

Lezione 9

Enrico Bertolazzi

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

> #
# procedura eulero che implementa
# lo schema di avanzamento di Eulero
# esplicito
#
eulero := proc (x0, # punto iniziale
                xf, # punto finale
                y0, # dato di partenza
                h, # passo di avanzamento
                f) # f(x,y)

    local i, L, xi, yi, xil, yil ;
    # inizializzazione
    i := 0 ;
    L := [ [x0, y0] ] ;
    xi := x0 ;
    yi := y0 ;
    # ciclo
    while xi <= xf do
        xil := evalf(xi+h) ;
        yil := evalf(yi+h*f(xi,yi)) ;
        L := [ op(L), [xil, yil] ] ;
        # aggiornamento
        xi := xil ; yi := yil ;
    end ;
    return L ;
end proc ;
eulero := proc(x0, xf, y0, h, f)

    local i, L, xi, yi, xil, yil;

    i := 0;

    L := [[x0, y0]];

    xi := x0;

    yi := y0;

    while xi <= xf do

        xil := evalf(xi + h);
```

(1)

```
    yi1 := evalf(yi + h*f(xi, yi));  
    L := [op(L), [xi1, yi1]];  
    xi := xi1;  
    yi := yi1
```

```
end do;
```

```
return L
```

```
end proc
```

```
> #  
# definizione del problema  
# con soluzione esatta  $\exp(-100*x)$   
#
```

```
fun := (x,y) -> -100*y ;  
x0 := 0 ;  
y0 := 1 ;  
xf := 0.1 ;
```

$fun := (x, y) \rightarrow -100y$

$x0 := 0$

$y0 := 1$

$xf := 0.1$

(2)

```
> # salvo in A il "plot" della soluzione esatta  
A := plot(exp(-100*x), x=0..0.1, color=blue, thickness=3) :
```

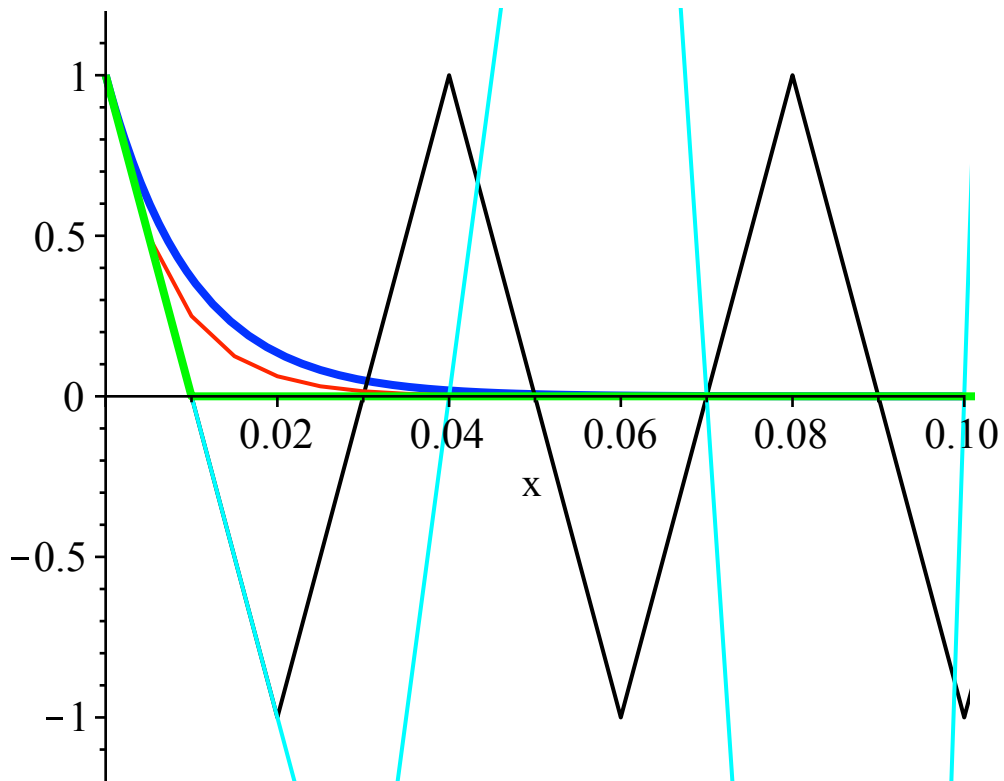
```
> # prima soluzione con h=0.005 (blue)  
h := 0.005 :  
L1 := eulero(x0,xf,y0,h,fun) :  
B1 := plot(L1,0..0.1,-1.2..1.2,color=red) :
```

```
> # seconda soluzione con h=0.01 (green)  
h := 0.01 :  
L2 := eulero(x0,xf,y0,h,fun) :  
B2 := plot(L2,0..0.1,-1.2..1.2,color=green,thickness=3) :
```

```
> # terza soluzione con h=0.02 (black)  
h := 0.02 :  
L3 := eulero(x0,xf,y0,h,fun) :  
B3 := plot(L3,0..0.1,-1.2..1.2,color=black) :
```

```
> # quarta soluzione con h=0.03 (cyan)  
h := 0.03 :  
L4 := eulero(x0,xf,y0,h,fun) :  
B4 := plot(L4,0..0.1,-1.2..1.2,color=cyan) :
```

```
> # combino le soluzioni in un unico grafico  
display({A,B1,B2,B3,B4}) ;
```



```

#
# procedura eulero che implementa
# lo schema di avanzamento di Eulero
# esplicito per un sistema di equazioni
#
eulero2D := proc (t0, # tempo iniziale
                  tf, # tempo finale
                  x0, # dato iniziale
                  y0, # dato iniziale
                  h, # passo di avanzamento
                  f, #
                  g) #
local i, L, ti, xi, yi, til, xil, yil ;
# inizializzazione
i := 0 ;
L := [ [x0, y0] ] ;
ti := t0 ;
xi := x0 ;
yi := y0 ;
# ciclo
while ti <= tf do
  til := evalf(ti+h) ;
  xil := evalf(xi+h*f(ti,xi,yi)) ;
  yil := evalf(yi+h*g(ti,xi,yi)) ;
  L := [ op(L), [xil, yil] ] ;
# aggiornamento

```

```

    ti := til ; xi := xil ; yi := yil ;
end ;
return L ;
end proc ;

```

```
eulero2D := proc(t0, tf, x0, y0, h, f, g)
```

(3)

```

    local i, L, ti, xi, yi, til, xil, yil;

```

```

    i := 0;

```

```

    L := [[x0, y0]];

```

```

    ti := t0;

```

```

    xi := x0;

```

```

    yi := y0;

```

```

    while ti <= tf do

```

```

        til := evalf(ti + h);

```

```

        xil := evalf(xi + h*f(ti, xi, yi));

```

```

        yil := evalf(yi + h*g(ti, xi, yi));

```

```

        L := [op(L), [xil, yil]];

```

```

        ti := til;

```

```

        xi := xil;

```

```

        yi := yil

```

```

    end do;

```

```

    return L

```

```
end proc
```

```

> #
# definizione del problema
# con soluzione esatta sul cerchio
#

```

```

fx := (t,x,y) -> y ;

```

```

fy := (t,x,y) -> -x ;

```

```

t0 := 0 ;

```

```

tf := 6 ;

```

```

x0 := 0 ;

```

```

y0 := 1 ;

```

```

fx := (t, x, y) -> y

```

(4)

```

fy := (t, x, y) -> -x

```

```

t0 := 0

```

```

tf := 6

```

```
x0 := 0
```

```
y0 := 1
```

```
> # salvo in C il "plot" del cerchio  
C := implicitplot(x^2+y^2-1,  
                 x=-2..2,y=-2..2,  
                 color=blue,thickness=3) :  
> # prima soluzione con h=0.01 (blue)  
h := 0.01 :  
L1 := eulero2D(t0,tf,x0,y0,h,fx,fy) :  
B1 := plot(L1,-2..2,-2..2,color=red) :  
> # seconda soluzione con h=0.05 (green)  
h := 0.05 :  
L2 := eulero2D(t0,tf,x0,y0,h,fx,fy) :  
B2 := plot(L2,-2..2,-2..2,color=green) :  
> # terza soluzione con h=0.1 (black)  
h := 0.1 :  
L3 := eulero2D(t0,tf,x0,y0,h,fx,fy) :  
B3 := plot(L3,-2..2,-2..2,color=black) :  
> # quarta soluzione con h=0.5 (cyan)  
h := 0.5 :  
L4 := eulero2D(t0,tf,x0,y0,h,fx,fy) :  
B4 := plot(L4,-2..2,-2..2,color=cyan) :  
> # combino le soluzioni in un unico grafico  
display({C,B1,B2,B3,B4}) ;
```

