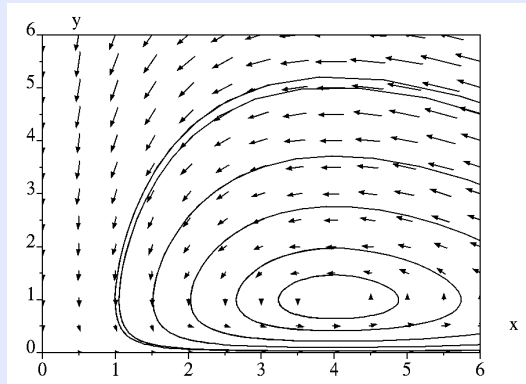




Résolution d'équations différentielles avec "SCILAB"

Marc Buffat



Université Claude Bernard Lyon I
UFR de Mécanique

ODE
PROBLEMES...
SCILAB Ex.
DAE

[Home Page](#)

[Title Page](#)



Page 1 of 22

[Go Back](#)

[Full Screen](#)

[Close](#)

[Quit](#)

1. ODE

Soit le système de n équations différentielles d'ordre 1

$$\begin{aligned}\frac{dY}{dt} &= F(Y,t) \\ Y(0) &= Y_0\end{aligned}\tag{1}$$

$Y(t)$ Y_0 et $F(Y,t)$ sont des vecteurs de dimension n

Remarque: une EDP d'ordre N

$$y^{(n)} = f(y, y', y'', \dots, y^{(n-1)})$$

se transforme en un système de N EDP d'ordre 1 avec le changement de variables

$$Y_1 = y, Y_2 = y', Y_3 = y'', \dots, Y_n = y^{(n-1)}$$

et le système à résoudre s'écrit

$$Y_1' = Y_2$$

$$Y_2' = Y_3$$

...

$$Y_n' = f(Y_1, Y_2, \dots, t)$$



ODE

PROBLEMES...

SCILAB Ex.

DAE

Home Page

Title Page

◀◀ ▶▶

◀ ▶

Page 2 of 22

Go Back

Full Screen

Close

Quit

1.1. Méthode d'Euler

Approximation de $Y(x)$ sur $[a,b]$
subdivise l'intervalle $[a,b]$, pas d'intégration h :

$$h = \frac{b - a}{N}$$

on note Y_n la solution approchée de $Y(x_n)$
La suite Y_n est obtenue par la relation de récurrence

$$Y_0 = Y(x_0) \text{ donné}$$

$$Y_{n+1} = Y_n + hf(x_n, Y_n) \text{ pour } n > 0$$

1. Interprétation graphique: approximation de la courbe par sa tangente
2. D.L. de $Y(x_{n+1})$ au voisinage de $Y(x_n)$
3. Approximation de $Y'(x_i)$ par différences décentrées amont
4. Méthode à un pas d'ordre 1



ODE

PROBLEMES...

SCILAB Ex.

DAE

[Home Page](#)

[Title Page](#)

[◀◀](#)

[▶▶](#)

[◀](#)

[▶](#)

Page 3 of 22

[Go Back](#)

[Full Screen](#)

[Close](#)

[Quit](#)

1.2. Méthodes de Runge Kutta

1.2.1. Méthode Runge Kutta d'ordre 2

$$\begin{aligned}K_1 &= f(x_n, Y_n) \\K_2 &= f\left(x_n + \frac{h}{2}, Y_n + \frac{h}{2}K_1\right) \\Y_{n+1} &= Y_n + hK_2\end{aligned}$$

1.2.2. Méthode Runge Kutta d'ordre 4

$$\begin{aligned}K_1 &= f(x_n, Y_n) \\K_2 &= f\left(x_n + \frac{h}{2}, Y_n + \frac{h}{2}K_1\right) \\K_3 &= f\left(x_n + \frac{h}{2}, Y_n + \frac{h}{2}K_2\right) \\K_4 &= f(x_n + h, Y_n + hK_3) \\Y_{n+1} &= Y_n + \frac{h}{6}(K_1 + 2K_2 + 2K_3 + K_4)\end{aligned}$$



ODE

PROBLEMES...

SCILAB Ex.

DAE

Home Page

Title Page

◀◀ ▶▶

◀ ▶

Page 4 of 22

Go Back

Full Screen

Close

Quit

1.3. Méthodes de Prédiction-Correction

1.3.1. Méthodes d'Adams, Adams-Moulton

On prédit une valeur Y_{n+1}^p de $Y(t^{n+1})$ par extrapolation, puis on corrige cette valeur pour obtenir $Y_{n+1} = Y_{n+1}^c$:

$$Y_{n+1}^p = Y_n + \frac{h}{12}(23F(Y_n, t^n) - 16F(Y_{n-1}, t^{n-1}) + 5F(Y_{n-2}, t^{n-2}))$$

$$Y_{n+1}^c = Y_n + \frac{h}{12}(5F(Y_{n+1}^p, t^{n+1}) + 8F(Y_n, t^n) - F(Y_{n-1}, t^{n-1}))$$

Méthodes à pas liés, plus rapides que les méthodes à pas séparées (R.K.), mais problème d'initialisation



ODE

PROBLEMES...

SCILAB Ex.

DAE

Home Page

Title Page

◀◀ ▶▶

◀ ▶

Page 5 of 22

Go Back

Full Screen

Close

Quit

2. PROBLEMES RAIDES

Méthodes implicites, qui nécessitent le calcul du Jacobien de $F(Y,t)$

Principe: calcul implicite de Y_{n+1}

$$Y_{n+1} = Y_n + h * F(Y_{n+1}, t^{n+1})$$

avec un calcul par Newton linéarisé de $F(Y_{n+1}, t^{n+1})$

$$F(Y_{n+1}, t^{n+1}) = F(Y_n, t^n) + \frac{\partial F}{\partial Y}(Y_n, t^n)(Y_{n+1} - Y_n)$$

d'où le calcul de Y_{n+1} par résolution du système linéaire d'ordre n:

$$(Id - h * \frac{\partial F}{\partial Y}(Y_n, t^n))(Y_{n+1} - Y_n) = h * F(Y_n, t^n)$$

Id est la matrice identité d'ordre n, et $\frac{\partial F}{\partial Y}$ la Jacobienne d'ordre n de F

Calcul de la Jacobienne

$$(\frac{\partial F}{\partial Y})_{i,j} = \frac{\partial F_i}{\partial Y_j}$$

Approximation numérique de la Jacobienne

$$(\frac{\partial F}{\partial Y})_{i,j} \approx \frac{F_i(Y_1, \dots, Y_j + dY_j, \dots, Y_n) - F_i(Y_1, \dots, Y_j, \dots, Y_n)}{dY_j}$$

Méthode plus stable (mais moins précise)!

BDF (Backward Diffrentation Formula) algorithme LSODA dans SCILAB



ODE

PROBLEMES...

SCILAB Ex.

DAE

[Home Page](#)

[Title Page](#)

[◀◀](#)

[▶▶](#)

[◀](#)

[▶](#)

Page 6 of 22

[Go Back](#)

[Full Screen](#)

[Close](#)

[Quit](#)

3. SCILAB Ex.

3.1. Un premier exemple en SCILAB

Résolution de l'EDP

$$\frac{dy}{dt} = y^2 - t \text{ avec } y(0) = y_0$$

Programme SCILAB : scilab -f ex1.sce

```
deff("dy=f(t,y)","dy=y^2-t")
// bornes
tmin=-3; tmax=5; ymin=-3; ymax=3;
t=tmin:tmax; y=ymin:ymax;
// trace le champ sur une grille
nt=size(t,2); ny=size(y,2);
// champ de vecteur de pente f(t,y)
ft=ones(nt,ny);
fy=feval(t,y,f);
xset("font",2,12)
champ(t,y,ft,fy)
xtitle("", "t", "y")
// resolution
t0=0; y0=0; tfin=5; dt=0.1; T=t0:dt:tfin;
```



ODE
PROBLEMES...

SCILAB Ex.

DAE

Home Page

Title Page

◀◀ ▶▶

◀ ▶

Page 7 of 22

Go Back

Full Screen

Close

Quit

```

sol=ode(y0,t0,T,f);
// trace de la solution
xset("thickness",2); plot2d(T,sol,1,"000")
// trace en interactif
while(%t)
    // lecture du pt de départ
    [c_i,t0,y0]=xclick();
    // test si bouton droit
    if c_i==2 then break end;
    if t0>=tmin & t0<=tmax & y0>=ymin & y0<=ymax then
        T=t0:dt:tmax; sol=ode(y0,t0,T,f);
        plot2d(T(1:size(sol,"*")),sol,1,"000")
        T=t0:-dt:tmin; sol=ode(y0,t0,T,f);
        plot2d(T(1:size(sol,"*")),sol,1,"000")
    end
end
end

```



ODE
PROBLEMES...

SCILAB Ex.

DAE

[Home Page](#)

[Title Page](#)

[◀◀](#)

[▶▶](#)

[◀](#)

[▶](#)

Page 8 of 22

[Go Back](#)

[Full Screen](#)

[Close](#)

[Quit](#)

3.2. Un système dynamique

Problème de Volterra

Problème dynamique proies/prédateurs.

À Trieste, pendant la première guerre mondiale, la pêche avait bien sûr diminué à cause des événements. La pêche consistait à lancer des filets et à récupérer tous les poissons. Le bureau des pêches avait constaté qu'alors la proportion de poissons du style requins, peu intéressants pour la consommation, avait considérablement augmenté par rapport aux poissons intéressants du style sardines. Ils demandèrent l'aide de Volterra qui modélisa le système requins-sardines par le système des deux équations différentielles suivantes où $x(t)$ représente le nombre de sardines et $y(t)$ représente le nombre de requins :

$$\begin{aligned}\frac{dx}{dt} &= ax(t) - bx(t) * y(t) \text{ avec } a, b > 0 \\ \frac{dy}{dt} &= cx(t) * y(t) - dy(t) \text{ avec } c, d > 0 \\ x(0) &= x_0 \text{ et } y(0) = y_0\end{aligned}$$

Ce modèle approché, appelé aussi système de Lotka-Volterra, signifie qu'en l'absence de requins les sardines prolifèrent $x'(t) = ax(t)$, qu'en l'absence de sardines les requins disparaissent $y'(t) = -dy(t)$ et le terme en $x(t)y(t)$, qui représente la rencontre des requins et des sardines, augmente le nombre de requins et diminue le nombre de sardines (car ces dernières sont mangées par les requins).



ODE

PROBLEMES...

SCILAB Ex.

DAE

Home Page

Title Page

◀◀ ▶▶

◀ ▶

Page 9 of 22

Go Back

Full Screen

Close

Quit

À partir de ce modèle Volterra en déduit, sans pouvoir faire les calculs numériques à l'époque, que plus on pêche de poissons, plus la proportion de sardines, donc de poissons intéressants, est importante !

C'est ce que nous allons essayer de retrouver par le calcul, en recherchant le point d'équilibre du système dynamique dans l'espace des phases.

On prendra par exemple pour l'application numérique
 $a=3$, $b=1$, $c=1$ et $d=2$
et un cas où on augmente la fréquence de pêche $a=a-\delta$ $d=d+\delta$
 $a=1$, $b=1$, $c=1$ et $d=4$



ODE

PROBLEMES...

SCILAB Ex.

DAE

Home Page

Title Page

◀◀

▶▶

◀

▶

Page 10 of 22

Go Back

Full Screen

Close

Quit

3.3. Voltera

Programme SCILAB : scilab -f requin.sce

```
xclear();
deff("dY=fv(t,Y)","dY=[a*Y(1)-b*Y(1)*Y(2); c*Y(1)*Y(2)-d*Y(2)]")
// trace des lignes de champs dans la plan de phase
a=3; b=1; c=1; d=2;
// cas ou on augmente la frequence de peche a=a-delta d=d+delta
a=1; b=1; c=1; d=4;
xmin=0; xmax=6; ymin=0; ymax=6;
xr=xmin:0.5:xmax; yr=ymin:0.5:ymax;
xset("font",2,12)
fchamp(fv,0,xr,yr)
xtitle("", "x", "y")
// calcul des trajectoires
xset("thickness",2)
t0=0; tmax=5; dt=0.05;
T=t0:dt:tmax;
while(%t)
    [c_i,x0,y0]=xclick();
    if c_i==2 then break end;
    if x0>=xmin & x0<=xmax & y0>=ymin & y0<=ymax then
        sol=ode([x0;y0],t0,T,fv);
        plot2d(sol(1,:),sol(2,:),1,"000")
    end
end
```



ODE

PROBLEMES...

SCILAB Ex.

DAE

Home Page

Title Page

◀ ▶

◀ ▶

Page 11 of 22

Go Back

Full Screen

Close

Quit

end
end



ODE

PROBLEMES...

SCILAB Ex.

DAE

Home Page

Title Page



Page 12 of 22

Go Back

Full Screen

Close

Quit

3.4. Un problème raide

On considère le problème de la réaction chimique de Robertson

$$y_1' = -0.04y_1 + 10^4 y_2 * y_3$$

$$y_2' = -y_1' - y_3'$$

$$y_3' = 3.10^7 y_2^2$$

Les méthodes RK4, ADAMS ne converge pas !!!

Utilisation de la méthode implicite

Calcul du Jacobien $J = \frac{\partial F}{\partial Y}$

$$J = \begin{pmatrix} -0.04 & 10^4 y_3 & 10^4 y_2 \\ 0.04 & -10^4 y_3 - 6.10^7 y_2 & -10^4 y_2 \\ 0 & 6.10^7 y_2 & 0 \end{pmatrix}$$

Programme SCILAB : scilab -f stiffode.sce



ODE

PROBLEMES...

SCILAB Ex.

DAE

Home Page

Title Page

◀◀ ▶▶

◀ ▶

Page 13 of 22

Go Back

Full Screen

Close

Quit

```

clear
// Equations definition
deff(' [yd]=chem(t,y)',[
    'yd(1)=-0.04*y(1) + 1d4*y(2)*y(3);';
    'yd(3)= 3d7*y(2)*y(2);';
    'yd(2)= -yd(1) - yd(3);'])
// Jacobien
deff(' [J]=Jac(t,y)',[
    'J=[-0.04,1.e4*y(3),1.e4*y(2);0.04,-1.e4*y(3)-6.e7*y(2),1.e4*y(2);0,6.e7*y(2),0]';
])
// Integration
t=[1.d-5:0.02:.4 0.41:.1:4 40 400 4000 40000 4d5 4d6 4d7 4d8 4d9 4d10];
rtol=1.d-4;atol=[1.d-6;1.d-10;1.d-6];
method="stiff";
timer();
y=ode(method,[1;0;0],0,t,rtol,atol,chem,Jac);
timer()
// Visualisation
rect=[1.d-5,-0.1,1d11,1.1];
xtitle("", "t", "Y")
plot2d1("oln",t',(diag([1 10000 1])*y)',(1:3),"111",' y1@10000 y2@y3',rect)
halt();

```



ODE
PROBLEMES...

SCILAB Ex.

DAE

Home Page

Title Page

◀ ▶

◀ ▶

Page 14 of 22

Go Back

Full Screen

Close

Quit

3.5. Faulkner Scan

L'équation de la couche limite laminaire en présence d'un gradient de pression externe est régie par l'EDP suivante:

$$\frac{\partial^3 f}{\partial \eta^3} + f(\eta) \frac{\partial^2 f}{\partial \eta^2} + \beta(1 - (\frac{\partial f}{\partial \eta})^2) = 0$$

avec les conditions aux limites

$$f(0) = 0, \frac{\partial f}{\partial \eta}(0) = 0, f(\infty) = 1$$

Le profil de vitesse est donné par

$$\frac{u}{u_e} = \frac{\partial f}{\partial \eta} \text{ avec } \eta = \frac{y}{\sqrt{2}} \frac{u_e}{\sqrt{\nu x}}$$

On introduit la condition aux limites

$$\frac{\partial^2 f}{\partial \eta^2}(0) = \alpha$$

où α doit être tel que $f(10) = 1$

Pour résoudre ce problème on peut utiliser

1. méthode de Tir, méthode des Aajoints, méthode de Continuation.



ODE
PROBLEMES...

SCILAB Ex.

DAE

Home Page

Title Page

◀◀ ▶▶

◀ ▶

Page 15 of 22

Go Back

Full Screen

Close

Quit

3.6. Méthode de Tir

On part d'une solution f_{α_0} calculée pour $\alpha = \alpha_0$ et on corrige la solution en calculant la correction $df(\eta)$ connaissant f_{α_0} par résolution du problème de gradient (i.e. calcul de la variation df de la solution pour une variation $d\alpha = 1$ de la C.L. $\frac{\partial^2 f}{\partial \eta^2}(0) = \alpha$)

$$\frac{\partial^3 df}{\partial \eta^3} + f_{\alpha_0}(\eta) \frac{\partial^2 df}{\partial \eta^2} + df(\eta) \frac{\partial^2 f_{\alpha_0}}{\partial \eta^2} - 2\beta \frac{\partial df}{\partial \eta} \frac{\partial f_{\alpha_0}}{\partial \eta} = 0$$

avec les conditions aux limites

$$df(0) = 0, \frac{\partial df}{\partial \eta}(0) = 0, \frac{\partial^2 df}{\partial \eta^2}(0) = 1$$

En utilisant la méthode de Newton, on obtiens la correction recherchée sur α

$$\alpha_1 = \alpha_0 - \frac{f_{\alpha_0}(10) - 1}{df(10)}$$



ODE

PROBLEMES...

SCILAB Ex.

DAE

Home Page

Title Page

◀◀ ▶▶

◀ ▶

Page 16 of 22

Go Back

Full Screen

Close

Quit

4. DAE

Equations différentielles algebriques (DAE) ou implicites

$$g(t, Y, Y') = 0 \text{ avec } Y(0) = Y_0$$

où sous la forme d'un problème sous contraintes

$$\begin{aligned}\dot{y} &= f(y, u) \\ 0 &= g(y, u)\end{aligned}$$

Problèmes de mécanique sous contrainte: équations de Lagrange

$$\begin{aligned}\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_k}\right) - \frac{\partial L}{\partial q_k} &= 0, \text{ pour } k = 0, \dots, n \\ 0 &= g(q)\end{aligned}$$

ou $g(q)$ est un vecteur de m contraintes associées à m multiplicateurs λ_i ,
et L le lagrangien

$$L(q, \dot{q}) = T(q, \dot{q}) - U(q, \dot{q}) - \sum_{i=1}^m \lambda_i g_i(q)$$



ODE

PROBLEMES...

SCILAB Ex.

DAE

Home Page

Title Page

◀◀ ▶▶

◀ ▶

Page 17 of 22

Go Back

Full Screen

Close

Quit

4.1. Index d'une DAE

Index d'une DAE: nombre de différenciation pour obtenir une EDO

– Index 1:

si g dépend explicitement de u , on différencie g pour obtenir une EDO

$$\dot{y} = f(y, u)$$
$$\dot{u} = -\left(\frac{\partial g}{\partial u}\right)^{-1} \left(\frac{\partial g}{\partial y}\right) f(y, u)$$

– Index 2: si g ne dépend pas de u , il faut différentier 2 fois pour obtenir une EDO,

i.e. calculer $\dot{g}(y) = 0$ et $\ddot{g}(y) = 0$

$$0 = \left(\frac{\partial g}{\partial y}\right) f(y, u)$$

$$\dot{u} = -(g_y(y) f_u(y, u))^{-1} (g_{yy}(y) f(y, u) * f(y, u) + g_y(u) f_y(y, u))$$

SCILAB peut résoudre les DAE d'ordre 1 et quelques DAE d'ordre 2 avec DASSL

$$G(t, Y, \dot{Y}) = 0$$

qui utilise des BDF d'ordre 1 à 4.



ODE

PROBLEMES...

SCILAB Ex.

DAE

Home Page

Title Page



Page 18 of 22

Go Back

Full Screen

Close

Quit

4.2. Exemple: mvt 3D d'un pendule

Soit $q = (x_1, x_2, x_3)$ l'extrémité d'un pendule de longueur L et de masse m .
On a

$$T = 0.5 * m(\dot{x}_1^2 + \dot{x}_2^2 + \dot{x}_3^2), U = -m\gamma x_3$$

sous la contrainte $L = x_1^2 + x_2^2 + x_3^2$

Les équations de Lagrange s'écrivent:

$$\begin{aligned} m\ddot{x}_1 &= -2x_1\lambda \\ m\ddot{x}_2 &= -2x_2\lambda \\ m\ddot{x}_3 &= -2x_3\lambda - m\gamma \\ L &= x_1^2 + x_2^2 + x_3^2 \end{aligned}$$

qui s'écrivent sous la forme matricielle avec $\phi(q, u) = (0, 0, -\gamma)^t$

$$\begin{pmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \dot{q} \\ \dot{u} \\ \dot{\lambda} \end{pmatrix} - \begin{pmatrix} 0 & I & 0 \\ 0 & 0 & g_g(q)^t \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} q \\ u \\ \lambda \end{pmatrix} - \begin{pmatrix} 0 \\ \phi(q, u) \\ g(q) \end{pmatrix} = 0$$



ODE
PROBLEMES...
SCILAB Ex.
DAE

[Home Page](#)

[Title Page](#)

[◀◀](#) [▶▶](#)

[◀](#) [▶](#)

Page 19 of 22

[Go Back](#)

[Full Screen](#)

[Close](#)

[Quit](#)

4.3. SCILAB ex.

Le problème est d'ordre 3 (donc non soluble avec DASSL)

Transformation en un pble DAS d'index 2

$$\begin{pmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \dot{q} \\ \dot{u} \\ \dot{\lambda} \end{pmatrix} - \begin{pmatrix} 0 & I & 0 \\ 0 & 0 & -g_g(q)^t \\ 0 & g_g(q) & 0 \end{pmatrix} \begin{pmatrix} q \\ u \\ \lambda \end{pmatrix} - \begin{pmatrix} 0 \\ \phi(q,u) \\ 0 \end{pmatrix} = 0$$

avec les conditions initiales

$$q(0) = (x_1(0), x_2(0), x_3(0)), \text{ et } u(0) = (\dot{x}_1(0), \dot{x}_2(0), \dot{x}_3(0))$$

Il faut en plus donner des conditions initiales pour λ , telle que

$$\dot{g}(0) = g_g(q(0))u(0) = 0 \text{ et } f(t, y(0), \dot{y}(0)) = 0$$

i.e. par résolution de

$$\begin{pmatrix} I & g_g(q_0)^t \\ -g_g(q_0) & 0 \end{pmatrix} \begin{pmatrix} \dot{u}(0) \\ \lambda(0) \end{pmatrix} = \begin{pmatrix} \phi(q_0, u_0) \\ 0 \end{pmatrix} = 0$$



ODE
PROBLEMES...
SCILAB Ex.

DAE

Home Page

Title Page

◀◀ ▶▶

◀ ▶

Page 20 of 22

Go Back

Full Screen

Close

Quit

4.4. programme SCILAB

Programme SCILAB : scilab -f pendule3D.sce

```
clear()
// parametres
n=3; nc=1; N=n+n+nc;
I=eye(n,n); Z=zeros(n,n); Zb=zeros(nc,N); Z0=zeros(n,nc);
Z1=zeros(n,1); Z2=zeros(nc,n); Z3=zeros(nc,nc); Z4=zeros(nc,1);
m=1; gama=9.8; l=1;
// fonctions
deff("res=phi(q,u)","res=[0;0;-gama]");
deff("res=g(q)","res=q'*q-l^2");
deff("res=g_g(q)","res=2*q''");
// equations
deff("res,ires=syst(t,y,yd)","q=y(1:n);","u=y(n+1:2*n);","ires=0;","...
res=[I,Z,Z0; Z,I,Z0; Zb ]*yd - [Z,I,Z0;Z,Z,-g_g(q)''; Z2,g_g(q),Z3]*y-[Z1;phi(q,u);Z4];1");
// conditions initiales
theta0=0.2;phi0=%pi/4;
q0=[sin(phi0)*cos(theta0);cos(phi0)*cos(theta0);sin(theta0)];
u0=[1;-1;0];
// calcul de lambda(0)
w=[I,g_g(q0)';g_g(q0),Z3]\[phi(q0,u0);Z4];
ud0=w(1:n);
lambda0=w(n+1,$);
```



ODE
PROBLEMES...

SCILAB Ex.

DAE

Home Page

Title Page

◀ ▶

◀ ▶

Page 21 of 22

Go Back

Full Screen

Close

Quit

```

// appel dass1
y0=[q0;u0;lambda0];
yd0=[u0;ud0;0];
info=list([],0,[],[],[],0,0);
atol=[0.0001;0.0001;0.0001;00.001;0.001;0.001;1];
rtol=atol;
t0=0;T=0:0.01:20;
y2=dass1([y0,yd0],t0,T,rtol,atol,syst,info);
q2=y2(1+1:1+n,:);
norm(q2(:,$),2)
// trace
xbasc();
xtitle("Trajectoire du pendule");
param3d(q2(1,:),q2(2,:),q2(3,:),31,78,"X@Y@Z",[2,4]);

```



ODE
PROBLEMES...

SCILAB Ex.

DAE

[Home Page](#)

[Title Page](#)

◀◀ ▶▶

◀ ▶

Page 22 of 22

[Go Back](#)

[Full Screen](#)

[Close](#)

[Quit](#)