Université Paris-Dauphine
Département MIDO Analyse numérique 3 , M1

T.P. 2 : Equation de la chaleur effet régularisant *Correction*

```
function TP2bis
close all

% Discretisation-----------------------------------------------------------------------------
L                           = 1 ;
T                           = 1 ;
N                           = 1000 ;
Mx                          = 30 ;
My                          = 30 ;
M                           = (Mx-1)*(My-1) ;
t                           = linspace(0,T,N) ;
x                           = linspace(0,L,Mx+1) ;
y                           = linspace(0,L,My+1) ;
dt                          = t(2)-t(1) ;
dx                          = x(2)-x(1) ;
dy                          = y(2)-y(1) ;

factgx                      = 1e3 ;
factgy                      = 1e3 ;
rho_ini                     = (ones(My+1,1)*[exp(-factgx*(x-.5).^2)])...
                              .* ([exp(-factgy*(y-.5).^2)]'*ones(1,Mx+1)) ;
rho_ini                     = rho_ini(2:Mx,2:My) ;
rho_ini                     = rho_ini/(dx*dy*sum(sum(rho_ini))) ;
rho_ini                     = rho_ini(:) ;
% Matrices N S / E O-------------------------------------------------------------------------
eps                         = 1e-1 ;
Pn                          = coords([2:Mx-1 Mx-1]   , [1:My-1]     , Mx-1,My-1) ;
Ps                          = coords([1 1:Mx-2]      , [1:My-1]     , Mx-1,My-1) ;
Pe                          = coords([1:Mx-1]        , [2:My-1 My-1], Mx-1,My-1) ;
Po                          = coords([1:Mx-1]        , [1 1:My-2]   , Mx-1,My-1) ;

Pnt                         = coords([2:Mx-1]        , [1:My-1]     , Mx-1,My-1) ;
Pst                         = coords([1:Mx-2]        , [1:My-1]     , Mx-1,My-1) ;
Pet                         = coords([1:Mx-1]        , [2:My-1]     , Mx-1,My-1) ;
Pot                         = coords([1:Mx-1]        , [1:My-2]     , Mx-1,My-1) ;

A= (1-2*eps*dt*(1/dx^2+1/dy^2))*  sparse(1:M,1:M,ones(M,1),M,M) ...
    + eps*dt/(dx^2)          * (sparse(Pn, 1:M,ones(M,1),M,M) + sparse(Ps,1:M,ones(M,1),M,M)) ...
    + eps*dt/(dy^2)          * (sparse(Pe, 1:M,ones(M,1),M,M) + sparse(Po,1:M,ones(M,1),M,M)) ;
 % A est la matrice du Laplacien

%-------------------------------------------------------------------------------------------------
equation_chaleur(A,N,Mx,My,x,y,dx,dy,rho_ini) ;
```

```
%-------------------------------------------------------------------------------------------------------
function equation_chaleur(A,N,Mx,My,x,y,dx,dy,rho_ini)
rho = rho2D1D(rho_ini) ;
for j=2:N
    rho                         = A * rho ;
    axis([x(1) x(Mx) y(1) y(My) 0 10])
    mesh(ones(My-1,1)*x(1,2:Mx), y(1,2:My)'*ones(1,Mx-1),rho1D2D(rho,Mx-1,My-1)) ;title(num2str(j));
    pause(.001) ;
end
%-------------------------------------------------------------------------------------------------------
function rho2D = rho1D2D(rho,k,l)
rho2D                           = reshape(rho,k,l) ;
%-------------------------------------------------------------------------------------------------------
function rho1D = rho2D1D(rho2D)
rho1D                           = rho2D(:) ;
%-------------------------------------------------------------------------------------------------------
function coord = coords(i,j,k,l)
c                               = reshape(1:k*l,k,l) ;
coord                           = c(i,j) ;
coord                           = coord(:) ;
```