

---

## Table of Contents

.....	1
Initialisation de la source de chaleur et de la conductivité thermique .....	2
Construction de la matrice des coefficients .....	2
Question 1 .....	6
Question 2 .....	7
Question 3 .....	11

```
% Distribution de la température dans un appartement d'un immeuble aux
plusieurs étages
```

```
clear all
```

```
clc
```

```
% Équation de transfert de chaleur:
```

```
%  $k \cdot (d^2 T(x,y)/dx^2 + d^2 T(x,y)/dy^2) + S = 0$ 
```

```
% Conditions aux limites:
```

```
% (1) Condition convective (de Robin) à  $x=0$  et à  $x=L_x$  (faces externes
du mur):
```

```
%  $-k \cdot dT(x=0,y)/dx = -h \cdot (T - T_a)$ 
```

```
%  $-k \cdot dT(x=L,y)/dx = h \cdot (T - T_a)$ 
```

```
Ta=-10; %oC
```

```
% Dimensions d'appartement
```

```
Lx=4; %[m]
```

```
Ly=2.4; %[m]
```

```
% Parametres d'un mur d'isolation thermique
```

```
Lm=0.4; %m ; Épaisseur du mur en brique
```

```
km=0.85;%W/(m*K); La conductivité thermique de la brique
```

```
h=20; %W/(m^2*K); Coefficient de transfert thermique sur les surfaces
extérieures du mur
```

```
% Paramètres de l'air qui remplit l'appartement
```

```
ka=0.024;
```

```
d_ar=[];tini_ar=[];tinv_ar=[];mem_ar=[];Tm_ar=[];tempMileu=[];
```

```
for fact=100e-2 * [1 1/2 1/4] % On veut : fact=10e-2 * [1 1/2 1/4],
mais le code plante
```

```
    d=0.1*fact; %Pas de discrétisation en [m]
```

```
    d_ar=[d_ar d];
```

---

```
Nx=round(Lx/d)+1;
Ny=round(Ly/d)+1;
Nm=round(Lm/d)+1;
```

```
tic
```

## Initialisation de la source de chaleur et de la conductivité thermique

```
S=zeros(Ny,Nx); k=zeros(Ny,Nx);
for i=1:Ny
    y=(i-1)*d;
    for j=1:Nx
        x=(j-1)*d;

        % Source volumique de chaleur q[W/m^3] d'épaisseur dL.
        % La source est intégrée dans les parties intérieures du
mur à x=Lm et à x=Lx-Lm et
        % il occupe les tiers du mur dans la direction verticale
        dL=0.1;
        q=1e4;% W/m^3;
        if (x<=Lm)&&(y<=Ly/3+Lm)&&(y>Lm)
            % À l'intérieur de l'élément chauffant
            S(i,j)=q*exp(-(x-Lm)/dL).^2);
        elseif (x>=(Lx-Lm))&&(y<=Ly/3+Lm)&&(y>Lm)
            % À l'intérieur de l'élément chauffant
            S(i,j)=q*exp(-(Lx-Lm-x)/dL).^2);
        else
            % À l'extérieur de l'élément chauffant
            S(i,j)=0;
        end

        % L'espace de vie de l'appartement est délimité par
        % les parois d'épaisseur Lm à tous les quatre côtés
        if (x<=Lm) || (x>=(Lx-Lm)) || (y<=Lm) || (y>=(Ly-Lm))
            % À l'intérieur du mur
            k(i,j)=km;
        else
            % À l'intérieur de l'appartement
            k(i,j)=ka;
        end
    end
end
end
```

## Construction de la matrice des coefficients

```
M=sparse(zeros(Nx*Ny,Nx*Ny));
b=zeros(Nx*Ny,1);

for i=1:Ny
```

---

```

for j=1:Nx
% remplir la ligne pl de la matrice M
index=(i-1)*Nx+j;

if (i==1) & (j~=1)& (j~=Nx)
% noeud sur le plafond y=0
pl = index;
pc = index;      M(pl,pc)=3+2*d*h/k(i,j);
pc=index+1*Nx;   M(pl,pc)=-4;
pc=index+2*Nx;   M(pl,pc)=1;
b(index)=2*d*h*Ta/k(i,j);
test{i,j} = 'ext';
elseif (i==Ny) & (j~=1)& (j~=Nx)
% noeud sur le plancher y=Ly
pl = index;
pc = index;      M(pl,pc)=3+2*d*h/k(i,j);
pc=index-1*Nx;   M(pl,pc)=-4;
pc=index-2*Nx;   M(pl,pc)=1;
b(index)=2*d*h*Ta/k(i,j);
test{i,j} = 'ext';
elseif (j==1)& (i~=1)& (i~=Ny)
% noeud à la surface externe du mur x=0
pl = index;
pc = index;      M(pl,pc)=3+2*d*h/k(i,j);
pc=index+1;      M(pl,pc)=-4;
pc=index+2;      M(pl,pc)=1;
b(index)=2*d*h*Ta/k(i,j);
test{i,j} = 'ext';
elseif (j==Nx)& (i~=1)& (i~=Ny)
% noeud à la surface externe du mur x=Nx
pl = index;
pc = index;M(pl,pc)=3+2*d*h/k(i,j);
pc=index-1;M(pl,pc)=-4;
pc=index-2;M(pl,pc)=1;
b(index)=2*d*h*Ta/k(i,j);
test{i,j} = 'ext';

% Coins extérieur
elseif (i==1)&&(j==1)
pl = index;
pc = index;      M(pl,pc)=3+2*d*h/k(i,j);
pc=index+1 + 1*Nx;      M(pl,pc)=-4;
pc=index+2 + 2*Nx;      M(pl,pc)=1;
b(index)=2*d*h*Ta/k(i,j);
elseif (i==1)&&(j==Nx)
pl = index;
pc = index;      M(pl,pc)=3+2*d*h/k(i,j);
pc=index-1 + 1*Nx;      M(pl,pc)=-4;
pc=index-2 + 2*Nx;      M(pl,pc)=1;
b(index)=2*d*h*Ta/k(i,j);
elseif (i==Ny)&&(j==1)
pl = index;
pc = index;      M(pl,pc)=3+2*d*h/k(i,j);

```

---

---

```

        pc=index+1 - 1*Nx;      M(pl,pc)=-4;
        pc=index+2 - 2*Nx;      M(pl,pc)=1;
        b(index)=2*d*h*Ta/k(i,j);
elseif (i==Ny)&&(j==Nx)
    pl = index;
    pc = index;      M(pl,pc)=3+2*d*h/k(i,j);
    pc=index-1 - 1*Nx;      M(pl,pc)=-4;
    pc=index-2 - 2*Nx;      M(pl,pc)=1;
    b(index)=2*d*h*Ta/k(i,j);

% Mur intérieur
elseif (i==Nm+1)& (j>Nm+1 & j<Nx-Nm)
    % noeud à la surface interne du mur y=Nm
    pl = index;
    pc = index;      M(pl,pc)=(3)*ka;
    pc = index+1*Nx; M(pl,pc)=(-4)*ka;
    pc = index+2*Nx; M(pl,pc)=ka;
    pc = index-1*Nx;  M(pl,pc)=- (3)*km;
    pc = index-2*Nx;  M(pl,pc)=- (-4)*km;
    pc = index-3*Nx;  M(pl,pc)=-km;
    b(index)=0;
    test{i,j} = 'int';
elseif (i==Ny-Nm)& (j>Nm+1 & j<Nx-Nm)
    % noeud à la surface interne du mur y = Ny-Nm
    pl = index;
    pc = index;      M(pl,pc)=(3)*ka;
    pc = index-1*Nx; M(pl,pc)=(-4)*ka;
    pc = index-2*Nx; M(pl,pc)=ka;
    pc = index+1*Nx;  M(pl,pc)=- (3)*km;
    pc = index+2*Nx;  M(pl,pc)=- (-4)*km;
    pc = index+3*Nx;  M(pl,pc)=-km;
    b(index)=0;
    test{i,j} = 'int';
elseif (j==Nm+1) &(i>Nm+1 & i<Ny-Nm)
    % noeud à la surface interne du mur x=Nm
    pl = index;
    pc = index;      M(pl,pc)=(3)*ka;
    pc = index+1; M(pl,pc)=(-4)*ka;
    pc = index+2; M(pl,pc)=ka;
    pc = index-1;  M(pl,pc)=- (3)*km;
    pc = index-2;  M(pl,pc)=- (-4)*km;
    pc = index-3;  M(pl,pc)=-km;
    b(index)=0;
    test{i,j} = 'int';
elseif (j==Nx-Nm) &(i>Nm+1 & i<Ny-Nm)
    % noeud à la surface interne du mur x=Nx-Nm
    pl = index;
    pc = index;      M(pl,pc)=(3)*ka;
    pc = index-1; M(pl,pc)=(-4)*ka;
    pc = index-2; M(pl,pc)=ka;
    pc = index+1;  M(pl,pc)=- (3)*km;
    pc = index+2;  M(pl,pc)=- (-4)*km;
    pc = index+3;  M(pl,pc)=-km;

```

---

---

```

b(index)=0;
test{i,j} = 'int';

% Coins intérieur
elseif (i==Nm+1)&&(j==Nm+1)
    pl = index;
    pc = index;          M(pl,pc)=(3)*ka;
    pc = index+1+1*Nx;    M(pl,pc)=(-4)*ka;
    pc = index+2+2*Nx;    M(pl,pc)=ka;
    pc = index-1-1*Nx;    M(pl,pc)=- (3)*km;
    pc = index-2-2*Nx;    M(pl,pc)=- (-4)*km;
    pc = index-3-3*Nx;    M(pl,pc)=-km;
    b(index)=0;
    test{i,j} = 'coin';
elseif (i==Nm+1)&&(j==Nm+1)
    pl = index;
    pc = index;          M(pl,pc)=(3)*ka;
    pc = index-1+1*Nx;    M(pl,pc)=(-4)*ka;
    pc = index-2+2*Nx;    M(pl,pc)=ka;
    pc = index+1-1*Nx;    M(pl,pc)=- (3)*km;
    pc = index+2-2*Nx;    M(pl,pc)=- (-4)*km;
    pc = index+3-3*Nx;    M(pl,pc)=-km;
    b(index)=0;
    test{i,j} = 'coin';
elseif (i==Ny-Nm)&&(j==Nm+1)
    pl = index;
    pc = index;          M(pl,pc)=(3)*ka;
    pc = index+1-1*Nx;    M(pl,pc)=(-4)*ka;
    pc = index+2-2*Nx;    M(pl,pc)=ka;
    pc = index-1+1*Nx;    M(pl,pc)=- (3)*km;
    pc = index-2+2*Nx;    M(pl,pc)=- (-4)*km;
    pc = index-3+3*Nx;    M(pl,pc)=-km;
    b(index)=0;
    test{i,j} = 'coin';
elseif (i==Ny-Nm)&&(j==Nx-Nm)
    pl = index;
    pc = index;          M(pl,pc)=(3)*ka;
    pc = index-1-1*Nx;    M(pl,pc)=(-4)*ka;
    pc = index-2-2*Nx;    M(pl,pc)=ka;
    pc = index+1+1*Nx;    M(pl,pc)=- (3)*km;
    pc = index+2+2*Nx;    M(pl,pc)=- (-4)*km;
    pc = index+3+3*Nx;    M(pl,pc)=-km;
    b(index)=0;
    test{i,j} = 'coin';

elseif ((i>1)&(i<Ny))&&((j>1)&(j<Nx))
    % noeud qui est strictement à l'intérieur de la
cellule de simulation
    pl=index;
    pc=index;M(pl,pc)=-4; % contribution de noeud (i,j)

```

---

---

```

        pc=(i-1)*Nx+j-1;M(pl,pc)=1; % contribution de noeud
    (i,j-1)
        pc=(i-1)*Nx+j+1;M(pl,pc)=1; % contribution de noeud
    (i,j+1)
        pc=(i-2)*Nx+j;M(pl,pc)=1; % contribution de noeud
    (i-1,j)
        pc=(i)*Nx+j;M(pl,pc)=1; % contribution de noeud (i
+1,j)
        b(pl)=-d^2*S(i,j)/k(i,j);
        test{i,j} = '';

    else
        display('Erreur dans la définition de la matrice de
coefficients');
        disp(i)
        disp(j)
    end
end
M = sparse(M);
end
tini_ar=[tini_ar toc];

tic
%T=M\b;
[L,U]=lu(M);T=U\ (L\b);

tinv_ar=[tinv_ar toc];

mem_ar=[mem_ar 8*(Nx*Ny)^2];
Delta = sqrt(sqrt(8./mem_ar)*Lx*Ly);

Tr=reshape(T,Nx,Ny)';
Tm_ar=[Tm_ar Tr(round(Ly/d/2+1),round(Lx/d/2+1))];
ErrTemp=abs(Tm_ar(1:end-1)-Tm_ar(2:1:end));

end

```

## Question 1

```

fprintf('La température au milieu est %.0f C \n',Tm_ar(end))
fprintf('L'erreur sur la température est de %.0f C \n',ErrTemp(end))

% figure(1)
% h=pcolor((0:d:Lx),(0:d:Ly),S);set(h,'LineStyle','none')
% colorbar
% xlabel('x [m]'); ylabel('y [m]'); title('S(x,y) [W/m^3]')
% axis equal
% axis tight

```

---

```

% figure(2)
% h=pcolor((0:d:Lx),(0:d:Ly),k);set(h,'LineStyle','none')
% colorbar
% xlabel('x [m]'); ylabel('y [m]'); title('k(x,y) [W/(m^2\cdotK)]')
% axis equal
% axis tight

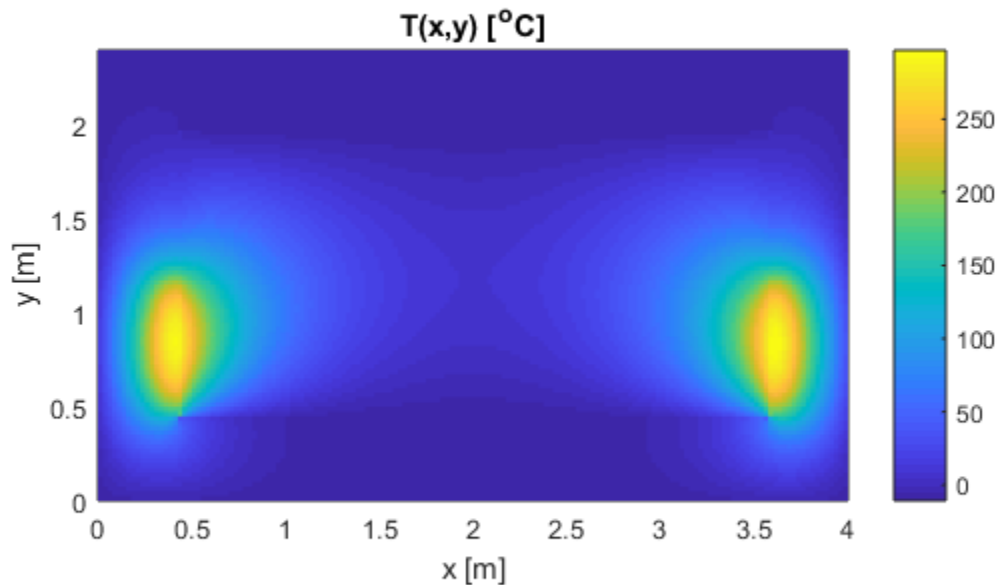
```

```

figure(3)
h=pcolor((0:d:Lx),(0:d:Ly),Tr);set(h,'LineStyle','none')
colorbar
% caxis([-20 20])
xlabel('x [m]'); ylabel('y [m]'); title('T(x,y) [^oC]')
axis equal
axis tight

```

*La température au milieu est 8 C*  
*L'erreur sur la température est de 3 C*



## Question 2

```

figure(4)
loglog(d_ar,mem_ar/1024^3,':o')
axis([1e-3 1 1e-4 128])
axis square
xlabel('\delta [m]')
ylabel('mem [Gb]')

```

---

```

title('Mémoire nécessaire selon le pas')

figure(5)
loglog(d_ar,tini_ar,':o')
axis([1e-3 1 1e-3 10000])
axis square
xlabel('\delta [m]')
ylabel('t_{ini} [s]')
title('Temps d''initialisation de la matrice selon le pas')

figure(6)
loglog(d_ar,tinv_ar,':o')
axis([1e-3 1 1e-3 10000])
axis square
xlabel('\delta [m]')
ylabel('t_{inv} [s]')
title('Temps d''inversion de la matrice selon le pas')

figure(7)
loglog(d_ar,Tm_ar,':o')
axis tight
xlabel('\delta [m]')
ylabel('Err [°C]')
title('Erreur sur la température au milieu selon le pas')

poly_mem = polyfit(log(d_ar),log(mem_ar/1024^3),1);
poly_ini = polyfit(log(d_ar),log(tini_ar),1);
poly_inv = polyfit(log(d_ar),log(tinv_ar),1);
poly_err = polyfit(log(d_ar),log(Tm_ar),1);

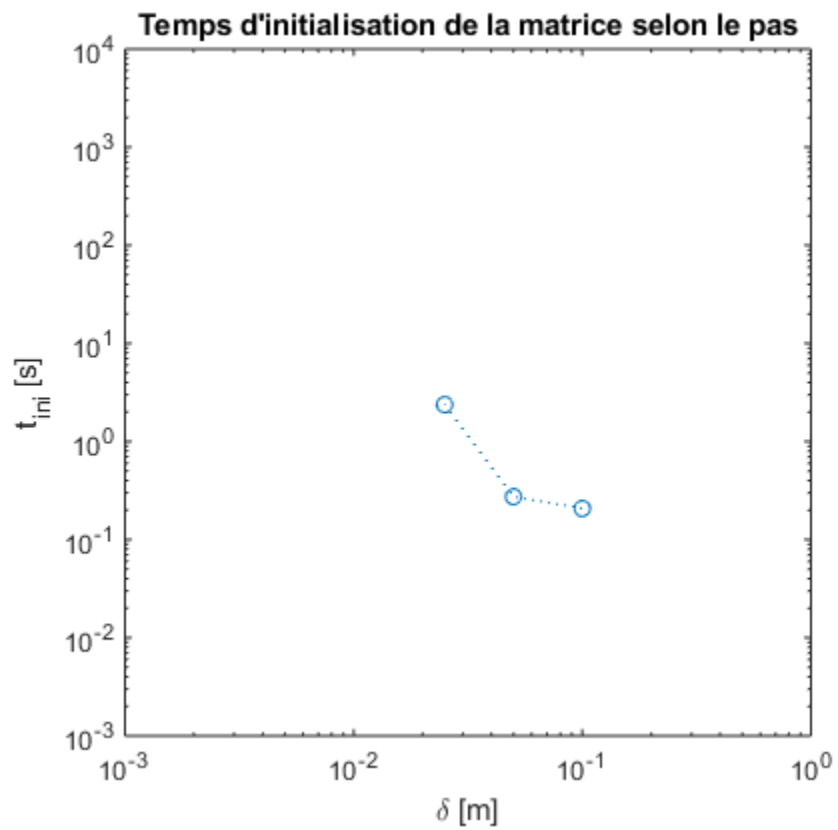
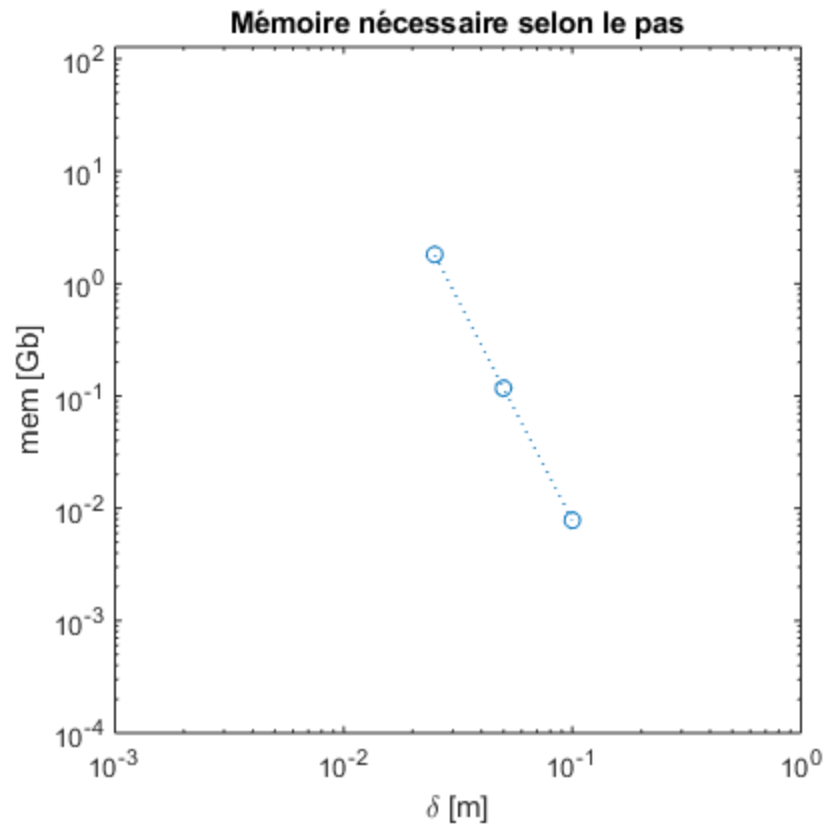
fprintf('Les coefficients pour la mémoire nécessaire sont P_mem = %.2f\n',poly_mem(1),poly_mem(2))
fprintf('Les coefficients pour le temps d''initialisation sont P_ini = %.2f et A_ini = %.2f\n',poly_ini(1),poly_ini(2))
fprintf('Les coefficients pour le temps d''inversion sont P_inv = %.2f et A_inv = %.2f\n',poly_inv(1),poly_inv(2))
fprintf('Les coefficients pour l''erreur sur la température au milieu sont P_err = %.2f et A_err = %.2f\n',poly_err(1),poly_err(2))

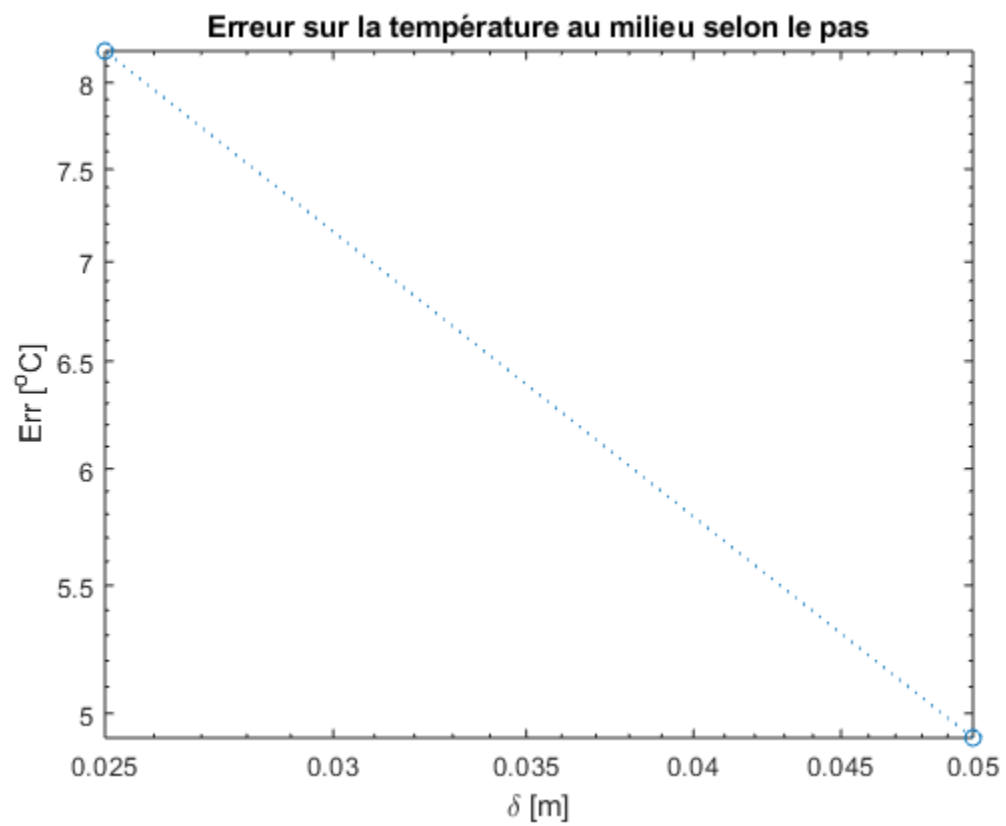
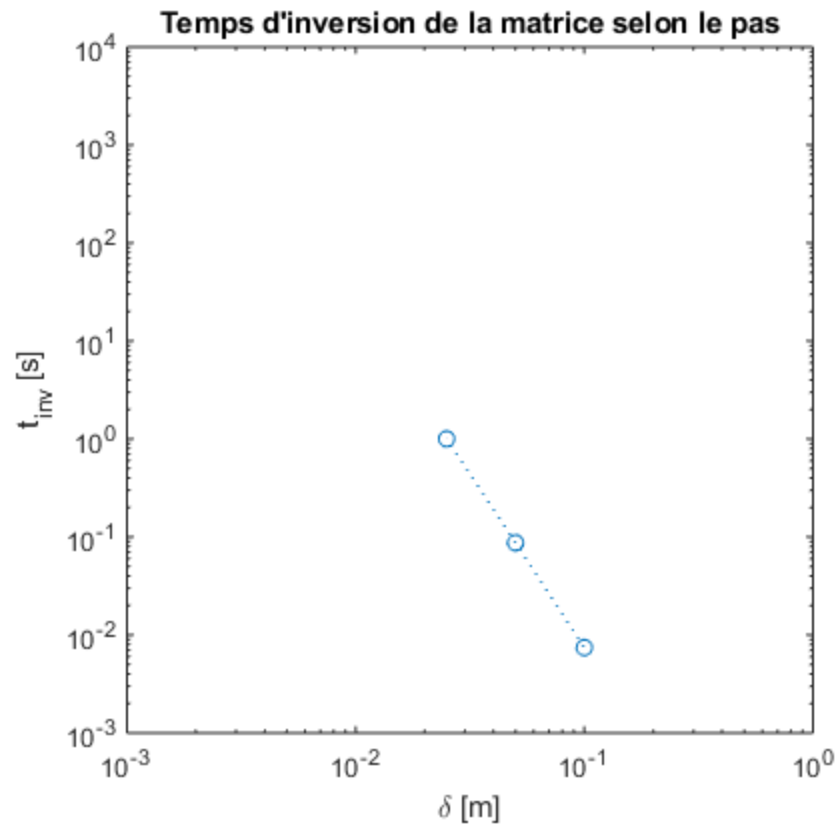
Warning: Negative data ignored
Les coefficients pour la mémoire nécessaire sont P_mem = -3.93 et
A_mem = -13.90
Les coefficients pour le temps d'initialisation sont P_ini = -1.76 et
A_ini = -5.94
Les coefficients pour le temps d'inversion sont P_inv = -3.54 et A_inv
= -13.04
Les coefficients pour l'erreur sur la température au milieu sont P_err
= -0.35 et A_err = 0.71

```

---







---

## Question 3

```
delta_min =sqrt(sqrt(8/(128e9))*Lx*Ly);
```

```
fprintf('Le pas de discrétisation minimal avec 128 Gb de RAM est delta  
= %.4f \n',delta_min)
```

```
Le pas de discrétisation minimal avec 128 Gb de RAM est delta =  
0.0087
```

*Published with MATLAB® R2017b*