

Méthode de Störmer-Verlet

1. Système à hamiltonien séparable

On reprend les notations introduites dans [Méthode d'Euler symplectique](#). On considère un système dynamique hamiltonien dont les équations du mouvement peuvent se mettre sous la forme canonique, et dont l'hamiltonien est de la forme :

$$H(p, q) = K(p) + V(q)$$

où K est l'énergie cinétique du système et V son énergie potentielle.

Les équations de Hamilton s'écrivent alors :

$$\frac{dq}{dt} = +\frac{\partial K}{\partial p} \quad (1)$$

$$\frac{dp}{dt} = -\frac{\partial V}{\partial q} \quad (2)$$

2. Méthode de Störmer-Verlet

On cherche une solution numérique des équations (1) et (2) avec les conditions initiales à $t = 0$:

$$q(0) = q_0 \quad (3)$$

$$p(0) = p_0 \quad (4)$$

Pour obtenir N points sur l'intervalle $[0, T]$, on définit le pas de temps $h = T/N$ et on cherche des valeurs approchées pour les instants :

$$t_n = nh \quad (5)$$

où l'entier n varie de 0 à N .

La méthode de Störmer-Verlet ([1]) est définie par :

$$p_i = p_n - \frac{1}{2}h \frac{\partial V}{\partial q}(q_n) \quad (6)$$

$$q_{n+1} = q_n + h \frac{\partial K}{\partial p}(p_i) \quad (7)$$

$$p_{n+1} = p_i - \frac{1}{2}h \frac{\partial V}{\partial q}(q_{n+1}) \quad (8)$$

Les quantités de mouvement intermédiaires (vecteur p_i) sont tout d'abord évaluées puis les nouvelles positions sont calculées avec ces quantités de mouvement. Enfin les nouvelles quantités de mouvement sont calculées avec les nouvelles positions.

Comparée aux méthodes d'Euler symplectiques, la méthode Störmer-Verlet a l'avantage d'être d'ordre 2. C'est l'une des méthodes les plus utilisées en dynamique moléculaire.

3. Programmation avec Scilab

3.a. Fonctions de calcul

On définit tout d'abord une fonction qui calcule la dérivée par rapport aux quantités de mouvement de l'énergie cinétique. L'exemple donné ici est une chaîne d'oscillateurs linéaires couplés :

```
n=4 //nombre d'oscillateurs
d=1.0/(n+1) // distance moyenne
function [dq]=dK(p)
    dq=p
endfunction
```

La fonction suivante calcule la dérivée de l'énergie potentielle, c'est-à-dire les forces (au signe près) :

```
function [dp]=dV(q)
    dp(1)=(q(1)-d)+(q(1)-q(2)+d)
    for k=2:n-1,
        dp(k)=(q(k)-q(k-1)-d)+(q(k)-q(k+1)+d)
    end;
    dp(n)=(q(n)-q(n-1)-d)+(q(n)-1.0+d)
endfunction
```

On fournit aussi une fonction qui calcule l'énergie du système :

```
function e=energie(q,p)
    e=0.5*p(1)^2+0.5*(q(1)-d)^2;
    for k=2:n-1,
        e=e+0.5*p(k)^2+0.5*(q(k)-q(k-1)-d)^2;
    end;
    e=e+0.5*p(n)^2+0.5*(q(n)-q(n-1)-d)^2+0.5*(q(n)-1.0+d)^2;
endfunction
```

La fonction suivante effectue un pas de la méthode de Störmer-Verlet :

```
function [q,p]=pasVerlet(dK,dV,h,qn,pn)
    pi=pn-0.5*h*dV(qn)
    q=qn+h*dK(pi)
    p=pi-0.5*h*dV(q)
endfunction
```

Pour stocker les valeurs obtenues sur l'intervalle $[0, T]$, on utilise la convention de la fonction `ode` de scilab : les valeurs de q sont stockées dans une matrice dont chaque ligne

correspond à une composante de q (de même pour p). La fonction suivante effectue le calcul complet, avec une condition initiale (instant 0) définie dans deux matrices colonne.

n_2 est le nombre de points mémorisés, n_1 le nombre de pas entre deux points mémorisés.

```
function [mq,mp,e,t]=verlet(q0,p0,T,n1,n2,dK,dV,energie)
    h=T/(n1*n2);
    s=size(q0);
    nv=s(1);
    mq = zeros(nv,n2+1);
    mp = zeros(nv,n2+1);
    e = zeros(1,n2+1);
    t=zeros(1,n2+1);
    mq(:,1)=q0;
    mp(:,1)=p0;
    q=q0;
    p=p0;
    e(1)=energie(q,p0);
    t(1)=0;
    for k2=1:n2,
        for k1=1:n1,
            [q,p]=pasVerlet(dK,dV,h,q,p)
        end;
        mq(:,k2+1)=q;
        mp(:,k2+1)=p;
        e(k2+1)=energie(q,p);
        t(k2+1)=t(k2)+n1*h;
    end;
endfunction
```

3.b. Exemple : oscillateurs couplés

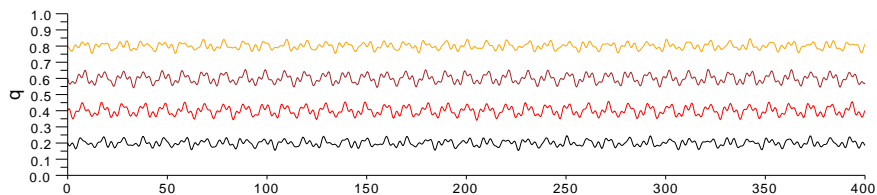
On teste le fonctionnement sur le système de 4 oscillateurs défini plus haut :

```
function v=vitesse()
    v=(rand()-0.5)*0.1;
endfunction
q0=zeros(n,1);
p0=zeros(n,1);
for k=1:n,
    q0(k)=d*k;
    p0(k)=vitesse();
end;
n1=1;
n2=1000;
T=400;
```

```
[mq,mp,e,t]=verlet(q0,p0,T,n1,n2,dK,dV,energie);
```

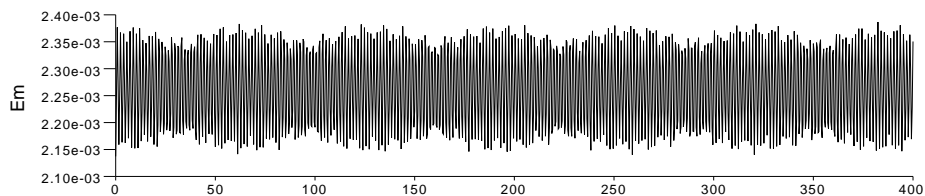
On trace les positions en fonction du temps :

```
figA=scf();
plot2d(t,mq(1,:),style=color('black'),rect=[0,0,T,1]);
plot2d(t,mq(2,:),style=color('red'));
plot2d(t,mq(3,:),style=color('brown'));
plot2d(t,mq(4,:),style=color('orange'));
xtitle('', 't', 'q');
```



On trace l'énergie mécanique du système en fonction du temps :

```
figB=scf();
plot2d(t,e);
xtitle('', 't', 'Em');
```

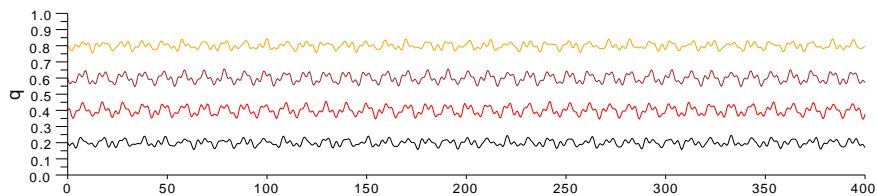


Pour comparaison, voyons le même système avec une méthode à pas multiples (Adams) :

```
function ydot=systeme(t,y)
    k=1;
    ydot(k)=y(k+n);
    ydot(k+n)=-(y(k)-d)-(y(k)-y(k+1)+d);
    for k=2:n-1,
        ydot(k)=y(k+n);
        ydot(k+n)=-(y(k)-y(k-1)-d)-(y(k)-y(k+1)+d);
    end;
    k=n;
    ydot(k)=y(k+n);
    ydot(k+n)=-(y(k)-y(k-1)-d)-(y(k)-1.0+d);
endfunction
y0=[q0;p0];
t0=0;
```

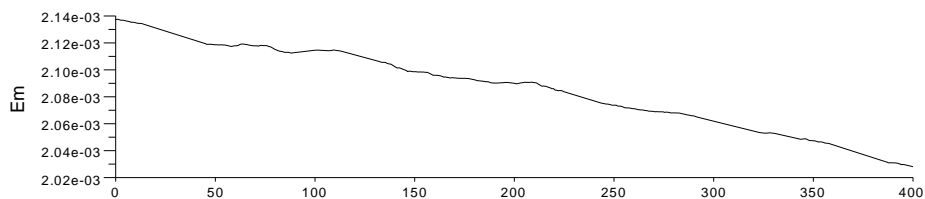
```
t=0:T/n2:T;
rtol=1e-5;
atol=1e-6;
[y,w,iw]=ode('adams',y0,t0,t,rtol,atol,systeme);
```

```
figC=scf();
plot2d(t,y(1,:),style=color('black'),rect=[0,0,T,1]);
plot2d(t,y(2,:),style=color('red'));
plot2d(t,y(3,:),style=color('brown'));
plot2d(t,y(4,:),style=color('orange'));
xtitle('', 't', 'q');
```



```
function e=energie2(y)
s=size(y);
e=zeros(1,s(2));
for j=1:s(2),
    e(j) = 0.5*y(1+n,j)^2+0.5*(y(1,j)-d)^2;
    for k=2:n-1,
        e(j)=e(j)+0.5*y(k+n,j)^2+0.5*(y(k,j)-y(k-1,j)-d)^2;
    end;
    e(j)=e(j)+0.5*y(n+n,j)^2+0.5*(y(n,j)-y(n-1,j)-d)^2+0.5*(y(n,j)-1.0+d)^2;
end;
endfunction
Em2=energie2(y);
```

```
figD=scf();
plot2d(t,Em2);
xtitle('', 't', 'Em');
```



Dans ce calcul, la tolérance locale (rtol et atol) a été ajustée de manière à obtenir une variation d'énergie similaire à la méthode de Störmer-Verlet.

On constate que la méthode à pas multiple conduit à une dérive de l'énergie alors que la méthode de Störmer-Verlet présente des fluctuations de l'énergie autour d'une valeur stable.

Le nombre d'évaluations du système est :

```
iw(12)
```

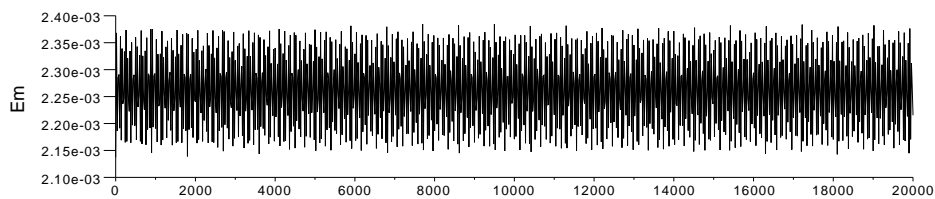
```
3215
```

à comparer aux 2000 évaluations de dV (2 fois le nombre de pas) pour la méthode de Störmer-Verlet (l'évaluation de dK est négligeable par rapport à celle de dV).

Voyons le comportement de la méthode de Störmer-Verlet sur un temps plus long (avec le même pas) :

```
n1=50;
n2=1000;
T=20000;
[mq,mp,e,t]=verlet(q0,p0,T,n1,n2,dK,dV,energie);
```

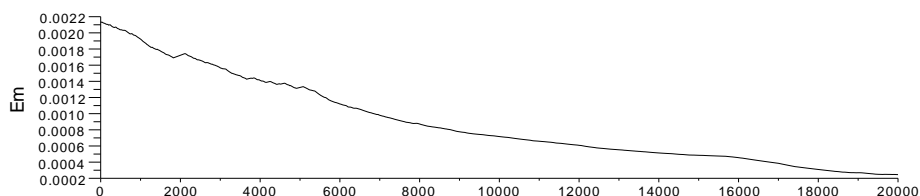
```
figE=scf();
plot2d(t,e);
xtitle('','t','Em');
```



La méthode à pas multiple sur le même temps, avec les mêmes tolérances que plus haut :

```
t=0:T/n2:T;
[y,w,iw]=ode('adams',y0,t0,t,rtol,atol,systeme);
Em2=energie2(y);
```

```
figF=scf();
plot2d(t,Em2);
xtitle('','t','Em');
```



La méthode à pas multiples présente une dérive très importante de l'énergie. Bien sûr, il est possible de réduire cette dérive en réduisant la tolérance locale, mais le comportement à long terme reste imprévisible a priori.

On voit ici le principal avantage de la méthode symplectique pour un système hamiltonien : l'amplitude des fluctuations d'énergie reste faible sur les temps longs, et cette amplitude est prévisible dès le début du calcul. La méthode de Störmer-Verlet est donc

particulièrement adaptée à la simulation des systèmes conservatifs sur les temps longs (système solaire, dynamique moléculaire, etc).

Références

[1] B. Leimkuhler, S. Reich, *Simulating hamiltonian dynamics*, (Cambridge university press, 2004)