

**Outils, modélisation et simulation en calcul numérique – Corrigé série 4**

12 avril 2005

**Exercice 1.**

i) Les équations gouvernant l'évolution de  $N_A(t)$  et  $N_B(t)$  sont

$$\frac{d}{dt}N_A(t) = -kN_A(t)N_B(t), \quad (1)$$

$$\frac{d}{dt}N_B(t) = -kN_A(t)N_B(t). \quad (2)$$

ii) Pour les conditions initiales  $N_A(0) = N_B(0) = N_0/2$ , la solution est

$$N_A(t) = N_B(t) = \frac{N_0/2}{1 + N_0kt/2}. \quad (3)$$

Le comportement asymptotique est donc

$$N_A(t) \sim N_B(t) \sim t^{-1}. \quad (4)$$

Notons que si les conditions initiales ne sont pas égales, i.e.,  $N_A(0) = a_0$ ,  $N_B(0) = b_0$ ,  $a_0 \neq b_0$ , alors la décroissance devient exponentielle.

iii) L'évolution des densités locales  $n_A(\mathbf{r}, t)$  et  $n_B(\mathbf{r}, t)$  est gouvernée par une équation de réaction-diffusion :

$$\frac{d}{dt}n_A(\mathbf{r}, t) = D\Delta n_A(\mathbf{r}, t) - kn_A(\mathbf{r}, t)n_B(\mathbf{r}, t), \quad (5)$$

$$\frac{d}{dt}n_B(\mathbf{r}, t) = D\Delta n_B(\mathbf{r}, t) - kn_A(\mathbf{r}, t)n_B(\mathbf{r}, t). \quad (6)$$

iv) En intégrant l'Eq. (5) sur le volume  $V$  on obtient

$$\frac{d}{dt} \underbrace{\frac{1}{V} \int_V d\mathbf{r} n_A(\mathbf{r}, t)}_{= N_A(t)} = \frac{D}{V} \int_V d\mathbf{r} \Delta n_A(\mathbf{r}, t) - \frac{k}{V} \int_V d\mathbf{r} n_A(\mathbf{r}, t)n_B(\mathbf{r}, t). \quad (7)$$

On peut faire porter la première intégrale du membre de droite de l'Eq. (7) sur la frontière  $\partial V$  du volume  $V$ , et sa contribution sera donc nulle à cause des conditions aux bords périodiques.

On obtient ainsi

$$\underbrace{\frac{d}{dt} N_A(t)}_{\substack{(4)_{t^{-1}} \\ \sim -t^{-2}}} = -\frac{k}{V} \int_V d\mathbf{r} \underbrace{n_A(\mathbf{r}, t)n_B(\mathbf{r}, t)}_{(4)_{t^{-2}}}. \quad (8)$$

En supposant que le comportement asymptotique des densités locales est toujours en  $t^{-1}$ , alors le membre de droite de l'Eq. (8) va asymptotiquement comme  $-t^{-2}$ , ce qui est bien compatible avec le comportement du membre de gauche de l'Eq. (8).

- v) Il existe plusieurs possibilités pour programmer cette dynamique. On peut par exemple adopter la méthode présentée dans le livre *Cellular Automata Modeling of Physical Systems* (Bastien Chopard et Michel Droz, Cambridge University Press, Cambridge, 1998), ou toute autre approche pertinente. Le critère final pour décider si un programme s'avère être une bonne modélisation de la dynamique est qu'il fournisse a posteriori les bonnes prédictions pour les grandeurs d'intérêt. Programme Matlab :

```

clear;
N=100; %size of the system NxN
density=1.0; %initial density
Nmin=10; %stops when <= Nmin particles A or B
Nplot=1; %plot every Nplot iterations
%black-gray-white-map
black=[1 1 1];
gray=[0.5 0.5 0.5];
white=[0 0 0];
bcmap(1,:)=black;
bcmap(2,:)=gray;
bcmap(3,:)=white;
colormap(bcmap);
%initial conditions
a=zeros(N);
b=zeros(N);
k=0;
for i=1:N %on average, (N^2)*density/2 particles A
    for j=1:N
        if (rand(1,1) < density/2.)
            a(i,j)=1;
            k=k+1;
        end
    end
end
l=0;
for i=1:k %place k particles B: uncorrelated with A
    while (l==0)
        i2=floor(rand(1,1)*N)+1;
        j2=floor(rand(1,1)*N)+1;
        if (b(i2,j2)==0)
            b(i2,j2)=1;
            l=1;
        end
    end
end
l=0;
end
for i=1:N %removing A+B particles
    for j=1:N
        if (a(i,j)+b(i,j) > 1)
            remove=min(a(i,j),b(i,j));
            a(i,j)=a(i,j)-remove;
            b(i,j)=b(i,j)-remove;
        end
    end
end
end
t=0;

```

```

NA=sum(sum(a));
NB=sum(sum(b));
while ((NA > Nmin) | (NB > Nmin)) %time loop
    t=t+1;
    a=AB1(a,N);
    b=AB1(b,N);
    for i=1:N %removing multiple particles
        for j=1:N
            if (a(i,j)+b(i,j) > 1)
                remove=min(a(i,j),b(i,j));
                a(i,j)=a(i,j)-remove;
                b(i,j)=b(i,j)-remove;
            end
        end
    end
    densityplot(t,1)=t;
    NA=sum(sum(a));
    NB=sum(sum(b));
    densityplot(t,2)=NA+NB;
    if (mod(t,Nplot)==0)
        imagesc((a>0) + 0.5*(b>0));
        axis square off;
        drawnow;
    end
end
plot(log10(densityplot(:,1)),log10(densityplot(:,2)),'kx-');

```

avec les fichiers externes *AB1.m* :

```

function x2=AB1(x,N)
x2=zeros(N);
for i=1:N %N^2 algorithm: very inefficient, can do much better
    for j=1:N
        if (x(i,j)>0) %if there is at least 1 particle
            for k=1:x(i,j)
                p=floor(rand(1,1)*4.)+1;
                i2=i+AB2(p+1); %destination x
                j2=j+AB2(p); %destination y
                if ((i2>N) | (i2<1) | (j2>N) | (j2<1)) %boundary conditions
                    if (i2==N+1), i2=1; end
                    if (i2==0), i2=N; end
                    if (j2==N+1), j2=1; end
                    if (j2==0), j2=N; end
                end
                x2(i2,j2)=x2(i2,j2)+1; %moving particle
            end
        end
    end
end
end

```

et *AB2.m* :

```
%create displacements {(1,0),(0,1),(-1,0),(0,-1)}={f(x+1),f(x)}, x in [1,4]
function f=AB2(x)
if((x==1) | (x==3) | (x==5)), f=0; end
if(x==2), f=1; end
if(x==4), f=-1; end
```

Programme Fortran 90 :

```
module fonctions
implicit none
integer, parameter :: N = 500 !size of the system NxN
real, parameter :: density = 1.0 !initial density
integer, parameter :: Nmin = 5 !stops when <= Nmin particles A or B
character*(*), parameter :: filename = 's4.txt' !name of the output file
character*(*), parameter :: filenameA = 's4a.txt' !output file: grid A
character*(*), parameter :: filenameB = 's4b.txt' !output file: grid B
integer, parameter :: Nsave = 10000 !saving grid in a file every Nsave timesteps
integer :: seed
contains
!*****
FUNCTION ran2(idum) !Random generator. Initialize idum with negative integer
INTEGER idum,IM1,IM2,IMM1,IA1,IA2,IQ1,IQ2,IR1,IR2,NTAB,NDIV
REAL :: ran2,AM,EPS,RNMX
PARAMETER (IM1=2147483563,IM2=2147483399,AM=1./IM1,IMM1=IM1-1,IA1=40014,&
&IA2=40692,IQ1=53668,IQ2=52774,IR1=12211,IR2=3791,NTAB=32,&
&NDIV=1+IMM1/NTAB,EPS=1.2e-7,RNMX=1.-EPS)
INTEGER :: idum2,j,k,iv(NTAB),iy
SAVE iv,iy,idum2
DATA idum2/123456789/, iv/NTAB*0/, iy/0/
if(idum.le.0)then
idum=max(-idum,1)
idum2=idum
do j=NTAB+8,1,-1
k=idum/IQ1
idum=IA1*(idum-k*IQ1)-k*IR1
if(idum.lt.0)idum=idum+IM1
if(j.le.NTAB)iv(j)=idum
enddo
iy=iv(1)
endif
k=idum/IQ1
idum=IA1*(idum-k*IQ1)-k*IR1
if(idum.lt.0)idum=idum+IM1
k=idum2/IQ2
idum2=IA2*(idum2-k*IQ2)-k*IR2
if(idum2.lt.0)idum2=idum2+IM2
j=1+iy/NDIV
iy=iv(j)-idum2
iv(j)=idum
if(iy.lt.1)iy=iy+IMM1
ran2=min(AM*iy,RNMX)
return
```

```

end function ran2
!*****
!create displacements {(1,0),(0,1),(-1,0),(0,-1)}={f(x+1),f(x)}, x in [1,4]
function f(x)
integer, intent(in) :: x
integer :: f
if((x==1) .or. (x==3) .or. (x==5)) f=0
if(x==2) f=1
if(x==4) f=-1
return
end function f
!*****
subroutine deplacement(x)
integer, dimension(1:N,1:N), intent(inout) :: x
integer, dimension(1:N,1:N) :: x2
integer :: i,j,k,p,i2,j2
x2=0
do i=1,N !N^2 algorithm: very inefficient, can do much better
  do j=1,N
    if (x(i,j)>0) then !if there is at least 1 particle
      do k=1,x(i,j)
        p = int(ran2(seed)*4.)+1
        i2=i+f(p+1) !destination x
        j2=j+f(p) !destination y
        if ((i2>N) .or. (i2<1) .or. (j2>N) .or. (j2<1)) then !boundary cond.
          if (i2==N+1) i2=1
          if (i2==0) i2=N
          if (j2==N+1) j2=1
          if (j2==0) j2=N
        end if
        x2(i2,j2)=x2(i2,j2)+1 !moving particle
      end do
    end if
  end do
end do
x=x2 !output
return
end subroutine deplacement
!*****
end module fonctions
!-----
program s4
use fonctions
implicit none
integer :: i,j,t,i2,j2
real :: logplot
integer, dimension(1:N,1:N) :: a,b !lattices
seed = -14731
a=0;b=0;t=0
do i=1,N !on average, (N^2)*density/2 particles A
  do j=1,N
    if (ran2(seed) < density/2.) then
      a(i,j) = 1
      t=t+1
    end if
  end do
end do

```

```

        end if
    end do
end do
do i=1,t !place t particles B: uncorrelated with A
    nouveau: do
        i2=int(ran2(seed)*real(N))+1
        j2=int(ran2(seed)*real(N))+1
        if (b(i2,j2)==0) then
            b(i2,j2)=1
            exit nouveau
        end if
    end do nouveau
end do
where(a+b==2)!removing A+B particles
    a=0;b=0
end where
t=0 !initializing
logplot=0.
open(unit=1,file=filename)
close(1,status='DELETE')
boucle: do !time loop
    t=t+1
    call deplacement(a)
    call deplacement(b)
    do i=1,N !removing multiple A+B particles
        do j=1,N
            if (a(i,j)+b(i,j) > 1) then
                i2=min(a(i,j),b(i,j))
                a(i,j)=a(i,j)-i2
                b(i,j)=b(i,j)-i2
            end if
        end do
    end do
    if (mod(t,Nsave)==0) then !saving the grid
        open(unit=1,file=filenameA)
        open(unit=2,file=filenameB)
        do i=1,N
            do j=1,N
                if (a(i,j)==1) write(1,*) i,j
                if (b(i,j)==1) write(2,*) i,j
            end do
        end do
        close(1)
        close(2)
    end if
    if ((log(real(t)) - logplot > 0.01) .or. (t==1)) then !lin. density in logplot
        logplot=log(real(t))
        open(unit=1,file=filename,position='APPEND') !saving n_A(t), n_B(t)
        write(1,*) t,sum(a),sum(b),sum(a+b)
        close(1)
    end if
    if ((sum(a) <= Nmin) .or. (sum(b) <= Nmin)) exit boucle
end do boucle
end program s4

```

On constate la formation par la dynamique d'amas de particules de type  $A$  ou  $B$  (cf. Figs. 1, 2, et 3).

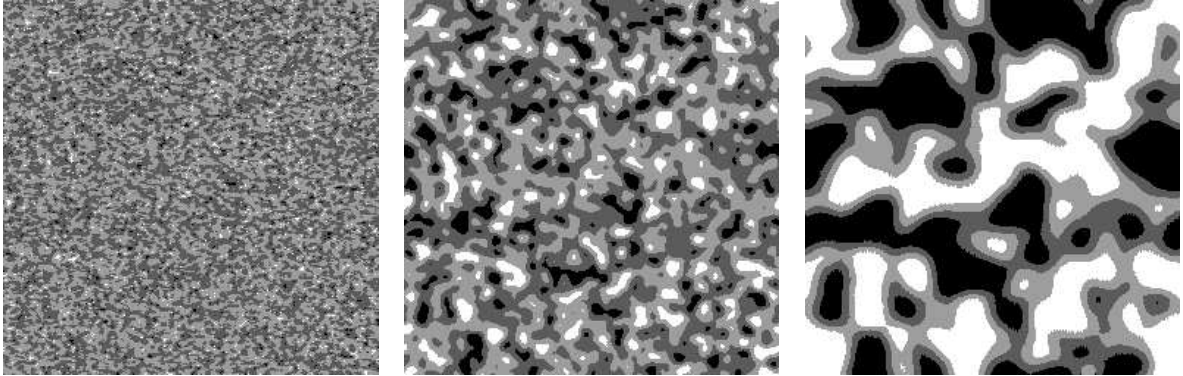


FIG. 1 – Apparition d'amas de particules de type  $A$  (blanc) et  $B$  (noir). Les zones en gris clair (foncé) sont telles que  $n_A > n_B$  ( $n_A < n_B$ ), respectivement. A gauche la condition initiale homogène, et les images suivantes représentent l'état du système à des temps ultérieurs. Ces images sont obtenues par une simulation Boltzmann sur réseau. Une animation correspondante peut être visualisée depuis l'adresse internet <http://cui.unige.ch/~chopard/CA/Animations/img-root.html> (en lien depuis le site du cours <http://mpej.unige.ch/~coppex/cours>). Les résultats pour une simulation d'automate cellulaire sont présentés dans les Figs. 2 et 3.

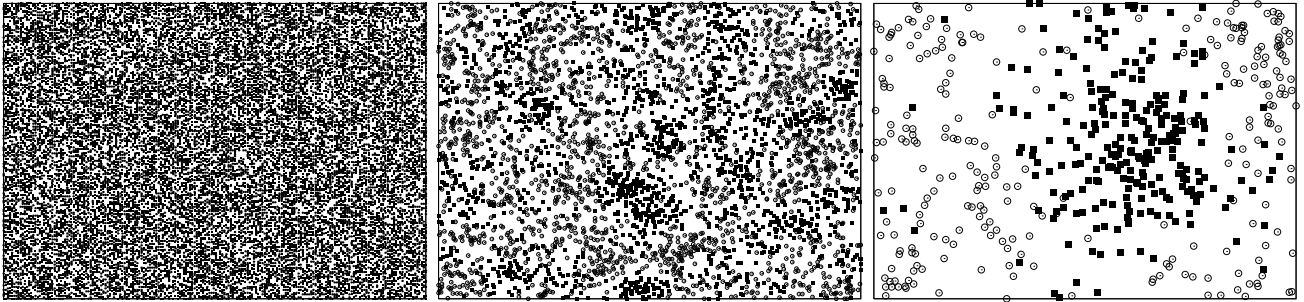


FIG. 2 – Dynamique de réaction-diffusion  $A + B \rightarrow \emptyset$  obtenue par automate cellulaire. Les particules de type  $A$  sont représentées par ■, et les particules  $B$  par ○. A gauche la condition initiale homogène, et les images suivantes représentent l'état du système à des temps ultérieurs.

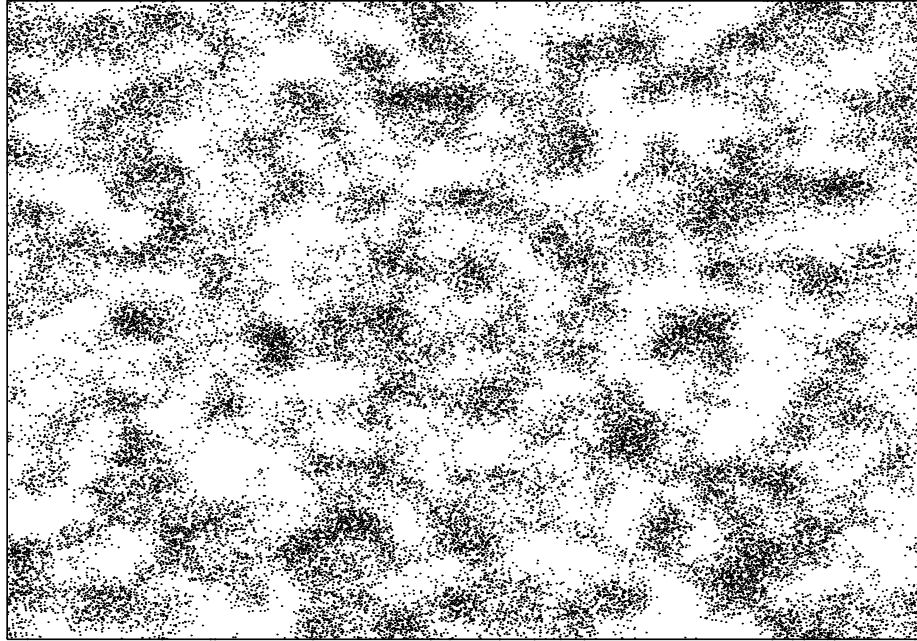


FIG. 3 – La formation d'inhomogénéités devient bien visible lorsqu'on ne représente qu'une seule espèce. La taille du système est de  $N = 6'000$ , et le nombre de pas de temps de  $20'000$ .

vi) Dans la limite des longs temps, on observe bien une loi de puissance (correspondant à une droite dans un graphique log-log) avec exposant proche de  $-1/2$  (cf. Fig. 4).

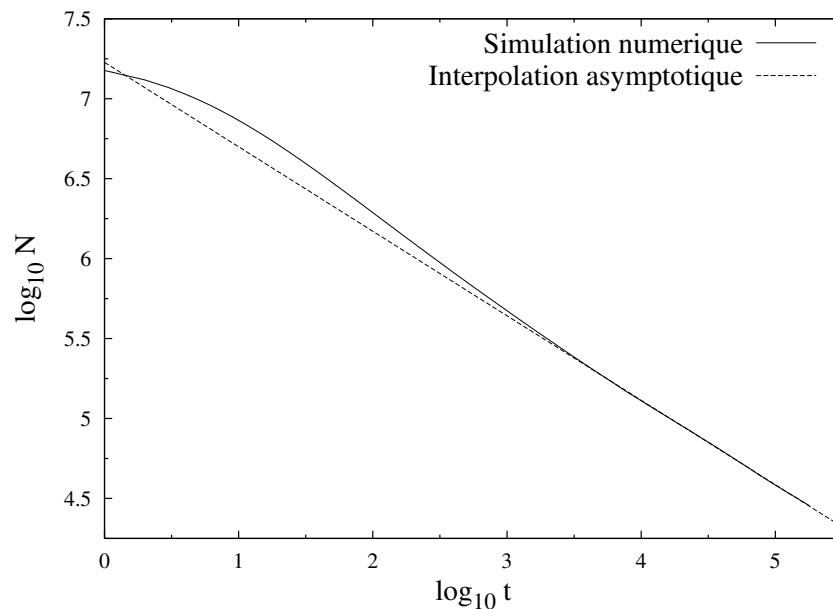


FIG. 4 – Nombre total de particules en fonction du temps dans un diagramme log-log pour un système de taille  $N = 6'000$  avec densité initiale 1. L'exposant (la pente) ainsi trouvé est  $\beta = -0.528\dots$



vii) Les simulations numériques fournissent un comportement asymptotique en  $t^{-1/2}$ , alors que les modèles de type champ-moyen fournissent tous deux un comportement en  $t^{-1}$ . On a constaté au point v) la formation par la dynamique d'amas de particules de type  $A$  ou  $B$ . Les fluctuations sont amplifiées par la dynamique et le système n'est donc plus homogène. La réaction d'annihilation ne se déroule donc plus de façon homogène dans le système, mais uniquement aux frontières entre ces amas de particules. La réaction n'est donc plus volumique, mais de surface, d'où le ralentissement de la dynamique.

viii) On peut comprendre la loi de décroissance en  $t^{-d/4}$  comme suit. Soit un intervalle temporel  $\tau$ . De par la diffusion, une particule parcourt durant  $\tau$  une distance de l'ordre de

$$\ell = \sqrt{D\tau}. \quad (9)$$

Divisons à présent le volume  $V$  en volumes  $\omega$  d'extension spatiale  $\ell$ ,  $\omega \simeq \ell^d$ . Soit  $L$  la taille du système, alors il existe  $V/\omega \simeq (L/\ell)^d$  tels volumes. Supposons les densités initiales de particules égales  $N_A(0) = N_B(0) = N_0/2$ , alors le nombre moyen  $N$  de particules  $A$  ou  $B$  dans les petits volumes  $\omega$  sera

$$N \simeq (N_0/2)\omega \simeq (N_0/2)\ell^d \simeq (N_0/2)(D\tau)^{d/2}. \quad (10)$$

Néanmoins, la grandeur  $N$  est soumise à des fluctuations statistiques qui sont de l'ordre de  $\sqrt{N}$ . Par conséquent, si les particules  $A$  et  $B$  sont non corrélées pour tout temps, les différences de densité entre  $N_A(t)$  et  $N_B(t)$  seront typiquement de l'ordre de

$$\sqrt{N} \simeq \sqrt{N_0/2}(D\tau)^{d/4}. \quad (11)$$

Si durant l'intervalle  $\tau$  toutes les particules qui se trouvaient initialement dans un volume  $\omega$  s'annihilent (ce qui est une hypothèse raisonnable car  $\tau$  est par définition le temps nécessaire aux particules pour explorer le volume  $\omega$ ), le nombre de particules restant après ce temps  $\tau$  sera donné par l'excès de particules d'une espèce. Ce nombre est de l'ordre des fluctuations  $\sqrt{N}$ , et donc

$$N_A(\tau) = N_B(\tau) \simeq \sqrt{N} \frac{V}{\omega} \simeq \sqrt{N} \left(\frac{L}{\ell}\right)^d \simeq \sqrt{N} L^d (D\tau)^{-d/2} \simeq L^d \sqrt{N_0/2} (D\tau)^{-d/4}. \quad (12)$$

On a ainsi bien le comportement  $N_A(t) \simeq N_B(t) \sim t^{-d/4}$ .