8. Yhtälöiden ratkaisuja Newtonilla, animaatioita

Käsitellään puhtaana Maple-työnä ja myös Maple-Matlab-yhteistyönä.

```
> restart
> with(plots):
> N := x->evalf(x - f(x)/D(f)(x));    # Tätä käytetään Maple-työssä
```

$$N := x \to evalf\left(x - \frac{f(x)}{D(f)(x)}\right) \quad (1.1)$$

```
> Nsymb := x->x - f(x)/D(f)(x);    # Tämä tarvitaan Maple-Matlab-yhteiskäyttöön.
```

$$Nsymb := x \to x - \frac{f(x)}{D(f)(x)} \quad (1.2)$$

```
> f := x->x*cos(x) - sin(x) - 1;
```

$$f := x \to x\cos(x) - \sin(x) - 1 \quad (1.3)$$

```
> Nsymb(x)
```

$$x + \frac{x\cos(x) - \sin(x) - 1}{x\sin(x)} \quad (1.4)$$

```
> lprint(Nsymb(x))
x+(x*cos(x)-sin(x)-1)/(x*sin(x))
```

1) Maple-Matlab:

Siirretään tämä kaava Matlab-ikkunaan (copy/paste)

Puhdas Matlab-työ vaatisi derivoimista ja N-kaavan sieventämistä ja kirjoittamista Matlabille. Sen kaiken hoitaa meille Maple.

Tehdään tässä Matlab-iterointi mahdollisimman yksinkertaisesti toistamalla komentoikkunassa nuoliylös-näppäimellä x=N(x)-riviä (tai editorissa CTR-ENTER). Toki voitaisiin hoitaa for-silmukalla tai vielä paremmin while-rakenteella. Newtonin menetelmä suppenee yleensä hyvin nopeasti tai sitten hajaantuu, joten iterointiaskeleita ei yleensä tarvita monta.

```
>> syms x
>> vectorize(x+(x*cos(x)-sin(x)-1)/(x*sin(x)))
%    vectorize tekee pisteet.

x+(cos(x).*x-sin(x)-1)./sin(x)./x

>> N=@(x) x+(cos(x).*x-sin(x)-1)./sin(x)./x

>> x=6;
>> format long
```

```
>> x=N(x)

x =
  11.299466443472243

>> x=N(x)

x =
  10.990110316129204
```

```
>> x=N(x)    % Toista n-nappaimella.

x =
   2.993463587028939
>> x=N(x)    % Tämä ampuu kauas. Vaihdetaan
alkupistettä.

>> x=6.5

x =
   6.500000000000000

>> x=N(x)

x =
  10.170723934666555

>> x=N(x)
```

```
>> x=N(x)

x =
  10.995575700223300
>> x=N(x)

x =
  11.299466443472243

>> x=N(x)

x =
  10.990110316129204

>> x=N(x)
```

2) Jatketaan nyt puhtaalla Maple-linjalla:

```
> fkuva := plot(f(x), x = 0..3·Pi, color = black);
```

$$fkuva := PLOT(\ldots) \tag{1.5}$$

```
> N(x);
```

$$x + \frac{x\cos(x) - 1.\sin(x) - 1.}{x\sin(x)} \tag{1.6}$$

```
> x[0] := Pi + .3;    # Otetaan toinen alkupiste kuin edellä.
```

$$x_0 := \pi + 0.3 \tag{1.7}$$

```
> for k from 1 to 5 do
    x[k] := N(x[k - 1])
  end do
```

$$x_1 := 7.366983641$$

$$x_2 := 7.607183306$$

$$x_3 := 7.592100585$$

$$x_4 := 7.592056182$$

$$x_5 := 7.592056182 \tag{1.8}$$

Tehtävässä ei pyydetty, mutta tehdään graafinen havainnollistus piirtämällä tangentteja sivuamispisteestä
x-akselin leikkauspisteeseen. Tässä on riemastuttavan kätevä mahdollisuus määritellä grafiikka-arvoinen funktio:

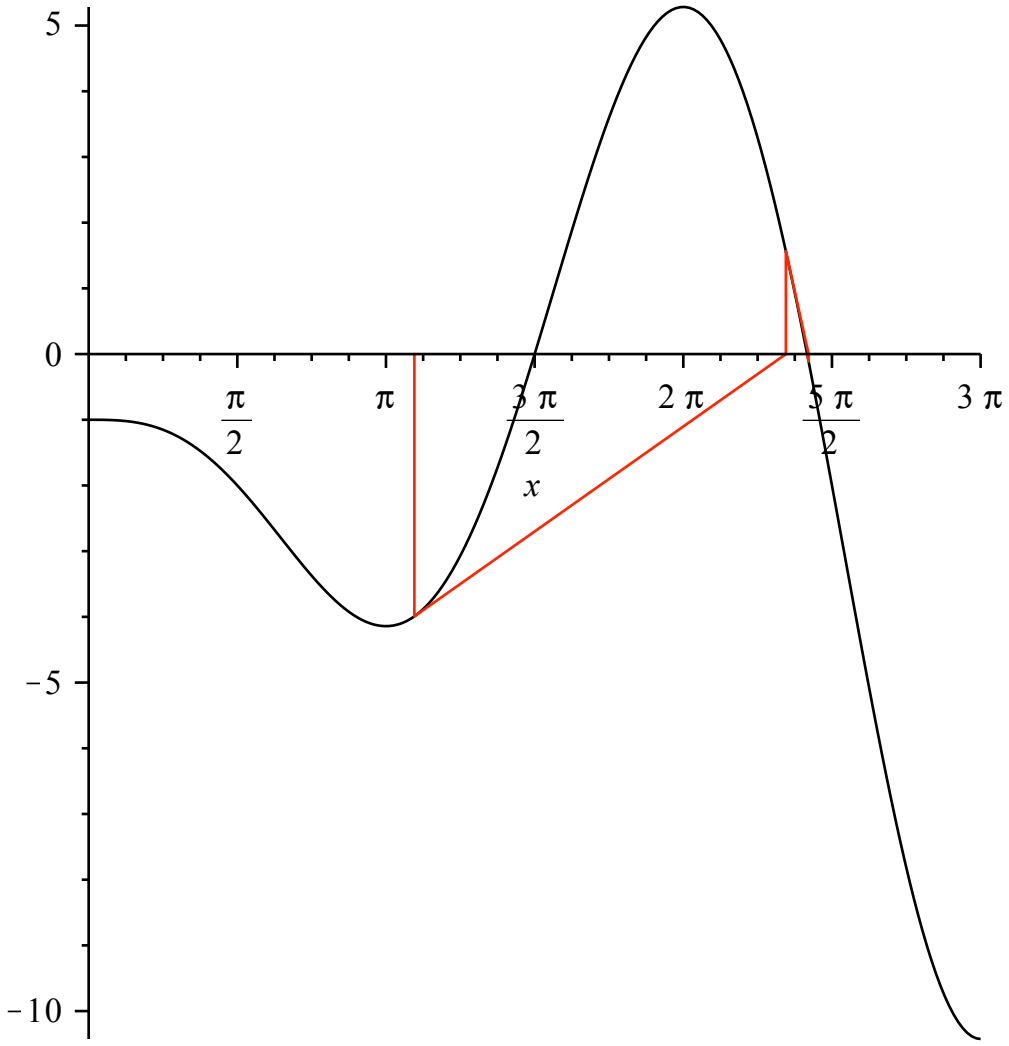
> $tangkuva := x0 \rightarrow plot([[x0, 0], [x0, f(x0)], [N(x0), 0]])$

$$tangkuva := x0 \rightarrow plot([[x0, 0], [x0, f(x0)], [N(x0), 0]]) \quad \textbf{(1.9)}$$

> $\#tang := (f, x0, x) \rightarrow f(x0) + D(f)(x0) \cdot (x - x0)$

display-komennolle annetaan jono näitä kuvia, kunkin kuvan "kantapiste" on ao. iteraatiojonon piste.

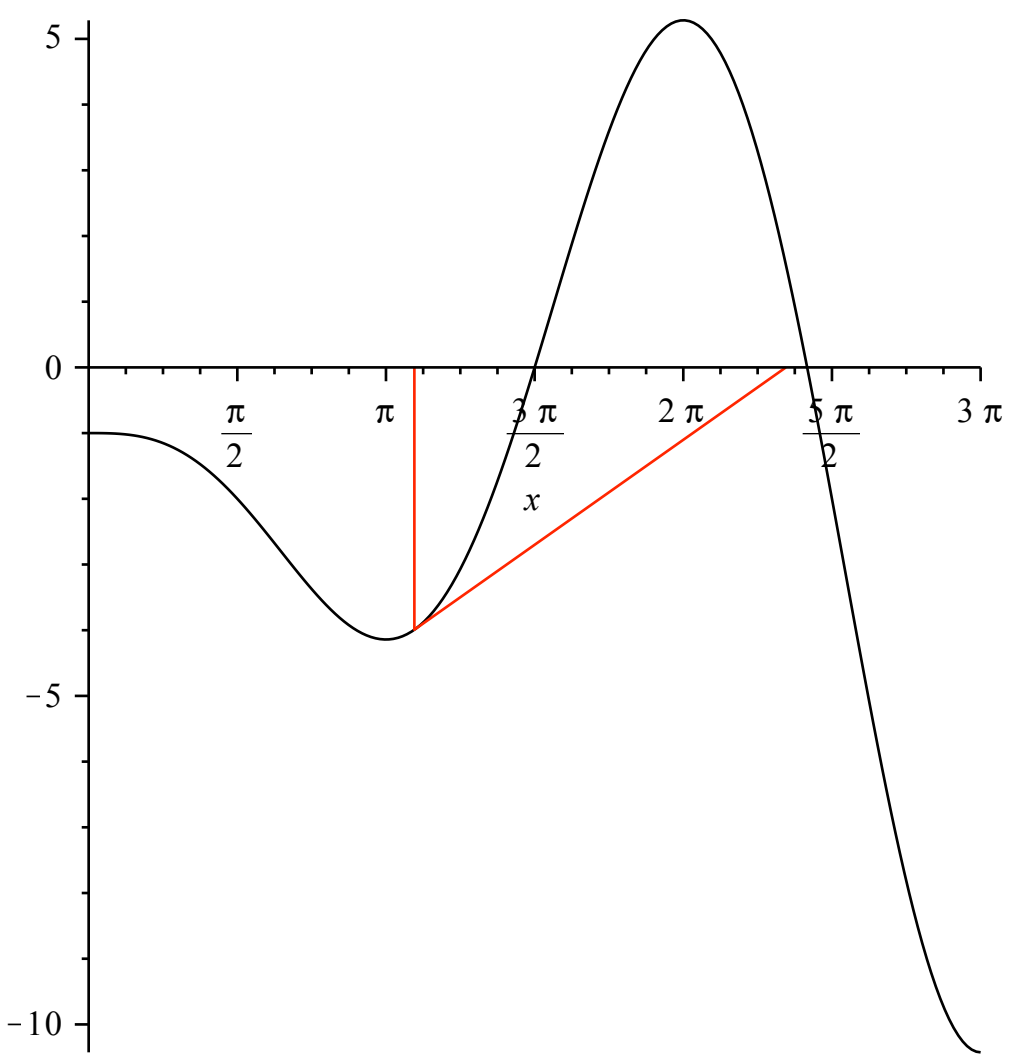
> $display(fkuva, seq(tangkuva(x[k-1]), k = 1..4))$



>

Selvemmin prosessi näkyy animaationa, se saadaan lisäämällä edelliseen valitsin insequence=true
Koska haluamme kuvaajan näkyvän jokaisessa animaatiokehyksessä, jouduimme lisäämään
display(fkuva,...)
muuten *fkuva* pyyhkiytyisi seuraavassa kehyksessä pois.

> $display(seq(display(fkuva, tangkuva(x[k-1])), k = 1..9), insequence = true)$



Klikkaamalla kuvaa, saat animaatiovalikon. Newton suppenee näillä arvoilla turhan nopeasti. Voi kokeilla muita
lähtöarvoja. Tässä on joka tapauksessa hyvin kätevä yleinen tekniikka, jolla erilaisia iteraatioita on tavattoman helppo
graafisesti havainnollistaa.

> $x[0] := 6.5$ *# Otetaan sama alkup. kuin Matlab;ssa*

$$x_0 := 6.5 \qquad \textbf{(1.10)}$$

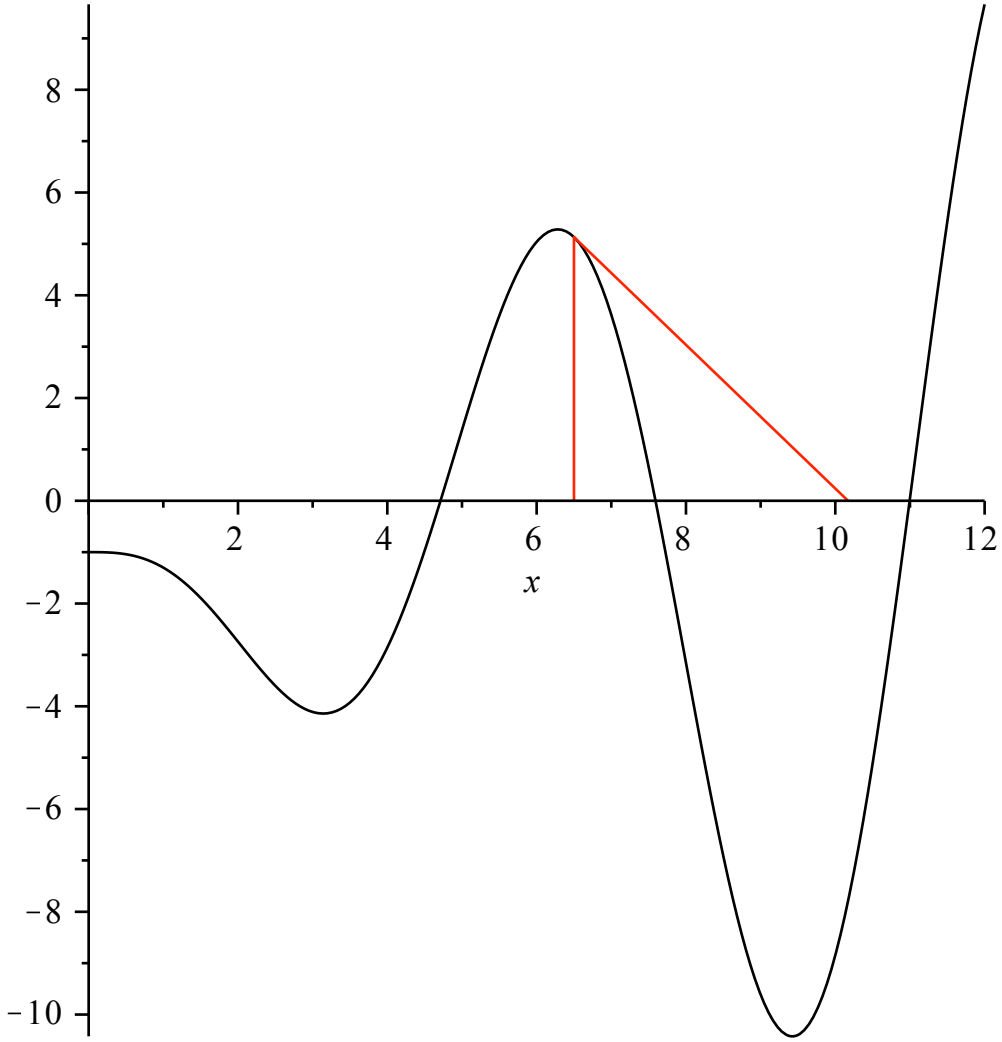
> **for** k **from** 1 **to** 6 **do**
$x[k] := N(x[k-1])$
end do

$$x_1 := 10.17072393$$
$$x_2 := 11.29946645$$
$$x_3 := 10.99011032$$
$$x_4 := 10.99557570$$
$$x_5 := 10.99557429$$
$$x_6 := 10.99557429 \qquad \textbf{(1.11)}$$

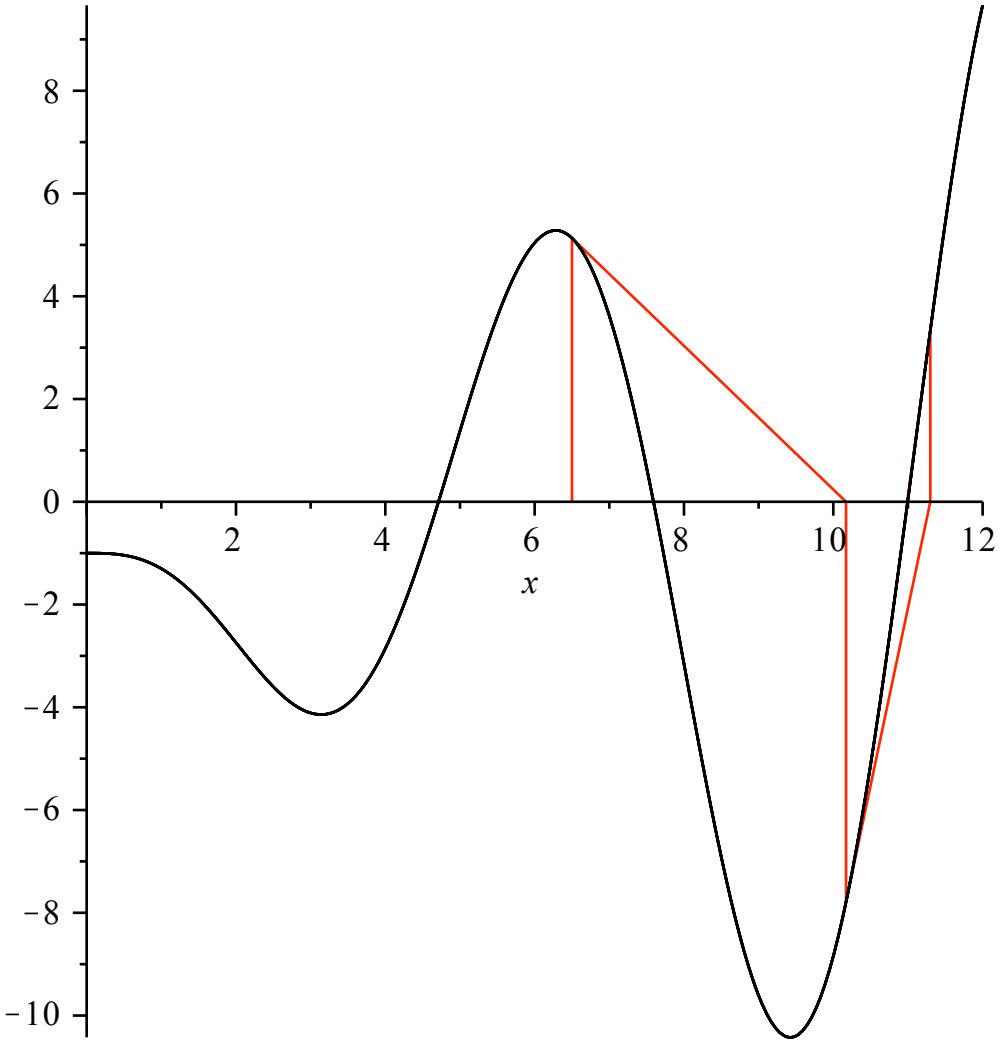
```
> fkuva := plot( f(x), x = 0 ..12, color = black);
```

$$fkuva := PLOT(\ldots) \qquad \textbf{(1.12)}$$

```
> display(seq(display( fkuva, tangkuva(x[k − 1])), k = 1 ..5), insequence = true)
```




```
> display(seq(display( fkuva, tangkuva(x[k − 1])), k = 1 ..5), insequence = false)
```

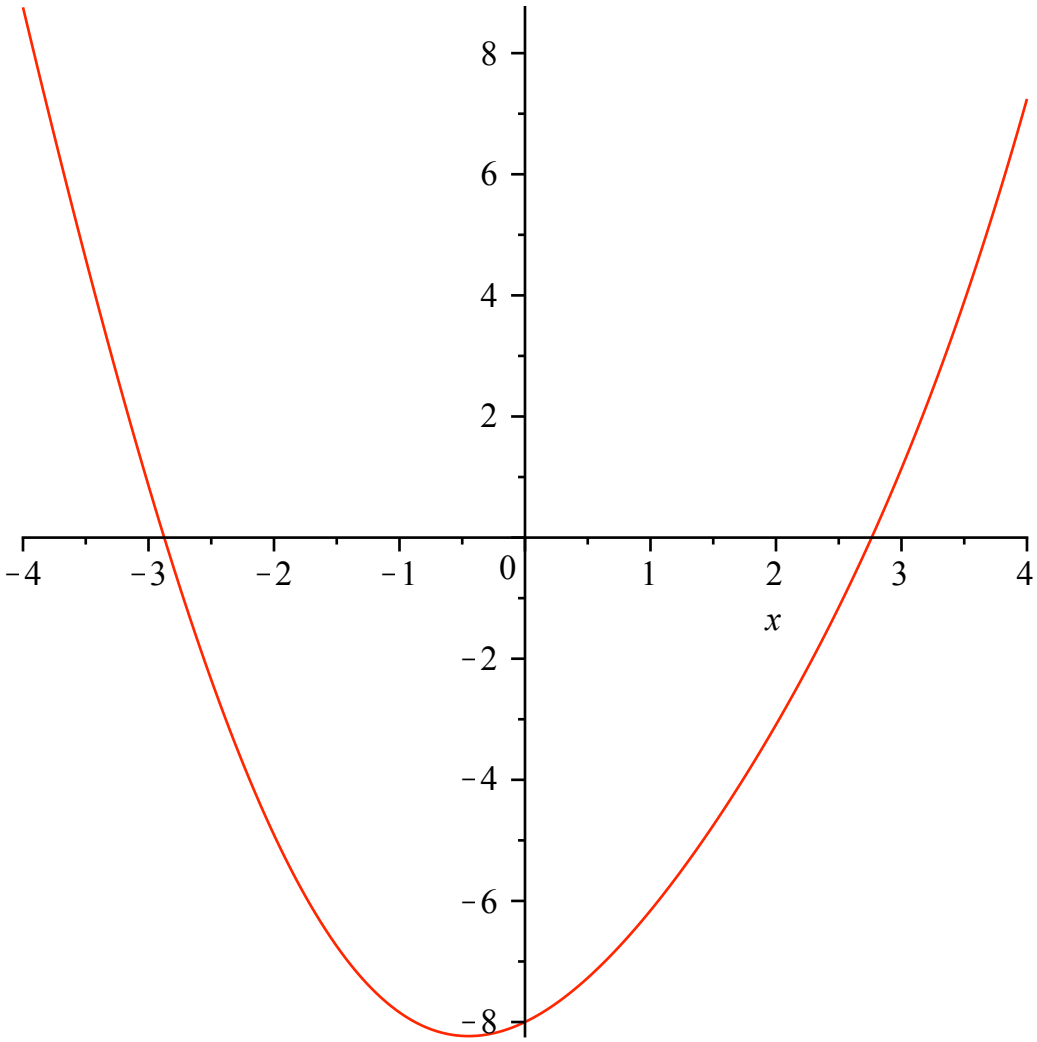


b)

> $f := x \to x^2 + \sin(x) - 8$

$$f := x \to x^2 + \sin(x) - 8 \qquad \textbf{(1.13)}$$

> $plot(f(x), x = -4 .. 4)$



> $x[0] := -1$

$$x_0 := -1 \qquad \textbf{(1.14)}$$

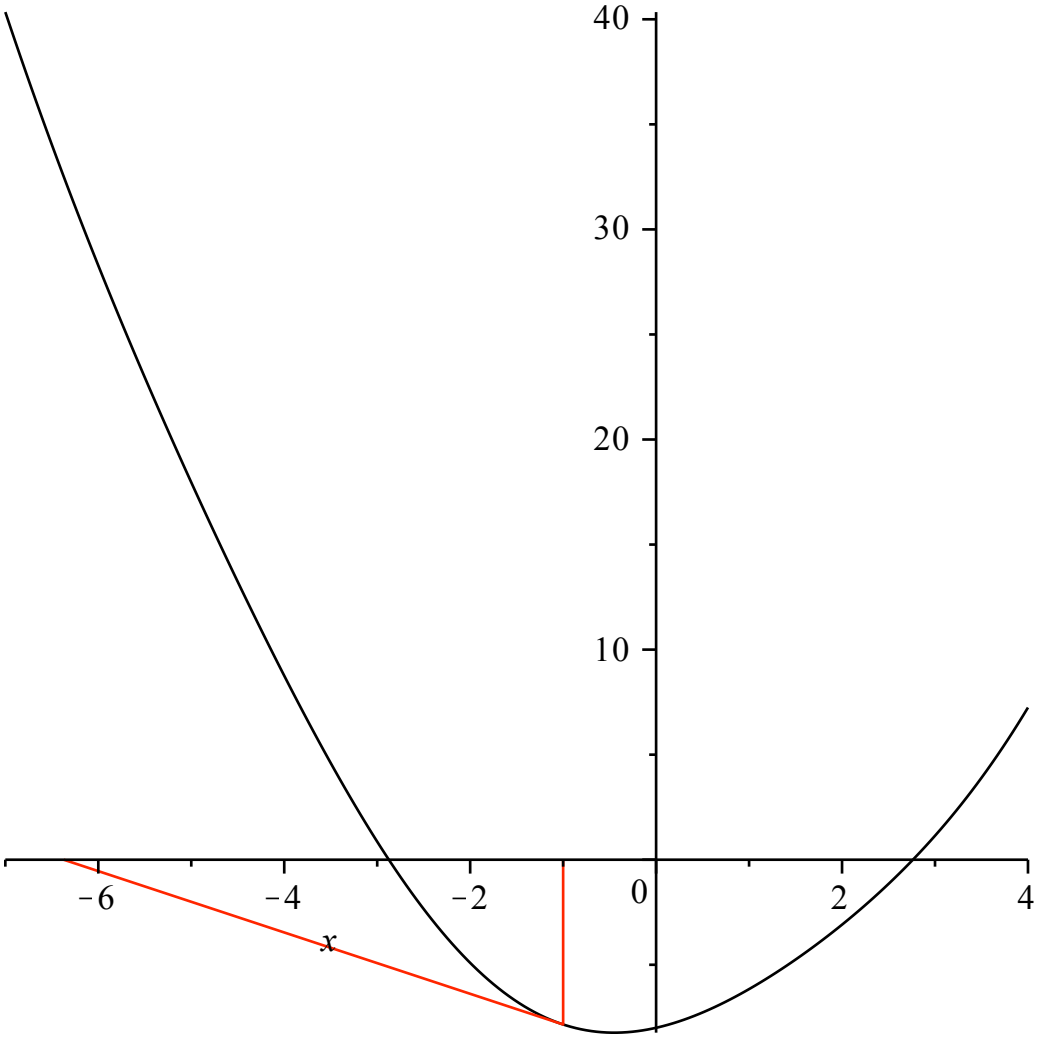
> **for** k **from** 1 **to** 7 **do**
$x[k] := N(x[k-1])$
end do

$$x_1 := -6.371982855$$
$$x_2 := -3.604384472$$
$$x_3 := -2.933318548$$
$$x_4 := -2.875234609$$
$$x_5 := -2.874675521$$
$$x_6 := -2.874675468$$
$$x_7 := -2.874675468 \qquad \textbf{(1.15)}$$

> $fkuva := plot(f(x), x = -7..4, color = black);$

$$fkuva := PLOT(\ldots) \qquad \textbf{(1.16)}$$

```
> display(seq(display(fkuva, tangkuva(x[k - 1])), k = 1 ..6), insequence = true)
```




```
>
```