Súrlódó harmonikus oszcillátor mozgása 1 dimenzióban

Bartha Ferenc, SZTE, 2003

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

A mozgásegyenlet: $me := m\left(\frac{\partial}{\partial t}\mathrm{v}(t)\right) = -a\,\mathrm{x}(t) - \frac{b\,\mathrm{v}(t)}{|\mathrm{v}(t)|} + F,\ \mathrm{v}(t) = \frac{\partial}{\partial t}\mathrm{x}(t)$

Az F erõ a tapadási súrlódást jelenti. Úgy értendõ, hogy F=0 amikor v(t) nem nulla, a fordulópontokban (v=0) pedig |F|<c, ahol c valamilyen állandó.

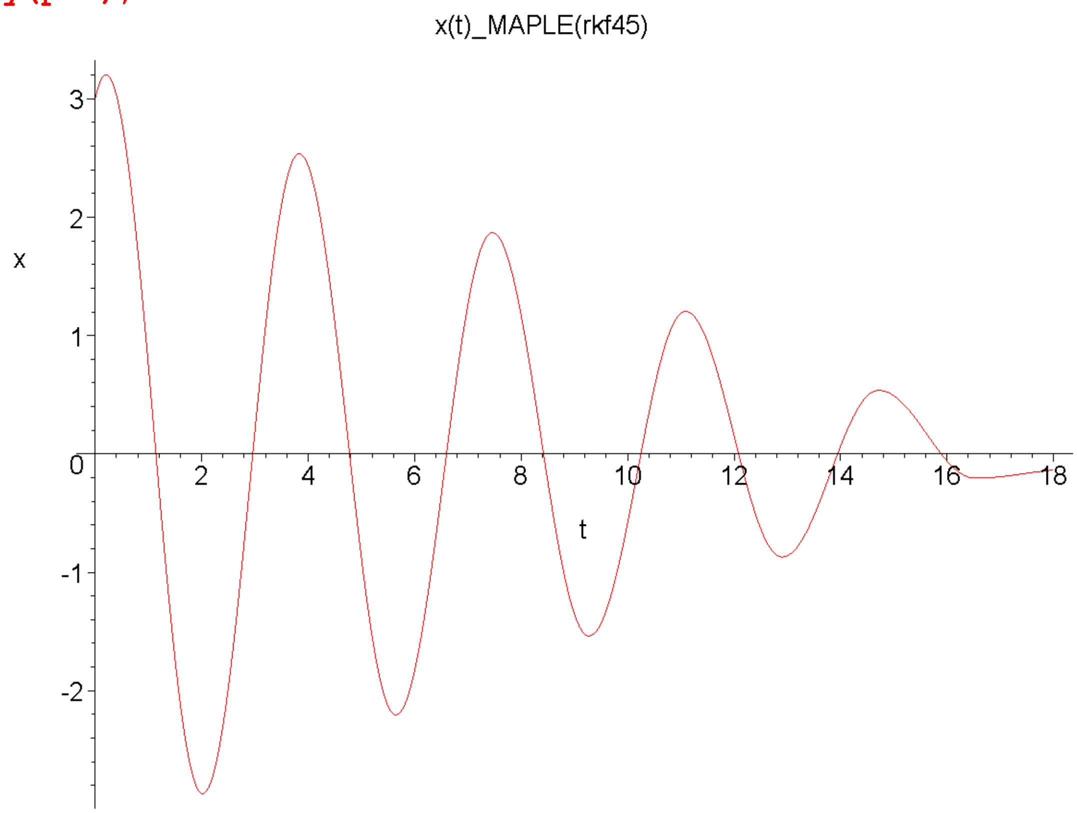
Paraméterek: `m:=1:a:=3:b:=0.5:c:=1:x0:=3:v0:=2:`

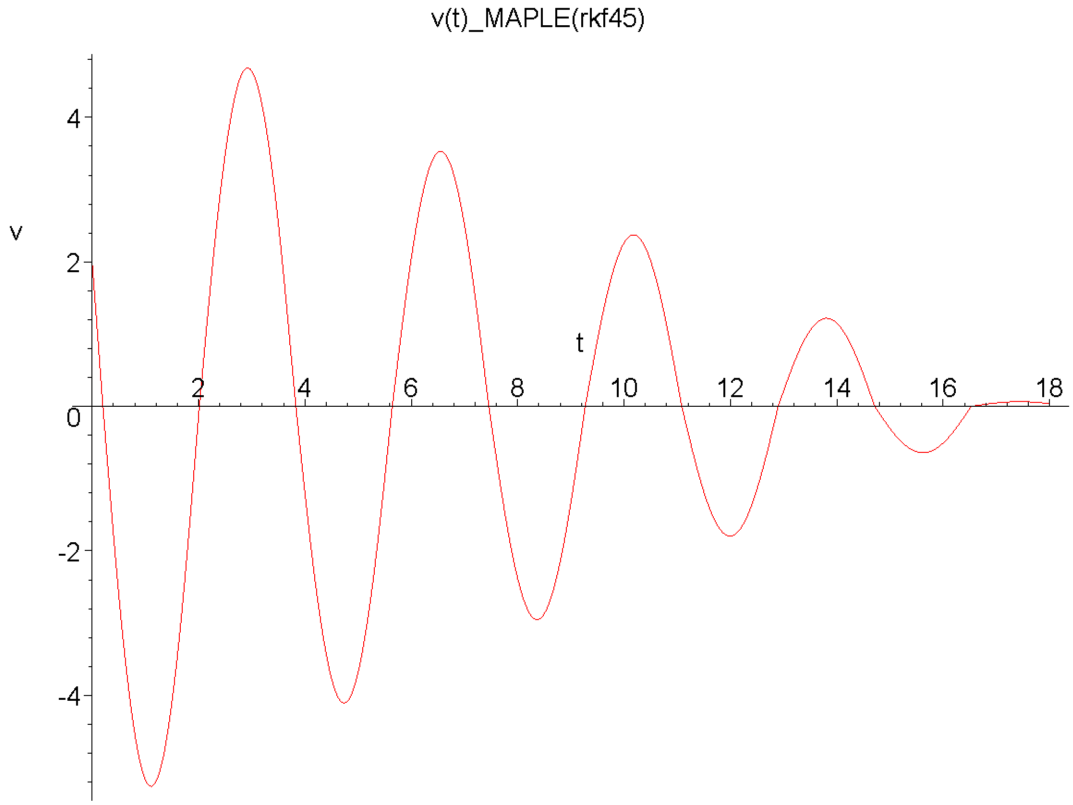
Kezdõfeltételek t=0-ban: `ini:=x(0)=x0,v(0)=v0:`

Szimulációs idõ: `tmax:=18:`

Elõször a MAPLE megoldását állítjuk elõ (tapadási súrlódás nélkül: F=0)

```
> F:=0:sol:=dsolve([me,ini],{x(t),v(t)},type=numeric):
> prx:=odeplot(sol,[t,x(t)],0..tmax,title=`x(t)_MAPLE(rkf45)`):display(prx);
  prv:=odeplot(sol,[t,v(t)],0..tmax,title=`v(t)_MAPLE(rkf45)`):display(prv);
```





Most pedig ugyanez tapadással, saját (Euler-féle) megoldással....
Figyeljük meg (az alább következõ h:=..... sor módosításával), hogy
a) h=0.05 (durva) beosztással a rezgés nemhogy nem csillapodik, hanem gerjed!
b) h=0.005 beosztásnál a megoldás látszólag egész jó, de elvétjük a tapadást! A hiba vizsgálatából (utolsó két ábra) látható, hogy a kvalitatíve jónak látszó megoldás nagyon pontatlan.

Legyen most a (jónak minósíthetõ) lépésköz: `h:=0.001:`

Ezzel durván `imx:=floor(tmax/h);`
pontra osztottuk a [0,tmax] intervallumot
Nemigen szükséges minden osztópontban eltárolni az ábrázoláshoz a megfelelõ függvényértékeket. Mondjuk elég lesz maximum `npt:=100:`
az összesbõl, ami azt jelenti, hogy elég minden
`step:=max(1,floor(imx/npt));`
step-edik pontban megõrizni x(t) és v(t) értékét

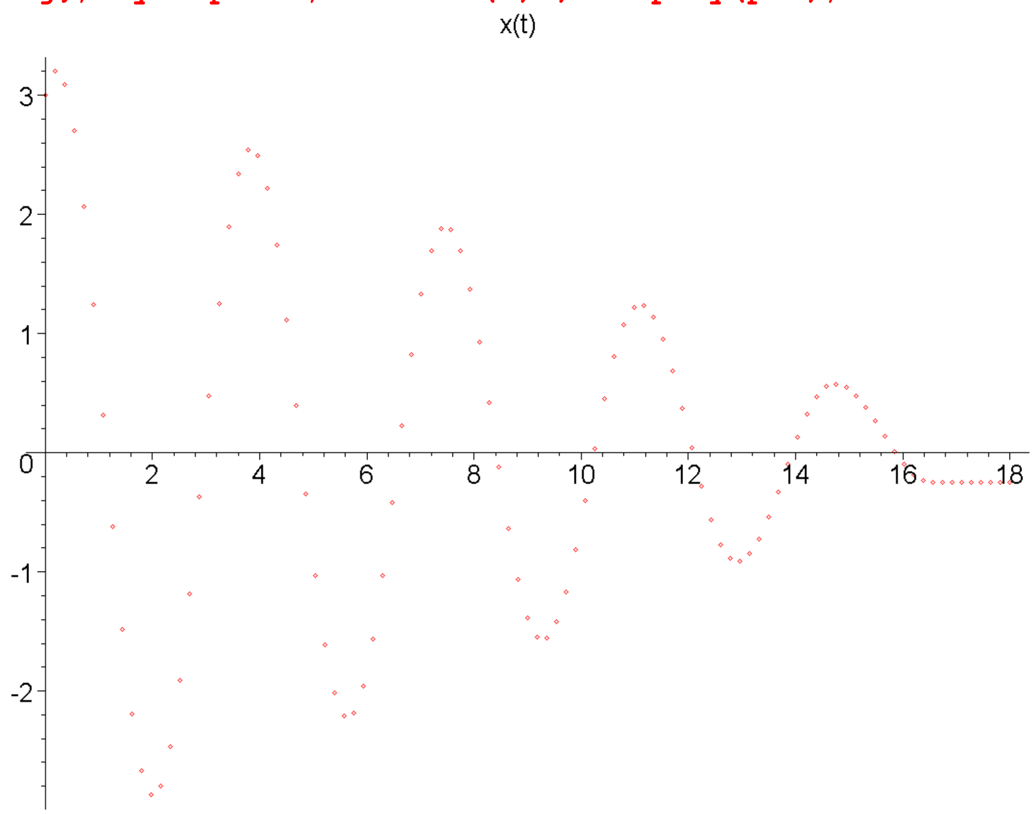
$$imx := 18000$$

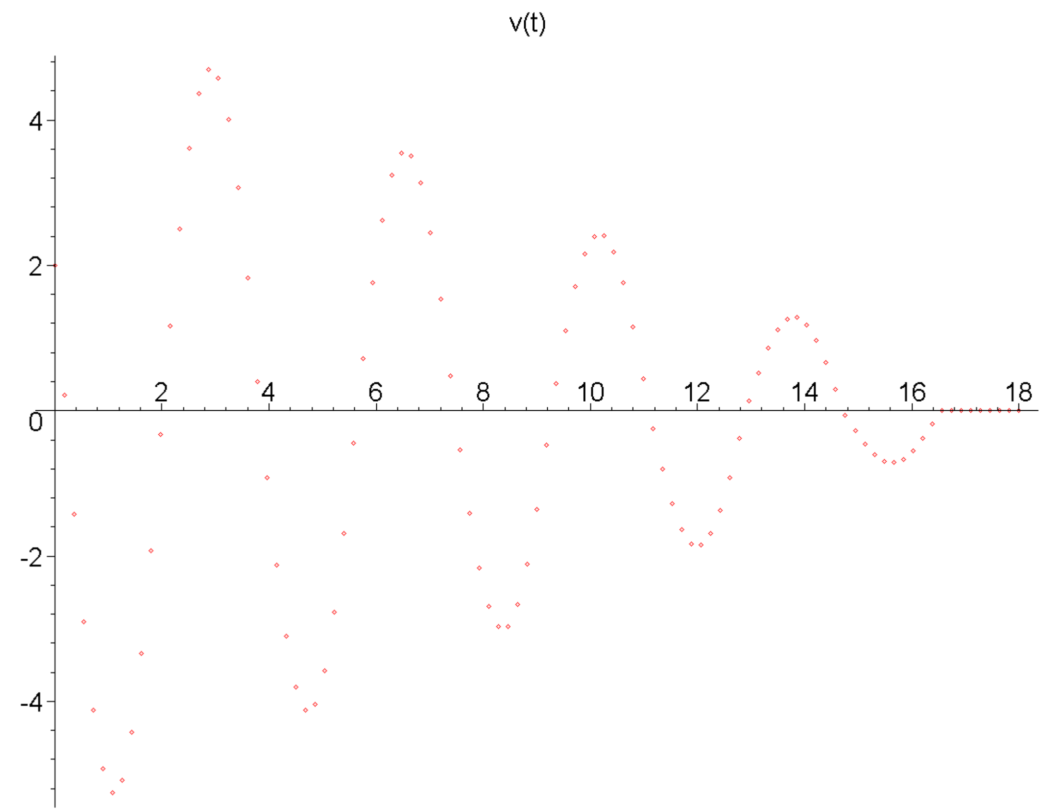
$$step := 180$$

Ime a rekurziós ciklus:

```
> t:=0:vi:=v0:xi:=x0:
  j:=0:
  for i from 0 to imx do
  if ((i mod step)=0) then  #mentjük a kiszámolt pontokat az
  ábrázoláshoz
   j:=j+1:
   ex[j]:=xi:ev[j]:=vi:et[j]:=t:
```

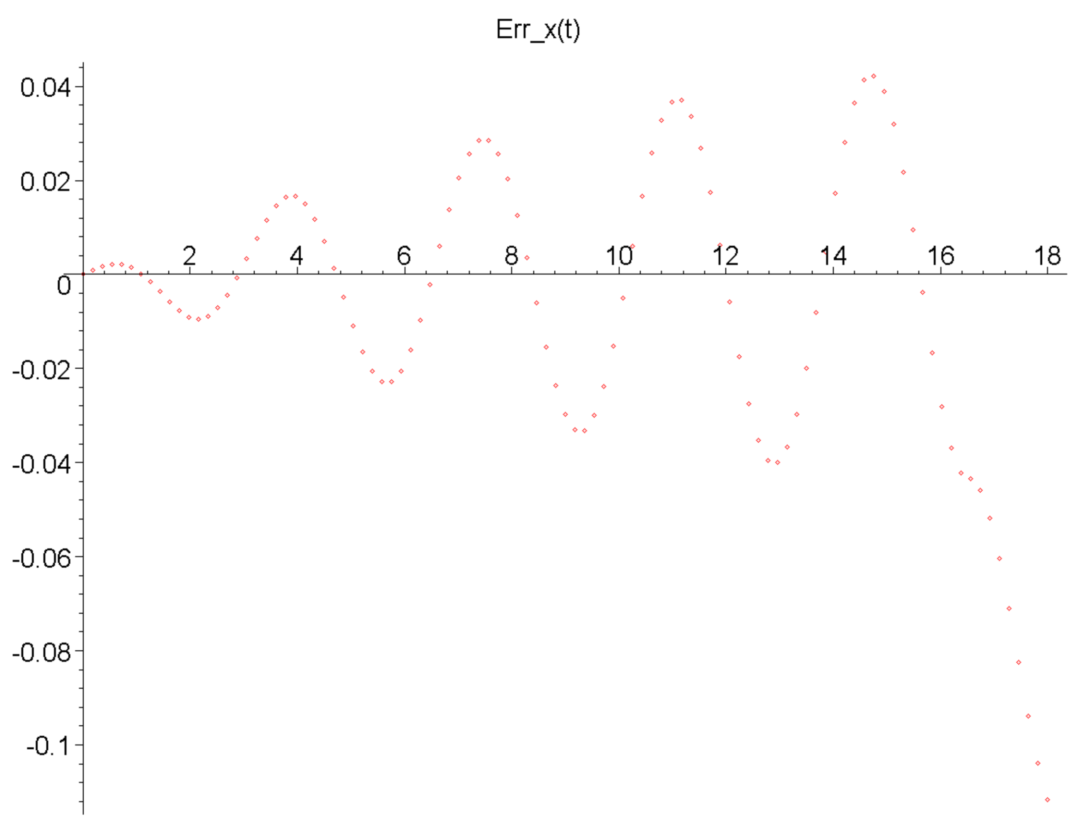
```
 rx[j]:=subs(sol(t),x('t')):rv[j]:=subs(sol(t),v('t'))
end if:
f1:=vi:f2:=-a/m*xi-b/m*signum(vi):
if(vi*(vi+h*f2)>0 or a*abs(xi)>c) then  #nem kell lépni, ha
tapad
  xi:=xi+h*f1:vi:=vi+h*f2:
end if:
t:=t+h:
od:
pex:=plot({[et[s],ex[s]]
$s=1..j},style=point,title=`x(t)`):display(pex);
pev:=plot({[et[s],ev[s]]
$s=1..j},style=point,title=`v(t)`):display(pev);
```

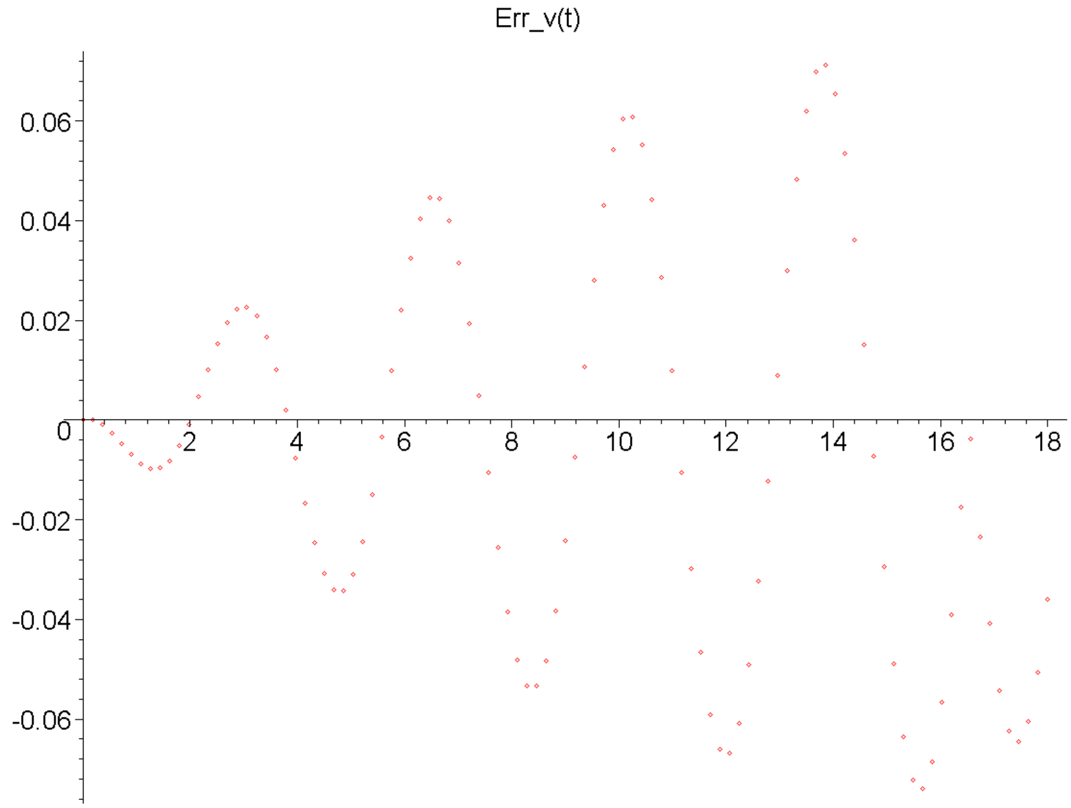




Összehasonlítások:

```
pdx:=plot({[et[s],ex[s]-rx[s]]
$s=1..j},style=point,title=`Err_x(t)`):display(pdx);
pdv:=plot({[et[s],ev[s]-rv[s]]
$s=1..j},style=point,title=`Err_v(t)`):display(pdv);
>
```




Err_v(t)
0.06
0.04
0.02
0
-0.02
-0.04
-0.06
2
4
6
8
10
12
14
16
18

>

>

>