

Outils, modélisation et simulation en calcul numérique – Corrigé série 5

19 avril 2005

Exercice 1.

Pour le modèle HPP, on peut décomposer le réseau en 2 sous-réseaux dissociés. Le nombre de particules dans chacun de ces réseaux est alors un invariant artificiel (c'est-à-dire non requis pour la modélisation de base). Dans le cas du réseau FHP, il existe 3 invariants artificiels reliés à l'impulsion des particules.

Considérons au temps t les particules se trouvant sur une ligne horizontale du réseau FHP de label "+". Au temps $t + \tau$, ces particules seront soit restées sur la même ligne, soit elles se seront déplacées sur les lignes supérieures et inférieures de label "-" (cf. Fig. 1).

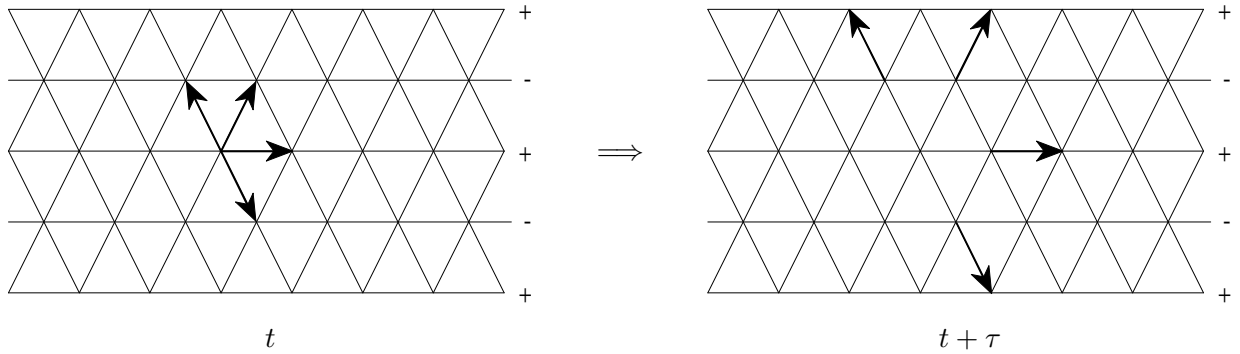


FIG. 1 – Invariant artificiel pour le modèle FHP : la composante verticale de l'impulsion issue des particules sur la ligne horizontale de label "+" au temps t est transférée, à l'itération suivante $t + \tau$, aux deux lignes supérieures et inférieures de labels "-".

Toute l'impulsion verticale initialement sur la ligne "+" (ou "-") en t sera répartie sur les deux lignes "-" (ou "+") en $t + \tau$. Il est ainsi possible de construire une grandeur globalement conservée en sommant sur toutes les lignes "+" et "-" :

$$\sum_{\mathbf{x} \in L_+} P_y(\mathbf{x}, t + \tau) = \sum_{\mathbf{x} \in L_-} P_y(\mathbf{x}, t), \quad (1)$$

$$\sum_{\mathbf{x} \in L_-} P_y(\mathbf{x}, t + \tau) = \sum_{\mathbf{x} \in L_+} P_y(\mathbf{x}, t), \quad (2)$$

où L_+ et L_- sont les deux sous-réseaux, et P_y représente la composante verticale de la quantité de mouvement. En additionnant les Eqs. (1) et (2) on obtient la loi de conservation physiquement requise de la quantité de mouvement totale. Par contre, en soustrayant les Eqs. (1) et (2) on obtient

$$\sum_{\mathbf{x} \in L_+} P_y(\mathbf{x}, t + \tau) - \sum_{\mathbf{x} \in L_-} P_y(\mathbf{x}, t + \tau) = - \left[\sum_{\mathbf{x} \in L_+} P_y(\mathbf{x}, t) - \sum_{\mathbf{x} \in L_-} P_y(\mathbf{x}, t) \right]. \quad (3)$$

En développant le membre de droite de l'Eq. (3) n fois (où $t = n\tau$) on obtient

$$\sum_{\mathbf{x} \in L_+} P_y(\mathbf{x}, t + \tau) - \sum_{\mathbf{x} \in L_-} P_y(\mathbf{x}, t + \tau) = (-1)^{t/\tau} \left[\sum_{\mathbf{x} \in L_+} P_y(\mathbf{x}, 0) - \sum_{\mathbf{x} \in L_-} P_y(\mathbf{x}, 0) \right], \quad (4)$$

et donc

$$(-1)^{-t/\tau} \left[\sum_{\mathbf{x} \in L_+} P_y(\mathbf{x}, t + \tau) - \sum_{\mathbf{x} \in L_-} P_y(\mathbf{x}, t + \tau) \right] = \sum_{\mathbf{x} \in L_+} P_y(\mathbf{x}, 0) - \sum_{\mathbf{x} \in L_-} P_y(\mathbf{x}, 0). \quad (5)$$

Le membre de droite de l'Eq. (5) est une constante déterminée par les conditions initiales, et par conséquent le membre de gauche de l'Eq. (5) est aussi une constante pour tout τ [et ceci indépendamment de la parité de n]. Nous avons donc trouvé un invariant artificiel lié à la structure même du réseau FHP. Bien entendu, il est possible de réaliser la même construction pour les lignes orientées à 60 et 120 degrés par rapport à la direction horizontale. On obtient ainsi un total de 3 invariants artificiels liés à la structure discrète du réseau.

L'existence d'invariants artificiels peut, selon les réseaux utilisés, mener à des prévisions erronées. Il n'existe pas non plus de méthode générale permettant de les trouver. Néanmoins, une fois un tel invariant identifié on peut en général l'éliminer par un choix approprié des conditions initiales.

Exercice 2.

On peut se convaincre que la seule règle de collision entre particules lentes et rapides qui conserve la densité, l'énergie, et l'impulsion est représentée par la Fig. 2 (ainsi que les rotations issues de cette règle).

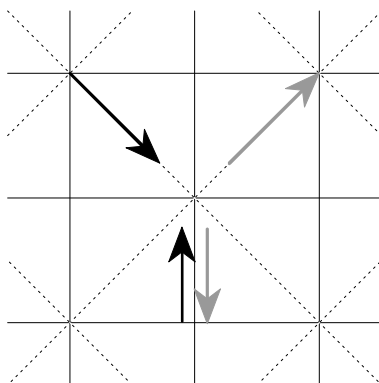


FIG. 2 – Réseau carré à deux vitesses. Les particules lentes suivent les trajectoires horizontales et verticales (trait plein), tandis que les particules rapides suivent les diagonales (trait discontinu). La seule règle de collision acceptable est représentée par les vitesses des particules incidentes en noir, et les vitesses des particules après collision en gris foncé.

Exercice 3.

S'agissant d'une visualisation en temps réel de la dynamique, il est naturel de recourir à un langage tel que Matlab plutôt que Fortran.

L'état $s_i(t)$ de chaque cellule peut être caractérisé par le quadruplet $s_i(t) = (x_1, x_2, x_3, x_4)$, où $x_k \in \{0, 1\}$ décrit la présence (ou l'absence $x_k = 0$) d'une particule arrivant sur le site i depuis la direction k (par exemple, $k = 1$ correspond à l'est, $k = 2$ au nord, $k = 3$ à l'ouest, et $k = 4$ au sud). Bien entendu, en fait les $x_k = x_k(i, t)$ dépendent du temps et du site considéré. Par exemple, pour le modèle HPP les collisions seront représentées par $(1, 0, 1, 0) \rightarrow (0, 1, 0, 1)$ et $(0, 1, 0, 1) \rightarrow (1, 0, 1, 0)$. Toute autre configuration restera inchangée. Il est alors utile d'interpréter le codage $s_i(t)$ comme

une représentation binaire :

$$s_i(t) = (x_1, x_2, x_3, x_4) = \sum_{k=0}^3 2^k x_k \in [0, 15]. \quad (6)$$

Les collisions peuvent alors être codées dans une table. Par exemple, la collision $(1, 0, 1, 0) \rightarrow (0, 1, 0, 1)$ sera décrite en notation décimale par $5 \rightarrow 10$, et la collision $(0, 1, 0, 1) \rightarrow (1, 0, 1, 0)$ par $10 \rightarrow 5$. On obtient ainsi la table pour les collisions :

entrée	sortie
1	1
2	2
3	3
4	4
5	10
6	6
7	7
8	8
9	9
10	5
11	11
12	12
13	13
14	14
15	15

L'automate cellulaire consiste alors en deux étapes : la collision et la propagation. Pour la phase de propagation, le bit k de la cellule i est transmis au voisin dans la direction k de la cellule i . Etant donné que les collisions ne font que de modifier les directions des particules, le principe d'exclusion (une particule par site au maximum) sera automatiquement satisfait s'il l'était au temps initial.

```
clear;
N=500; %size of the system NxN
density=0.4; %initial average density
tmax=N/2; %stops when tmax iterations reached
Nplot=1; %plot every Nplot iterations (speed)
radius=20; %radius for the initial explosion
black=[1 1 1]; %black-white-map
white=[0 0 0];
bmap(1,:)=black;
bmap(2,:)=white;
colormap(bmap);
lattice=zeros(N); %random initial conditions
for i=1:N
    for j=1:N
        if (rand(1,1) < density) %on average, (N^2)*density particles
            lattice(i,j)=2^(floor(rand(1,1)*4)); %single particle
        end
    end
end
end
```

```

for i=1:360 %initial condition: isotropic explosion in the center
  for j=0:radius
    x=round(j*cos((2*pi/360)*i));
    y=round(j*sin((2*pi/360)*i));
    for k=0:3
      if ((i>=k*90) & (i<(k+1)*90))
        if (rand(1,1)<(90-(i-k*90))/90),p=2^k;else,p=mod(2^(k+1),15);end
      end
    end
    lattice(N/2+x,N/2+y)=p;
  end
end
l=[N 1:N-1]; %array: N,1,2,...,N-1
r=[2:N 1]; %array: 2,3,4,...,N,1
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
  end
  bin(:, :, 1)=bin(1, :, 1); %propagation
  bin(:, :, 2)=bin(:, 1, 2);
  bin(:, :, 3)=bin(r, :, 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
  end
  if (mod(t,Nplot)==0) %plot
    imagesc(lattice>0);
    axis square off;
    drawnow
  end
end
end

```

En partant d'une condition initiale sphérique, le réseau est tel que la dynamique HPP engendre des anisotropies (cf. Fig. 3).

Une animation correspondante se trouve à l'adresse internet :
<http://cui.unige.ch/~chopard/CA/Animations/HPP/hpp-wave.html>

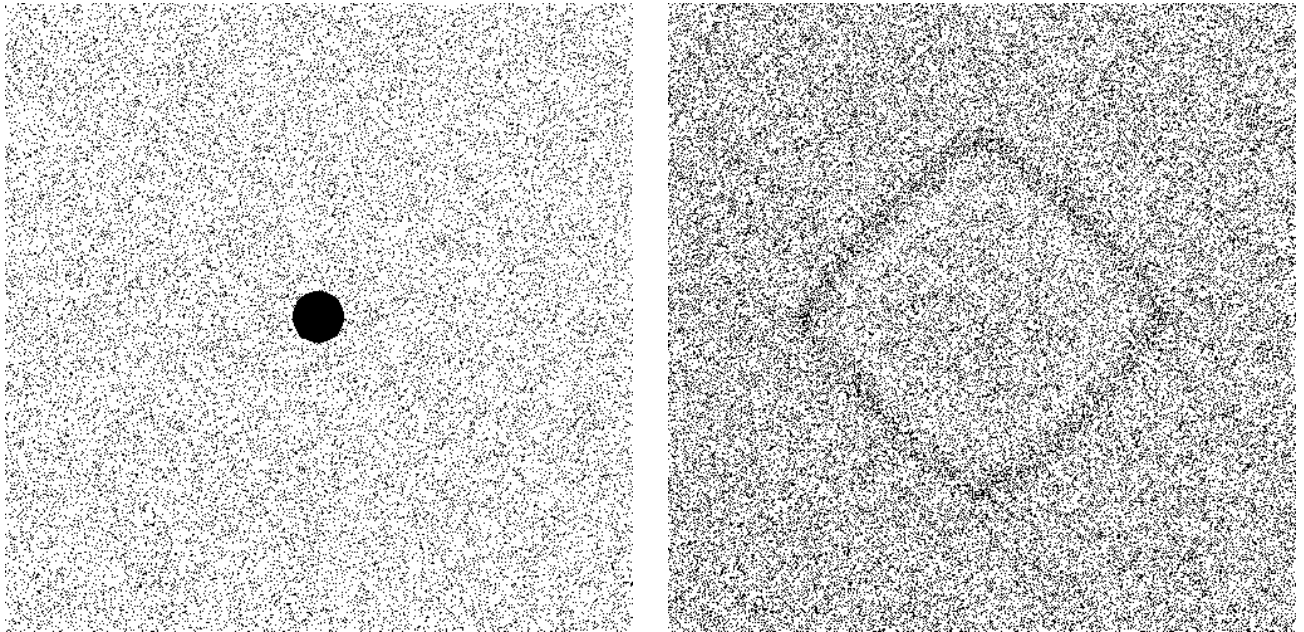


FIG. 3 – Dynamique HPP en partant d'une condition initiale à symétrie sphérique (à gauche), pour un réseau de taille $N = 500$. La figure de droite représente le système à un temps ultérieur. On observe la formation d'anisotropies par la dynamique.