Équation de VAN DER POL

L'**équation de Van der Pol** est une équation différentielle du second ordre de la forme :

$$y'' - c(1-y^2)y' + y = 0$$

où c est une constante. Elle est équivalente au système différentiel :

$$\begin{cases} x'(t) &= c(1-y^2(t))x(t) - y(t) \\ y'(t) &= x(t) \end{cases}$$

Sa traduction en code Scilab, pour $c = 0.5$, est :

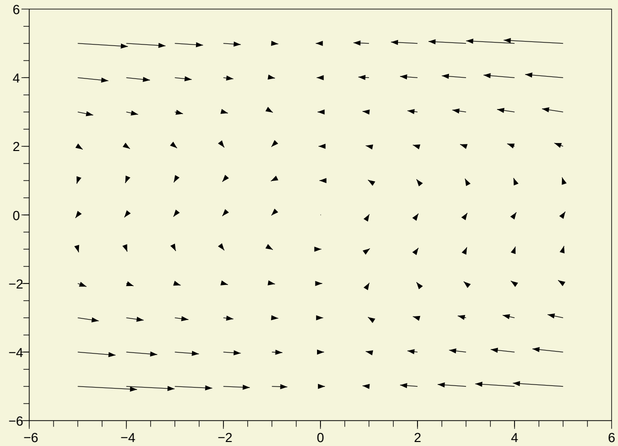
```
--> function [Xprime] = VanDerPol(t,X)
-->   Xprime(1) = 0.5 * (1-X(2)^2) * X(1) - X(2)
-->   Xprime(2) = X(1)
--> endfunction
```

En réalité la forme que nous donnons là est une forme simplifiée de l'équation qui est utilisée en physique pour modéliser un *oscillateur entretenu*. Cette équation n'est pas linéaire et nous ne pouvons pas en donner de solution explicite.

Pour des compléments, voir l'article sur Wikipédia ; plus généralement une recherche sur *Internet* livre une littérature abondante sur le sujet ainsi que des animations.

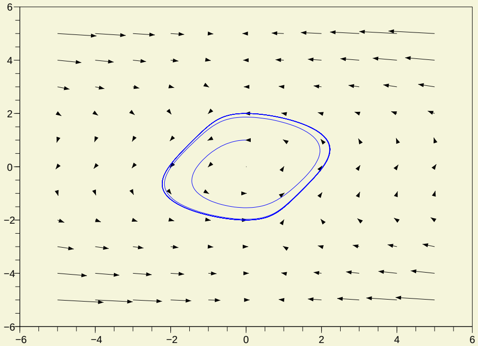
Représentons le champ de vecteur dans l'espace des phases (x, y) :

```
--> s=linspace(-5,5,11);
--> fchamp(VanDerPol,0,s,s)
```



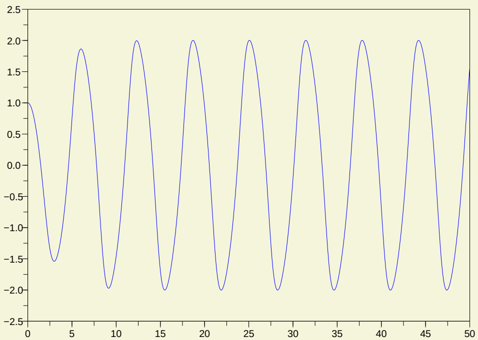
Superposons la solution avec les conditions initiales $x(0) = 0$, $y(0) = 1$, autrement dit $y(0) = 1$ et $y'(0) = 0$.

```
--> t=linspace(0,50,1000); X0=[0;1];
--> [u]=ode(X0,0,t,VanDerPol);
--> plot(u(1,:)',u(2,:)',2)
```



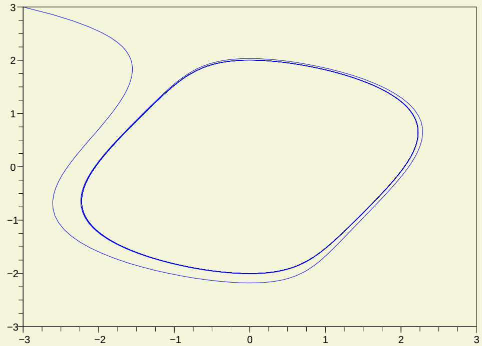
Représentons la solution y en fonction de t :

```
--> clf; plot(t,u(2,:))
```



Comme nous pouvons le constater la solution déterminée tend vers un *cycle limite* dans l'espace des phases, elle devient quasiment *périodique*. C'est la cas, pour la valeur de c choisie ici, quelles que soient les conditions initiales.

```
--> clf;
--> X0=[-3;3];
--> [u]=ode(X0,0,t,VanDerPol);
--> plot(u(1,:)',u(2,:)',2);
```



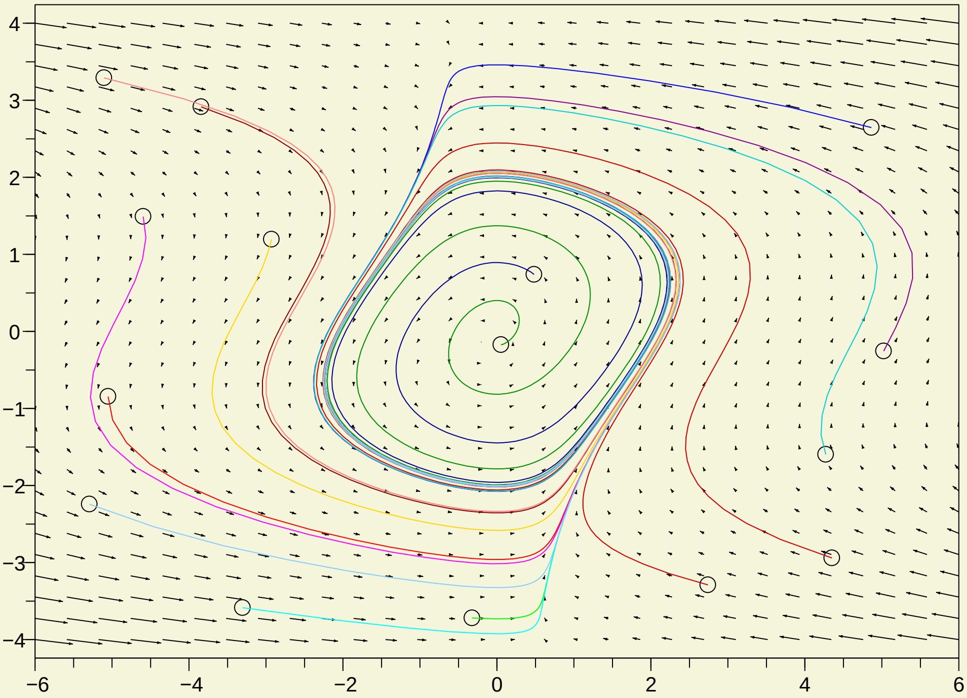
Représentons plusieurs solutions sur le même champ.

1. Tracé du champs de vecteur issu de l'équation de VAN DER POL :

```
--> n = 30; dx = 6; dy = 4;
--> x = linspace(-dx,dx,n);
--> y = linspace(-dy,dy,n);
--> clf();
--> fchamp(VanDerPol,0,x,y,1,[-dx,-dy,dx,dy],"031")
--> xselect()
```

2. Résolution de l'équation différentielle :

```
--> m = 500 ; T = 30; t = linspace(0,T,m);
--> couleurs = [21 2 3 4 5 6 19 28 32 9 13 22 18 21 12 30 27];  // 17 couleurs
--> num = -1;
--> while %t
-->  [c_i,c_x,c_y]=xclick();
-->  if c_i == 0 then
-->    plot2d(c_x, c_y, -9, "000")
-->    u0 = [c_x;c_y];
-->    [u] = ode(u0, 0, t, VanDerPol);
-->    num = modulo(num+1,length(couleurs));
-->    plot2d(u(1,:)',u(2,:)',couleurs(num+1),"000")
-->  elseif c_i == 2 then
-->    break
-->  end
--> end
```



Ce document est largement inspiré de celui que vous trouverez ici :
http://www.iecn.u-nancy.fr/~pincon/scilab/Doc/node80.html
En particulier, le dernier segment de code permettant de choisir, à la souris, les conditions initiales est repris intégralement.