Rezolvarea ecuaţiilor algebrice neliniare

Prof.dr.ing. Gabriela Ciuprina

Universitatea "Politehnica" Bucureşti, Facultatea de Inginerie Electrică

Suport didactic pentru disciplina *Algoritmi Numerici*, 2017-2018

Cuprins

Formularea problemei

Enunţ

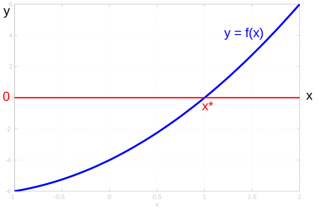
Se dă $f : [a, b] \to \mathbb{R}$, continuă.
Se cere x pentru care

$$f(x) = 0$$

Buna formulare matematică

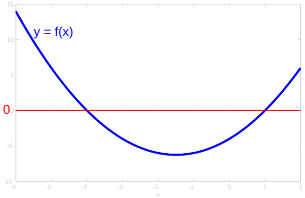
Există o soluţie $x^* \in [a, b]$ şi aceasta este unică.

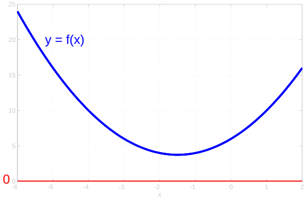
$$f(x^*) = 0$$



Formularea problemei

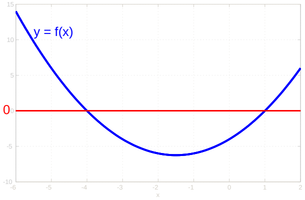
Exemple de probleme prost formulate:

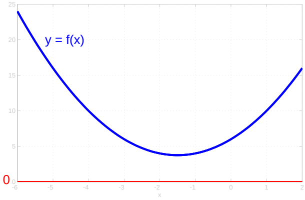




Formularea problemei

Exemple de probleme prost formulate:

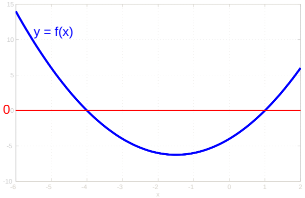




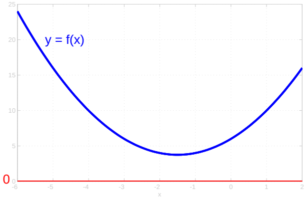
Soluţia nu este unică.

Formularea problemei

Exemple de probleme prost formulate:



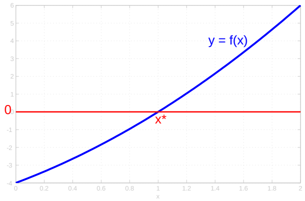
Soluţia nu este unică.

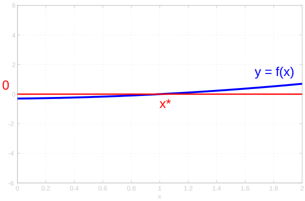


Nu există soluţie.

Condiţionarea problemei

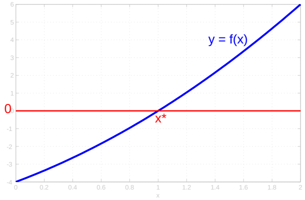
Condiţionarea depinde de panta lui *f* în apropierea soluţiei.

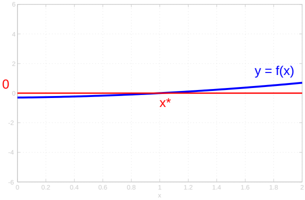




Condiţionarea problemei

Condiţionarea depinde de panta lui *f* în apropierea soluţiei.

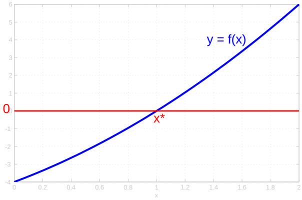


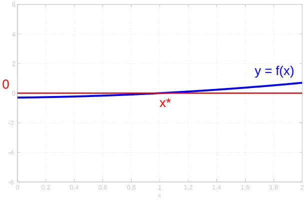


Bine condiţionată.

Condiţionarea problemei

Condiţionarea depinde de panta lui *f* în apropierea soluţiei.





Bine condiţionată.

Prost condiţionată.

Condiţionarea problemei

Numărul de condiţionare (revedeţi cursul despre erori):

Formulare implicită

$$f(x) = y$$

(y - date, x - rezultat), aici $y = 0$

Formulare explicită

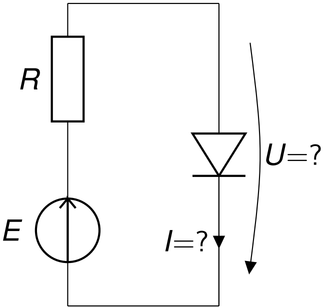
$$x = g(y)$$

$(g = f^{-1})$

$$\hat{k} = \|J(g(y))\| = |g'(y)| = \frac{1}{|f'(x)|}$$

Dacă $|f'(x^*)| \approx 0 \Rightarrow \hat{k}$ e mare $\Rightarrow$ prost condiţionată.

Exemplul 1



Se dau: E, R şi caracteristica $i = g(u)$

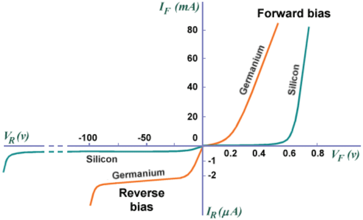
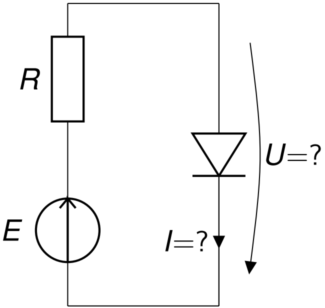


Figura este preluată de la

https://www.technologyuk.net/physics/

Se cere: punctul static de funcţionare al diodei (I, U)

Exemplul 1



$$\begin{aligned} u &= -Ri + E \\ i &= g(u) \end{aligned}$$

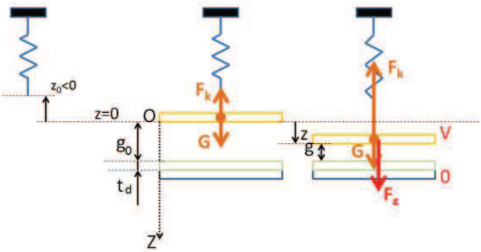
$$u + Rg(u) - E = 0$$

$$f(u) = 0$$

unde

$$f(u) = u + Rg(u) - E$$

Exemplul 2



Se dau:

g_0, A, t_d
k, ε_r
V

Se cere: g

$$k(g_0-g) = \frac{\varepsilon_0 A V^2}{2\left(g + \frac{t_d}{\varepsilon_r}\right)^2}$$

$$f(g) = 0$$

unde

$$f(g) = (g - g_0)\left(g + \frac{t_d}{\varepsilon_r}\right)^2 + \frac{\varepsilon_0 A V^2}{2k}$$

Metoda bisecţiei - ideea

Ipoteză suplimentară:

$$f(a)f(b) < 0$$

Ideea

$$x_0, x_1, \ldots, x_k, \ldots \to x^*$$

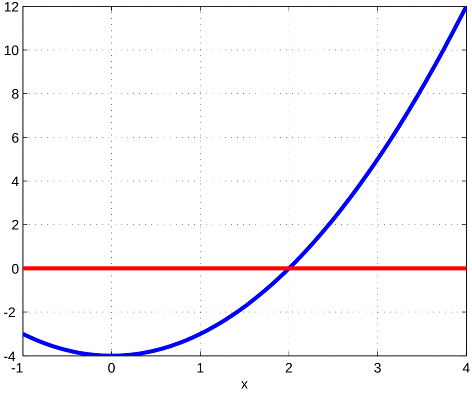
Prin înjumătăţirea intervalului:

1. $x_m = (a+b)/2$
2. se va selecta dintre intervalele $[a, x_m]$ şi $[x_m, b]$ pe acela care conţine soluţia
3. se renotează cu $[a, b]$ jumătatea aleasă şi se reia de la pasul 1.

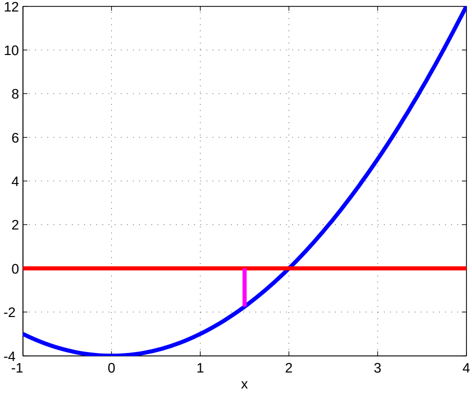
Algoritmul se opreşte atunci când $|b-a| < \varepsilon$

ε este o eroare absolută impusă de utilizator.

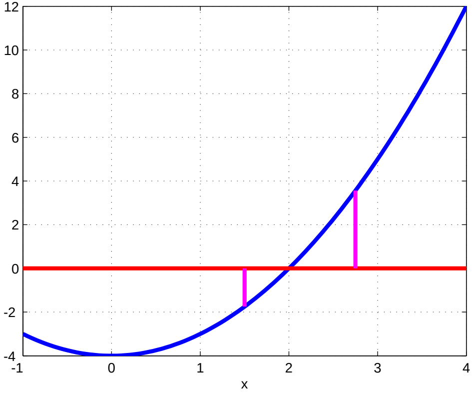
Metoda bisecţiei - ideea



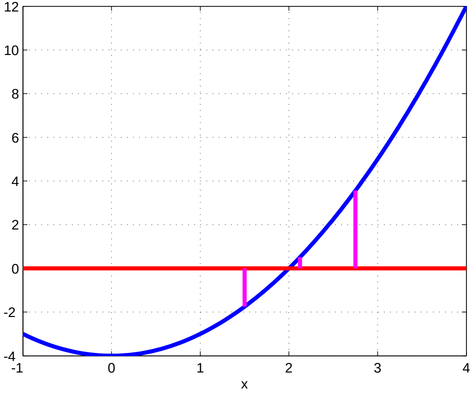
Metoda bisecţiei - ideea



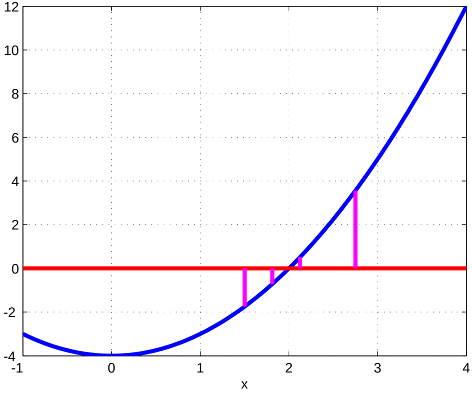
Metoda bisecţiei - ideea



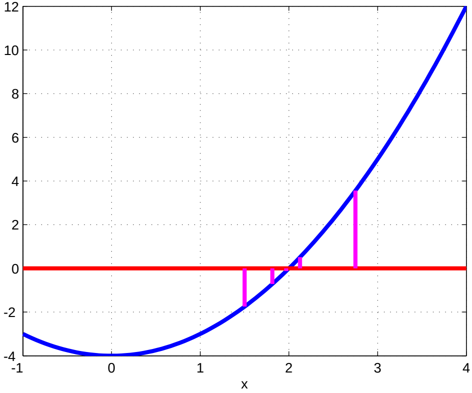
Metoda bisecţiei - ideea



Metoda bisecţiei - ideea



Metoda bisecţiei - ideea



Metodei bisecţiei - algoritm

funcţie bisecţie (*a*, *b*, *eps*, *nit*)
    real *a*, *b* ; domeniul de definiţie al funcţiei f
    real ε ; eroarea impusă
    întreg *nit* ; număr maxim de iteraţii
    real *xm* ; soluţia
    întreg $k = 0$; contor iteraţii
    repetă
        $k = k + 1$
        $xm = (a + b)/2$
        dacă $f(xm)f(a) > 0$ **atunci**
            $a = xm$
        altfel
            $b = xm$
    până când $(b - a) < eps$ **sau** $k > nit$
    dacă $k > nit$
        scrie Eroarea impusă nu a fost atinsă.
    întoarce *xm* ; soluţie
retur

Metoda bisecţiei - erori

La fiecare iteraţie, eroarea absolută se înjumătăţeste:

$$
\begin{aligned}
|x_0 - x^*| &< l \\
|x_1 - x^*| &< l/2 \\
|x_2 - x^*| &< l/2^2 \\
&\vdots \\
|x_k - x^*| &< l/2^k \\
&\vdots
\end{aligned}
$$

$l = b - a$

În ipotezele făcute, procedura este garantat convergentă!

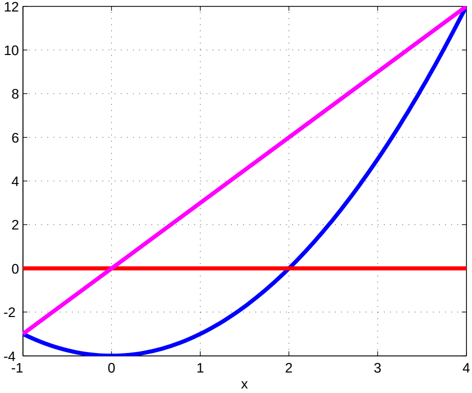
Metoda falsei poziţii (*regula falsi*) - ideea

Ideea: similară cu a bisecţiei - alegerea unui interval în care funcţia îşi schimbă semnul

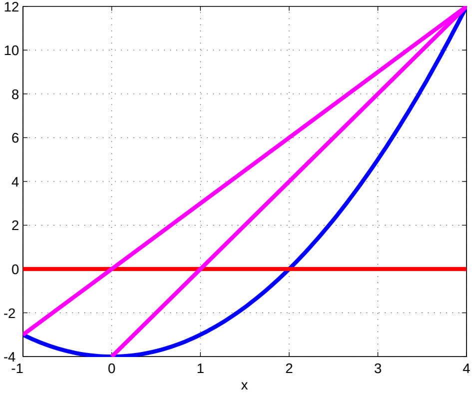
DAR nu se înjumătăţeşte intervalul

Intervalul se împarte în două părţi, determinate de intersecţia coardei determinată de punctele $(a, f(a))$ şi $(b, f(b))$ cu axa Oy.

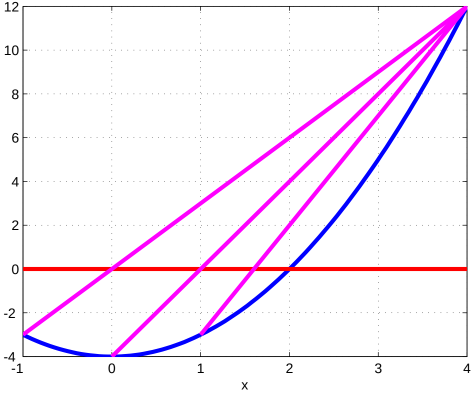
Metoda falsei poziţii (*regula falsi*) - ideea



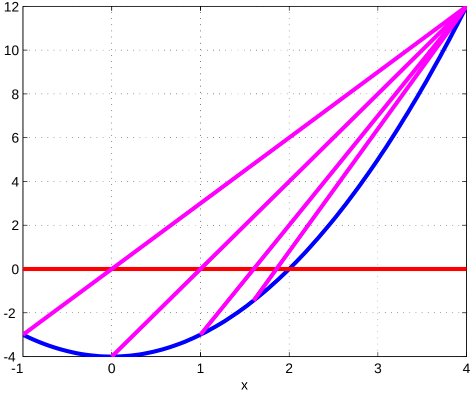
Metoda falsei poziţii (*regula falsi*) - ideea



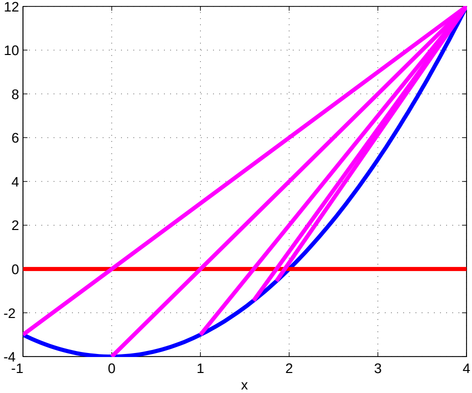
Metoda falsei poziţii (*regula falsi*) - ideea



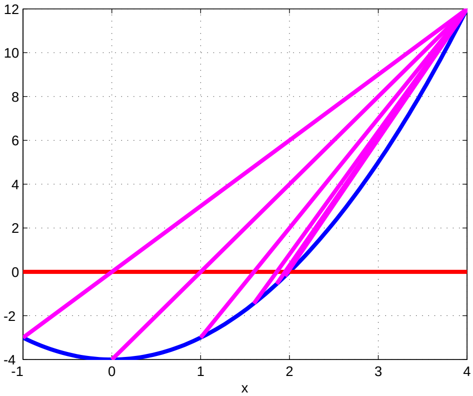
Metoda falsei poziţii (*regula falsi*) - ideea



Metoda falsei poziţii (*regula falsi*) - ideea



Metoda falsei poziţii (*regula falsi*) - ideea



Metoda falsei poziţii (*regula falsi*) - algoritm

$$y(x) = \frac{f(b) - f(a)}{b - a}(x - a) + f(a)$$

$$y(x) = 0 \quad \Rightarrow \quad x = \frac{af(b) - bf(a)}{f(b) - f(a)}$$

1. $x_m = (af(b) - bf(a))/(f(b) - f(a))$
2. se va selecta dintre intervalele $[a, x_m]$ şi $[x_m, b]$ pe acela care conţine soluţia
3. se renotează cu $[a, b]$ intervalul ales şi se reia de la pasul 1.

Algoritmul se opreşte atunci când $|b - a| < \varepsilon$

ε este o eroare absolută impusă de utilizator.

Metoda falsei poziţii (*regula falsi*) - convergenţă

- Convergenţa este asigurată dacă funcţia este continuă şi îşi schimbă semnul pe intervalul [a,b];
- Convergenţa ar putea fi mai rapidă decât la bisecţie dacă întotdeauna intervalul ales este mai mic decât jumătate din intervalul anterior;
- Ideea poate fi folosită în combinaţie cu metoda secantelor[1] pentru a preveni divergenţa acesteia din urmă.
- O variantă mai rapid convergentă a acestei metode este cunoscută sub numele de metoda lui Ridder detalii aici. Se foloseşte valoarea funcţiei f la mijlocul intervalului pentru a determina o altă funcţie h care are aceeaşi rădăcină ca şi f. Metoda falsei poziţii se aplică pentru h.

[1] Metoda secantelor este prezentată în slide-urile care urmează

Ideea metodelor bazate pe interpolare

Interpolare directă:

- Se folosesc valorile funcţiei deja calculate pentru a găsi o interpolare g a funcţiei f.
- Aproximaţia rădăcinii este dată de zeroul polinomului de interpolare g.

$f(x) \approx g(x)$

$f(x^*) = 0 \quad \Rightarrow \quad g(x^*) \approx 0$

Algoritm:

$g(x_k) = 0 \quad \Rightarrow \quad f(x_k) \approx 0$

Ideea metodelor bazate pe interpolare

Interpolare inversă:

- Se folosesc valorile funcţiei deja calculate pentru a găsi o interpolare h a funcţiei f^{-1}.
- Aproximaţia rădăcinii este dată de $h(0)$.

$$f(x) = y \quad \Rightarrow \quad x = f^{-1}(y)$$
$$f(x^*) = 0 \quad \Rightarrow \quad x^* = f^{-1}(0)$$
$$f^{-1}(y) \approx h(y) \Rightarrow \quad x^* \approx h(0)$$

Algoritm:

$$x_k = h(0) \quad \Rightarrow \quad x_k \approx x^*$$

Metode bazate pe interpolare liniară

Metode care folosesc interpolări de ordinul 1 (două puncte)

1. Metoda falsei poziţii[2] - cele două puncte nu sunt neaparat ultimele două aproximaţii calculate.
2. Metoda secantelor[3] - cele două puncte sunt exact ultimele două aproximaţii calculate.

[2] Prezentată anterior, ca o metodă de încadrare

[3] Prezentată în slide-urile următoare, ca o metodă de punct fix

Metode bazate pe interpolarea pătratică

Metode care folosesc interpolări de ordinul 2 (trei puncte)

1. Metoda Muller
   Se aproximează f cu o parabolă folosind ultimele trei aproximaţii calculate.
   Se calculează o rădăcină a acestei parabole, cea mai apropiată de ultima aproximaţie calculată.
   Detalii la
   Eric W. "Muller's Method." From MathWorld–A Wolfram Web Resource.
   http://mathworld.wolfram.com/MullersMethod.html
2. Interpolarea pătratică inversă
   Se aproximează f^{-1} cu o parabolă folosind ultimele trei aproximaţii calculate.
   Se evaluează acest polinom de interpolare în 0.
   Este folosită mai des în combinaţie cu alte metode.

Interpolarea pătratică inversă

Polinomul de interpolare pentru f^{-1}, folosind punctele: (x_{k-2}, f_{k-2}), (x_{k-1}, f_{k-1}), (x_k, f_k) este:

$$\begin{aligned} f^{-1}(y) &= \frac{(y-f_{k-1})(y-f_k)}{(f_{k-2}-f_{k-1})(f_{k-2}-f_k)}x_{k-2} + \frac{(y-f_{k-2})(y-f_k)}{(f_{k-1}-f_{k-2})(f_{k-1}-f_k)}x_{k-1} + \\ &+ \frac{(y-f_{k-2})(y-f_{k-1})}{(f_k-f_{k-2})(f_k-f_{k-1})}x_k \end{aligned}$$

Aproximaţia următoare $x_{k+1} = f^{-1}(0)$

$$\begin{aligned} x_{k+1} &= \frac{f_{k-1}f_k}{(f_{k-2}-f_{k-1})(f_{k-2}-f_k)}x_{k-2} + \frac{f_{k-2}f_k}{(f_{k-1}-f_{k-2})(f_{k-1}-f_k)}x_{k-1} + \\ &+ \frac{f_{k-2}f_{k-1}}{(f_k-f_{k-2})(f_k-f_{k-1})}x_k \end{aligned} \tag{1}$$

Ideea metodei iteraţiei simple

Ecuaţia de rezolvat:

$$f(x) = 0 \tag{2}$$

Ideea:

$$x_0, x_1, \ldots, x_k, \ldots \to x^*$$

Calcul recursiv:

$$x_k = g(x_{k-1}), \tag{3}$$

g se numeşte *funcţie de iteraţie*

Ideea metodei iteraţiei simple

Ecuaţia de rezolvat:

$$f(x) = 0 \tag{2}$$

Ideea:

$$x_0, x_1, \ldots, x_k, \ldots \to x^*$$

Calcul recursiv:

$$x_k = g(x_{k-1}), \tag{3}$$

g se numeşte *funcţie de iteraţie*

1. $g = ?$

Ideea metodei iteraţiei simple

Ecuaţia de rezolvat:

$$f(x) = 0 \tag{2}$$

Ideea:

$$x_0, x_1, \ldots, x_k, \ldots \to x^*$$

Calcul recursiv:

$$x_k = g(x_{k-1}), \tag{3}$$

g se numeşte *funcţie de iteraţie*

1. $g = ?$
2. $x_0 = ?$

Ideea metodei iteraţiei simple

Ecuaţia de rezolvat:

$$f(x) = 0 \tag{2}$$

Ideea:

$$x_0, x_1, \ldots, x_k, \ldots \to x^*$$

Calcul recursiv:

$$x_k = g(x_{k-1}), \tag{3}$$

g se numeşte *funcţie de iteraţie*

1. $g =?$
2. $x_0 =?$
3. Şirul este convergent?

Ideea metodei iteraţiei simple

Ecuaţia de rezolvat:

$$f(x) = 0 \tag{2}$$

Ideea:

$$x_0, x_1, \ldots, x_k, \ldots \to x^*$$

Calcul recursiv:

$$x_k = g(x_{k-1}), \tag{3}$$

g se numeşte *funcţie de iteraţie*

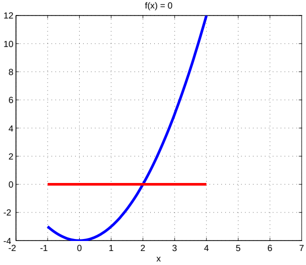
1. $g = ?$
2. $x_0 = ?$
3. Şirul este convergent?
4. Care este criteriul de oprire?

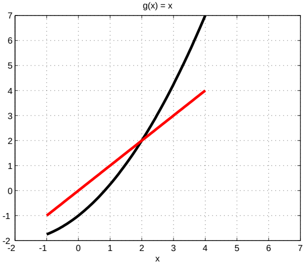
Metoda iteraţiei simple - construcţia lui g

$$x_k = g(x_{k-1}), \qquad \lim_{k\to\infty} x_k = x \tag{4}$$

$$x = g(x) \tag{5}$$

Soluţia ecuaţiei $f(x) = 0$ este punct fix al aplicaţiei g





Metoda iteraţiei simple - construcţia lui *g*

$$f(x) = 0$$

Metoda iteraţiei simple - construcţia lui g

$$f(x) = 0$$

$$cf(x) = 0$$

Metoda iteraţiei simple - construcţia lui *g*

$$f(x) = 0$$

$$cf(x) = 0$$

$$x + cf(x) = x$$

Metoda iteraţiei simple - construcţia lui g

$$f(x) = 0$$

$$cf(x) = 0$$

$$x + cf(x) = x$$

$$g(x) = x + cf(x) \tag{6}$$

Metoda iteraţiei simple - construcţia lui g

$$f(x) = 0$$

$$cf(x) = 0$$

$$x + cf(x) = x$$

$$g(x) = x + cf(x) \tag{6}$$

$$x_k = x_{k-1} + cf(x_{k-1}) \qquad k = 1, 2, \ldots, n. \tag{7}$$

Metoda iteraţiei simple - convergenţa

$$g(x) = x + cf(x) \tag{8}$$

$$x_k = x_{k-1} + cf(x_{k-1}) \qquad k = 1, 2, \ldots, n. \tag{9}$$

Iniţializare arbitrară $x_0 \in [a, b]$.

Obs: Constanta c influenţează puternic convergenţa.

Teoremă - condiţie suficientă de convergenţă

Dacă g este o contracţie, atunci şirul iteraţiilor este convergent.

g este contracţie, dacă:

$$|g(x_1) - g(x_2)| < |x_1 - x_2|, \forall x_1, x_2 \in [a, b] \tag{10}$$

Metoda iteraţiei simple - convergenţa

$$g(x) = x + cf(x) \tag{8}$$

$$x_k = x_{k-1} + cf(x_{k-1}) \qquad k = 1, 2, \ldots, n. \tag{9}$$

Iniţializare arbitrară $x_0 \in [a, b]$.
Obs: Constanta c influenţează puternic convergenţa.

Teoremă - condiţie suficientă de convergenţă

Dacă g este o contracţie, atunci şirul iteraţiilor este convergent.

g este contracţie, dacă:

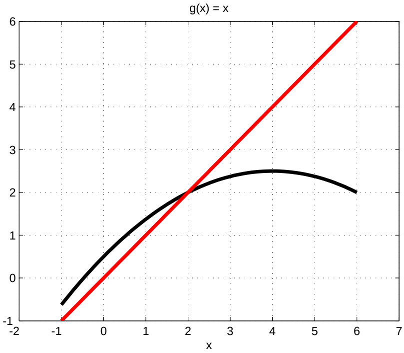
$$|g(x_1) - g(x_2)| \leq L|x_1 - x_2|, \forall x_1, x_2 \in [a, b] \tag{10}$$

$L < 1$ (strict!)

Metoda iteraţiei simple - convergenţa

Teoremă - condiţie suficientă de convergenţă

Dacă f este derivabilă şi $|g'| < 1 \Rightarrow$ iteraţii convergente.

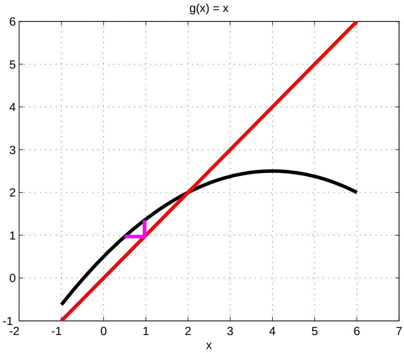


Convergent

Metoda iteraţiei simple - convergenţa

Teoremă - condiţie suficientă de convergenţă

Dacă f este derivabilă şi $|g'| < 1 \Rightarrow$ iteraţii convergente.

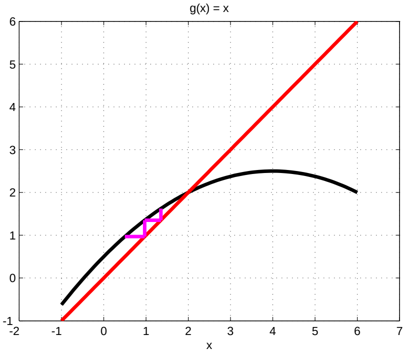


Convergent

Metoda iteraţiei simple - convergenţa

Teoremă - condiţie suficientă de convergenţă

Dacă f este derivabilă şi $|g'| < 1 \Rightarrow$ iteraţii convergente.

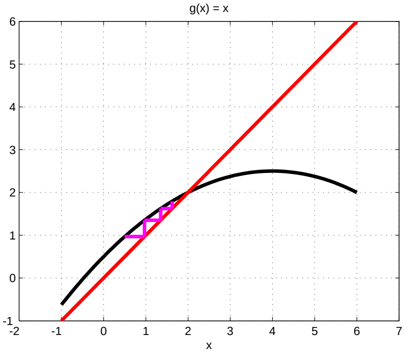


Convergent

Metoda iteraţiei simple - convergenţa

Teoremă - condiţie suficientă de convergenţă

Dacă f este derivabilă şi $|g'| < 1 \Rightarrow$ iteraţii convergente.

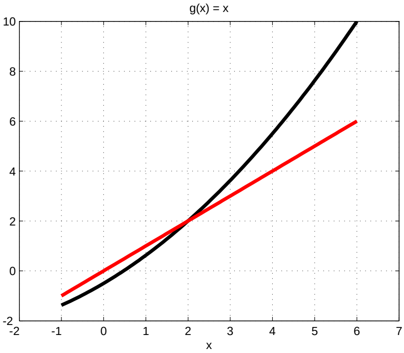


Convergent

Metoda iteraţiei simple - convergenţa

Teoremă - condiţie suficientă de convergenţă

Dacă f este derivabilă şi $|g'| < 1 \Rightarrow$ iteraţii convergente.

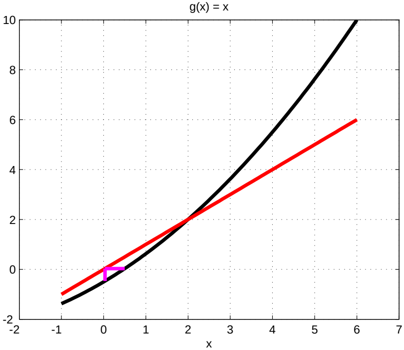


Divergent

Metoda iteraţiei simple - convergenţa

Teoremă - condiţie suficientă de convergenţă

Dacă f este derivabilă şi $|g'| < 1 \Rightarrow$ iteraţii convergente.

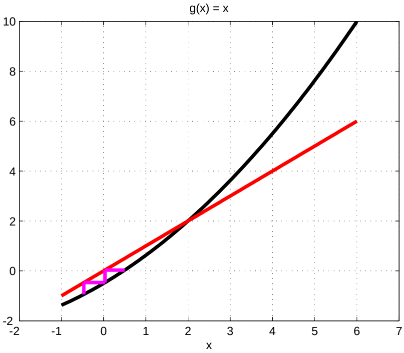


Divergent

Metoda iteraţiei simple - convergenţa

Teoremă - condiţie suficientă de convergenţă

Dacă f este derivabilă şi $|g'| < 1 \Rightarrow$ iteraţii convergente.

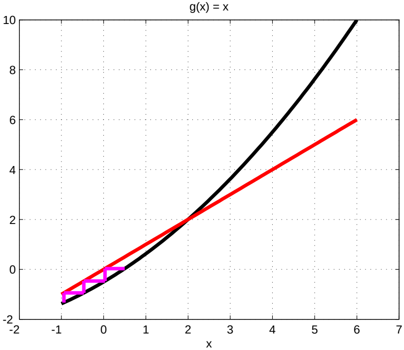


Divergent

Metoda iteraţiei simple - convergenţa

Teoremă - condiţie suficientă de convergenţă

Dacă f este derivabilă şi $|g'| < 1 \Rightarrow$ iteraţii convergente.



Divergent

Metoda iteraţiei simple - convergenţa

Condiţia $|g'| < 1$ este echivalentă cu:

$$|1 + cf'(x)| < 1, \quad x \in [a, b]. \tag{11}$$

$\Rightarrow$ importanţa constantei c

Cu cât $|g'| = |1 + cf'(x)|$ este mai mic, cu atât şirul iterativ este mai rapid convergent.

Notăm L o margine a derivatei $|g'|(x) \leq L$.

$$|x_1 - x^*| = |g(x_0) - g(x^*)| = |g'(\zeta)(x_0 - x^*)| \leq L|x_0 - x^*|$$

$$|x_2 - x^*| \leq L|x_1 - x^*| \leq L^2|x_0 - x^*|$$

$$|x_3 - x^*| \leq L|x_2 - x^*| \leq L^3|x_0 - x^*|$$

$$\vdots$$

$$|x_k - x^*| \leq L^k|x_0 - x^*| \tag{12}$$

Metoda iteraţiei simple - condiţia de oprire

Eroarea $|x_n - x^*|$ - nu se poate calcula
Reziduul $|f(x_n)|$ - se poate calcula, dar trebuie corelat cu numărul de condiţionare

$$f(x_n) - f(x^*) = f'(\zeta)(x_n - x^*)$$

$$|x_n - x^*| = \frac{1}{f'(\zeta)}|f(x_n)|$$

$$|x_n - x^*| \leq \hat{k}|f(x_n)|$$

Metoda iteraţiei simple - condiţia de oprire

Dar

$$|x_n - x_{n-1}| = |g(x_{n-1}) - x_{n-1}| = |x_{n-1} + cf(x_{n-1}) - x_{n-1}| = |cf(x_{n-1})|$$

Dacă c e corelat cu inversa derivatei, atunci cel mai natural criteriu de oprire este

$$|x_n - x_{n-1}| < \varepsilon$$

unde ε este parametru de intrare (impus de utilizator).

Metoda Newton

c se alege a.î. viteza de convergenţă să fie maximă:

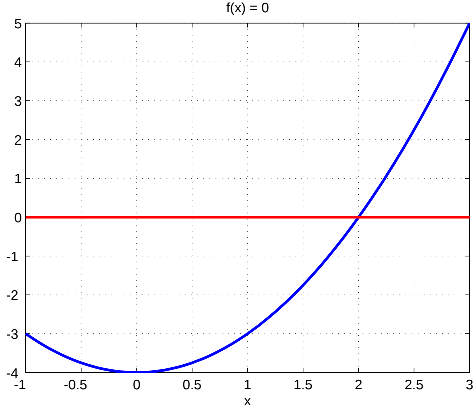
$$1 + c_k f'(x_k) = 0$$

$$c_k = -\frac{1}{f'(x_k)}. \tag{13}$$

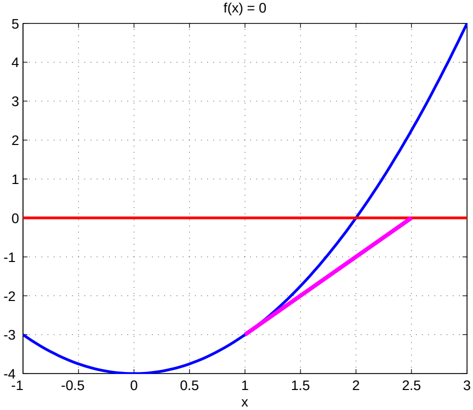
$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} \quad k = 1, 2, \ldots \tag{14}$$

Semnificaţie geometrică: La fiecare iteraţie graficul funcţiei este aproximat cu tangenta dusă în punctul de coordonate $x_k, f(x_k)$.
OBS: Metoda eşuează dacă tangenta are panta zero.

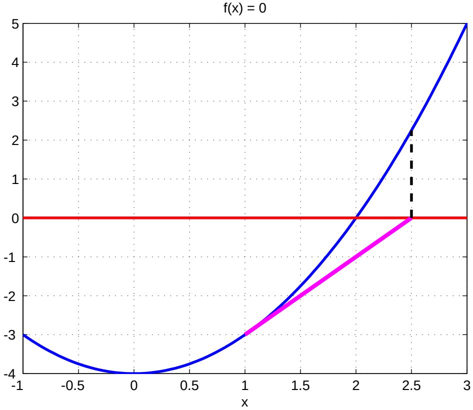
Metoda Newton



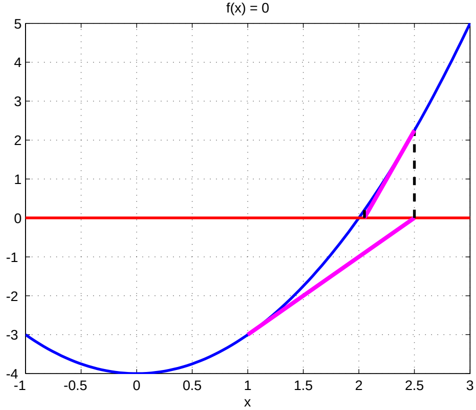
Metoda Newton



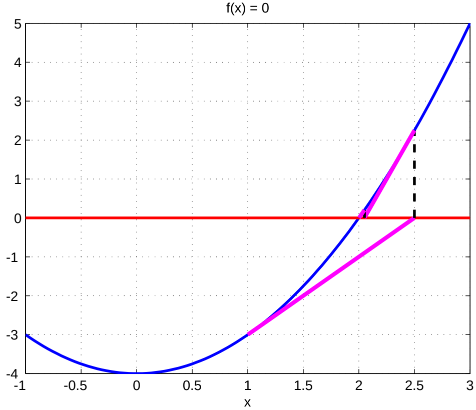
Metoda Newton



Metoda Newton



Metoda Newton



Metoda Newton

Justificare: Ecuaţia dreptei tangente:

$$y = f'(x)(x - x_k) + f(x_k), \tag{15}$$

Intersecţia tangentei cu axa orizontală:

$$y = 0 \Rightarrow x = x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$

Obs: la fiecare iteraţie trebuie evaluată derivata $f'(x_k)$, ceea ce poate necesita un efort mare de calcul.
:(
Ce se poate face pentru diminuarea efortului de calcul?

Metoda tangentelor paralele

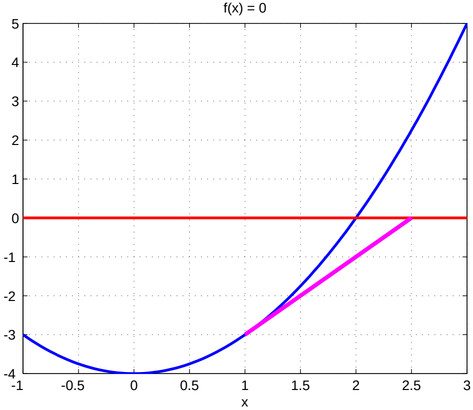
Variantă simplificată metoda **Newton-Kantorovici (a tangentelor paralele)**

$$c = -1/f'(x_0)$$

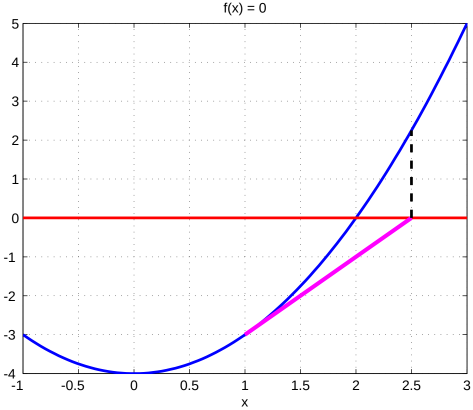
$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_0)} \tag{16}$$

Semnificaţia geometrică?

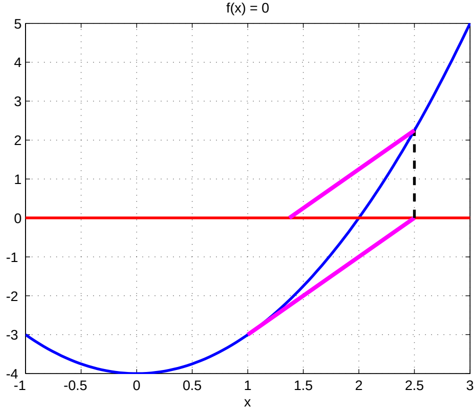
Metoda tangentelor paralele



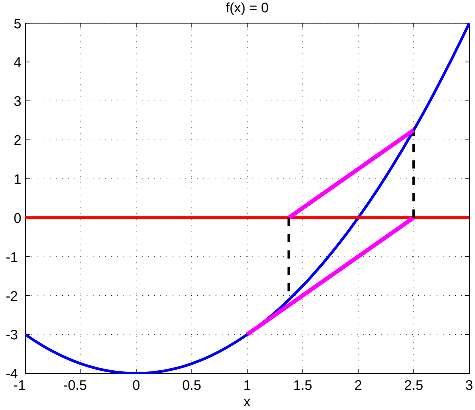
Metoda tangentelor paralele



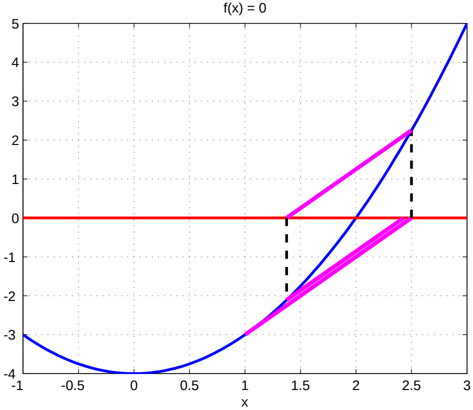
Metoda tangentelor paralele



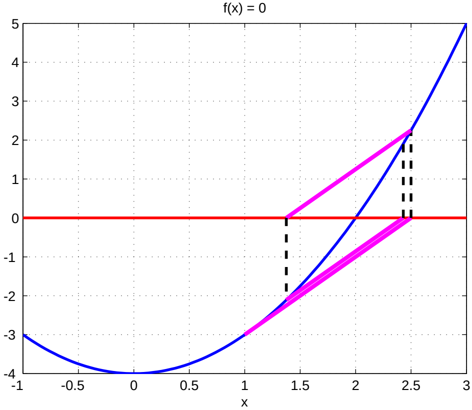
Metoda tangentelor paralele



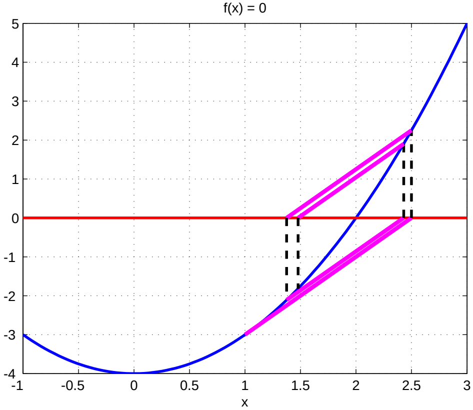
Metoda tangentelor paralele



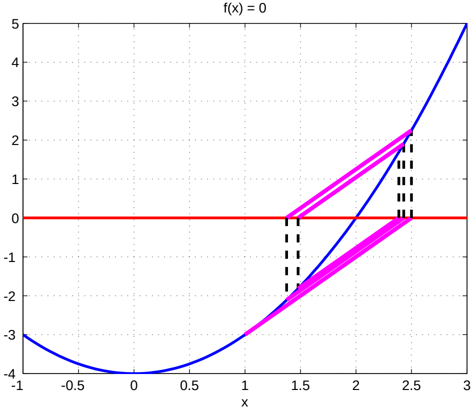
Metoda tangentelor paralele



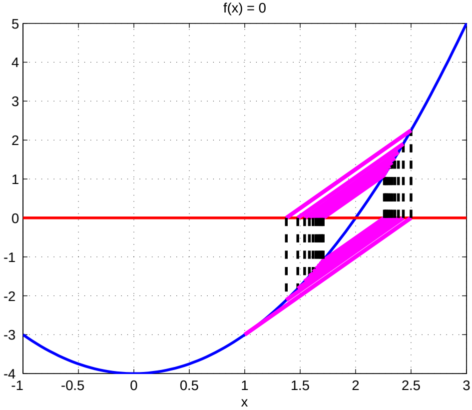
Metoda tangentelor paralele



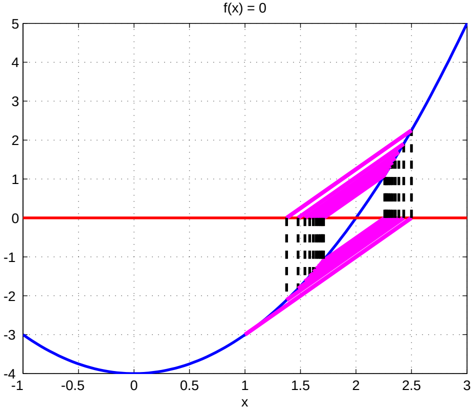
Metoda tangentelor paralele



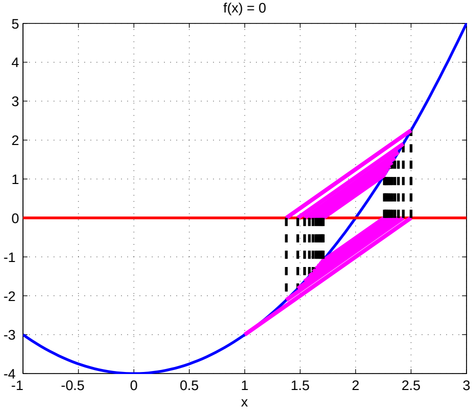
Metoda tangentelor paralele



Metoda tangentelor paralele

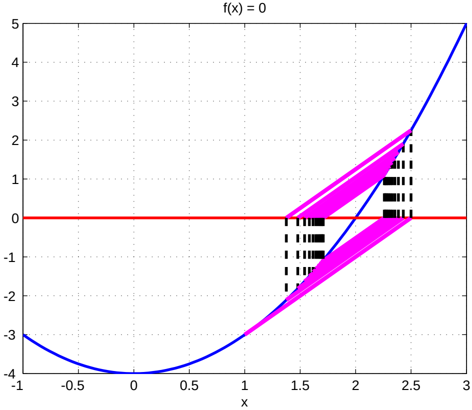


Metoda tangentelor paralele



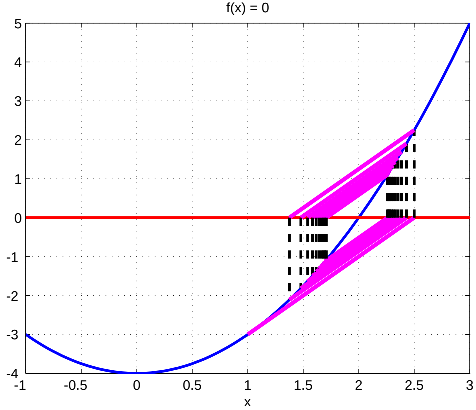
- S-a redus efortul de calcul, dar doar pentru o iteraţie

Metoda tangentelor paralele



- S-a redus efortul de calcul, dar doar pentru o iteraţie
- Rămâne necesitatea de a avea o expresie analitică pentru derivată $f'(x)$.

Metoda tangentelor paralele



- S-a redus efortul de calcul, dar doar pentru o iteraţie
- Rămâne necesitatea de a avea o expresie analitică pentru derivată $f'(x)$.
- Ce se poate face dacă nu există o astfel de expresie?

Metoda secantelor

Foloseşte o aproximare numerică a derivatei

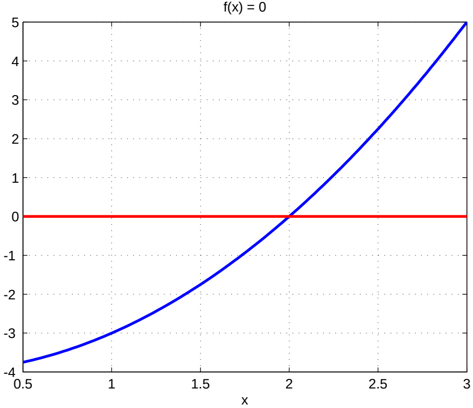
$$f'(x_k) \approx \frac{f(x_k) - f(x_{k-1})}{x_k - x_{k-1}}, \tag{17}$$

$$x_{k+1} = x_k - \frac{f(x_k)(x_k - x_{k-1})}{f(x_k) - f(x_{k-1})} \quad k = 1, 2, \dots \tag{18}$$

Semnificaţie geometrică: La fiecare iteraţie graficul funcţiei este aproximat cu secanta ce uneşte ultimele două puncte din şirul iterativ, având coordonatele $x_{k-1}, f(x_{k-1})$ şi respectiv $x_k, f(x_k)$.
OBS: Metoda eşuează dacă secanta are panta zero.

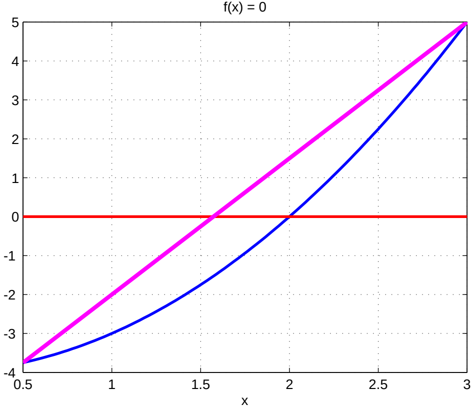
Metoda secantelor

Obs: funcţia de iteraţie are două variabile $x_{k+1} = g(x_k, x_{k-1})$, deci necesită o iniţializare dublă x_0, x_1 (ex: $x_0 = b, x_1 = a$).



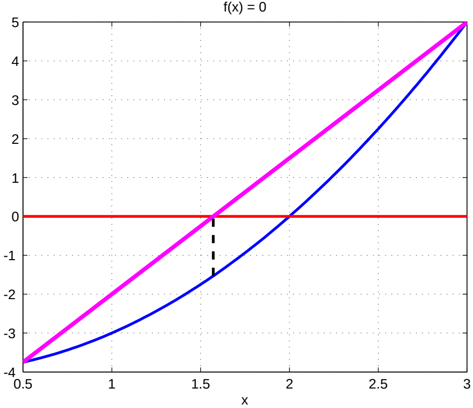
Metoda secantelor

Obs: funcţia de iteraţie are două variabile $x_{k+1} = g(x_k, x_{k-1})$, deci necesită o iniţializare dublă x_0, x_1 (ex: $x_0 = b, x_1 = a$).



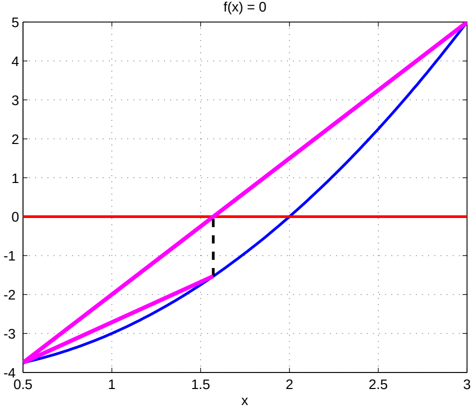
Metoda secantelor

Obs: funcţia de iteraţie are două variabile $x_{k+1} = g(x_k, x_{k-1})$, deci necesită o iniţializare dublă x_0, x_1 (ex: $x_0 = b, x_1 = a$).



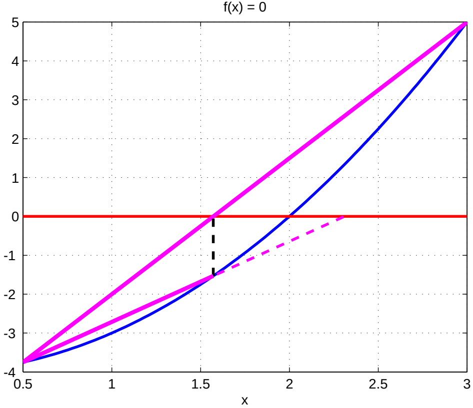
Metoda secantelor

Obs: funcţia de iteraţie are două variabile $x_{k+1} = g(x_k, x_{k-1})$, deci necesită o iniţializare dublă x_0, x_1 (ex: $x_0 = b, x_1 = a$).



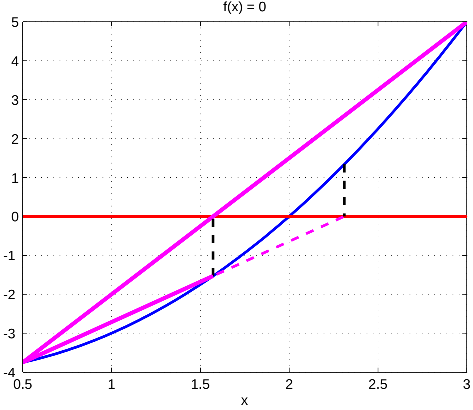
Metoda secantelor

Obs: funcţia de iteraţie are două variabile $x_{k+1} = g(x_k, x_{k-1})$, deci necesită o iniţializare dublă x_0, x_1 (ex: $x_0 = b, x_1 = a$).



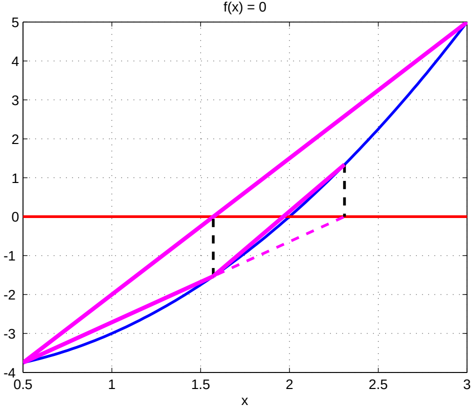
Metoda secantelor

Obs: funcţia de iteraţie are două variabile $x_{k+1} = g(x_k, x_{k-1})$, deci necesită o iniţializare dublă x_0, x_1 (ex: $x_0 = b, x_1 = a$).



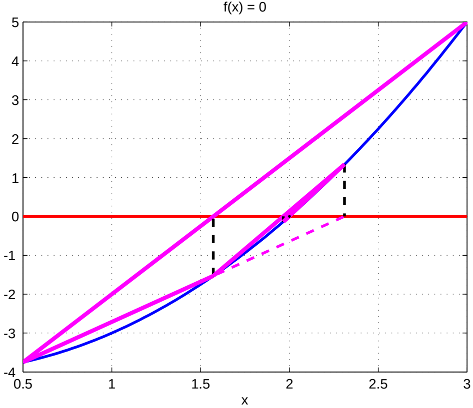
Metoda secantelor

Obs: funcţia de iteraţie are două variabile $x_{k+1} = g(x_k, x_{k-1})$, deci necesită o iniţializare dublă x_0, x_1 (ex: $x_0 = b, x_1 = a$).



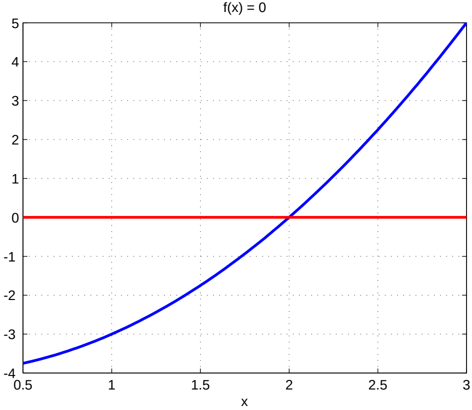
Metoda secantelor

Obs: funcţia de iteraţie are două variabile $x_{k+1} = g(x_k, x_{k-1})$, deci necesită o iniţializare dublă x_0, x_1 (ex: $x_0 = b, x_1 = a$).



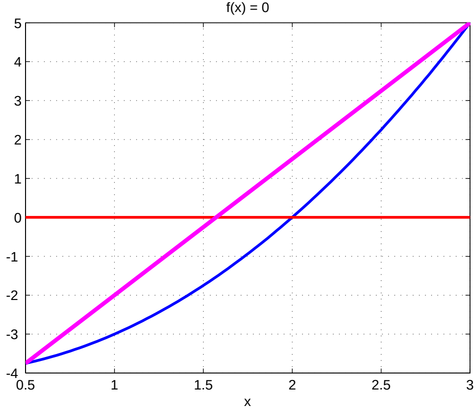
Metoda secantelor

Obs: dacă: $x_0 = a, x_1 = b$.



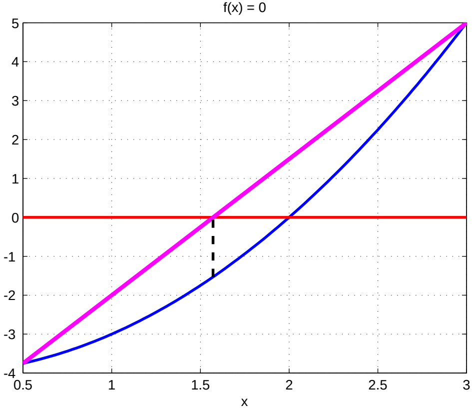
Metoda secantelor

Obs: dacă: $x_0 = a, x_1 = b$.



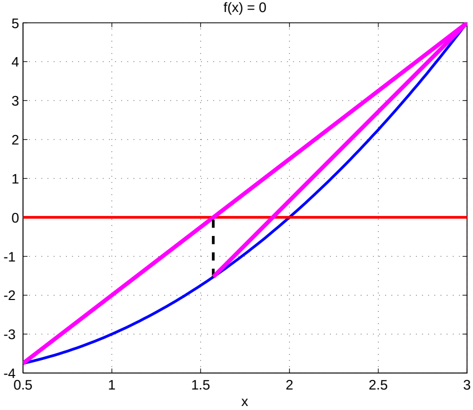
Metoda secantelor

Obs: dacă: $x_0 = a, x_1 = b$.



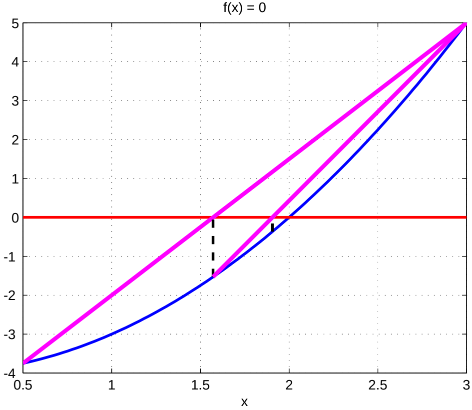
Metoda secantelor

Obs: dacă: $x_0 = a, x_1 = b$.



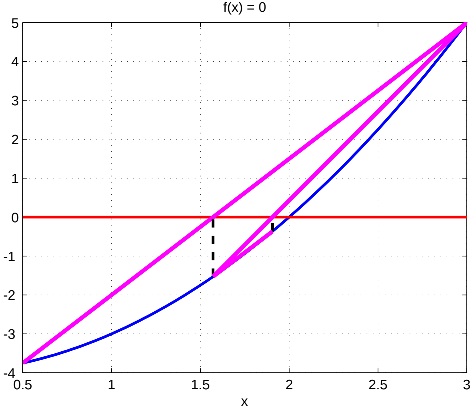
Metoda secantelor

Obs: dacă: $x_0 = a, x_1 = b$.



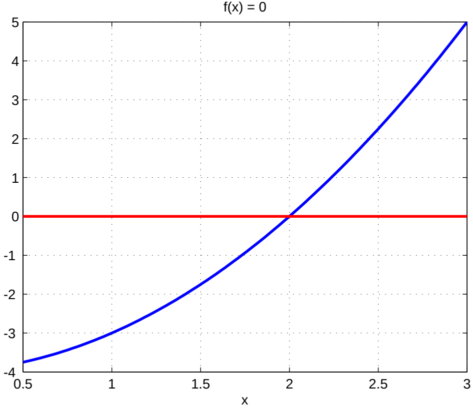
Metoda secantelor

Obs: dacă: $x_0 = a, x_1 = b$.



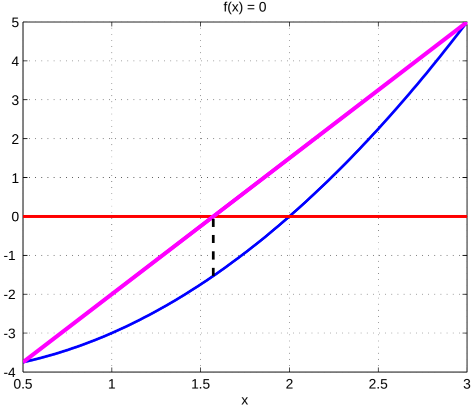
Metoda secantelor

Obs: variantă modificată (pentru a preveni divergenţa), se alege secanta corespunzătoare unei schimbări a semnului.



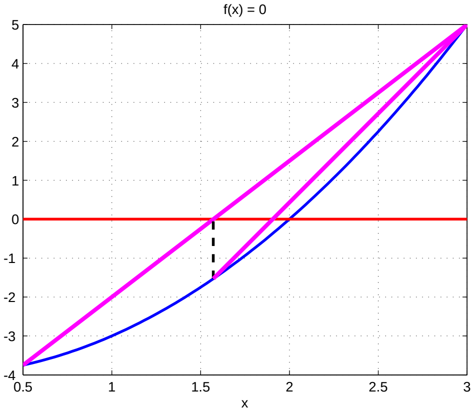
Metoda secantelor

Obs: variantă modificată (pentru a preveni divergenţa), se alege secanta corespunzătoare unei schimbări a semnului.



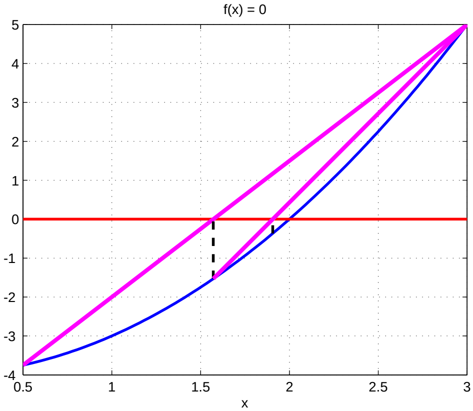
Metoda secantelor

Obs: variantă modificată (pentru a preveni divergenţa), se alege secanta corespunzătoare unei schimbări a semnului.



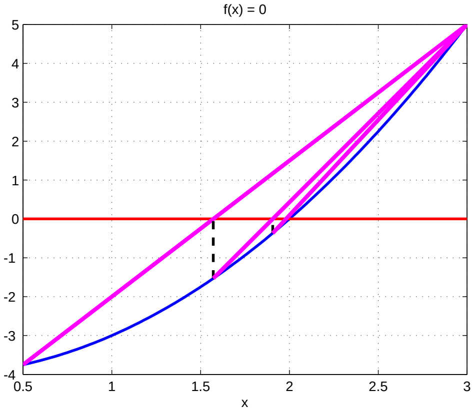
Metoda secantelor

Obs: variantă modificată (pentru a preveni divergenţa), se alege secanta corespunzătoare unei schimbări a semnului.



Metoda secantelor

Obs: variantă modificată (pentru a preveni divergenţa), se alege secanta corespunzătoare unei schimbări a semnului.



Algoritmi

procedura iteraţie simplă (x_0, eps, nit)
real x_0 ; iniţializare soluţie
real eps ; eroarea impusă
întreg nit ; număr maxim de iteraţii
întreg $k = 0$; contor iteraţii
real $xvechi = x_0$; iniţializarea soluţiei
repetă
    $k = k + 1$
    $xnou = g(xvechi)$; unde g(x)=x+cf(x)
    $d = |xnou - xvechi|$
    $xvechi = xnou$
până când $d < eps$ **sau** $k > nit$
dacă $k \leq nit$
    scrie $xnou$
retur

Algoritmi

procedura Newton (x_0, eps, nit)
real x_0 ; iniţializare soluţie
real eps ; eroarea impusă
întreg nit ; număr maxim de iteraţii
întreg $k = 0$; contor iteraţii
real $xvechi = x_0$; iniţializarea soluţiei
repetă
    $k = k + 1$
    $xnou = xvechi - f(xvechi)/fder(xvechi)$
    $d = |xnou - xvechi|$
    $xvechi = xnou$
până când $d < eps$ **sau** $k > nit$
dacă $k \leq nit$
    scrie $xnou$
retur

Algoritmi

procedura tangente paralele (x_0, eps, nit)
real x_0 ; iniţializare soluţie
real eps ; eroarea impusă
întreg nit ; număr maxim de iteraţii
real $xvechi = x_0$; iniţializarea soluţiei
real $fd = fder(x_0)$; valoarea derivatei în x_0
repetă
    $k = k + 1$
    $xnou = xvechi - f(xvechi)/fd$
    $d = |xnou - xvechi|$
    $xvechi = xnou$
până când $d < eps$ **sau** $k > nit$
dacă $k \leq nit$
    scrie $xnou$
retur

Algoritmi

procedura secante (a, b, eps, nit)
real a, b ; domeniul de definiţie al funcţiei
real eps ; eroarea impusă
întreg nit ; număr maxim de iteraţii
întreg $k = 0$; contor iteraţii
real $xv = a$; iniţializări ale soluţiei
real $xvv = b$
repetă
  $k = k + 1$
  $xnou = xv - (xv - xvv)f(xv)/(f(xv) - f(xvv))$
  $d = |xnou - xv|$
  $xvv = xv$
  $xv = xnou$
până când $d < eps$ **sau** $k > nit$
dacă $k \leq nit$
  scrie $xnou$

Comparaţie - efortul de calcul

- Depinde de eroarea impusă soluţiei.
- Efortul pentru o iteraţie depinde de metodă.
- Operaţiile de referinţă: evaluarea funcţiei f sau a derivatei acesteia.

Metoda	Număr de evaluări pe iteraţie
Bisecţiei	2 pentru f (poate fi redusă la o evaluare)
Falsei poziţii	2 pentru f (poate fi redusă la o evaluare)
Muller	3 pentru f (poate fi redusă la o evaluare)
Interpolarea pătratică inversă	3 pentru f (poate fi redusă la o evaluare)
Iteraţia simplă	1 pentru f
Tangente paralele	1 pentru f
Newton	1 pentru f şi 1 pentru f'
Secante	2 pentru f (poate fi redusă la o evaluare)

Comparaţie - convergenţă

Bisecţia

- garantat convergentă în ipoteza schimbării semnului;
- deoarece[4] $a_k = 1/2 a_{k-1}$ se spune că are convergenţă liniară.

Metoda falsei poziţii

- garantat convergentă în ipoteza schimbării semnului;
- convegenţă liniară, poate converge mai repede decât metoda bisecţiei pentru că alegerea punctului care împarte intervalul depinde de valorile funcţiei.

[4] a_k = marginea erorii absolute - revedeţi cursul despre erori

Comparaţie - convergenţă

Metodele bazate pe iteraţii

- nu sunt garantat convergente;
- viteza de convergenţă diferă de la o metodă la alta;
- metoda Newton e cea mai rapid convergentă, are convergenţă pătratică (demo pe slide-ul următor).
- metoda secantelor are o viteză de convergenţă între cea liniară şi cea pătratia ("superliniară"): $a_k \approx C a_{k-1}^{\alpha}$, $\alpha = (1+\sqrt{5})/2 \approx 1.61$. [Cheney]
- metoda Muller are o viteză de convergenţă superliniară, între secante şi Newton: $\alpha \approx 1.84$.
- metoda interpolării pătratice inverse are o viteză de convergenţă superliniară, între secante şi Newton: $\alpha \approx 1.8$.

Metoda Newton ar putea avea o eficienţă globală superioară, chiar dacă la fiecare iteraţie timpul de calcul este mai mare.

Convergenţa metodei Newton - demonstraţie

Funcţia de iteraţie $g(x) = x - \frac{f(x)}{f'(x)}$
Se verifică uşor că $g'(x^*) = 0$

$$g(x) = g(x^*) + (x - x^*)g'(x^*) + (x - x^*)^2 g''(\zeta)/2$$

Convergenţa metodei Newton - demonstraţie

Funcţia de iteraţie $g(x) = x - \frac{f(x)}{f'(x)}$
Se verifică uşor că $g'(x^*) = 0$

$$g(x) = g(x^*) + (x - x^*)g'(x^*) + (x - x^*)^2 g''(\zeta)/2$$

$$g(x) = g(x^*) + (x - x^*)^2 g''(\zeta)/2$$

Convergenţa metodei Newton - demonstraţie

Funcţia de iteraţie $g(x) = x - \frac{f(x)}{f'(x)}$
Se verifică uşor că $g'(x^*) = 0$

$$g(x) = g(x^*) + (x - x^*)g'(x^*) + (x - x^*)^2 g''(\zeta)/2$$

$$g(x) = g(x^*) + (x - x^*)^2 g''(\zeta)/2$$

$$g(x_{k-1}) = g(x^*) + (x_{k-1} - x^*)^2 g''(\zeta)/2$$

Convergenţa metodei Newton - demonstraţie

Funcţia de iteraţie $g(x) = x - \frac{f(x)}{f'(x)}$
Se verifică uşor că $g'(x^*) = 0$

$$g(x) = g(x^*) + (x - x^*)g'(x^*) + (x - x^*)^2 g''(\zeta)/2$$

$$g(x) = g(x^*) + (x - x^*)^2 g''(\zeta)/2$$

$$g(x_{k-1}) = g(x^*) + (x_{k-1} - x^*)^2 g''(\zeta)/2$$

$$x_k = x^* + (x_{k-1} - x^*)^2 g''(\zeta)/2$$

Convergenţa metodei Newton - demonstraţie

Funcţia de iteraţie $g(x) = x - \frac{f(x)}{f'(x)}$
Se verifică uşor că $g'(x^*) = 0$

$$g(x) = g(x^*) + (x - x^*)g'(x^*) + (x - x^*)^2 g''(\zeta)/2$$

$$g(x) = g(x^*) + (x - x^*)^2 g''(\zeta)/2$$

$$g(x_{k-1}) = g(x^*) + (x_{k-1} - x^*)^2 g''(\zeta)/2$$

$$x_k = x^* + (x_{k-1} - x^*)^2 g''(\zeta)/2$$

$$|x_k - x^*| \leq M|(x_{k-1} - x^*)|^2 \qquad (19)$$

Convergenţa metodei Newton - demonstraţie

Funcţia de iteraţie $g(x) = x - \frac{f(x)}{f'(x)}$
Se verifică uşor că $g'(x^*) = 0$

$$g(x) = g(x^*) + (x - x^*)g'(x^*) + (x - x^*)^2 g''(\zeta)/2$$

$$g(x) = g(x^*) + (x - x^*)^2 g''(\zeta)/2$$

$$g(x_{k-1}) = g(x^*) + (x_{k-1} - x^*)^2 g''(\zeta)/2$$

$$x_k = x^* + (x_{k-1} - x^*)^2 g''(\zeta)/2$$

$$|x_k - x^*| \leq M|(x_{k-1} - x^*)|^2 \qquad (19)$$

q.e.d.

Metode hibride

Metoda Brent-Dekker

- Combină 3 metode: bisecţia, secantelor şi interpolarea pătratică inversă;
- Are robusteţea dată de bisecţie dar poate fi rapid convergentă ca metoda secantelor sau interpolarea pătratică inversă.

Pentru detalii consultaţi
https://en.wikipedia.org/wiki/Brent%27s_method
Această metodă este implementată în funcţia `fzero` din Matlab.

Metode hibride

Cea mai rapid convergentă metodă este metoda Newton, dar ea necesită:

1. o iniţializare in interiorul razei de convergenţă (suficient de aproape de soluţie);
2. o expresie pentru evaluarea derivatei.

Presupunând că există o expresie care permite evaluarea derivatei, o altă idee de a combina metodele prezentate este de a folosi la început un algoritm care nu necesită evaluarea derivatei (metode de ordin zero), urmând a comuta în final pe iteraţii Newton (metode de ordinul unu).

Referinţe

- [Ioan98] D. Ioan et al., *Metode numerice în ingineria electrică*, Ed. Matrix Rom, Bucuresti, 1998. (Capitolul 16)
- [Cheney] Ward Cheney and David Kincaid, *Numerical Mathematics and Computing*, Brooks/Cole publishing Company,2000.
- [Heath] Michael Heath, *Scientific computing. An Introductory Survey*, McGraw Hill 2002 (capitolul 5 din carte) şi alte resurse de la

  disponibilă la http://heath.cs.illinois.edu/scicomp/