Metodo di Newton (delle tangenti)

Enrico Bertolazzi

Introduzione

Scopo: data una approssimazione x* di uno zero di f(x) migliora la approssimazione applicando il metodo delle tangenti di Newton

$$x_{n+1} = x_n - \frac{f(x_n)}{D(f)(x_n)}$$

Carica le librerie

```
> restart:
  with(plots):
Warning, the name changecoords has been redefined
```

Procedura Newton

```
> newton_step := proc (x0, f, epsi, max_iter)
    # definisco le variabili locali
    local i, x, df, dx ;
    df := D(f) ; # calcola la derivata della
                 # funzione f(x) in alternativa
                 # df := diff(f(x),x) ;
                 # df := unapply(f,x) ;
    x  := x0 ;   # inizializza x
    for i from 1 to max_iter do
      dx := evalf(f(x)/df(x),12) ;
      x  := x - dx ;
      # controllo per terminare le iterazioni
      if ( abs(dx) < epsi ) then break ; end if ;
    end do;
    return x ;
  end:
```

Procedura di Newton con grafica

```
> newton_graph := proc (x0, f, epsi, max_iter)
    local i, x, xi, xmax, xmin, df, dx, vert, tang, A, B, C ;
    vert := [ ] ;
    tang := [ ] ;
```

```
    df   := D(f) ;
    x    := x0 ;
    xmax := x0 ;
    xmin := x0 ;
    for i from 1 to max_iter do
      dx := evalf(f(x)/df(x),20) ;
      xi := x - dx ;
      if xi > xmax then xmax := xi ; end if ;
      if xi < xmin then xmin := xi ;end if ;
      vert := [ op(vert), [ [x,0], [x,f(x)] ] ] ;
      tang := [ op(tang), [ [x,f(x)], [xi,0 ] ] ] ;
      x    := xi ;
      if abs(dx) < epsi then break ; end if ;
    end do ;
    dx := (xmax - xmin)/20 ;
    A  := plot(vert, color=blue):
    B  := plot(tang, color=red):
    C  := plot(f,xmin-dx..xmax+dx, color=green, thickness=2):
    display({A,B,C});
  end:
```

Esempio d'uso

> Definisce alcuni parametri comuni

```
> epsi     := 1e-5 :
  max_iter := 10 :
```

> Esempio di loop infinito

```
> f := (x) -> (x/(x^2)^(1/2))*(x^2)^(1/4) ;
```

Risolve il problema e stampa i risultati

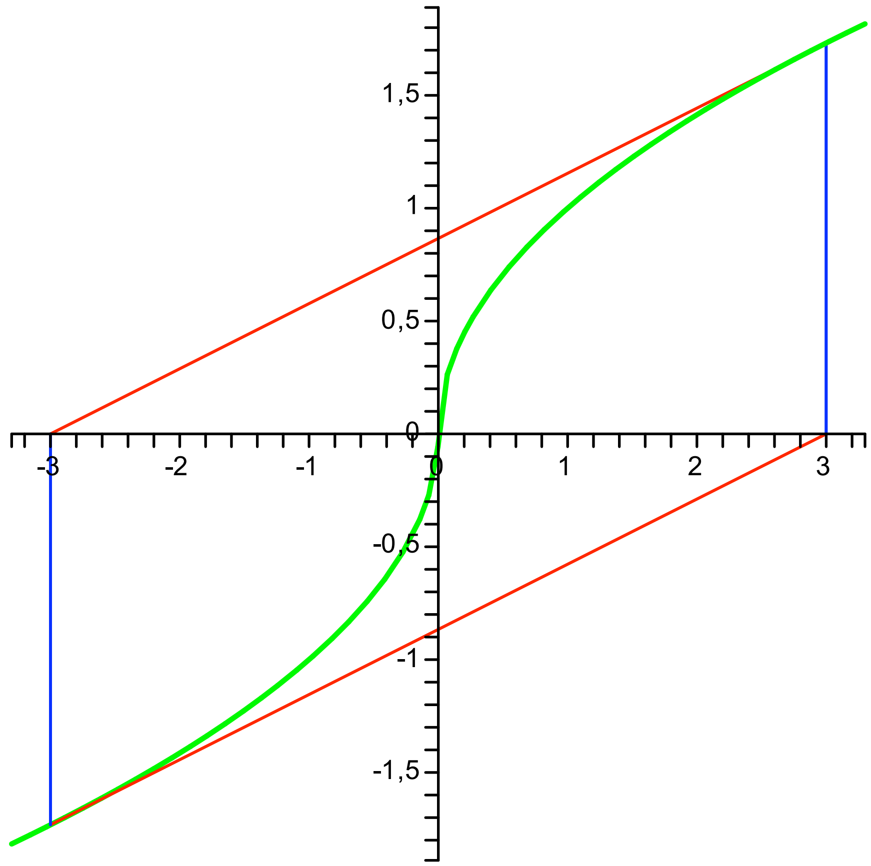
```
> newton_step(3,f,epsi,max_iter) ;
```

Risolve il problema e grafica la procedura

```
> newton_graph(3,f,epsi,max_iter) ;
```

$$f := x \to \frac{x\,(x^2)^{(1/4)}}{\sqrt{x^2}}$$

$$3.000000000$$



> Esempio di loop divergente

```
> f := (x) -> (x/(x^2)^(1/2))*(x^2)^(1/5) ;
```

Risolve il problema e stampa i risultati

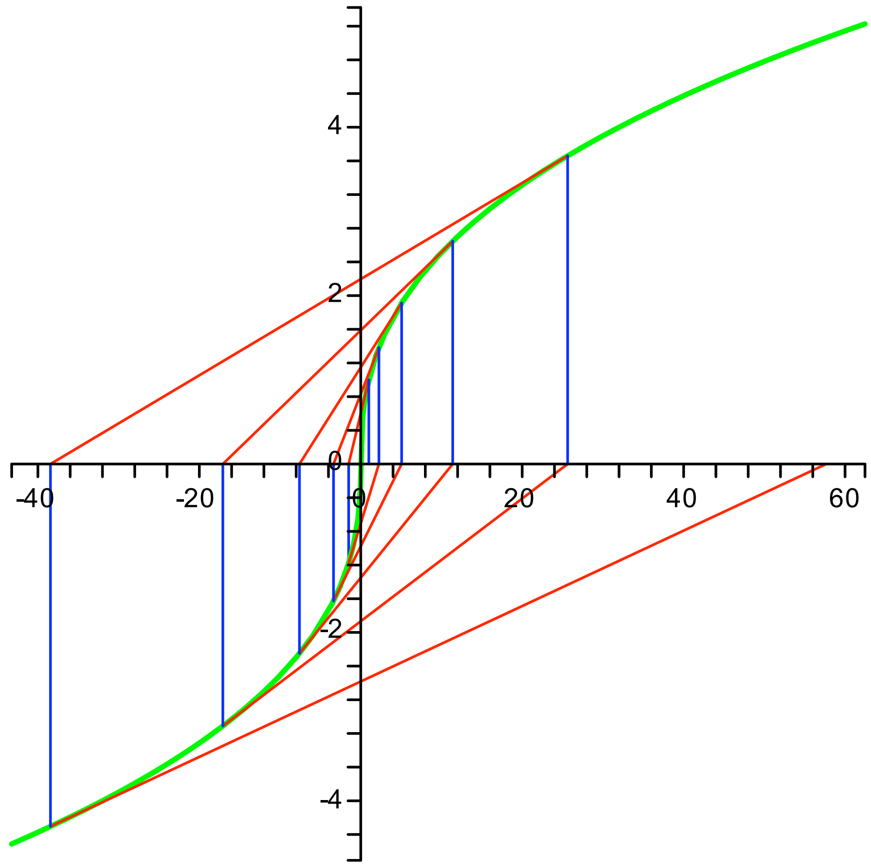
```
> newton_step(1,f,epsi,max_iter) ;
```

Risolve il problema e grafica la procedura

```
> newton_graph(1,f,epsi,max_iter) ;
```

$$f := x \to \frac{x\,(x^2)^{(1/5)}}{\sqrt{x^2}}$$

$$57.66503905$$



> Esempio di convergenza non monotona

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

Risolve il problema e stampa i risultati

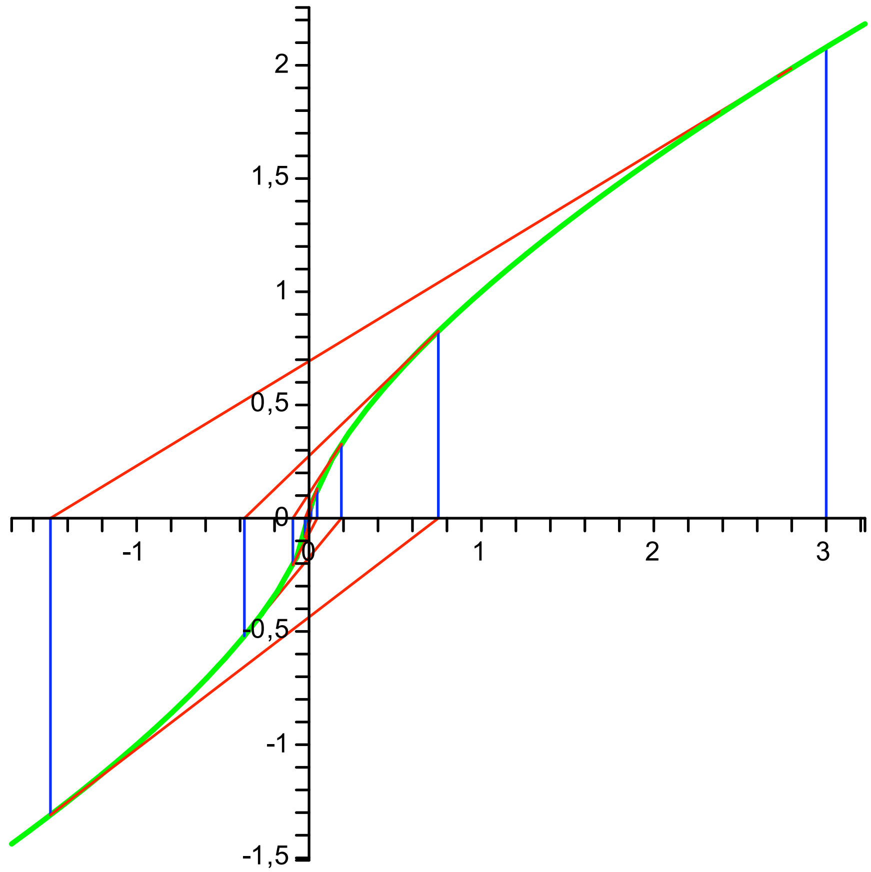
```
> newton_step(3,f,epsi,max_iter) ;
```

Risolve il problema e grafica la procedura

```
> newton_graph(3,f,epsi,max_iter) ;
```

$$f := x \to \frac{x\,(x^2)^{(1/3)}}{\sqrt{x^2}}$$

$$0.002929687500$$



> Esempio di convergenza monotona

```
> f := (x) -> x^2-3 ;
```

Risolve il problema e stampa i risultati

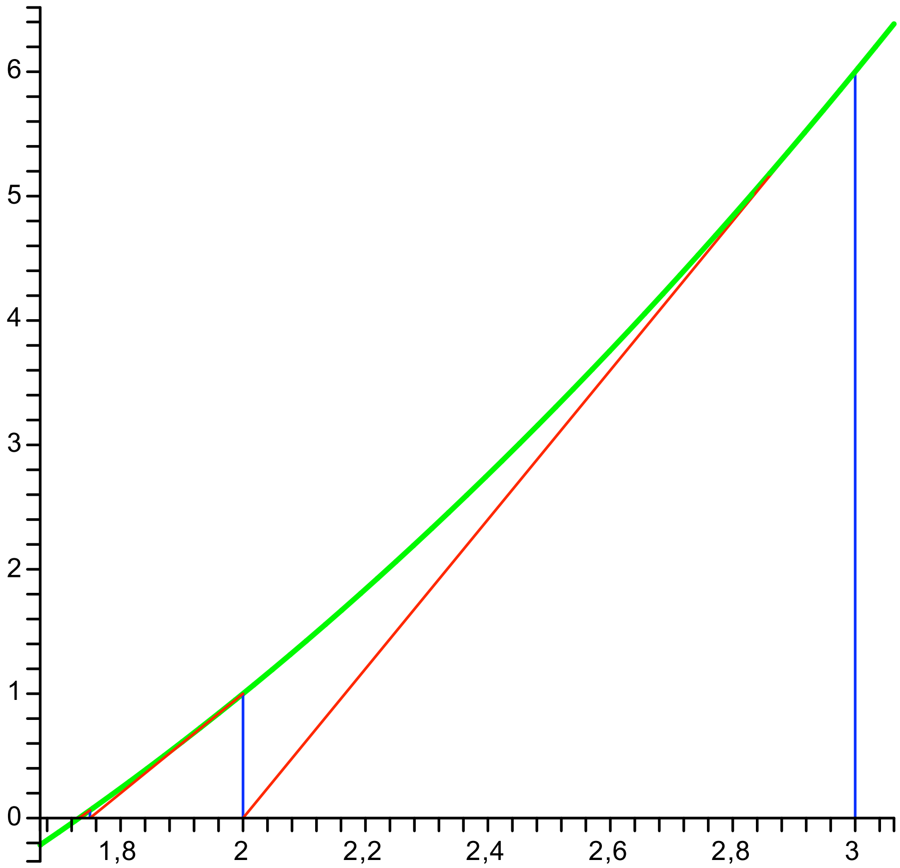
```
> newton_step(3,f,epsi,max_iter) ;
```

Risolve il problema e grafica la procedura

```
> newton_graph(3,f,epsi,max_iter) ;
```

$$f := x \rightarrow x^2 - 3$$

$$1.732050808$$



6
5
4
3
2
1
0
1,8
2
2,2
2,4
2,6
2,8
3

>