FIRST NAME : 26 April 2012
LAST NAME : .

Answer-sheet 2
Euler and Runge-Kutta of order 2

Exercice 1. : Consider the differential equation $y' = -y$. To be able to plot the corresponding vector field, we will need to define, in addition to the function `ff(t,y)=f(y)` for the `ode` instruction, a function `fff(t,(t,y))=[1,-y]'`.

1. Execute the following code and explain how `fchamp` compute the direction of the arrows. `clear ;`
```
xset("window",0) ;
function z=f(y) ; z=-y ; endfunction ;
function z=ff(t,y) ; z=f(y) ; endfunction ;
function z=fff(t,v) ; z=[1,-v(2)]' ; endfunction ;
fchamp(fff,0,0 :0.2 :2,0 :0.4 :4) ;
```

2. With the following constantes, compute the solution $y(t)$ s.t. $y(0) = 4$ with `y=ode(y0,0,t,ff)` and add it to your figure with `plot(t,y)`
```
Tmax=2 ;y0=4 ;
N=100 ;smallstep=Tmax/N ;
M=10 ;largestep=Tmax/M ;
t=0 :smallstep :Tmax ;
```
How is the solution with respect to the vector field ?

3. The following function allow to compute the Euler approximation.
```
y(t0)=y0.
function y=Euler(t0,y0,step) ;
      y=y0+step*ff(t0,y0) ;
endfunction ;
```
The following line allow to drow the segment which is the first step of the approximate solution.
```
plot([0,largestep],[y0, Euler(0,y0,largestep)],'r-o') ;
```
Outline the figure you obtain and comment.

4. When doing this again up to $T = 2 =$`Tmax` one get the complete approximate solution.
```
step=largestep ;
tt=0 :step :Tmax ;
yy=zeros(1+Tmax/step) ;yy(1)=y0 ;
for i=1 :1 :Tmax/step ;
      yy(i+1)=Euler(tt(i),yy(i),step) ;
end ;
plot(tt,yy,'g->') ;
disp(y(N+1)-yy(1+Tmax/step),'difference=',step,'step=') ;
```

What is the decrepency Δ you observe between the value $y(2)$ of the solution and its approximation y_M =`yy(1+M)` ?

$\Delta_{0.2} =$

Explain why $\tilde{y}(T)$ =`yy(M+1)`=`yy(1+Tmax/step)`.

Drow carefully the picture you obtain showing precisely on your picture the decrepency $\tilde{y}(0.4)$. Compute its exacte value.

$\tilde{y}(0.4) =$

5. Compute again the approximate solution but now with `step=smallstep;` and comment. Notice that you can get the new curve en red using the instruction `plot(tt,yy,'r-.');`

6. Let $\Delta_{0.2}$ and $\Delta_{0.02}$ be the errors, at $t = 2$, between exacte and approximate solution for $h = 0.2$ et $h = 0.02$. Compute :

   $\Delta_{0.2} =$

   $\Delta_{0.02} =$

   $\Delta_{0.02}/\Delta_{0.2} =$

   How does this error change with the size of the step ?

Exercice 2. **:** We want to see in what extend the RK2 method is indeed better then the Euler's one. Here is the Scilab code for the RK2 algorithm.

```
function y=RK(t0,y0,step);
// local variables : k1,k2;
      k1=step*ff(t0,y0);
      k2=step*ff(t0+step/2,y0+k1/2);
      y=y0+k2;
endfunction;
```

1. Consider again the equation $y' = -y$. Drow the vector field in the same window as for exercice 1 and the solution such that $y(0) = 4$. Then add the first segment or the approximate solution. Comment.

2. Drow the approximate solution for `step=largestep` first and then for `step=smallstep`. Comment.

3. Let again $\Delta_{0.2}$ and $\Delta_{0.02}$ be the decrepency at $t = 2$ between the exact and the approximate solution for $h = 0.2$ et $h = 0.02$. Report your results :

   $\Delta_{0.2} =$

   $\Delta_{0.02} =$

   $\Delta_{0.02}/\Delta_{0.2} =$

4. Explain if your results confirm or not that RK2 is a order 2 method.

Remark : The `ode` instruction does not compute the exact solution but it computes obviously an approximation. As we know that the exact solution is $y(t) = 4e^{-t}$ compute the error using for example

```
disp(4*exp(-2)-y(N+1),'difference=4exp(-2)-ode(2)');
```

$\Delta =$

Using the "help" of `scilab` find which algorithm is used by `ode`.