## 2. Het benaderen van nulpunten

### 2.1 Benaderen van vierkantswortels

Als we met een numerieke rekenmachine $\sqrt{7}$ berekenen, krijgen we als resultaat een benadering, $\sqrt{7} = 2,645751311$.

Het numeriek benaderen kan met een recursieve rij opgebouwd aan de hand van een iteratief proces.

Neem een startwaarde $x_0 > 0$ als eerste benadering voor $\sqrt{7}$.

Dan geldt dat $x_0 < \sqrt{7}$ ofwel dat $x_0 > \sqrt{7}$. Veronderstel dat $x_0 < \sqrt{7}$.

Bovendien geldt $\sqrt{7}x_0 < 7 \Leftrightarrow \sqrt{7} < \dfrac{7}{x_0}$ zodat $x_0 < \sqrt{7} < \dfrac{7}{x_0}$.

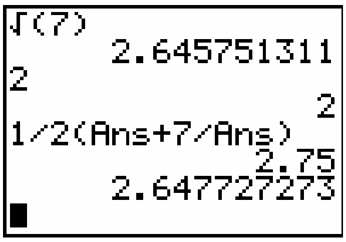
Als tweede benadering nemen we het midden van het interval $\left[x_0, \dfrac{7}{x_0}\right]$, $x_1 = \dfrac{1}{2}(x_0 + \dfrac{7}{x_0})$.

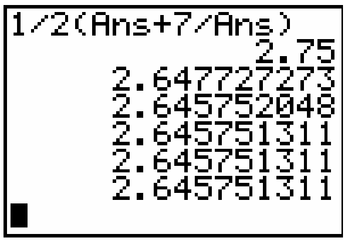
Deze werkwijze verderzettend bekomen we een rij die convergeert naar $\sqrt{7}$:

$$x_0, x_1 = \frac{1}{2}(x_0 + \frac{7}{x_0}), x_2 = \frac{1}{2}(x_1 + \frac{7}{x_1}), x_3 = \frac{1}{2}(x_2 + \frac{7}{x_2}), \ldots\ldots$$

#1: $\text{ITERATES}\left(\dfrac{1}{2}\cdot\left(x + \dfrac{7}{x}\right), x, 2, 5\right)$

#2: [2, 2.75, 2.647727272, 2.645752048, 2.645751311, 2.645751311]




De bovenstaande rij is gebaseerd op de iteratiefunctie $F(x) = \dfrac{1}{2}(x + \dfrac{7}{x})$ $(x > 0)$.

De vaste punten van $F$ zijn: $F(x) = \dfrac{1}{2}(x + \dfrac{7}{x}) = x \Leftrightarrow \dfrac{7}{x} = x \Leftrightarrow x^2 = 7 \Leftrightarrow x = \sqrt{7}$.

En $F'(x) = \dfrac{1}{2}(1 - \dfrac{7}{x^2})$ zodat $-1 < F'(\sqrt{7}) = 0 < 1$.

Dit verklaart de convergentie naar $\sqrt{7}$.

## 2.2 Numeriek bepalen van nulpunten

### 2.2.1 De bisectiemethode

De tussenwaardestelling vertelt ons dat voor een continue functie $f$ op $[x_0, x_1]$ met $f(x_0)f(x_1) < 0$ er een wortel of nulpunt bestaat in het interval $[x_0, x_1]$.

Voor het berekenen van de wortel kunnen we het volgende algoritme gebruiken:

Voor I gaande van 1 tot $n$ doe:

$$\frac{1}{2}(x_0 + x_1) = x_0 + \frac{x_1 - x_0}{2} \to x$$

Als $f(x_0)f(x) < 0$ dan $x \to x_1$ en als $f(x_0)f(x) > 0$ dan $x \to x_0$

Voor $n = 20$ vinden we de volgende numerieke benadering voor het nulpunt van $f(x) = x^2 - 7$, een benadering van $\sqrt{7}$.

| n | X0 | X | X1 | f(X0) | f(X) | f(X1) |
|---|---|---|---|---|---|---|
| 2 | 1 | 2.5 | 4 | -6 | -0.75 | 9 |
| 3 | 2.5 | 3.25 | 4 | -0.75 | 3.5625 | 9 |
| 4 | 2.5 | 2.875 | 3.25 | -0.75 | 1.265625 | 3.5625 |
| 5 | 2.5 | 2.6875 | 2.875 | -0.75 | 0.222656 | 1.265625 |
| 6 | 2.5 | 2.59375 | 2.6875 | -0.75 | -0.272461 | 0.222656 |
| 7 | 2.59375 | 2.640625 | 2.6875 | -0.272461 | -0.0271 | 0.222656 |
| 8 | 2.640625 | 2.664063 | 2.6875 | -0.0271 | 0.097229 | 0.222656 |
| 9 | 2.640625 | 2.652344 | 2.664063 | -0.0271 | 0.034927 | 0.097229 |
| 10 | 2.640625 | 2.646484 | 2.652344 | -0.0271 | 0.00388 | 0.034927 |
| 11 | 2.640625 | 2.643555 | 2.646484 | -0.0271 | -0.011619 | 0.00388 |
| 12 | 2.643555 | 2.64502 | 2.646484 | -0.011619 | -0.003872 | 0.00388 |
| 13 | 2.64502 | 2.645752 | 2.646484 | -0.003872 | 3.4E-06 | 0.00388 |
| 14 | 2.64502 | 2.645386 | 2.645752 | -0.003872 | -0.001934 | 3.4E-06 |
| 15 | 2.645386 | 2.645569 | 2.645752 | -0.001934 | -0.000965 | 3.4E-06 |
| 16 | 2.645569 | 2.64566 | 2.645752 | -0.000965 | -0.000481 | 3.4E-06 |
| 17 | 2.64566 | 2.645706 | 2.645752 | -0.000481 | -0.000239 | 3.4E-06 |
| 18 | 2.645706 | 2.645729 | 2.645752 | -0.000239 | -0.000118 | 3.4E-06 |
| 19 | 2.645729 | 2.645741 | 2.645752 | -0.000118 | -5.72E-05 | 3.4E-06 |
| 20 | 2.645741 | 2.645746 | 2.645752 | -5.72E-05 | -2.69E-05 | 3.4E-06 |

Uit de constructie van de bisectiemethode volgt dat ze altijd convergeert naar de wortel $\alpha$, al dan niet altijd even snel. Het algoritme van de bisectiemethode genereert een rij $x_0, x_1, x_2, x_3, \ldots$ die voldoet aan

$$|x_2 - \alpha| < \frac{x_1 - x_0}{2}$$

$$|x_3 - \alpha| < \frac{x_1 - x_0}{2^2} \qquad \text{en} \quad \frac{x_1 - x_0}{2^n} \to 0 \text{ voor } n \to +\infty.$$

$$\vdots$$

$$|x_n - \alpha| < \frac{x_1 - x_0}{2^n}$$

### 2.2.2 De regula falsi

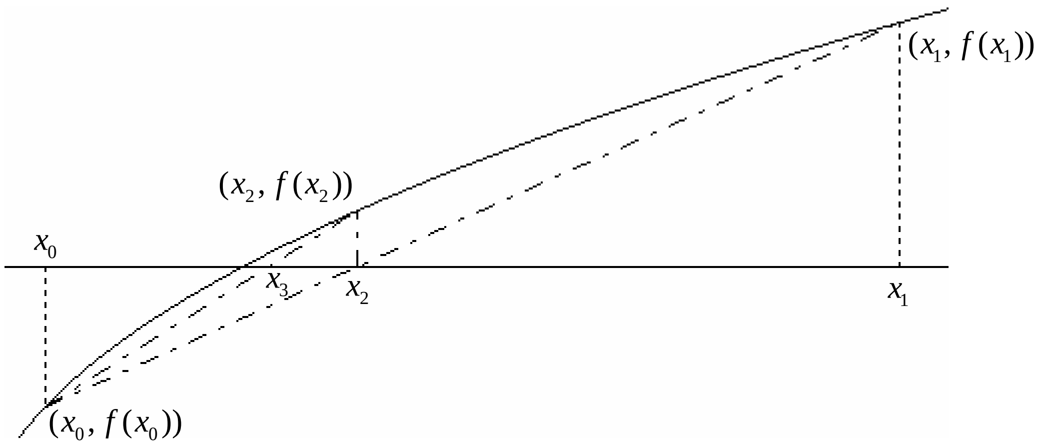
De bisectiemethode houdt enkel rekening met het teken van $f$. De regula falsi voegt daar de functiewaarde als volgt aan toe.

Vertrekkende van $x_0$ en $x_1$ bepalen we $x_2$ als het snijpunt van de rechte door de punten $(x_0, f(x_0))$ en $(x_1, f(x_1))$.

Deze rechte heeft als vergelijking $y - f(x_0) = \dfrac{f(x_1) - f(x_0)}{x_1 - x_0}(x - x_0)$.

Het snjipunt met de $x$-as is

$$x = \frac{x_1 - x_0}{f(x_0) - f(x_1)} f(x_0) + x_0 = \frac{f(x_0)(x_1 - x_0) + x_0(f(x_0) - f(x_1))}{f(x_0) - f(x_1)} = \frac{x_1 f(x_0) - x_0 f(x_1)}{f(x_0) - f(x_1)}.$$




In een volgende stap kijken we naar het teken van $f(x_2)$.

Als $f(x_0)f(x_2) < 0$ herhalen we het voorgaande voor $(x_0, f(x_0))$ en $(x_2, f(x_2))$ en als $f(x_0)f(x_2) > 0$ voor $(x_1, f(x_1))$ en $(x_2, f(x_2))$ en bekomen zo $x_3$.

We generenen een recursieve rij met de iteratiefunctie $F(x, y) = \dfrac{x\,f(y) - y\,f(x)}{f(y) - f(x)}$:

$$x_i = F(x_{i-1}, x_{i-2}) = \frac{x_{i-1}\,f(x_{i-2}) - x_{i-2}\,f(x_{i-1})}{f(x_{i-2}) - f(x_{i-1})}.$$

Een gelijkaardig algoritme als voor de bisectiemethode kan gebruikt worden voor het programmeren van de regula falsi vertrekkende van $x_0$ en $x_1$ met $f(x_0)f(x_1) < 0$:

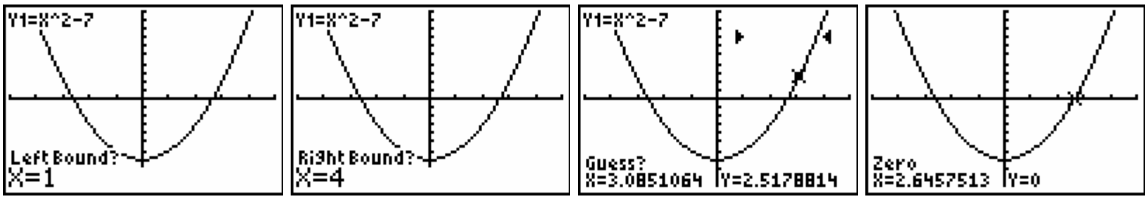
Voor I gaande van 1 tot $n$ doe:

$F(x_1, x_0) \to x$

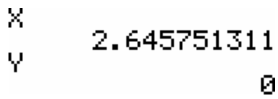
Als $f(x_0)f(x) < 0$ dan $x \to x_1$ en als $f(x_0)f(x) > 0$ dan $x \to x_0$

Gelijkaardige algoritmen worden gebruikt om nulpunten te bepalen met een grafische rekenmachine.

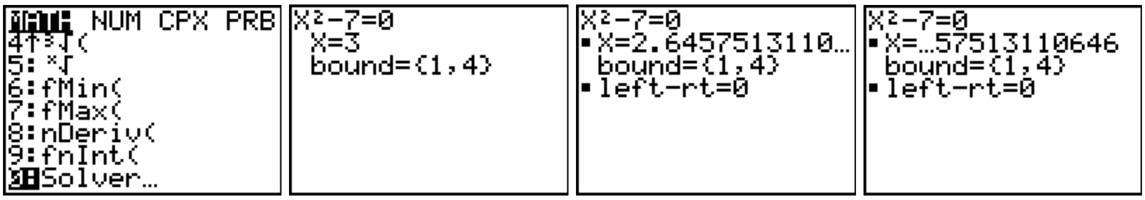
<u>GRAFISCH</u>




De berekende X- en Y-waarden kunnen via het rekenscherm nauwkeuriger bekeken worden.




<u>NUMERIEK SOLVER</u>

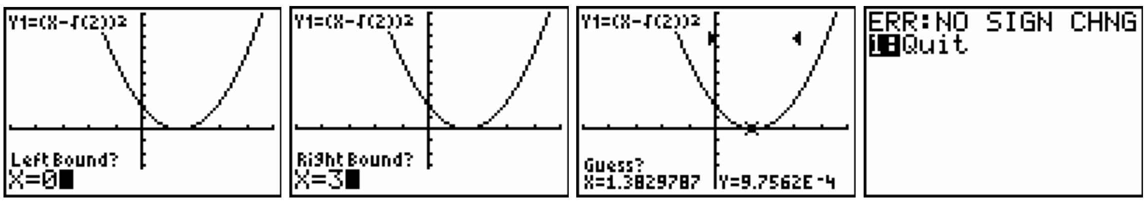



Voor left-rt wordt voor het gevonden resultaat, X, het verschil berekend tussen het linkerlid en rechterlid van de vergelijking voor X.

Men kan aantonen dat voor een continue functie $f$ op $[x_0, x_1]$ met $f(x_0)f(x_1) < 0$ de regula falsi convergeert.

### 2.2.3 De methode van Newton-Raphson

Voor de twee voorgaande methodes staat de voorwaarde $f(x_0)f(x_1) < 0$ centraal. Dit heeft tot gevolg dat een eenvoudige vergelijking zoals $(x-\sqrt{2})^2 = 0$ niet numeriek op te lossen is, als we veronderstellen dat we de oplossing $\sqrt{2}$ niet kennen en niet gebruiken als startwaarde.




De methode van Newton is een numerieke benaderingmethode van nulpunten die gebruik maakt van de raaklijn aan de grafiek van de functie. Afleidbaarheid is noodzakelijk voor het gebruiken van deze methode.

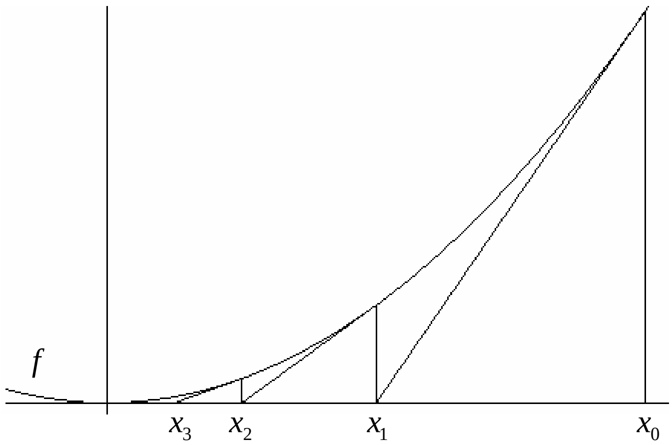
Als startwaarde bepalen we grafisch een $x_0$ in de buurt van een nulpunt.

In een volgende stap berekenen we het snijpunt van de raaklijn aan de grafiek in het punt $(x_0, f(x_0))$ en de $x$-as.

De *x*-coördinaat van dit snijpunt is de volgende benadering, $x_1$, van het nulpunt.

We herhalen deze werkwijze voor $x_1$ en bekomen zo het punt $x_2$ als snijpunt van de raaklijn in het punt $(x_1, f(x_1))$ en de $x$-as.

Het steeds verder zetten van dit proces genereert een rij $x_0, x_1, x_2, x_3, x_4, \ldots$ die voor heel wat functies convergeert. De exacte voorwaarde voor convergentie behandelen we hier niet.




We bepalen de iteratiefunctie die aan de basis ligt van de methode van Newton-Raphson.

De vergelijking van de raaklijn aan de grafiek van $f$ in $(x_n, f(x_n))$ is van de vorm:

$$y - f(x_n) = f'(x_n)(x - x_n) \Leftrightarrow y = f(x_n) + f'(x_n)(x - x_n).$$

Voor het snijpunt van de raaklijn met de $x$-as geldt $y = 0$ zodat $x = x_n - \dfrac{f(x_n)}{f'(x_n)}$.

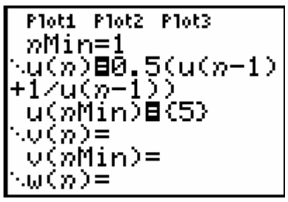
Deze $x$-waarde is de volgende stap in het iteratieproces, $x_{n+1}$.

De iteratiefunctie is $N(x) = x - \dfrac{f(x)}{f'(x)}$ en genereert de volgende rij:

$$x_0,\quad x_1 = x_0 - \frac{f(x_0)}{f'(x_0)},\quad x_2 = x_1 - \frac{f(x_1)}{f'(x_1)},\quad x_3 = x_2 - \frac{f(x_2)}{f'(x_2)},\quad x_4 = x_3 - \frac{f(x_3)}{f'(x_3)}, \ldots$$

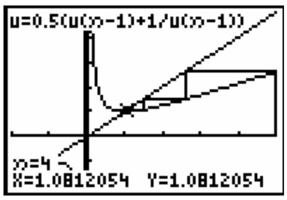
Dit iteratieproces noemt men de methode van Newton-Raphson

<u>VOORBEELD 1</u>




Voor $f(x) = x^2 - 1$ is $N(x) = x - \dfrac{x^2-1}{2x} = \dfrac{x^2+1}{2x} = \dfrac{1}{2}(x + \dfrac{1}{x})$.

Voor de vaste punten van $N(x)$ geldt $\dfrac{1}{x} = x \Leftrightarrow x^2 - 1 = 0$.




M.a.w. de vaste punten van $N(x)$ zijn de nulpunten van $f$.

Grafische analyse toont dat elk getal $(\neq 0)$ door iteratie met $N$ convergeert naar één van beide vaste punten. Ook geldt:

$$N'(x) = \frac{2x \cdot 2x - 2(x^2+1)}{4x^2} \quad \text{en} \quad N'(1) = N'(-1) = 0.$$

<u>VOORBEELD 2</u>

Voor $f(x)=x^2(x-1)$ geldt dat $N(x)=x-\dfrac{x^3-x^2}{3x^2-2x}=x-\dfrac{x^2-x}{3x-2}=\dfrac{2x^2-x}{3x-2}$.

Merk op dat ook hier de vaste punten van $N$ gelijk zijn aan de nulpunten van $f$.

#1: $N(x) := \dfrac{2\cdot x^2 - x}{3\cdot x - 2}$

#2: SOLVE(N(x) = x, x)

#3: $x = 1 \vee x = 0$

#4: $G(x) := \dfrac{d}{dx}\, N(x)$

#5: [G(0), G(1)]

#6: $\left[\dfrac{1}{2},\ 0\right]$

Ook voor voorbeeld 2 leidt de methode van Newton Rapshon tot convergentie naar het gewenste nulpunt bij goed gekozen startwaarden.

```
0.3
                 .3
(2*Ans²-Ans)/(3*
Ans-2)
       .1090909091
       .0509881423
       .0247902973
```

```
       .0247902973
       .0122355751
       .0060796605
       .0030305046
       .0015129458
    7.558993667E-4
    3.778066752E-4
```

```
.7
                 .7
(2*Ans²-Ans)/(3*
Ans-2)
                2.8
             2.0125
        1.507817337
```

```
        1.507817337
        1.204385458
        1.051790907
        1.004643174
        1.000042526
        1.000000004
                  1
```

De volgende voorbeelden tonen dat de methode van Newton niet altijd leidt tot een oplossing of tot een correcte oplossing.

<u>Voorbeeld 3</u>

De functie $f(x)=\sqrt[3]{x}$ is niet afleidbaar in 0.

Voor iedere $x\neq 0$ is $N(x)=x-\dfrac{\sqrt[3]{x}}{1/3\sqrt[3]{x^2}}=x-3x=-2x$.

Voor $N(x)=-2x$ geldt dat $N'(0)=-2$, wat laat aanvoelen waarom het fout gaat.

<u>Voorbeeld 4</u>

Het al dan niet convergeren is afhankelijk van de startwaarde.

Voor $f(x)=x^3-5x$ bepaalt $N(x)=x-\dfrac{x^3-5x}{3x^2-5}=\dfrac{2x^3}{3x^2-5}$ voor de startwaarde $x_0=1$ de, weliswaar afstotende, 2-cyclus $1\to -1\to 1\to -1\to ...$

#1: $N(x) := \dfrac{2\cdot x^3}{3\cdot x^2 - 5}$

#2: SOLVE(N(x) = x, x)

#3: $x = -\sqrt{5} \vee x = \sqrt{5} \vee x = 0$

#4: NN(x) := N(N(x))

#5: SOLVE(NN(x) = x, x, Real)

#6: $x = -\sqrt{5} \vee x = \sqrt{5} \vee x = -1 \vee x = 1 \vee x = 0$

#7: $D(x) := \dfrac{d}{dx}\, NN(x)$

#8: D(1)

#9: 36

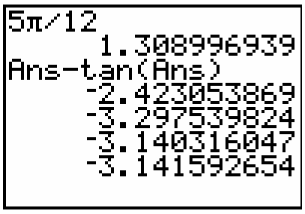
Voorbeeld 5

Het benaderen van nulpunt $x = \pi$ van $f(x) = \sin x$ met de methode van Newton-Raphson met als startwaarde $x_0 = \frac{5\pi}{12}$ geeft het volgende resultaat.

$$N(x) = x - \frac{\sin x}{\cos x} = x - \text{tg}(x)$$

$$N'(x) = 1 - \frac{1}{\cos^2 x}$$

$$N'(-\pi) = N'(\pi) = 0$$




Dit voorbeeld toont dat het belangrijk is de startwaarde dicht bij het te benaderen nulpunt te kiezen, hetgeen best grafisch kan gebeuren.

### 2.2.4 Het programmeren van de methode van Newton-Raphson

**(i) TI BASIC**

Voor een introductie over programmeren in TI BASIC verwijzen we naar het boekje *Programmeren met de TI-83 Plus* van Henk Pfaltzgraff.

Een programma kan men ruwweg opsplisten in drie onderdelen: de input, de verwerking en de output.

INPUT

De minimaal noodzakelijke input is het functievoorschrift, de startwaarde en het aantal iteratiestappen dat je wil zetten. We gebruiken hiervoor het commando `INPUT`.

*Input van het functievoorschrift*

Het functievoorschrift moet eerst ingegeven worden als een woord (string) en nadien omgezet worden in een vergelijking die we in de variabele $Y_1$ plaatsen.

```
Input "Y1=",Str1
String▸Equ(Str1,Y1)
```

*Input van de startwaarde en het aantal iteratiestappen*

```
Input "X0",A
Input "N=",N
```

VERWERKING

Voor het uitvoeren van de N iteratiestappen gebruiken we een `For`-lus. Een `For`-lus moet altijd afgesloten worden met een `End`-commando.

```
For(I,1,N)
A→B
nDeriv(Y1,X,B)→C
A-(Y1(A)/C)→A
End
```

OUTPUT

Na het uitvoeren van de `For`-lus bevindt de gewenste benadering zich in de variabele A. Om de inhoud van de variable A te tonen gebruiken we het `Disp`-commando

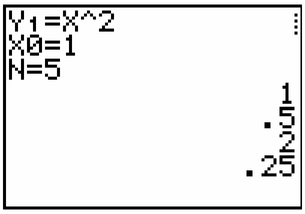
```
Disp A
```

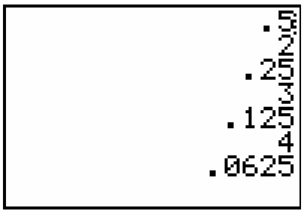
De input en output gebeurt via het rekenscherm. Om met een proper rekenscherm te starten, laten we het programma beginnen met `ClrHome`.

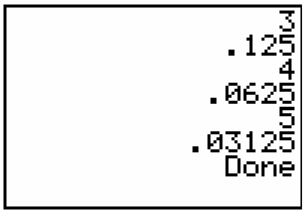
Als output verschijnt er slecht het resultaat van de laaste iteratiestap. Het is ook interessant om de iteratie stap voor stap te bekijken.

Dit kan door voor de `End` de volgende commando's, `Disp I,A:Pause,` in te voegen en de laatste regel weg te laten.

Het `Pause`-commando zorgt er voor dat iedere iteratiestap getoond wordt. Het drukken op `ENTER` genereert telkens de volgende iteratiestop tot N bereikt is.








Naar eigen wens kan het programma uitgebreid worden. Na de lus kan bijvoorbeeld een menu toegevoegd worden dat toelaat het functievoorschrift, de startwaarde of het aantal iteratiestappen te wijzigen.

```
Menu("OPTIES", "FUNCTIE",A,"STARTWAARDE",B,"ITERATIE",C,"STOP",D)
```

Het gebruik van labels is hier noodzakelijk.

```
ClrHome
Input "Y1=",Str1
String▸Equ(Str1,Y1)
Input "X0",A
A→P
Input "N=",N
Lbl E
For(I,1,N)
A→B
nDeriv(Y1,X,B)→C
A-(Y1(A)/C)→A
Disp I,A
Pause
End
Menu("OPTIES", "FUNCTIE",A,"STARTWAARDE",B,"ITERATIE",C,"STOP",D)
Lbl A
ClrHome
Input "Y1=",Str1
String▸Equ(Str1,Y1)
P→A
Goto E
Lbl B
ClrHome
Input "X0=",A
Goto E
Lbl C
ClrHome
Input "N=",N
P→A
Goto E
LbL D
ClrHome
Disp I,A
```

Merk op dat om goed te functioneren er nog een test moet toegevoegd worden, nl. nagaan of dat steeds `nDeriv(Y1,X,B)` $\neq 0$. Het toevoegen van deze test laten we als oefening voor de lezer.

Vanzelfsprekend kunnen ook de bisectiemethode, de regula falsi ende methode van Euler geprogrammeerd worden in TI BASIC. Ook dit zijn goede oefeningen in het schrijven van algoritmen om het rekenwerk te laten uitvoeren door een machine.

### (ii) DERIVE

Met de commando's ITERATE en ITERATES is het vrij eenvoudig numeriek de methode van Newton-Raphson uit te voeren met DERIVE.

#1: $f(x) := e^{-x} - (x-1)^2$

#2: $N(x) := x - \dfrac{f(x)}{\frac{d}{dx} f(x)}$

#3: newton(a, n) := ITERATE(N(x), x, a, n)

#4: newtons(a, n) := ITERATES(N(x), x, a, n)

#5: newton(1, 3)

#6: 1.486519702

#7: newtons(1, 3)

#8: [1, 2, 1.595068407, 1.486519702]

Met het commando `newton` kan je als volgt een tabel aanmaken, inclusief een kolom die de iteratiestappen aangeeft.

#9: tab(a, n) := APPEND([[n, approx]], VECTOR([k, newton(a, k)], k, 0, n))

#10: tab(1, 7)

#11:

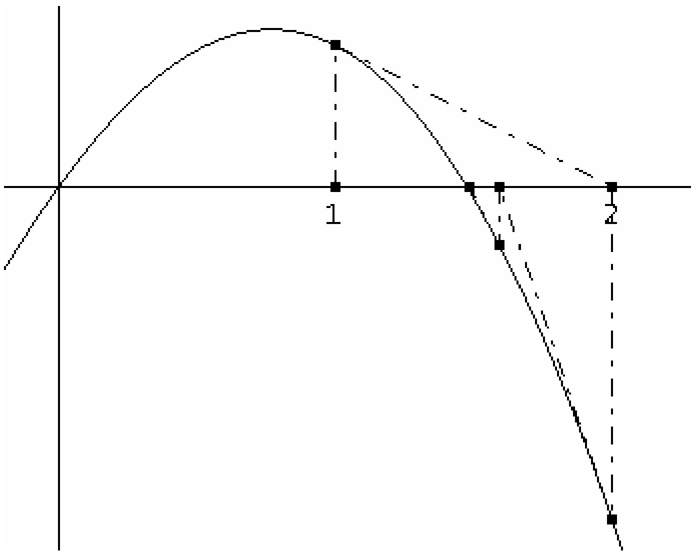
| n | approx |
|---|---|
| 0 | 1 |
| 1 | 2 |
| 2 | 1.595068407 |
| 3 | 1.486519702 |
| 4 | 1.477727962 |
| 5 | 1.477670064 |
| 6 | 1.477670062 |
| 7 | 1.477670062 |

Met de commando's `step` en `graf` kan de methode van Newton-Raphson grafisch voorgesteld worden.

```
                     ┌  a     0   ┐
#12:   step(a) :=    │  a    f(a) │
                     └ N(a)   0   ┘

       graf(a, n) := VECTOR(step((newtons(a, n)) ), k, 1, n)
#13:                                            k
```

De onderstaande tekening, `graf(1,3)`, geeft de eerste drie stappen in de methode van Newton-Raphson weer voor de functie $f(x) = e^{-x} - (x-1)^2$.




### 2.2.5 Vaste punten

Het bepalen van een nulpunt van een functie $f$ komt vaak overeen met het zoeken van een vast punt voor een goed gekozen functie $\varphi$.

Er geldt: als $\alpha$ een nulpunt is van $f$ is $\alpha$ een vast punt is van $\varphi(x) = x - f(x)g(x)$ met $g(x)$ een willekeurige functie en $g(\alpha) \neq 0$.

<u>VOORBEELD 1</u>

Het bepalen van een oplossing van $x^2 - 7 = 0$ met de bisectiemethode, een benadering voor $\sqrt{7}$, komt overeen met het zoeken van een vastpunt voor $\varphi(x) = x - f(x)g(x)$ met $g(x) = \dfrac{1}{2x}$, $\varphi(x) = x - f(x)g(x) = x - (x^2 - 7)\dfrac{1}{2x} = \dfrac{1}{2}x + \dfrac{7}{2x} = \dfrac{1}{2}(x + \dfrac{7}{x})$.

VOORBEELD 2

Het zoeken naar nulpunten van een functie $f$ met de methode van Newton-Raphson komt neer op het bepalen van de vaste punten van $\varphi(x) = x - f(x)g(x)$ met

$$g(x) = \frac{1}{f'(x)}.$$

Er geldt dat $f(x) = 0 \Leftrightarrow \varphi(x) = x$, hetgeen verklaart wat bleek uit de voorgaande voorbeelden, voor de methode van Newton-Raphson zijn de nulpunten van $f$ de vaste punten van $N$ en omgekeerd.