

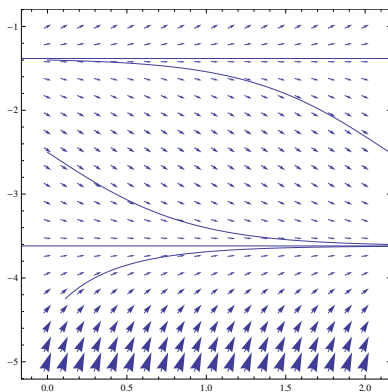
### Exercice I.1

Calculer les points d'équilibre et tracer approximativement les trajectoires  $\chi(x_0, \cdot)$  pour les trois équations différentielles suivantes :

1.  $\dot{x} = x^2 + 5x + 5$
2.  $\dot{x} = (x - 3)(x + 6)(x - 12)(x + 24)$
3.  $\dot{x} = -\text{sgn}(x)\sqrt{|x|}$

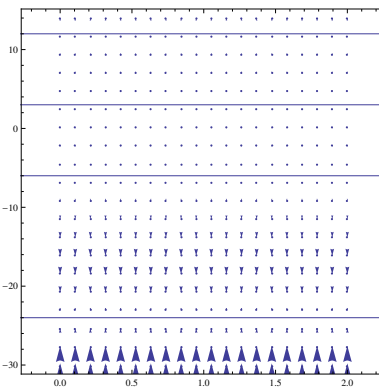
**Corrigé :** Dans les graphiques suivants, l'axe horizontal est le temps  $t$  [s] et l'axe vertical est la variable  $x$  :

Système 1



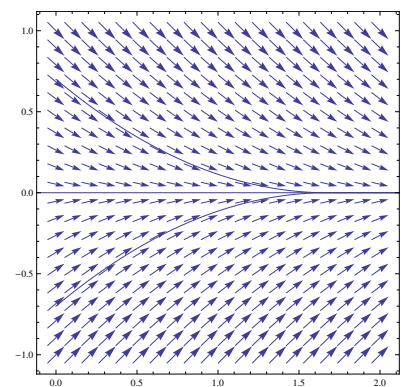
Deux équilibres  
 $\bar{x} = \frac{1}{2}(-5 \pm \sqrt{5})$  et (par ex.) trois trajectoires  $\mathcal{X}(-4.5, \cdot)$ ,  $\mathcal{X}(-2.5, \cdot)$  et  $\mathcal{X}(-1.4, \cdot)$ .

Système 2



Quatre équilibres  
 $\bar{x} = -24, -6, 3, 12$

Système 3



Un équilibre  $\bar{x} = 0$   
 et (par ex.) deux trajectoires  $\mathcal{X}(0.7, \cdot)$  et  $\mathcal{X}(-0.7, \cdot)$ .

### Exercice I.2

Soit le système

$$\begin{aligned}\dot{x}_1 &= 2x_1 + 3x_2 + 6x_1^2 + 6x_2^3 \\ \dot{x}_2 &= x_1 - x_2 + 7x_2^2\end{aligned}$$

Dessiner les éléments de pentes (portrait de phase type "limaille de fer") et déterminer approximativement le portrait de phase. Calculer la matrice issue de la linéarisation aux points d'équilibre. Calculer les valeurs et vecteurs propres et dessiner approximativement les trajectoires proche du point d'équilibre.

**Corrigé** Il y a deux points d'équilibre :

1.  $\bar{x}_1 = \bar{x}_{11} = -0.550$  et  $\bar{x}_2 = \bar{x}_{12} = -0.218$ ;
2.  $\bar{x}_1 = \bar{x}_{21} = 0$  et  $\bar{x}_2 = \bar{x}_{22} = 0$ .

La matrice

$$A_1 = \frac{\partial f}{\partial x} = \begin{pmatrix} 12x_1 + 2 & 18x_2^2 + 3 \\ 1 & 14x_2 - 1 \end{pmatrix}$$

est calculée, à partir de laquelle, on déduit la matrice

$$A_{11} = \frac{\partial f}{\partial x} \Big|_{x_1=\bar{x}_{11}, x_2=\bar{x}_{12}} = \begin{pmatrix} -4.60 & 3.86 \\ 1 & -4.05 \end{pmatrix}$$

dont les valeurs propres sont  $\lambda_1(A_{11}) = \lambda_{11} = -6.31$  et  $\lambda_2(A_{11}) = \lambda_{12} = -2.34$  (noeud stable) ainsi que la matrice

$$A_{12} = \frac{\partial f}{\partial x} \Big|_{x_1=\bar{x}_{21}, x_2=\bar{x}_{22}} = \begin{pmatrix} 2 & 3 \\ 1 & -1 \end{pmatrix}$$

dont les valeurs propres sont  $\lambda_1(A_{12}) = \lambda_{21} = 2.79$  et  $\lambda_2(A_{12}) = \lambda_{22} = -1.79$  (point selle). Calcul des vecteurs propres pour  $A_{11}$ . On a que  $A_{11}v_j = \lambda_j v_j$ ,  $j = 1, 2$  conduit en posant  $v_{j1} = 1$ , à

$$\begin{aligned} -4.60 + 3.86 \cdot v_{12} &= -6.31 \\ -4.60 + 3.86 \cdot v_{22} &= -2.34 \end{aligned}$$

ce qui donne

$$v_1 = \begin{pmatrix} 1 \\ -0.44 \end{pmatrix} \quad v_2 = \begin{pmatrix} 1 \\ 0.585 \end{pmatrix}$$

et pour  $A_{12}$  :

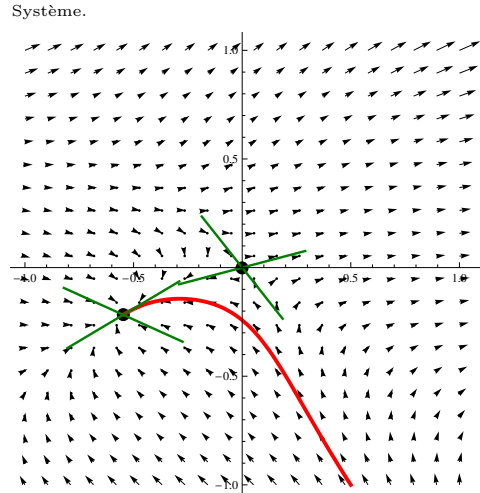
$$\begin{aligned} 2 + 3 \cdot v_{12} &= 2.79 \\ 2 + 3 \cdot v_{22} &= -1.79 \end{aligned}$$

et ainsi

$$v_1 = \begin{pmatrix} 1 \\ 0.26 \end{pmatrix} \quad v_2 = \begin{pmatrix} 1 \\ -1.26 \end{pmatrix}$$

En ce qui concerne l'index, pour autant que la matrice  $A$  ne dégénère pas (pas de valeur propre nulle), l'index du point d'équilibre peut se calculer par

$$\text{sgn det } A,$$



**Figure 1:** Portrait de phase, par champ de vecteurs, du système planaire. Deux équilibres sont présents. L'équilibre de gauche est un noeud stable et celui de droite (origine) est un point selle. L'index du point selle est  $-1$  (que l'on obtient graphiquement) et celui du point d'équilibre gauche est  $+1$  (que l'on obtient aussi graphiquement). En vert, sont représentées les directions des vecteurs propres. En rouge, une trajectoire solution correspondant à la condition initiale  $x_1(0) = 0.5$  et  $x_2(0) = -1$  est dessinée.

sans nécessiter d'avoir recours au calcul topologique. C'est un théorème important qui lie la notion algébrique du déterminant à la notion topologique. On constate, d'une part,  $\text{sgn det } A_{11} = +1$  et, d'autre part,  $\text{sgn det } A_{12} = -1$ , ce qui coïncide étrangement avec les index obtenus graphiquement sur la Figure 1.

### Exercice I.3

Soit la méthode d'intégration numérique de Euler qui consiste à approximer la dérivée par une différence finie :

$$\dot{x} \approx \frac{x(t) - x(t-h)}{h}$$

Utiliser cette technique pour intégrer numériquement l'équation de l'oscillateur

$$\begin{aligned}\dot{x}_1 &= x_2 \\ \dot{x}_2 &= -x_1\end{aligned}$$

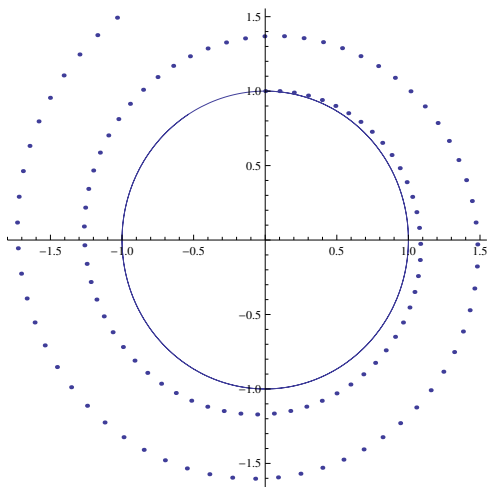
Répéter la simulation en changeant le système en ajoutant une force de frottement proportionnelle à la vitesse. La deuxième équation devient alors

$$\dot{x}_2 = -x_1 - 0.02x_2$$

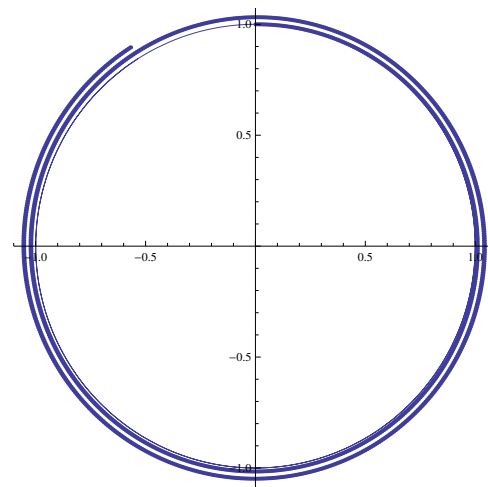
Changer la constante 0.02 pour la rendre soit plus grande ou plus petite et comparer avec la solution exacte (solution analytique). Que constatez-vous ?

**Corrigé :** Dans les deux premières simulations, il n'y pas de frottement. La courbe continue fine représente la vraie trajectoire dans le plan de phase. Les points sont les valeurs obtenues à chaque période d'échantillonnage. Les conditions initiales ne jouent pas un rôle important dans le cas d'un foyer d'un système linéaire, et on a choisi  $x_1(0) = 0$  et  $x_2(0) = 1$ .

Période d'intégration : 0.1 [s].  
La simulation est instable ( $\rightarrow \infty$ )

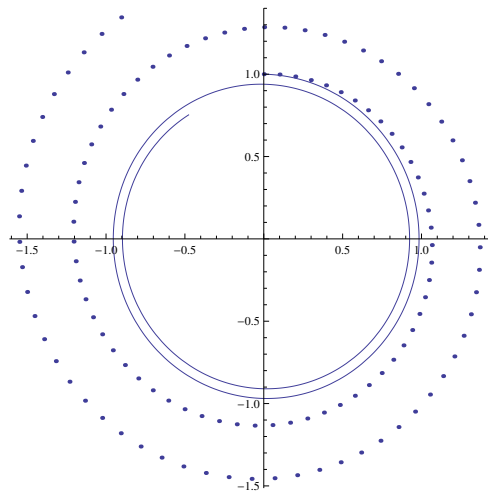


Période d'intégration : 0.01 [s].  
La simulation est instable ( $\rightarrow \infty$ )

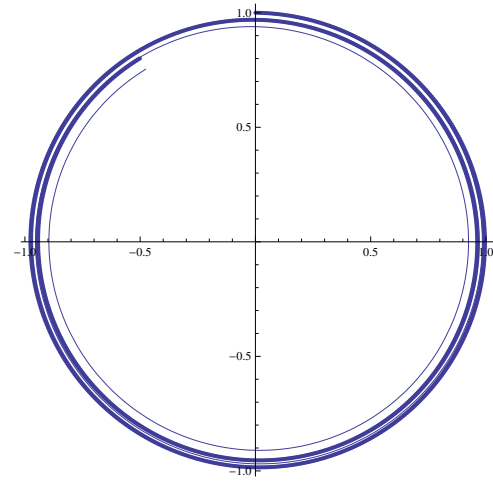


Pour les deux simulations ci-dessous, le coefficient du frottement est de 0.02 ; à nouveau la vraie solution dans le plan de phase est représentée en trait fin et les résultats de la méthode de Euler constituent un ensemble de points (un point par période d'intégration, conditions initiales  $x_1(0) = 0$ ,  $x_2(0) = 1$ ).

Période d'intégration : 0.1 [s]  
La simulation est instable ( $\rightarrow \infty$ )

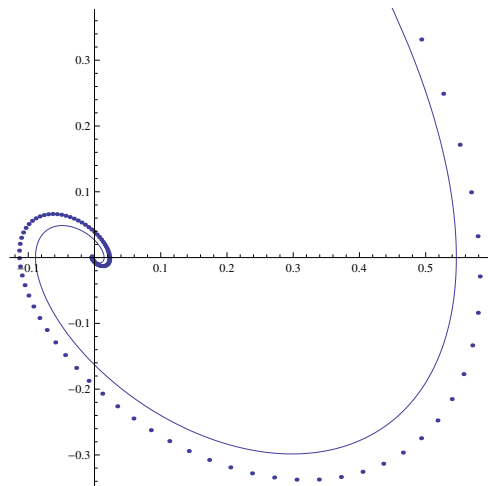


Période d'intégration : 0.01 [s]  
La simulation est stable et la précision est très bonne.

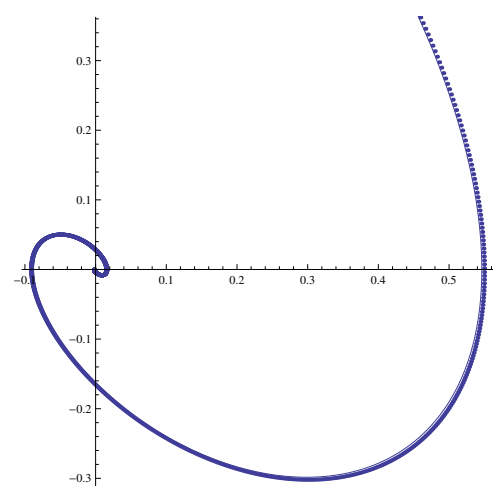


Pour les deux dernières simulations le coefficient de frottement est poussé à 1 et les conditions initiales restent inchangées ( $x_1(0) = 0$ ,  $x_2(0) = 1$ ).

Période d'échantillonnage : 0.1 [s]  
La simulation est stable mais la précision est mauvaise.



Période d'échantillonnage : 0.01 [s]  
La simulation est stable et la précision est très bonne.



Voici le code pour la simulation en Mathematica pour le frottement 0.02 et la période d'intégration de 0.01 [s] (La durée de la simulation est fixée à 12 [s]) :

```
Fosc02 = 0.01 {#2, -#1 - 0.02 #2} + {#1, #2} &
resultats = NestList[Fosc02 @@ # &, {0, 1}, 1200];
plotOsc001temp = ListPlot[resultats, AspectRatio -> 1]
```

Pour l'intégration numérique exacte :

```
solCercle =  
  NDSolve[{x1'[t] == x2[t], x2'[t] == -x1[t], x1[0] == 0,  
    x2[0] == 1}, {x1[t], x2[t]}, {t, 0, 12}]  
pCercle = ParametricPlot[{x1[t], x2[t]} /. solCercle, {t, 0, 12}]
```

Voici en Matlab, il faut créer la fonction Fosc.m contenant

```
function xdot = Fosc(x)  
xdot = [x(2); -x(1)];
```

Puis un autre fichier SimulEuler.m avec

```
x0=[0;1];  
resultats=zeros(2,120);  
resultats(:,1)=x0;  
for i=2:120  
    resultats(:,i) = 0.1*Fosc(resultats(:,i-1)) + resultats(:,i-1);  
end;  
plot(resultats(1,:),resultats(2,:))
```

Finalement taper dans la console : SimulEuler.

### Exercice 1.4

Même question que I.3 mais en utilisant la méthode de Runge-Kutta à pas fixe pour intégrer l'équation

$$\dot{x} = f(x, t)$$

La méthode consiste à calculer quatre grandeurs intermédiaires  $k_1$ ,  $k_2$ ,  $k_3$  et  $k_4$  pour effectuer un pas d'intégration :

$$x_{k+1} = x_k + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \quad \text{avec} \quad t_{k+1} = t_k + h$$

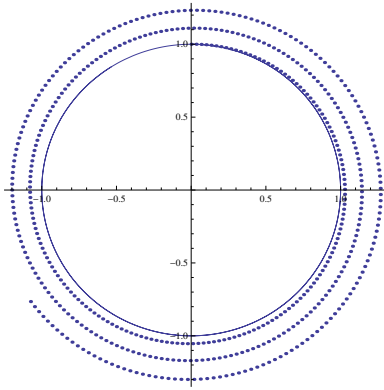
et

$$\begin{aligned} k_1 &= hf(x_k, t_k) \\ k_2 &= hf\left(x_k + \frac{1}{2}k_1, t_k + \frac{1}{2}h\right) \\ k_3 &= hf\left(x_k + \frac{1}{2}k_2, t_k + \frac{1}{2}h\right) \\ k_4 &= hf(x_k + k_3, t_k + h) \end{aligned}$$

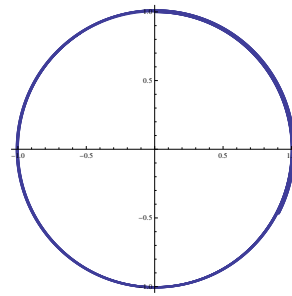
Comparer vos résultats avec la commande 'ode45' de MATLAB.

**Corrigé :** Les six figures suivantes sont les figures correspondantes aux six figures de l'exercice précédant. Il faut constater que les résultats s'améliorent avec l'intégrateur de Runge Kutta. Cependant il y a toujours de très grandes différences lorsque le pas d'intégration est choisi de manière trop grande (en l'occurrence 0.1). Le pas d'intégration de 0.01 donne de très bons résultats mais il subsiste encore un décalage avec l'intégrateur à pas variable utilisé dans les intégrateurs plus sophistiqués. Le code pour  $h = 0.1$  et  $b = 0.02$  en Mathematica est donné ci-dessous :

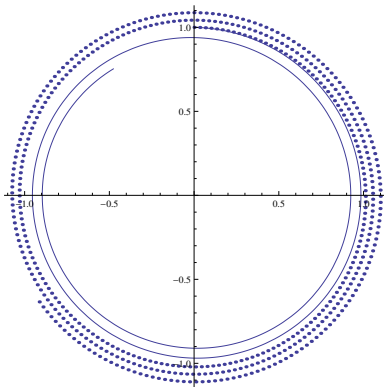
```
resultats = NestList[Module[{xk, h, Fosc},
  xk = #1;
  h = #2;
  Fosc = {#2, -#1 - 0.02 #2} &;
  {xk +
    Inner[#1 #2 &, 1/6 {1, 2, 2, 1},
      Append[#, Last[Fosc @@ (xk + #) &]] & [
        NestList[h Fosc @@ (xk + 1/2 #) &, {0, 0},
          2]] , #1 + #2 &], h} ] & @@ # &, {{0, 1}, 0.1}, 500];
Show[ListPlot[Part[#, 1] & /@ resultats, AspectRatio -> 1], pCercle]
```



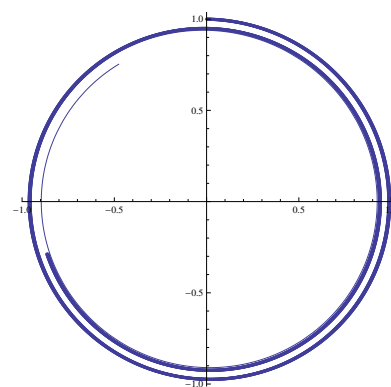
$h = 0.1 \quad b = 0$



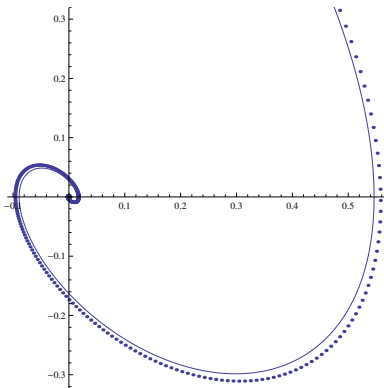
$h = 0.01 \quad b = 0$



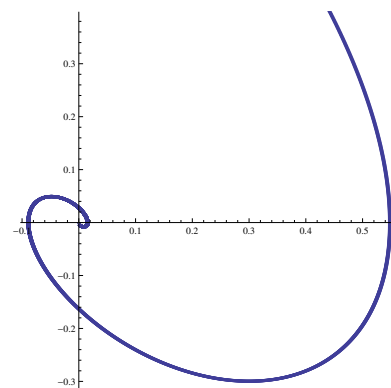
$h = 0.1 \quad b = 0.02$



$h = 0.01 \quad b = 0.02$



$h = 0.1 \quad b = 1$



$h = 0.01 \quad b = 1$

En MATLAB :

Il faut sauver le texte suivant dans un fichier nommé `SimulRK.m`

`x0=[0;1];`



```

resultats=zeros(2,1200);
resultats(:,1)=x0;
h = 0.01;
for i=2:120
    xk = resultats(:,i-1);
    k1 = h*Fosc(xk);
    k2 = h*Fosc(xk+0.5*k1);
    k3 = h*Fosc(xk+0.5*k2);
    k4 = h*Fosc(xk+k3);
    resultats(:,i) = xk + 1/6*(k1+2*k2+2*k3+k4);
end;
plot(resultats(1,:),resultats(2,:))

```

et taper dans la console : SimulRK