UNIVERSITÉ DE GENÈVE
Sections d'Informatique et de Physique

FACULTÉ DES SCIENCES
Profs. B. Chopard et M. Droz

Outils, modélisation et simulation en calcul numérique – Corrigé série 7

3 mai 2005

Exercice 1.

i) De part la définition du réseau FHP, il est évident que $\sum_{i=1}^{6} v_{i\alpha} = 0$. Il s'agit d'une conséquence immédiate de la relation $\mathbf{v}_i = -\mathbf{v}_{i+3}$.

ii) Soit $\mathbf{c} = \mathbf{v}/v$. Soit la matrice $\boldsymbol{A} = \{a_{\alpha\beta}\}_{\alpha,\beta=1}^{6}$ d'éléments $a_{\alpha\beta} = \sum_{i=1}^{6} c_{i\alpha}c_{i\beta}$. Formons le vecteur $\mathbf{b} = \boldsymbol{A} \cdot \mathbf{c}_k$, alors en adoptant la convention de Einstein de sommation sur les indices répétés :

$$\begin{aligned} b_\alpha &= a_{\alpha\beta}c_{k\beta} \\ &= c_{i\alpha}c_{i\beta}c_{k\beta} \\ &= c_{i\alpha}\mathbf{c}_i \cdot \mathbf{c}_k. \end{aligned} \tag{1}$$

Par changement de variables et sans restriction de généralité, b_α peut être écrit sous la forme

$$b_\alpha = c_{i\alpha}\mathbf{c}_i \cdot \mathbf{c}_1. \tag{2}$$

Or par définition du réseau FHP on a

$$\begin{aligned} \mathbf{c}_k \cdot \mathbf{c}_{k+1} &= 1/2, && (3) \\ \mathbf{c}_k \cdot \mathbf{c}_{k+2} &= -1/2, && (4) \\ \mathbf{c}_k + \mathbf{c}_{k+2} + \mathbf{c}_{k+4} &= 0, && (5) \\ & && (6) \end{aligned}$$

ainsi on trouve

$$b_\alpha = 2c_{1\alpha} + c_{2\alpha} - c_{3\alpha}. \tag{7}$$

Or comme pour le réseau FHP $\mathbf{c}_2 - \mathbf{c}_3 = \mathbf{c}_1$ on en déduit

$$b_\alpha = 3c_{1\alpha}, \qquad \forall\alpha, \tag{8}$$

et donc ceci étant vrai pour tout α et direction $\mathbf{c}_k$ on a

$$\mathbf{b} = 3\mathbf{c}_k. \tag{9}$$

Par conséquent

$$a_{\alpha\beta} = c_{i\alpha}c_{i\beta} = 3\delta_{\alpha\beta}, \tag{10}$$

ce qui dans les variables $\mathbf{v}$ donne

$$\sum_{i=1}^{6} v_{i\alpha}v_{i\beta} = 3v^2\delta_{\alpha\beta}. \tag{11}$$

iii) A nouveau, il s'agit d'une conséquence immédiate de la relation $\mathbf{v}_i = -\mathbf{v}_{i+3}$.

iv) En sommant sur $i=1,\ldots,6$ il vient

$$\begin{aligned}
\sum_{i=1}^{6} N_i^{(0)} &= \rho\Big[1+\frac{1}{3}\mathbf{w}\cdot\underbrace{\sum_{i=1}^{6}\mathbf{c}_i}_{=0} - 3G(\rho)w^2 + G(\rho)w_\alpha w_\beta \underbrace{\sum_{i=1}^{6} c_{i\alpha}c_{i\beta}}_{=3\delta_{\alpha\beta}}\Big] \\
&= \rho\Big[1-3G(\rho)w^2+3G(\rho)\underbrace{w_\alpha w_\beta\delta_{\alpha\beta}}_{\substack{=w_\alpha w\alpha \\ =w^2}}\Big] \\
&= \rho. \qquad (12)
\end{aligned}$$

v) En procédant de même, on a

$$\begin{aligned}
\sum_{i=1}^{6} N_i^{(0)} v_{i\gamma} &= \rho\Big[\underbrace{\sum_{i=1}^{6} v_{i\gamma}}_{=0} + \frac{1}{3}w_\alpha \underbrace{\sum_{i=1}^{6} c_{i\alpha}v_{i\gamma}}_{=3v\delta_{\alpha\gamma}} - 3G(\rho)w^2 \underbrace{\sum_{i=1}^{6} v_{i\gamma}}_{=0} + G(\rho)w_\alpha w_\beta \underbrace{\sum_{i=1}^{6} c_{i\alpha}c_{i\beta}v_{i\gamma}}_{=0}\Big] \\
&= \rho v w_\alpha \delta_{\alpha\gamma}, \qquad \mathbf{w}=\mathbf{u}/v, \\
&= \rho u_\gamma, \qquad (13)
\end{aligned}$$

ce qui est vrai $\forall\gamma$, d'où

$$\sum_{i=1}^{6} N_i^{(0)}\mathbf{v}_i = \rho\mathbf{u}. \qquad (14)$$

Exercice 2.

En pratique en physique on étudie souvent les corrélations entre plus proches voisins pour une dynamique isotrope et invariante par translation, où alors $c_{ij}(\mathbf{x}_i,\mathbf{x}_j)=c_{ij}(r)$, $r=|\mathbf{x}_i-\mathbf{x}_j|$. Néanmoins, dans notre cas on peut aussi étudier sans distinction les corrélations entre toutes les particules prises deux à deux (plus rapide numériquement).

En principe, un langage de programmation tel que Fortran serait plus approprié dans ce cas. Néanmoins, il suffit d'apporter de légères modifications au programme HPP de la série précédente, c'est pourquoi nous présentons la solution dans un code Matlab.

```
clear;
N=50; %size of the system NxN
density=1.0; %initial average density
tmax=5*N; %stops when tmax iterations reached
average=100; %number of runs for average
geometry=zeros(N,N,4); %creates the geometry (boundaries)
geometry(1:N,1,1)=1;
geometry(1,1:N,4)=1;
geometry(1:N,N,3)=1;
geometry(N,1:N,2)=1;
l=[N 1:N-1]; %array: N,1,2,...,N-1
r=[2:N 1]; %array: 2,3,4,...,N,1
theplot=zeros(tmax,2);
theplot(1:tmax)=[1:tmax];
for avg=1:average %loop on the runs for average
  den=0;
```

```
lattice=zeros(N); %random initial conditions
for i=1:N %on average, density particles
  for j=1:N
    if (rand(1,1) < density)
      den=den+1;
      lattice(i,j)=2^(floor(rand(1,1)*4)); %one particle per site
    end
  end
end
for t=1:tmax %time loop
  i = find(lattice == 5); %collision
  j = find(lattice == 10);
  lattice(i)=10;
  lattice(j)=5;
  bin=zeros(N,N,4); %binary decomposition
  for i=1:N
    for j=1:N
      d1 = lattice(i,j);
      for k=0:3
        nn=3-k;
        d2 = 2^nn;
        if (d1-d2 >= 0)
          d1 = d1-d2;
          bin(i,j,nn+1)=1;
        end
      end
    end
  end
  for i=1:N %geometry: bounce back
    for j=1:N
      for k=1:4
        if ((bin(i,j,k)==geometry(i,j,k)) & (geometry(i,j,k)==1))
          bin(i,j,k)=0;
          bin(i,j,mod(k+1,4)+1)=1;
        end
      end
    end
  end
  bin(:,:,1)=bin(:,r,1); %propagation
  bin(:,:,2)=bin(l,:,2);
  bin(:,:,3)=bin(:,l,3);
  bin(:,:,4)=bin(r,:,4);
  for i=1:N %decimal recomposition
    for j=1:N
      lattice(i,j)=bin(i,j,1)+2*bin(i,j,2)+4*bin(i,j,3)+8*bin(i,j,4);
    end
  end
  corr=0; %global correlations
  for i=1:N
    if (any(bin(i,j,:))>0),ni=1;else,ni=0;end
    for j=1:N
      if (any(bin(i,j,:))>0),nj=1;else,nj=0;end
      corr=corr+ni*nj;
    end
```

```
    end
    theplot(t,2)=theplot(t,2) + corr/(N^2)-(den/(N^2))^2;
  end
  tmp(:,1)=theplot(:,1);
  tmp(:,2)=theplot(:,2)/avg;
  save HPP3.txt tmp -ascii;
end
plot(theplot(:,1),theplot(:,2)/average,'kx-');
```

La Fig. 1 représente la fonction de corrélation (intégrée sur toutes les distances) en fonction du nombre d'itérations.

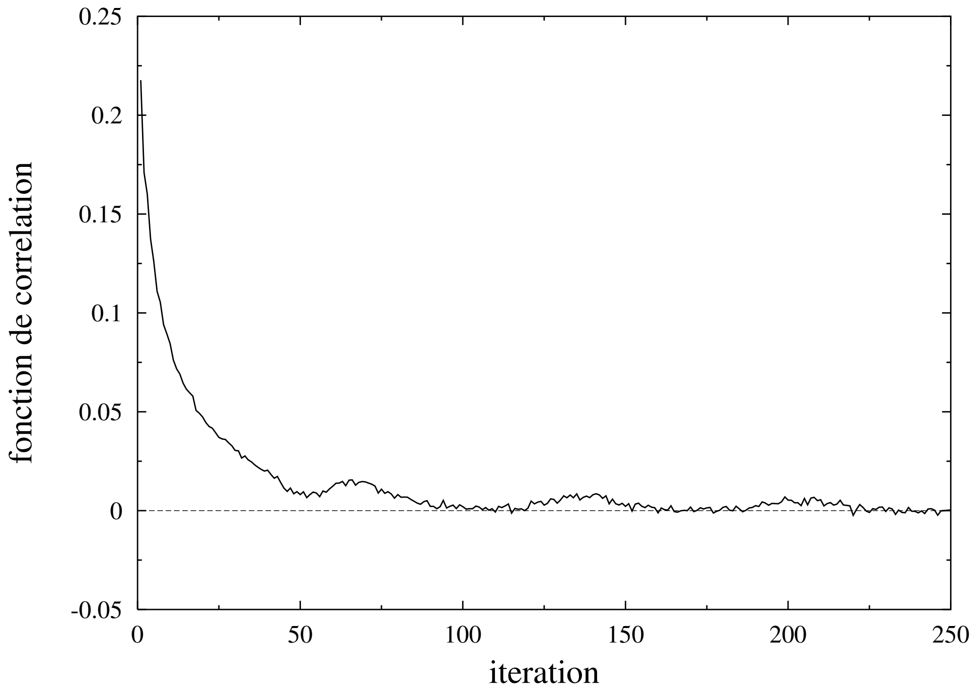

FIG. 1 – Fonction de corrélation du modèle HPP en fonction du temps pour une condition initiale aléatoire, une taille $N = 50$, et une moyenne sur 10^3 réalisations.