Oscillateur harmonique sans et avec thermostat

Context

Dans cet exemple on traitera la dynamique d'un oscillateur harmonique en deux dimensions en présence d'un thermostat qui le maintient à une énergie cinétique constante. On compare en particulier sa trajectoire à celle produite par un oscillateur harmonique standard, sans thermostat, avec la même fréquence et les mêmes conditions initiales pour la trajectoire.

Quelques instructions pour le rapport

- Donner une introduction synthétique à la mécanique de Gauss et sa capacité de tenir compte des contraintes non-holonomes (voir cours mécanique analytique). Expliquer ce que veut dire "contrainte non-holonome".
- Décrire la solution des équations de mouvement donnée en bas et expliquer pourquoi le plot paramétrique, (x(t),y(t)) pour $t \in [0,t_{max}]$, donne une ellipse dans le cas de l'oscillateur harmonique standard (sans thermostat). L'algorithme utilisé pour la solution numérique des équations de moiuvement n'est pas important pour votre rapport. Vous pouvez utiliser les figures ("cut and paste") dans ce document afin d'illustrer votre rapport.

Equations de mouvement

Avec thermostat (voir cours mécanique analytique)

In[1]:=
```
TraditionalForm[μ = -M ω^2 / (2 kB T) (x[t] × x'[t] + y[t] × y'[t])]
```

Out[1]//TraditionalForm=

$$-\frac{M\,\omega^2\,(x(t)\,x'(t)+y(t)\,y'(t))}{2\,T\,k_B}$$

La forme générale

In[2]:=
```
TraditionalForm[deq1 = x''[t] + ω^2 x[t] + μ x'[t] == 0]
```

Out[2]//TraditionalForm=

$$-\frac{M\,\omega^2\,x'(t)\,(x(t)\,x'(t)+y(t)\,y'(t))}{2\,T\,k_B}+x''(t)+\omega^2\,x(t)=0$$

In[3]:=
```
TraditionalForm[deq2 = y''[t] + ω^2 y[t] + μ y'[t] == 0]
```

Out[3]//TraditionalForm=

$$-\frac{M\,\omega^2\,y'(t)\,(x(t)\,x'(t)+y(t)\,y'(t))}{2\,T\,k_B}+y''(t)+\omega^2\,y(t)=0$$

Une forme particulière pour la solution numérique

In[4]:=
```
replace = {ω → 1, M → 4, T → 1, k_B → 1}
```
Out[4]=
```
{ω → 1, M → 4, T → 1, k_B → 1}
```

In[5]:=
```
deq1N = deq1 /. replace
```
Out[5]=
```
x[t] - 2 x′[t] (x[t] x′[t] + y[t] y′[t]) + x″[t] == 0
```

In[6]:=
```
deq2N = deq2 /. replace
```
Out[6]=
```
y[t] - 2 y′[t] (x[t] x′[t] + y[t] y′[t]) + y″[t] == 0
```

Pour comparaison sans thermostat

La forme générale

In[7]:=
```
TraditionalForm[deq1free = x''[t] + ω^2 x[t] == 0]
```
Out[7]//TraditionalForm=

$$x''(t) + \omega^2\, x(t) = 0$$

In[8]:=
```
TraditionalForm[deq2free = y''[t] + ω^2 y[t] == 0]
```
Out[8]//TraditionalForm=

$$y''(t) + \omega^2\, y(t) = 0$$

La forme particulière pour la solution numérique

In[9]:=
```
deq1freeN = deq1free /. replace
```
Out[9]=
```
x[t] + x″[t] == 0
```

In[10]:=
```
deq2freeN = deq2free /. replace
```
Out[10]=
```
y[t] + y″[t] == 0
```

Solution numérique des équations de mouvement

Avec thermostat

Créer vitesse initiale arbitraire compatible avec la contrainte imposée. Comme $E_{cin} = \mathrm{M}v^2/2 = k_B\,\mathrm{T}$ est imposée, cette vitesse est définie par un point arbitraire sur un cercle de module $v = \sqrt{2\, k_B\, T/M}$.

In[11]:=
```
v = Sqrt[2 k_B T / M];
```

In[12]:=
```
vN = v /. replace
```
Out[12]=

$$\frac{1}{\sqrt{2}}$$

In[13]:=
```
SeedRandom[1234]
```
Out[13]=
```
RandomGeneratorState[ Method: ExtendedCA
                      State hash: 6155169294560358013 ]
```

In[14]:=
```
ϕ = RandomReal[{0, 2 π}]
```
Out[14]=
```
5.50789
```

In[15]:=
```
vx0 = vN * Cos[ϕ]
```
Out[15]=
```
0.505028
```

In[16]:=
```
vy0 = vN * Sin[ϕ]
```
Out[16]=
```
-0.494921
```

Condition initiale pour la position

In[17]:=
```
x0 = 1.; y0 = 0.;
```

Solution numérique des équations de mouvement (le résultat sont deux fonctions, x(t), y(t), qui interpolent la solution numérique pour $t \in [0, t_{max}]$).

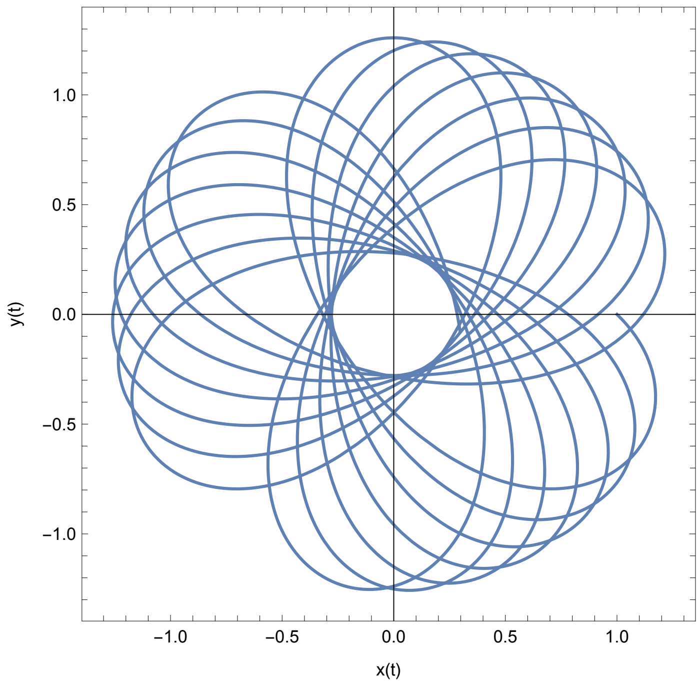
In[18]:=
```
tmax = 100.0;
```

In[19]:=
```
solution = NDSolve[{deq1N, deq2N, x[0] == x0, y[0] == y0, x'[0] == vx0, y'[0] == vy0},
   {x[t], y[t]}, {t, 0., tmax}]
```

Out[19]=
```
{{x[t] → InterpolatingFunction[ Domain: {{0., 100.}} Output: scalar ][t],
  y[t] → InterpolatingFunction[ Domain: {{0., 100.}} Output: scalar ][t]}}
```

In[20]:=
```
Fig1 = ParametricPlot[Evaluate[{x[t], y[t]} /. solution], {t, 0, tmax},
   Frame → True, AspectRatio → 1, FrameLabel → {"x(t)", "y(t)"}]
```

Out[20]=



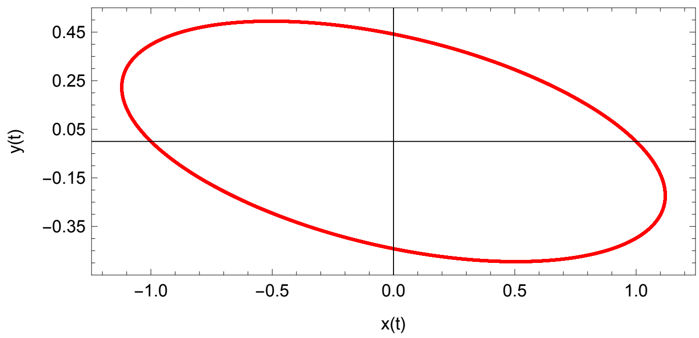
Pour comparaison sans thermostat

In[21]:=
```
x0
```
Out[21]=
```
1.
```

In[22]:=
```
y0
```
Out[22]=
```
0.
```

In[23]:=
```
solutionFree = NDSolve[{deq1freeN, deq2freeN, x[0] == x0,
   y[0] == y0, x'[0] == vx0, y'[0] == vy0}, {x[t], y[t]}, {t, 0., tmax}]
```
Out[23]=
```
{{x[t] → InterpolatingFunction[ Domain: {{0., 100.}} Output: scalar ][t],
  y[t] → InterpolatingFunction[ Domain: {{0., 100.}} Output: scalar ][t]}}
```

In[24]:=
```
Fig2 = ParametricPlot[Evaluate[{x[t], y[t]} /. solutionFree],
  {t, 0, tmax}, PlotStyle → Red, Frame → True, FrameLabel → {"x(t)", "y(t)"}]
```
Out[24]=



In[25]:=
```
Show[{Fig1, Fig2}]
```
Out[25]=