	x_1	…	x_n
y_0	y_1	…	y_n

$f(x_i) = y_i$, i = 0,1,…,n

.

Date de ieşire:

f12 – tablou ce conține pe prima linie valorile aproximative ale lui f' şi pe a doua linie valorile aproximative ale lui f".

≈f'(x_1)	≈f'(x_2)	…	≈f'(x_n)
≈f"(x_1)	≈f"(x_1)	…	≈f"(x_n)

pentru i = 1,n-1,1 executa

$$h := x_i - x_{i-1};\ a := (x_{i+1} - x_i)/h;$$

$$f12[1,i] := \frac{y_{i+1} + y_i\left(a^2 - 1\right) - a^2 y_{i-1}}{ha(a+1)}$$

$$f12[2,i] := \frac{2\left(y_{i+1} - (1+\alpha)y_i + ay_{i-1}\right)}{h^2 a(a+1)}$$

Procedură MAPLE pentru determinarea valorilor aproximative ale derivatelor de ordinul 1 şi 2

```
>derivare12 := proc(xy, n)
local f12, i, a, h;
    f12 := array(1 .. 2, 1 .. n - 1);
    for i to n - 1 do
        h := xy[1, i] - xy[1, i - 1];
        a := (xy[1, i + 1] - xy[1, i])/h;
        f12[1, i] := (xy[2, i + 1] +
                      (a^2 - 1)*xy[2, i]-
                       a^2*xy[2, i - 1])
                       /(h*a*(a + 1));
        f12[2, i] := 2*(xy[2, i + 1] -
                        (a + 1)*xy[2, i] +
                        a*xy[2, i - 1])
                      /(h^2*a*(a + 1))
    od;
    RETURN(evalm(f12))
end;
```

Pentru a exemplifica utilizarea procedurii `derivare12` utilizăm procedura `tabelvalori`. Pentru a putea compara valorile obţinute prin aproximare cu valorile exacte utilizăm procedura `tabelexact`.

Procedura `tabelvalori` are ca parametri o funcţie f, un tablou unidimensional x cu n+1 puncte distincte din domeniul de definiţie al lui f şi numărul natural n. Procedura întoarce un tablou cu două linii şi n+1 coloane ce conţine pe prima linie cele n+1 puncte distincte din domeniul de definiţie al lui f, iar pe a doua linie valorile corespunzătoare ale funcţiei.

Procedura `tabelexact` are ca parametri a funcţie f un tablou x cu n+1 puncte distincte din domeniul de definiţie al lui f şi numărul natural n. Procedura întoarce tabelul cu valorile "exacte" ale derivatelor de ordinul întâi şi doi calculate în x_1, x_2, …x_{n-1}. Cuvântul "exacte" apare între ghilimele deoarece valorile derivatelor se obţin prin evaluarea în virgulă mobilă a lui f' şi f'' în x_1, x_2, …x_{n-1}.

Procedurile `tabelvalori` şi `tabelexact` sunt definite mai jos.

```
>tabelvalori := proc(f, x, n)
local txy, i;
    txy := array(1 .. 2, 0 .. n);
    for i from 0 to n do
        txy[1, i] := evalf(x[i]);
        txy[2, i] := evalf(f(x[i]))
    od;
    RETURN(evalm(txy))
end;

>tabelexact := proc(f, x, n)
local tf12, i, f1, f2;
    tf12 := array(1 .. 2, 1 .. n - 1);
    f1 := D(f);
    f2 := D(f1);
    for i to n - 1 do
        tf12[1, i] := evalf(f1(x[i]));
        tf12[2, i] := evalf(f2(x[i]))
    od;
    RETURN(evalm(tf12))
end;
```

Exemple.

```
> f:=(x->x^2*ln(x)+x/(1+x^(1/3))*sin(x));
```

$$f := x \to x^2 \ln(x) + \frac{x \sin(x)}{1 + x^{1/3}}$$

```
> x1:=array(0..4,[1,0.2,0.5,0.6,2]);
x1 := array(0 .. 4, [
    (0) = 1
    (1) = .2
    (2) = .5
    (3) = .6
    (4) = 2
    ])

> xy1:=tabelvalori(f,x1,4);
                        xy1 := txy
```

```
> derivare12(xy1,4);
```

$$\begin{bmatrix} -.3468421371 & .2961754511 & .5392795229 \\ 2.304734144 & 1.982049777 & 2.880031658 \end{bmatrix}$$

```
> tabelexact(f,x1,4);
```

$$\begin{bmatrix} -.2101526077 & .2793411783 & .5152246459 \\ .7569197273 & 2.232381348 & 2.473377763 \end{bmatrix}$$

```
> x2:=array(0..4,[1,2,4,10,20]);
x2 := array(0 .. 4, [
    (0) = 1
    (1) = 2
    (2) = 4
    (3) = 10
    (4) = 20
    ])
> xy2:=tabelvalori(f,x1,4);
                    xy2 := txy

> derivare12(xy2,4);
```

$$\begin{bmatrix} -.3468421371 & .2961754511 & .5392795229 \\ 2.304734144 & 1.982049777 & 2.880031658 \end{bmatrix}$$

```
> tabelexact(f,x2,4);
```

$$\begin{bmatrix} 4.731889938 & 13.84717387 & 53.25852832 \\ 3.245781283 & 6.554466881 & 8.922360828 \end{bmatrix}$$

```
> x3:=array(0..4,[0.01,0.02,0.032,0.04,0.42]);

x3 := array(0 .. 4, [
    (0) = .01
    (1) = .02
    (2) = .032
    (3) = .04
    (4) = .42
    ])
```

```
> xy3:=tabelvalori(f,x3,4);
                          xy3 := txy

> derivare12(xy3,4);
```

$$\begin{bmatrix} -.1042783238 & -.1407156534 & -.1509977381 \\ -3.416008656 & -2.656879604 & .08635845496 \end{bmatrix}$$

```
> tabelexact(f,x3,4);
```

$$\begin{bmatrix} -.1061441109 & -.1416789068 & -.1604604080 \\ -3.430234506 & -2.560903525 & -2.149939013 \end{bmatrix}$$

Algoritm pentru determinarea valorilor aproximative ale derivatelor de ordinul 1 şi 2 în cazul punctelor echidistante

Date de intrare:

n – numărul de puncte distincte echidistante din [a, b] este n +1

h - pasul reţelei

y – tablou ce conţine valorile funcţiei în punctele considerate.

y_0	y_1	…	y_n

$f(x_0+ih) = y_i$, i = 0,1,…,n

.

Date de ieşire:

f12 – tablou ce conţine pe prima linie valorile aproximative ale lui f' şi pe a doua linie valorile aproximative ale lui f''.

$\approx f'(x_1)$	$\approx f'(x_2)$	…	$\approx f'(x_n)$
$\approx f''(x_1)$	$\approx f''(x_1)$	…	$\approx f''(x_n)$

pentru i = 1,n-1,1 executa

$$f12[1,i] := \frac{y_{i+1} - y_{i-1}}{2h}$$

$$f12[2,i] := \frac{y_{i+1} - 2y_i + y_{i-1}}{h^2}$$

Procedură MAPLE pentru determinarea valorilor aproximative ale derivatelor de ordinul 1 şi 2 – cazul punctelor echidistante

```
>deriv12 := proc(y, h, n)
```

```
local f12, i;
    f12 := array(1 .. 2, 1 .. n - 1);
    for i to n - 1 do
     f12[1, i] := 1/2*(y[i + 1] - y[i - 1])/h;
     f12[2, i] := (y[i + 1] - 2*y[i] + y[i - 1])/h^2
    od;
    RETURN(evalm(f12))
end;
```

Pentru a exemplifica utilizarea procedurii `deriv12` utilizăm procedura `tabval` analoagă procedurii `tabelvalori` din cazul neechidistant. Pentru a putea compara valorile obținute prin aproximare cu valorile exacte utilizăm procedura `tabexact` analoagă procedurii `tabelexact`.

Procedura `tabval` are ca parametri o funcție f, punctul x0 din domeniul de definiție al lui f, pasul rețelei h şi numărul natural n semnificând faptul că rețeaua conține n+1 puncte: x_i = x0 + hi, i = 0,1,..n.. Procedura întoarce un tablou unidimensional ce conține valorile funcției în punctele x_i.

Procedura `tabexact` are ca parametri a funcție f, punctul x0 din domeniul de definiție al lui f, pasul rețelei h şi numărul natural n cu aceeaşi semnificație ca la `tabval`. Procedura întoarce tabelul cu valorile "exacte" ale derivatelor de ordinul întâi şi doi calculate în x_1, x_2, …x_{n-1}.

Procedurile `tabval` şi `tabexact` sunt definite mai jos.

```
>tabval := proc(f, x0, h, n)
local ty, i;
    ty := array(0 .. n);
    for i from 0 to n do
      ty[i] := evalf(f(x0 + h*i))
    od;
    RETURN(evalm(ty))
end;
```

```
>tabexact := proc(f, x0, h, n)
local tf12, i, f1, f2;
    tf12 := array(1 .. 2, 1 .. n - 1);
    f1 := D(f);
    f2 := D(f1);
    for i to n - 1 do
        tf12[1, i] := evalf(f1(x0 + h*i));
```

```
        tf12[2, i] := evalf(f2(x0 + h*i))
    od;
    RETURN(evalm(tf12))
end;
```

Exemple

```
> f:=(x->x^2*ln(x)+x/(1+x^(1/3))*sin(x));
```

$$f := x \to x^2 \ln(x) + \frac{x \sin(x)}{1+x^{1/3}}$$

```
> y1:=tabval(f,1,0.1,4);
```

$$y1 := ty$$

```
> deriv12(y1,0.1,4);
```

$$\begin{bmatrix} 1.920215738 & 2.223175515 & 2.529391665 \\ 3.010663240 & 3.048532300 & 3.075790700 \end{bmatrix}$$

```
> tabexact(f,1,0.1,4);
```

$$\begin{bmatrix} 1.919474938 & 2.222646607 & 2.529005501 \\ 3.011911688 & 3.049410073 & 3.076345947 \end{bmatrix}$$

```
> y2:=tabval(f,1,10,4);
```

$$y2 := ty$$

```
> deriv12(y2,10,4);
```

$$\begin{bmatrix} 67.34439040 & 150.5151940 & 244.6872025 \\ 7.742624191 & 8.891536527 & 9.942865170 \end{bmatrix}$$

```
> tabexact(f,1,10,4);
```

$$\begin{bmatrix} 63.52994516 & 145.9780642 & 250.6815499 \\ 11.21546631 & 4.192579015 & 13.22308820 \end{bmatrix}$$

```
> y3:=tabval(f,1,0.01,4);
```

$$y3 := ty$$

```
> deriv12(y3,0.01,4);
```

$$\begin{bmatrix} 1.650398460 & 1.680082315 & 1.709823940 \\ 2.965448000 & 2.971323000 & 2.977002000 \end{bmatrix}$$

```
> tabexact(f,1,0.01,4);
```

$$\begin{bmatrix} 1.650388516 & 1.680072680 & 1.709814621 \\ 2.965463472 & 2.971337038 & 2.977020077 \end{bmatrix}$$

Probleme propuse

Scrieți proceduri MAPLE pentru aproximarea derivatelor unei funcții f dată printr-un tabel

x_0	x_1	…	x_n
y_0	y_1	…	y_n

$f(x_i) = y_i$, $i = 0,1,\ldots,n$

cu nodurile $x_0, x_1,\ldots, x_n$ echidistante. Utilizați formule de aproximare:

$$f'(x_i) \approx \frac{f(x_{i+1}) - f(x_i)}{h}$$

$$f''(x_i) \approx \frac{f(x_{i+2}) - 2f(x_{i+1}) + f(x_i)}{h^2}$$

sau

$$f'(x_i) \approx \frac{f(x_i) - f(x_{i-1})}{h}$$

$$f''(x_i) \approx \frac{f(x_i) - 2f(x_{i-1}) + f(x_{i-2})}{h^2}$$

III.2. Metode de derivare numerică folosind interpolarea

Fie $f : [a, b] \to \mathbf{R}$ o funcție, fie $x_0, x_1, \ldots, x_n$ n+1 puncte distincte din intervalul $[a, b]$, şi $y_i = f(x_i)$ pentru orice $i = 0,1,\ldots n$. Aproximăm $f^{(k)}$ prin derivata de ordinul k a polinomului de interpolare asociat lui f şi $x_0, x_1, \ldots, x_n$. Ținând cont de eroarea cu care polinomul de interpolare aproximează funcția se poate stabili eroarea cu care derivata polinomului de interpolare aproximează funcția. Să presupunem că $n = 4$ şi că punctele x_0, x_1, x_2, x_3, x_4 sunt echidistante şi să construim polinomul Newton ascendent.

$$P_4(x) = P_4(x_0 + th)$$

$$= f(x_0) + \frac{\Delta f(x_0)}{1!} t + \frac{\Delta^2 f(x_0)}{2!} t(t-1) + \frac{\Delta^3 f(x_0)}{3!} t(t-1)(t-2) +$$

$$+ \frac{\Delta^4 f(x_0)}{4!} t(t-1)(t-2)(t-3)$$

$$= f(x_0) + \frac{\Delta f(x_0)}{1!} t + \frac{\Delta^2 f(x_0)}{2!}(t^2 - t) + \frac{\Delta^3 f(x_0)}{3!}(t^3 - 3t^2 + 2t) +$$

$$+ \frac{\Delta^4 f(x_0)}{4!}(t^4 - 6t^3 + 11t^2 - 6t)$$

Notăm $g(t) = P_4(x_0 + th)$. Avem

$$g'(t) = P_4'(x_0 + th)h$$
$$g''(t) = P_4''(x_0 + th)h^2.$$

Pe de altă parte

$$g'(t) = \Delta f(x_0) + \frac{\Delta^2 f(x_0)}{2}(2t - 1) + \frac{\Delta^3 f(x_0)}{6}(3t^2 - 6t + 2) +$$

$$+ \frac{\Delta^4 f(x_0)}{24}(4t^3 - 18t^2 + 22t - 6)$$

$$= \Delta f(x_0) + \Delta^2 f(x_0)\frac{(2t-1)}{2} + \Delta^3 f(x_0)\frac{(3t^2 - 6t + 2)}{6} +$$

$$+ \Delta^4 f(x_0)\frac{(2t^3 - 9t^2 + 11t - 3)}{12}$$

iar

$$g''(t) = \frac{\Delta^2 f(x_0)}{2} 2 + \frac{\Delta^3 f(x_0)}{6}(6t - 6) +$$

$$+ \frac{\Delta^4 f(x_0)}{12}(6t^2 - 18t + 11)$$

$$= \Delta^2 f(x_0) + \Delta^3 f(x_0)(t - 1) + \Delta^4 f(x_0)\frac{(6t^2 - 18t + 11)}{12}.$$

În consecință avem:

$$f'(x) = f'(x_0 + ht) \approx \frac{1}{h}\left(\Delta f(x_0) + \Delta^2 f(x_0)\frac{(2t-1)}{2} + \Delta^3 f(x_0)\frac{(3t^2 - 6t + 2)}{6} + \right.$$

$$\left. + \Delta^4 f(x_0)\frac{(2t^3 - 9t^2 + 11t - 3)}{12} \right)$$

$$f''(x) = f''(x_0 + ht) \approx \frac{1}{h^2}\left(\Delta^2 f(x_0) + \Delta^3 f(x_0)(t - 1) + \Delta^4 f(x_0)\frac{(6t^2 - 18t + 11)}{12} \right).$$

Algoritm pentru determinarea valorii aproximative a derivatei unei într-un punct (aproximarea funcției se face cu polinomul Newton ascendent de grad4)

Date de intrare:

x0- primul punct din rețea

h – pasul rețelei

y – tablou ce conține valorile funcției în punctele rețelei.

y_0	y_1	…	y_n

$y_i = f(x_i) = f(x0 + ih)$, i = 0,1,2,3,4

x – punctul în care se aproximează derivată lui f.

Date de ieşire:

f1x - valoarea aproximativă a f'(x).

```
t : = (x-x0)/h;
pentru i = 0,3,1 executa
    pentru j =3,i,0,-1 executa
        y_{j+1} := y_{j+1} –y_i;
```

$f1x := y_1 + y_2*(t-1/2) + y_3/6*(3*t^2-6t+2) + y_4/12*(2t^3-9t^2+11t-3)$;

f1x:=f1x/h;

Analog se calculează aproximația derivatei de ordinul 2.

Proceduri MAPLE pentru calculul valorilor aproximative a derivatelor de ordinul 1 şi 2 utilizând polinomului Newton ascendent de grad 4

```
>deriv1 := proc(y, h, x0, x)
local i, j, t, yl, f1x;
    yl := array(0 .. 4);
    for i from 0 to 4 do yl[i] := y[i] od;
    t := (x - x0)/h;
    for i from 0 to 3 do
        for j from 3 by -1 to i do
            yl[j + 1] := yl[j + 1] - yl[j]
        od
    od;
    f1x := yl[1] + yl[2]*(t - 1/2) +
            1/6*yl[3]*(3*t^2 - 6*t + 2)+
```

```
          1/12*yl[4]*(2*t^3 - 9*t^2 + 11*t - 3);
    f1x := f1x/h;
    RETURN(f1x)
end;


>deriv2 := proc(y, h, x0, x)
local i, j, t, yl, f2x;
    yl := array(0 .. 4);
    for i from 0 to 4 do yl[i] := y[i] od;
    t := (x - x0)/h;
    for i from 0 to 3 do
        for j from 3 by -1 to i do
          yl[j + 1] := yl[j + 1] - yl[j]
        od
    od;
    f2x := yl[2] + yl[3]*(t - 1) +
           1/12*yl[4]*(-12*t^2 + 11);
    f2x := f2x/h^2;
    RETURN(f2x)
end;
```

Pentru a exemplifica utilizarea procedurilor `deriv1` şi `deriv2` utilizăm procedura `tabvalori` Procedura `tabvalori` are ca parametri o funcţie f, punctul x0 din domeniul de definiţie al lui f şi pasul reţelei h. Procedura întoarce un tablou unidimensional ce conţine valorile funcţiei în punctele $x_i = x0 + hi$, i = 0,1,2,3,4.

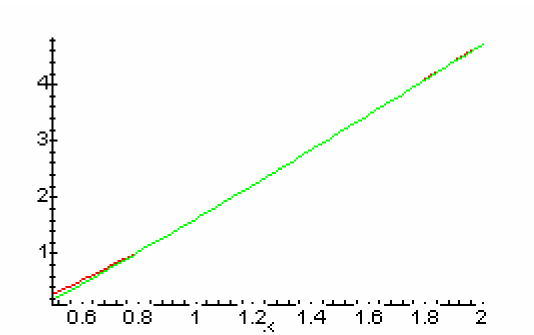
```
>tabvalori := proc(f, h, x0)
local i, ty;
    ty := array(0 .. 4);
    for i from 0 to 4 do
       ty[i] := evalf(f(x0 + h*i))
    od;
    RETURN(evalm(ty))
end;
```

Exemple

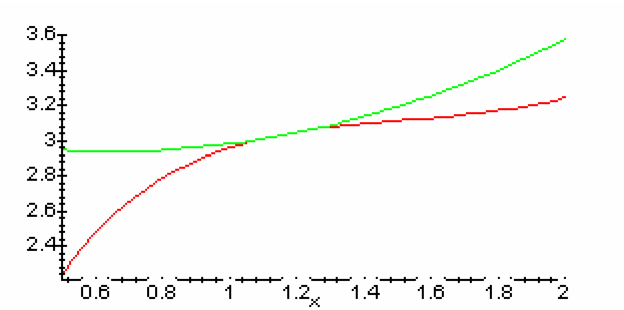
```
> f:=(x->x^2*ln(x)+x/(1+x^(1/3))*sin(x));
```

$$f := x \to x^2 \ln(x) + \frac{x \sin(x)}{1 + x^{1/3}}$$

```
> y1:=tabvalori(f,0.2,1);
                        y1 := ty
> deriv1(y1,0.2,1,1.5);
                     3.148195264
> evalf(D(f)(1.5));
                     3.148216949
> deriv2(y1,0.2,1,1.5);
                     3.194501068
> evalf(D(D(f))(1.5));
                     3.113593341
> with(plots);
> plot([diff(f(x),x),deriv1(y1,0.2,1,x)],x=0.5..2);
```

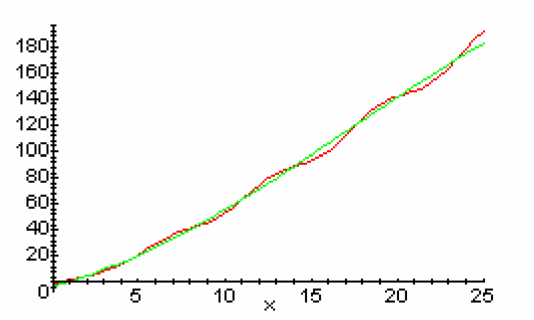



```
>plot([diff(diff(f(x),x),x),deriv2(y1,0.2,1,x)],x=0.
5..2);
```

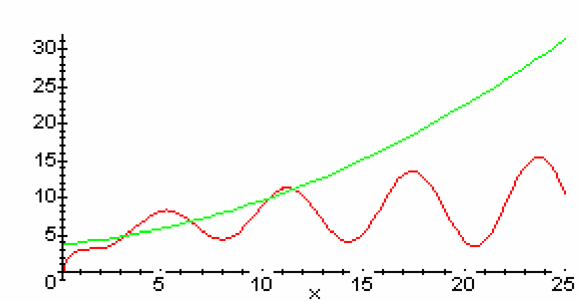



```
> y2:=tabvalori(f,5,1);
```

```
                              y2 := ty
> deriv1(y2,5,1,1.5);
                           2.546042046
> evalf(D(f)(1.5));
                           3.148216949
> deriv2(y2,5,1,1.5);
                           4.172638336
> evalf(D(D(f))(1.5));
                           3.113593341
> plot([diff(f(x),x),deriv1(y2,5,1,x)],x=0.1..25);
```

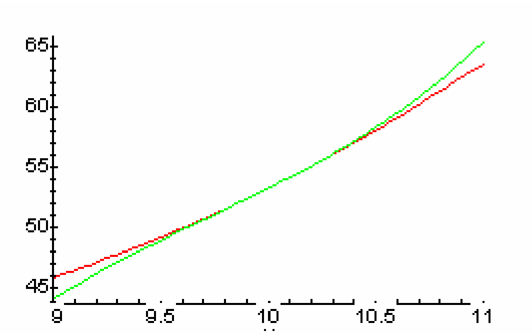



```
>plot([diff(diff(f(x),x),x),deriv2(y2,5,1,x)],x=0.1.
.25);
```




```
> y3:=tabvalori(f,0.01,10);
                              y3 := ty
> deriv1(y3,0.01,10,10.025);
                           53.48260625
> evalf(D(f)(10.025));
                           53.48260830
> deriv2(y3,0.01,10,10.025);
```

```
                    8.997666667
> evalf(D(D(f))(10.025));
                    9.003934377
> plot([diff(f(x),x),deriv1(y3,0.01,10,x)],x=9..11);
```




```
>plot([diff(diff(f(x),x),x),deriv2(y3,0.01,10,x)],x=
9..11);
```