```
% PAQUOT YVAN

% QUESTION 1

% L'équation  x = 1 - 5cos(x.x.x) possède beaucoup de racines. % Déterminez
une approximation de celle qui est la plus proche
% de x = 2 avec au moins 10 chiffres exacts.
% Utilisez l'une des quatre méthodes suivantes :
% a) Dichotomie
% b) Point fixe
% c) Sécante
% d) Tangente
% Justifiez le choix de la méthode numérique employée et
% estimez sa vitesse de convergence.
%===========================================================================

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!                                          !!
!!  ATTENTION !!!!!  Pour cette question, je me suis trompé de méthode :    !!
!!  J'ai confondu les noms de "sécante" et "tangente". Il fallait utiliser  !!
!!  La méthode de la tangente qui n'est autre que celle de NEWTON.          !!
!!  Dans le choix de la méthode, c'est donc la sécante qui est instable     !!
!!  numériquement et la tangente qui est la meilleure.                      !!
!!                                          !!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!


%Choix de la méthode:
% 1°) La méthode de la tangente est rejetée, car elle souffre d'instabilité
% numérique en convergence avancée. Or,la précision demandée est très
% grande
% 2°) La méthode du Point fixe est rejetée, car g' > 1 aux alentours de 2
% 3°) Les méthodes de dichotomie et de la sécante peuvent toutes deux
% s'appliquer (racine simple).
% 4°) La méthode de la sécante est plus rapide. Elle est donc choisie.


%équation à résoudre : 1 - 5cos(x3) -x =0
clear
clc

% 1°) Graphe de la fonction
g=inline('1 - 5*cos(x.*x.*x) -x');
x=linspace(0,3,50000);
y=g(x);
plot(x,y);
grid on;
axis equal;

Nitermax=20;
clear x;
format long;

%Cherchons graphiquement 2 précurseurs de part et d'autre de la racine:
x(1)=1.95;
```

```matlab
x(2)=2.05;

f=g;
for i=3:Nitermax  % Condition d'arrêtpour le cas où ...
   Niter=i;
   x(i) = x(i-1) - f(x(i-1)) / (f(x(i-1))-f(x(i-2))) * (x(i-1)-x(i-2));
         ecart=abs(( x(i)-x(i-1) )/x(i));
   if (ecart < 1e-10)
       break;  % arrêt si l'écart relatif < 1e-10
   end
   end

RACINE=x(Niter)
Nombre_d_iterations=Niter-2  % -2 , car les 2 précurseurs étaient comptés.
Erreur=ecart

K1 = (x(Niter)-x(Niter-1)) / (x(Niter-1)-x(Niter-2))
Nombre_de_chiffres_signif_par_iteration = -log10(abs(K1))






% PAQUOT YVAN

%QUESTION 2
clear
clc

% Calculez une solution approchée du problème aux limites suivant
%
% y''(x) - 4.y'(x) + 4.y(x) = cos(x)
% y(0) = 0,  y(pi) = 0 %
% Donnez votre appréciation sur la qualité de la solution obtenue.
%==========================================================================

a=0;
b=pi;
y0=0;
yf=0;

N=1000;  % Nombre de pas
h=(b-a)/N;  % pas


%Construisons la matrice tridiagonale A
x=linspace(h,(N-1)*h,N-1);  % vecteur des Xn

Dprinc =(h*h*4-2) .* diag(ones(N-1,1));  % Diagonale principale de A
Dsup=(1+(h/2)*(-4)).*diag(ones(N-2,1),1);% Diagonale supérieure de A
Dinf=((1-(h/2)*(-4))).*diag(ones(N-2,1),-1); % Diagonale inférieure de A
A=Dprinc+Dsup+Dinf;
B=h2 * cos(x);  % vecteur B
B(1)=B(1)-y0*(1-(h/2)*(-4));
```

```
B(N-1)=B(N-1)-yf*(1+(h/2)*(-4));

% Résolvons le système A*y = B

y=A\B'

Y=[y0; y; yf];  % Ajout des limites
X=[a x b]';

plot(X,Y);
grid on
axis equal
```

```
% PAQUOT YVAN

% QUESTION 3

% Calculez deux valeurs propres de la matrice ci-dessous, ainsi que les
% vecteurs propres associés, par la méthode des puissances
%
%  A=[ 9 -2  1 -3  1 -2
%     -2 16 -1 -3  2  1
%      1 -1 -8 -1  0  2
%     -3 -3 -1  2 -6  3
%      1  2  0 -6  1  0
%     -2  1  2  3  0  4]
%
% Expliquez la démarche que vous utilisez.
%===========================================================================

clear
clc

A=[ 9 -2  1 -3  1 -2
   -2 16 -1 -3  2  1
    1 -1 -8 -1  0  2
   -3 -3 -1  2 -6  3
    1  2  0 -6  1  0
   -2  1  2  3  0  4]

N=size(A,1);  % A est supposée carrée
Niter=100;

%RECHERCHE DE LA + GRANDE VALEUR PROPRE
```

```
%Posons le vecteur propre initial zmax:
zmax(:,1)=ones(N,1);

for i=1:Niter
   zmax(:,i+1)=A*zmax(:,i); % Itération du vecteur
   [VPmax indice]=max(abs(zmax(:,i+1))); % Localisation de la + grande valeur
   if VPmax ~= zmax(indice,i+1)  % Si la valeur propre est négative
       VPmax=-VPmax;
   end
      zmax(:,i+1)=zmax(:,i+1)./VPmax;
   end

%RECHERCHE DE LA + GRANDE VALEUR PROPRE
% Modifions A de manière à trouver non plus la + gde val. p. (en valeur
% absolue), mais la plus petite.
Abis=A - VPmax .* diag(ones(N,1));

%Posons le vecteur propre initial zmin:
zmin(:,1)=ones(N,1);



for i=1:Niter
   zmin(:,i+1)=Abis * zmin(:,i); % Itération du vecteur
   [VPmin indice]=max(abs(zmin(:,i+1)));  % Localisation de la + grande valeur
   if VPmin ~= zmin(indice,i+1)  % Si la valeur propre est négative
       VPmin=-VPmin;
   end
      zmin(:,i+1)=zmin(:,i+1)./VPmin;

end

VPmin = VPmin + VPmax;

Valeurs_propres=[VPmax VPmin]

Vecteurs_propres_correspondant=[zmax(:,Niter+1) zmin(:,Niter+1) ]
```