Scilab

→**utiliser l'interface de développement de Scilab (version 5.4.1.)**

→ **fenêtre de commande, éditeur de scripts(Scinotes) , appel de fonction …**

→ **créer des signaux discrets dans les vecteurs et tableaux Scilab**

→ **écrire et utiliser scripts et fonctions Scilab.**

Avril 2014

Liberté • Égalité • Fraternité
RÉPUBLIQUE FRANÇAISE

académie Nantes

Scilab

Le matin applications diverses sur :

→Etude d'un signal temporel et fréquentiel

→ Courbes paramétriques

→ Equations différentielles (pendule)

→ Transformée de Fourier (diffraction d'une fente rectangulaire avec une approche physique puis mathématique)

→ Propagation des ondes (progressive et stationnaire).

→ L'après midi sera consacré à des exercices d'applications (Physique , Math et Sti)

Avril 2014

Scilab

- **Matlab (Matrix Laboratory) est un logiciel très connu et très utilisé dans le domaine du calcul scientifique industriel.**
- **On utilise ici Scilab (pour Science Laboratory) qui n'a rien à lui envier, mais qui est :**
  - **→ libre et gratuit, open source, maintenu et évolutif.**
  - **→ doté d'une large communauté d'utilisateurs**
  - **→ utilisable hors de l'école et dans l'école, sous Windows et/ou Unix/Linux.**
  - **→ très semblable à Matlab.**

Liberté • Égalité • Fraternité
RÉPUBLIQUE FRANÇAISE

académie Nantes

Bureau et environnement de Scilab

Diffraction d'une fente.sci (G:\Scilab\programme\Diffraction d'une fente.sci) - SciN

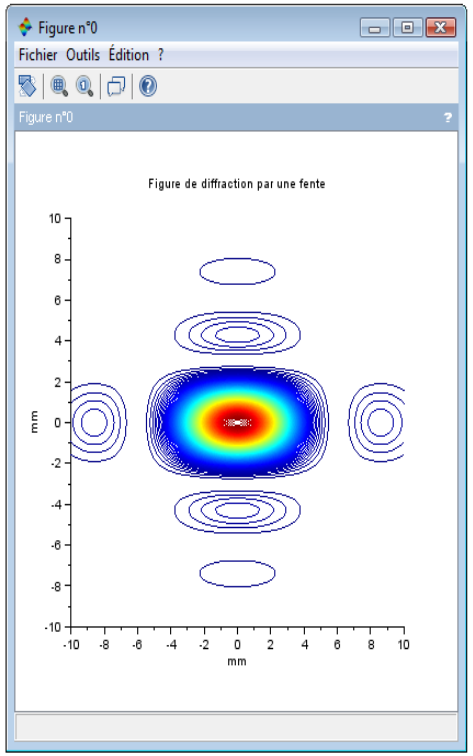
FR Français (France)

Fichier Édition Format Options Fenêtre Exécuter ?

Diffraction d'une fente.sci (G:\Scilab\programme\Diffraction d'une fente.sci) - SciNotes

Diffraction d'une fente.sci

```
clear;
function [I] = Intensite(x,y)
  I = (sinc((K*a*x))^2)*(sinc((K*b*y))^2);
endfunction;
printf("Diffraction par une fente rectangulaire\n\n");
a = 0.1//("largeur de la fente (en mm) : ");
b = 0.2//("hauteur de la fente (en mm) : ");
lambda = 600//("longueur d''onde (en nm) : ");
D = 1.//("distance de l''écran (en m) : ");
K = %pi/(D*10^3*lambda*10^-6);
//Paramètres de visualisation
NN = 100;    //nombre de niveaux
XMIN = -10;
XMAX = 10;
YMIN = -10;
YMAX = 10;
PAS = (XMAX - XMIN)/100;
//Définition des variables
x = XMIN:PAS:XMAX;
y = YMIN:PAS:YMAX;
for i=1:length(x)
  for j=1:length(y)
    z(i,j) = Intensite(x(i),y(j));
  end;
end;
//Tracé de la courbe de niveau;
clf();f=gcf();
xset("fpf"," ");
f.color_map = jetcolormap(NN);
contour2d(x,y,z,0:1/NN:1,frameflag=3, rect=[XMIN YMIN XMAX YMAX]);
//grayplot(x,y,z,rect=[XMIN YMIN XMAX YMAX]);
xtitle("Figure de diffraction par une fente","mm","mm");
```



Avril 2014

Liberté • Égalité • Fraternité
RÉPUBLIQUE FRANÇAISE

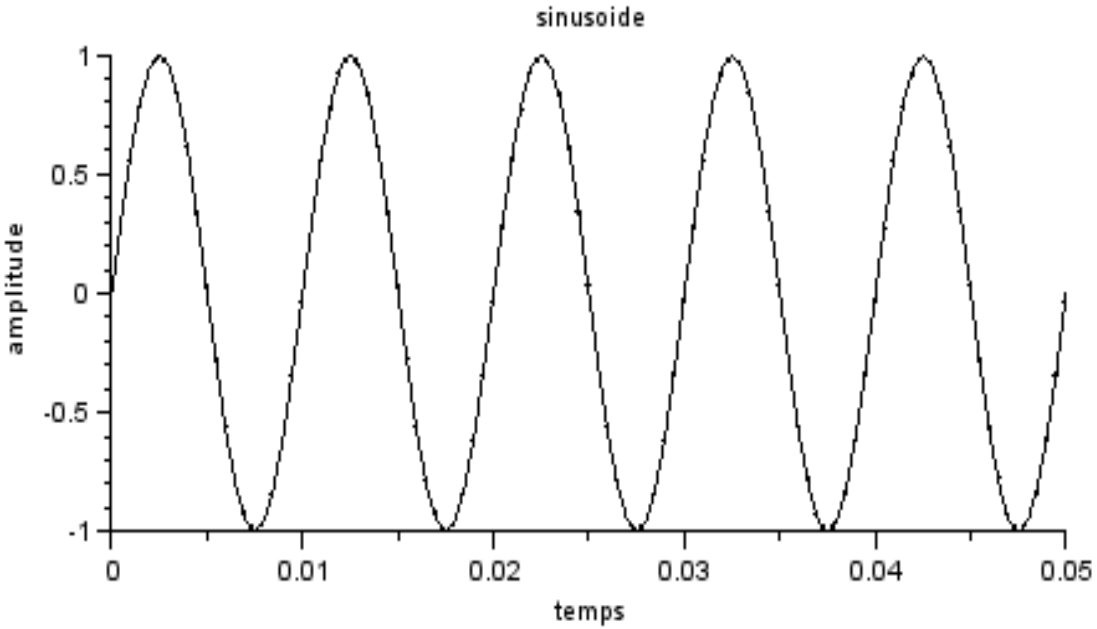
académie Nantes

Etude temporelle d'un signal sinusoidal

```
1 Te=5e-4                  //période d'échantillonnage fe=2 kHz) //
2 nbpts=100                //nbrs affichés sur la courbe//
3 t=Te*(0:1:nbpts)  //0 temps initial
                            1 pas d'incrémentation (1*Te))//
4 A=1;f=100                //Amplitude 1V et fréquence 100 Hz
    condition de Shannon respecté)//
5 y=A*sin(2*%pi*f*t) //Signal sinusoidal//
6 plot2d(t,y)              //représentation temporelle//
7 xtitle( "sinusoide","temps","amplitude");
```

Avril 2014

Etude temporelle d'un signal sinusoidal



Avril 2014

Liberté • Égalité • Fraternité
RÉPUBLIQUE FRANÇAISE

académie Nantes

Etude fréquentielle d'un signal sinusoidal

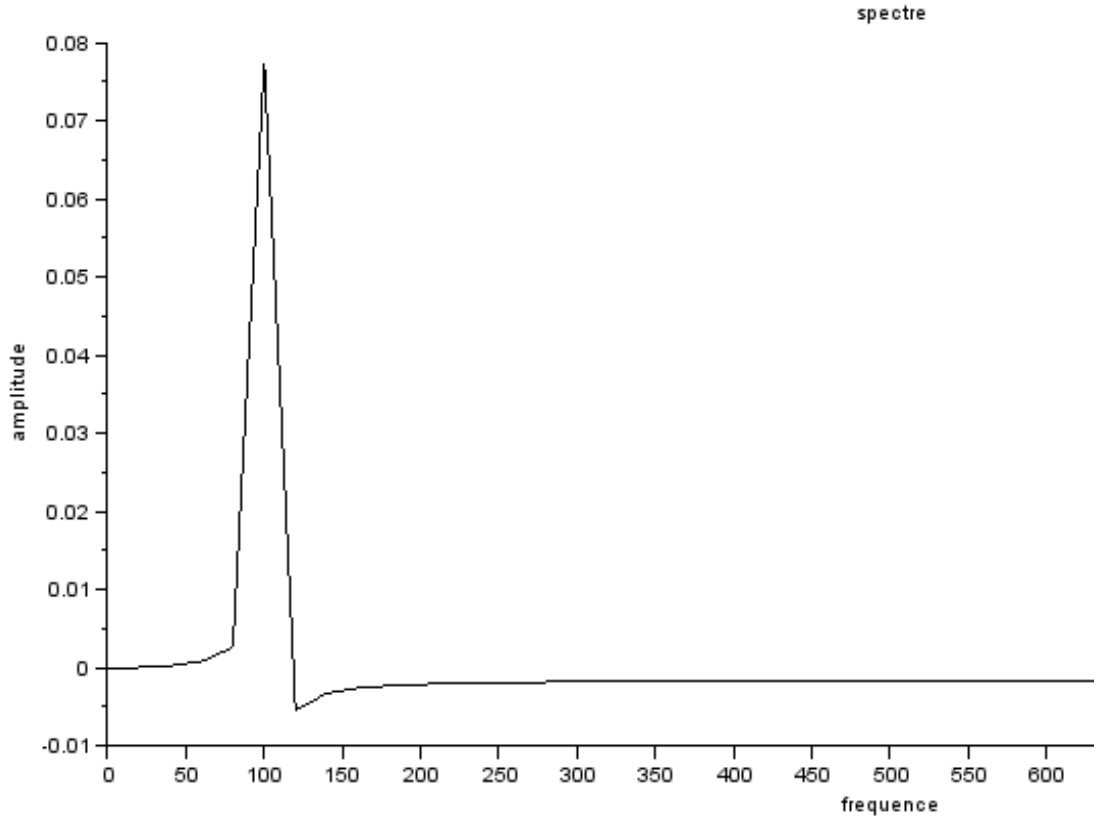
```
7 n=100                          //nbrs de points de calculs //
8 freq =1/(n*Te)*(0:n-1) //définition de la variable freq//
9 spectre=fft(y,-1)        //calcul du spectre fréquentiel//
10 plot2d(freq (1:n/2,1/n*spectre(1:n/2)
7 xtitle( « Analyse spectrale",
    " fréquence","amplitude");
```

Avril 2014

Liberté • Égalité • Fraternité
RÉPUBLIQUE FRANÇAISE

académie Nantes

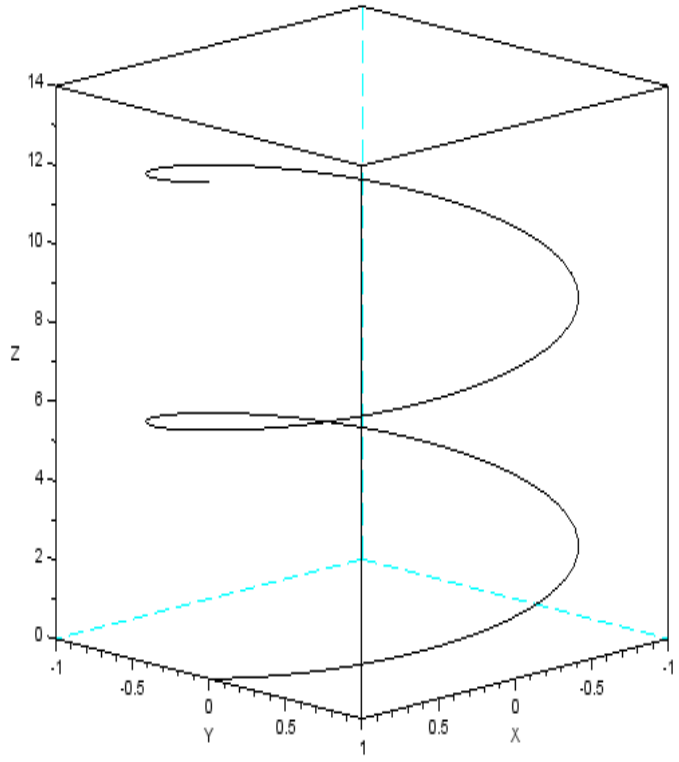
Etude fréquentielle d'un signal sinusoidal



Avril 2014

Courbes paramétriques

```
1 t=linspace(0,4*%pi ,100) ;
//définition de la variable t //
2 X=cos(t);
3 y=sin(t);
4 param3d(x,y,t);
 //Tracé de la courbe dans l'espace//
```



Avril 2014

Liberté • Égalité • Fraternité
RÉPUBLIQUE FRANÇAISE

académie Nantes

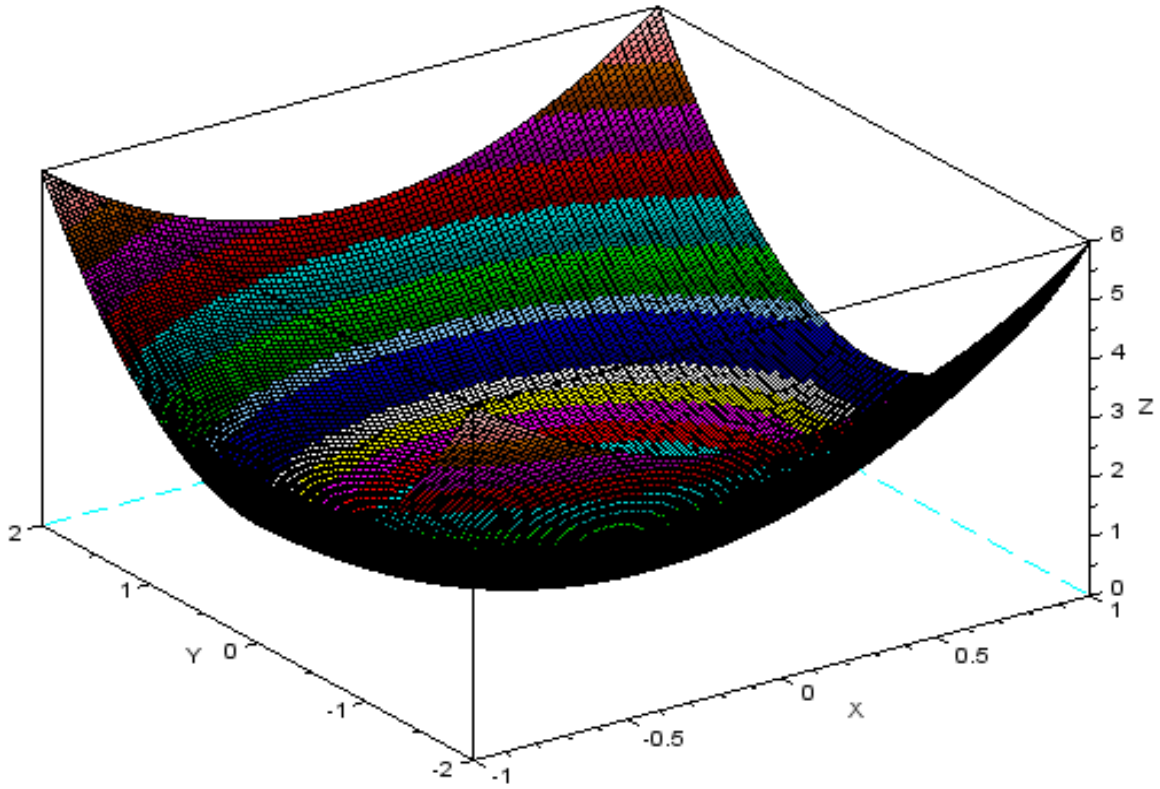
Surface d'une fonction en 3 dimensions

```
1 function z=f(x, y)        //définition de la fonction//
2 z=2*x^2+y^2 ;
3 endfunction
4 x=linspace(-1,1,100) //paramètres de calculs//
5 y=linspace(-2,2,200) //paramètres de calculs//
6 z=feval(x,y,f)';
7 clf;                      //efface l'écran (clear figure)//
8 surf(x,y,z);              //tracer de la surface//
```

Liberté • Égalité • Fraternité
RÉPUBLIQUE FRANÇAISE

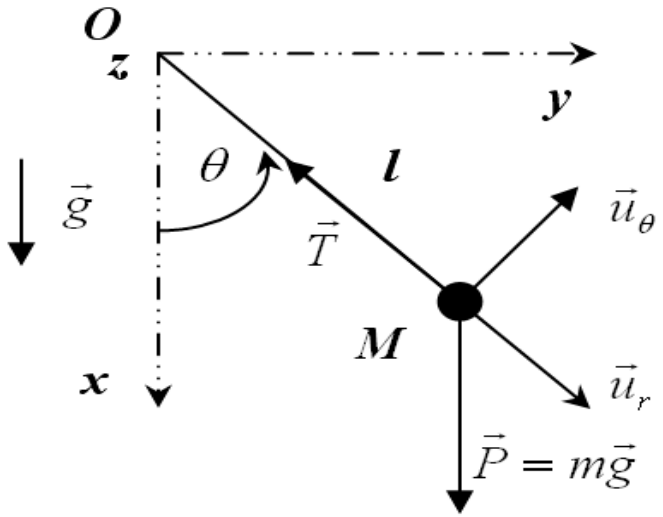
académie Nantes

Surface d'une fonction en 3 dimensions



Avril 2014

Oscillateur harmonique (équations différentielles)



$$\ddot{\theta} + \omega_o^2\theta = 0 \quad avec \quad \omega_o = \sqrt{\frac{g}{l}}$$

équation du type y''(t)+ω0^2*y(t)=0

On associe y'à yprim(1) , y'' à yprim(2) , y(1) à y et y(2) à y'.

On associe une matrice A 2×2

$$\begin{pmatrix} y' \\ y'' \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} y \\ y' \end{pmatrix}$$

Soient 2 équations linéaires

$y'=a_{11}y+a_{12}y'$ et $y''=a_{21}y+a_{22}y'$

soit pour notre équation $a_{11}=0$ $a_{12}=1$

$a_{21}=-\omega o^2$ et $a_{22}=0$

Oscillateur harmonique (équations différentielles)

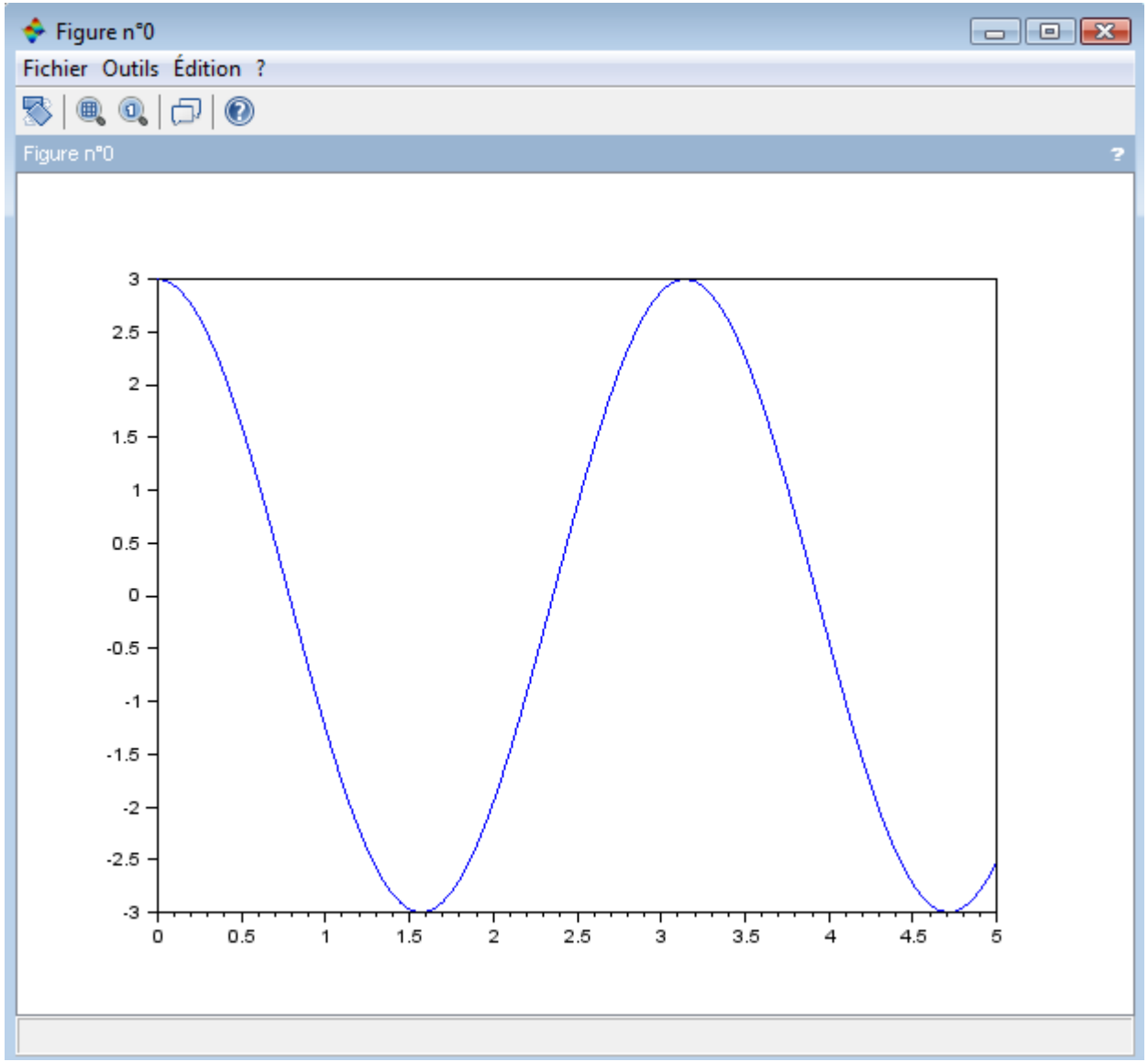
```
1 function yprim=f(t, y)      //définition de la fonction//
2         yprim(1)=y(2);      //   a12=1      //
3         yprim(2)=-4*y(1);   //   a21=-ωo^2  //
4 endfunction
5 t0=0; tmax=5;
6 t=t0:0.05:tmax;
7 y0=3; yprim0=0;
8 y=ode([y0;yprim0],t0,t,f);
9 clf; plot(t,y(1,:))
```

Avril 2014

Liberté • Égalité • Fraternité
RÉPUBLIQUE FRANÇAISE

académie Nantes

Oscillateur harmonique (équations différentielles)



Avril 2014

Diffraction d'une fente. (transformée de Fourier)

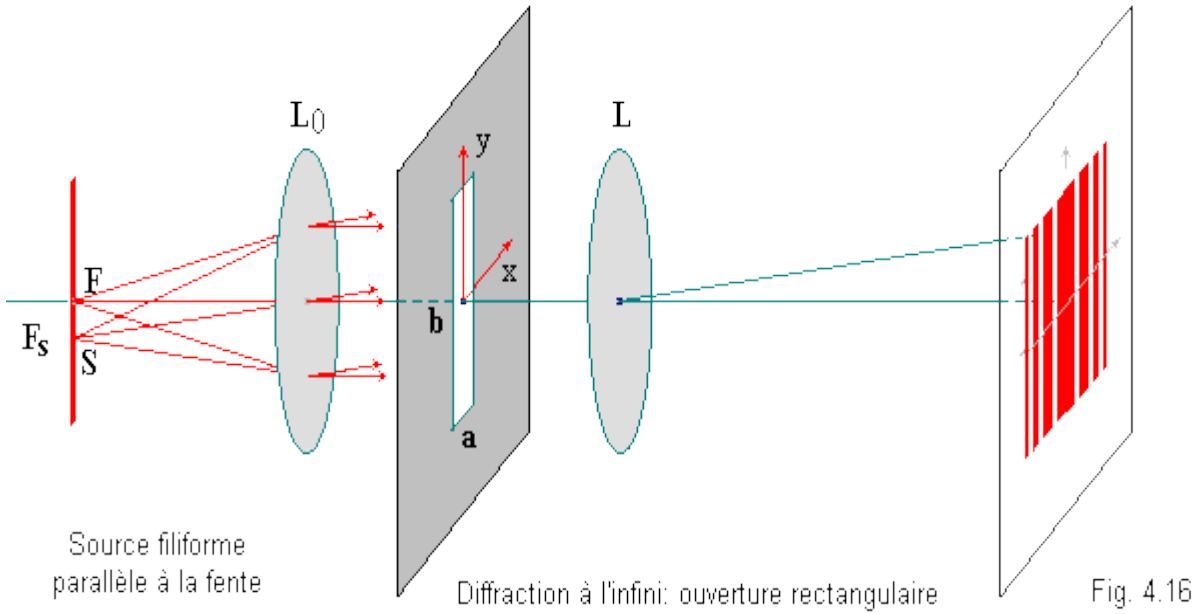


Fig. 4.16

Avril 2014

Liberté • Égalité • Fraternité
RÉPUBLIQUE FRANÇAISE

académie Nantes

Diffraction d'une fente.
(transformée de Fourier)

```
1 clear;
2 function [I]=Intensite(x, y) // Calcul de l'intensité lumineuse en
                                  fonction de la position //
3 I = (sinc((K*a*x))^2)*(sinc((K*b*y))^2);
                                  // b, a : longueur et largeur de la fente//
4 endfunction;
5 printf("Diffraction par une fente rectangulaire\n\n");
6 a = 0.1                         //("largeur de la fente (en mm): ")//;
7 b = 0.2                         //("hauteur de la fente (en mm) : ")//;
8 lambda = 600                    //("longueur d''onde (en nm) : ")//;
9 D = 1                           //("distance de l''écran (en m) : ")//;
10 K = %pi/(D*10^3*lambda*10^-6);
11 NN = 100;                      // nombre de niveaux//
12 XMIN = -10;XMAX = 10;YMIN = -10;YMAX = 10;
13 PAS = (XMAX - XMIN)/100;
14 x = XMIN:PAS:XMAX;
15 y = YMIN:PAS:YMAX;
```

Avril 2014

Liberté • Égalité • Fraternité
RÉPUBLIQUE FRANÇAISE

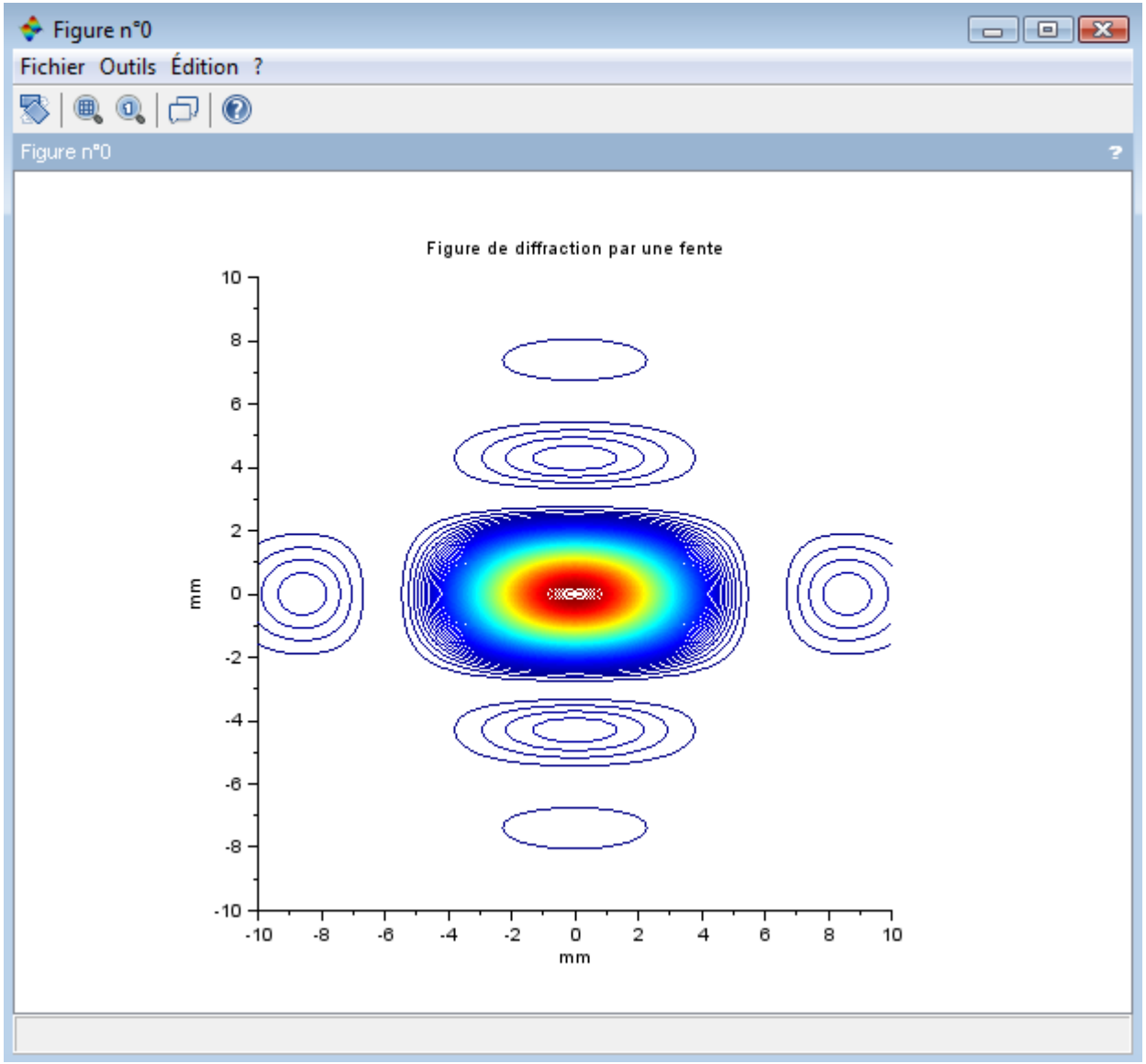
académie Nantes

Diffraction d'une fente. (transformée de Fourier)

```
16 for i=1:length(x)
17       for j=1:length(y)
18 z(i,j) = Intensite(x(i),y(j));
19       end;
20 end;
21 // Tracé de la courbe de niveau//
22 clf();f=gcf();
23 xset("fpf"," ");
24 f.color_map = jetcolormap(NN);
25 contour2d(x,y,z,0:1/NN:1,frameflag=3, rect=[XMIN YMIN XMAX
    YMAX]);
26 //grayplot(x,y,z,rect=[XMIN YMIN XMAX YMAX]);
27  xtitle("Figure de diffraction par une fente","mm","mm");
```

Avril 2014

Diffraction d'une fente. (transformée de Fourier)



Avril 2014

Approche Mathématique. (transformée de Fourier)

```
1 // TFD
2 // définition d'une fonction "porte" de largeur 1
3 function s=f(t)
4         if t<0 then s=0
5         else if t<1 then s=1
6                         else s=0
7                 end
8         end
9 endfunction
10 // ---------------------------------------------
11 utilisateur=x_mdialog('intervalle [a;b] et fréquence
d''échantillonnage',['a';'b';'fe'],['0';'4';'100']);
12 a=evstr(utilisateur(1));
13 b=evstr(utilisateur(2));
14 fe=evstr(utilisateur(3));
15 // nombre d'échantillon N du signal
16 N=(b-a)*fe;
```

Liberté • Égalité • Fraternité
RÉPUBLIQUE FRANÇAISE

académie Nantes

Diffraction d'une fente. (transformée de Fourier)

```
17 // vecteur X contenant les échantillons
18 X=zeros(1,N);
19 for k=1:N
20       X(k)=f(a+(k-1)/fe);
21 end
22 // vecteur Y contenant la TFD de la suite des valeurs dans X
23 Y=zeros(1,N);
24 v=exp(-2*%i*%pi/N);
25 w=1;
26 for n=1:N
27       wk=1;
28       for k=1:N
29       Y(n)=Y(n)+wk*X(k);
30       wk=wk*w;
31       end
32 w=w*v
33 end
```

Avril 2014

académie Nantes

Approche Mathématique. (transformée de Fourier)

```
34 // ------------------------------------------------------
35 // module des complexes de la suite Y
36 Z=abs(Y);
37 // ------------------------------------------------------
38 // échantillon F des fréquences
39 F=zeros(1,N);
40 for k=1:N
41       F(k)=(k-1)*fe/N;
42 end
43 // graphique
44 clf(0);
45 figure(0);
46 plot2d(F,Z); par une fente","mm","mm");
```

Propagation d'une onde. (Onde progressive)

Le cas que nous présentons ici est la propagation d'une onde de marée de période de 12 heures dans un canal infini (ici limité à m, ceci pour illustrer plusieurs longueurs d'onde sur une même figure).

Equation de propagation de l'onde d'amplitude h : $\frac{\partial^2 h}{\partial t^2} - gh\frac{\partial^2 h}{\partial x^2} = 0$

avec une célerité $c = \sqrt{gH}$

Une solution de cette équation pour l'évolution de la **surface libre** est la suivante: $h(x,t) = a\cos(kx - \sigma t)$

Les caractéristiques de cet exemple sont les suivants:

→période de la marée : 12 heures

→amplitude des vagues : a=5m

→profondeur : H=500 m (la profondeur sur les représentations graphiques a été limitée à 50 m pour des questions de lisibilité)

→Les calculs conduisent à : longueur d'onde : 2897 m , célérité : 67 m/s

Liberté • Égalité • Fraternité
RÉPUBLIQUE FRANÇAISE

académie Nantes

Propagation d'une onde.
(Onde progressive)

```
1 // propagation d'onde en eaux peu profondes
2 // Canal ouvert à droite (istat=0) ou stationnaire (istat=1)
3 clf()
4 LONG=10000000; // longueur du canal
5 fact=1000000 // facteur d'échelle pour représentation des vecteurs vitesse
6 istat=0 // onde stationnaire (0) ou non (1)
7 a=5;// amplitude de l'onde
8 yminaffich=-50 //affiche la profondeur jusqu'à une profondeur de 50 m
9 b=get("current_axes");
10 b.data_bounds=[0,yminaffich;LONG/fact,6];
11 d=gca() //définition des axes
12 d.data_bounds=[0,-50;10,20];
13 d.axes_visible = 'on';
14 d.labels_font_size=2;
15 d.x_label.font_size = 3;d.y_label.font_size = 3;
16 d.x_label.text="abscisse en millions de km "
17 d.y_label.text="profondeur d''eau totale : 500 m"
```

Avril 2014

Liberté • Égalité • Fraternité
RÉPUBLIQUE FRANÇAISE

académie Nantes

Propagation d'une onde. (Onde progressive)

```
18 // fixer la couleur du champ de vecteur à "blanc"
19 f=gcf()
20 f.color_map=wintercolormap(32)
21                    // définit les dimensions de la figure
22 f.figure_position
23 f.figure_size=[1500,500]
24                    // paramètres
25 title('Propagation de la marée en milieu infini','position',[0.5
     20],'fontsize',5)
26                     // données
27 g=9;81             // gravite
28 T=12*3600;     // période de l'onde
29 d=500;           //profondeur d'eau
30 C=sqrt(g*d);  // célérité de l'onde
31 L=T*C;           //longueur d'onde
32 k=2*%pi/L ;    // nombre d'onde
33 x=0:LONG/1000:LONG //discrétisation suivant x
```

Avril 2014

Propagation d'une onde.
(Onde progressive)

```
34 sigma=2*%pi/T                          // fréquence
35 coefu=a*g*k/sigma                      // utilisé pour la vitesse
36          //couleurs des aires des courbes
37 id1=color('white')
38 id2=color(0,191,255)
39 i=0
40 for t=0:3600:12*3600
41          // fixer la couleur du champ de vecteur à "blanc
42 i=i+1
43 if i<>1 then yprec=y; end
44 y=a*cos(k*x-sigma*t)                          //Onde progressive
45 if istat==1 then y=y+a*cos(k*x+sigma*t) //Onde stationnaire
46 end
47 if i==1 then yprec=y; end
48 xfpolys([x'/fact;LONG/fact;0],[yprec';yminaffich;yminaffich],[id1])
49 plot(x'/fact,yprec',"w")
50 xfpolys([x'/fact;LONG/fact;0],[y';yminaffich;yminaffich],[id2])
```

Liberté • Égalité • Fraternité
RÉPUBLIQUE FRANÇAISE

académie Nantes

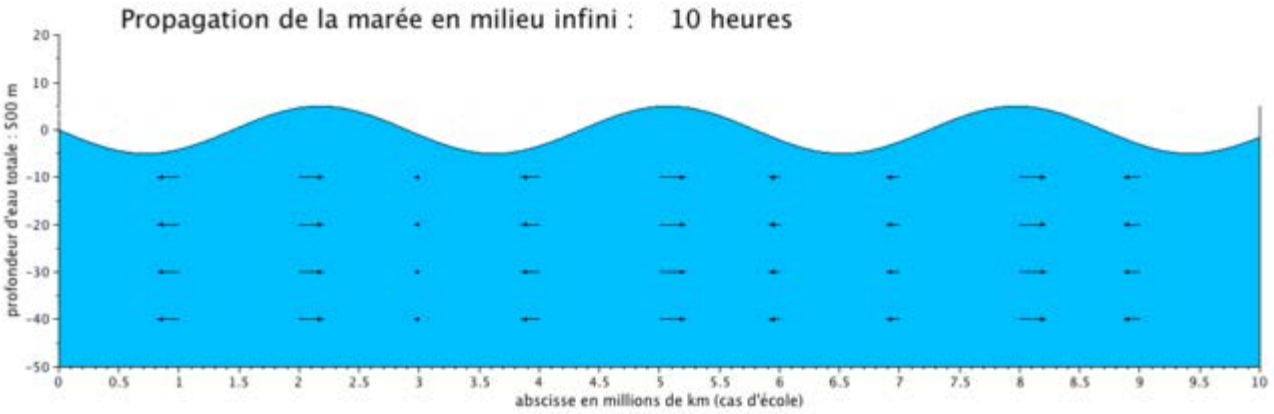
Propagation d'une onde. (Onde progressive)

```
51 deltay=max(y,yprec)
52 num=string(i)
53 xpause(100000);
54 title('Propagation de la marée en milieu infini :...
55 '+num+ ' heures','position',[0.5 20],'fontsize',5)
56                    // dessin des vecteurs vitesse
57 xvect=[1 2 3 4 5 6 7 8 9 20];yvect=[-40:10:-10]
58 fvarx=coefu*cos(k*xvect*fact-sigma*t);
59 if istat==1 then fvarx=fvarx-coefu*cos(k*xvect*fact+sigma*t); end
60 fvarx(10)=3;
61 fx=fvarx'*ones(1,4);fy=zeros(10,4);
62                    // tracé des vecteurs vitesse
63 champ(xvect',yvect',fx,fy,arfact=1,rect=[0,-50,10,20])
64                    //GIF export
65 xs2gif(0,'Marée_'+string(i)+'.gif');
66 end
```

Liberté • Égalité • Fraternité
RÉPUBLIQUE FRANÇAISE

académie Nantes

Propagation d'une onde. (Onde progressive)

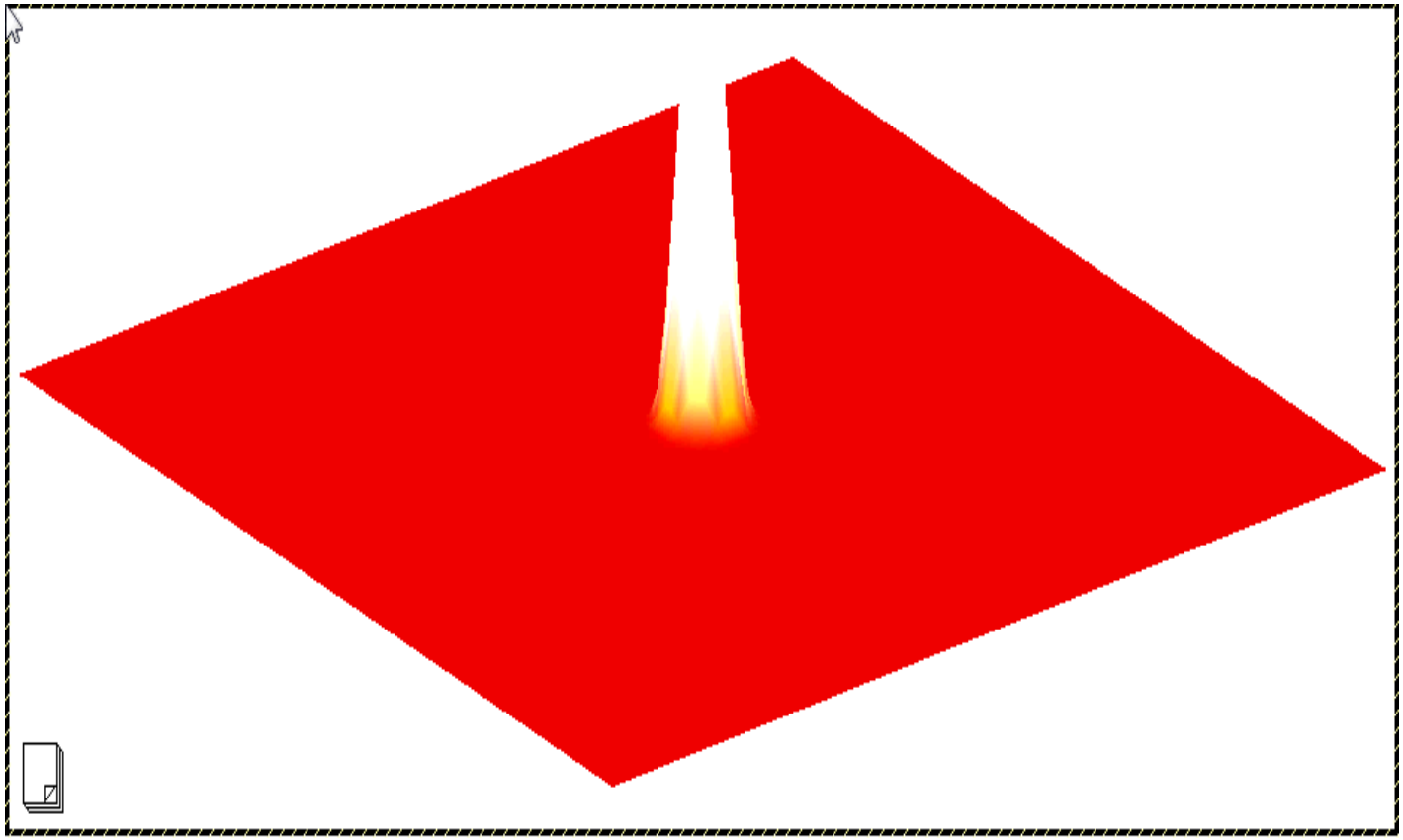


Avril 2014

Liberté • Égalité • Fraternité
RÉPUBLIQUE FRANÇAISE

académie Nantes

Propagation d'une onde. (Onde progressive)



Avril 2014

Liberté • Égalité • Fraternité
RÉPUBLIQUE FRANÇAISE

académie Nantes

XCOS.
(modélisation sous forme d'icones)

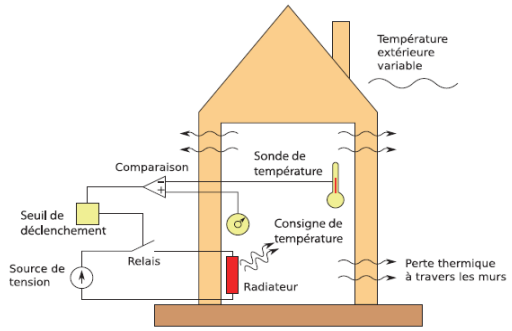
→ **Xcos est l'outil de Scilab dédié à la modélisation et à la simulation de systèmes dynamiques hybrides incluant à la fois des modèles continus et discrets.**

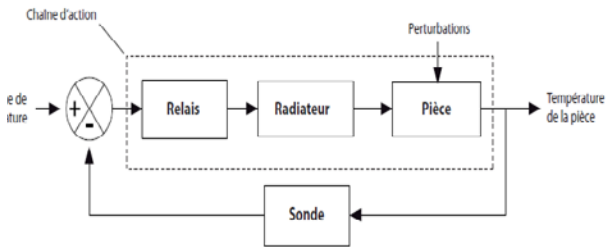
→ **Xcos inclut un éditeur graphique permettant de représenter facilement des modèles sous forme de schémas fonctionnels (diagrammes) en connectant des blocs entre eux.**

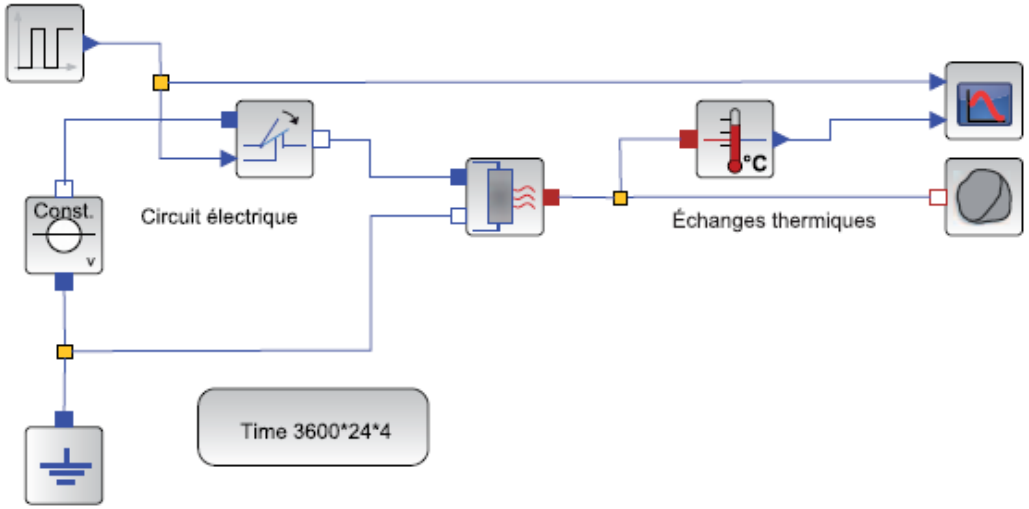
→ **Chaque bloc représente une fonction de base prédéfinie ou une fonction définie par l'utilisateur.**

Avril 2014

XCOS.
(modélisation sous forme d'icones)







Avril 2014