Détection et localisation d'un canon

PROGRAMME (OCTAVE)

```
%alpha = a
%beta = b
a = 0.56;
b = 1.18;
D1 = 0.179;                %Dab
D2 = 0.930;                %Dad

global X;
global deltX;
global deltY;
global a2;
global b2;
global C;
global Xnew;
global D1new;
global D2new;

%%%%%%%%%%%%%%%%%%entrée de la fonction%%%%%%%%%%%%%%%%%%

function f=cool(xy)
a = 0.56;
b = 1.18;
D1 = 0.179;                %Dab
D2 = 0.930;                %Dad
f = [(4*D1^2-4)*xy(1)^2+4*D1^2*xy(2)^2-(4*D1^2-4)*xy(1)-D1^4+2*D1^2-1;
(4*a^2-4*D2^2)*xy(1)^2+(4*b^2-4*D2^2)*xy(2)^2+(4*a*D2^2-4*a^3-
4*a*b^2)*xy(1)+(4*b*D2^2-4*b^3-
4*b*a^2)*xy(2)+8*a*b*xy(1)*xy(2)+a^4+b^4+2*a^2*b^2-2*a^2*D2^2-
2*b^2*D2^2+D2^4];

endfunction

  xy0=[0 1.5];
  [X,valeur_en_X]=fsolve('cool',xy0)          %%%x et y seront en unité
de notre repère%%%



%%%%%%%%itération pour trouver si le système a d'autres solution%%%%%%%%

  for x0=-1:2
      for y0=-2:3
            xy0=[x0 y0];
            [X,valeur_en_X]=fsolve('cool',xy0);
            if (abs(valeur_en_X) < 1e-10)
            X,
            else
            0,
            endif
      endfor
   endfor
```

```
%%%%%%%%%%%%%%%incertitudes%%%%%%%%%%%%%%%

    function g=gizz(uv)
    global X;
    global deltX;                               %delta x = u
    global deltY;                               %delta y = v

    a = 0.56;
%%%%%le programme doit nous renvoyer u et v%%%%%

    b = 1.18;
    D1 = 0.179;         %Dab
    D2 = 0.930;         %Dad
    w = 0.02;             %a donner: variation de D1
    z = 0.02;             %a donner: variation de D2

 g = [((-8*D1^2+8).*X(1)+(4*D1^2-4))*uv(1)-(8*D1^2.*X(2))*uv(2)+(-
8.*X(1)^2*D1-8.*X(2)^2*D1+8*D1.*X(1)+4*D1^3-4*D1)*w;((8*a^2-
8*D2^2).*X(1)+(4*a*D2^2-4*a^3-4*a*b^2)+8*a*b.*X(2))*uv(1)+((8*b^2-
8*D2^2).*X(2)+(4*b*D2^2-4*b^3-4*b*a^2)+8*a*b.*X(1))*uv(2)+(-8*D2.*X(1)^2-
8*D2.*X(2)^2+8*a.*X(1)*D2+8*b*D2.*X(2)-(4*a^2+4*b^2)*D2+4*D2^3)*z];
    endfunction

      uv0=[0.5 1];
      [deltaX,valeur_en_deltaX]=fsolve('gizz',uv0);

      deltX = abs(deltaX(1));
      deltY = abs(deltaX(2));

      delX = (deltX*1.84)*1000;
      delY = (deltY*1.84)*1000;

      disp('deltaX =');
      disp(delX),              %affiche delta x en mètres
      disp('deltaY =');        %affiche delta y en mètres
      disp(delY),


%%%%%%%%%%%%%%%%partie graphique%%%%%%%%%%%%%%%


    a =0.56;
    b =1.18;
    D1 = 0.179;              %Dab
    D2 = 0.930;              %Dad

clf
    x = [-3:0.001:3];

%%%%%Suppression de la partie complexe de y0 et -y0%%%%%

%    [l,c] = size(x);
%     j=0;
%     for i=1:1:c
```

```
%         tmp = (-2*D1^2+D1^4+1+(4*D1^2-4).*x+(-4*D1^2+4).*x.^2)/(4*D1^2);
%
%         if ( tmp(i) < 0.0 )
%             j = i;
%     break;
%         else
%     y0(i) = sqrt(tmp(i));
%         endif
%      endfor

%    y0_new = y0(1,1:j-1);
%    x_new  = x(1,1:j-1);
%     plot(x_new,y0_new, '*')
%     plot(x_new,-y0_new,'*')
%     plot(x_new+deltX,y0_new+deltY,'k')        %x+deltX
%     plot(x_new-deltX,y0_new-deltY,'k')
%     plot(x_new+deltX,-y0_new+deltY,'k')        %x+deltX
%     plot(x_new-deltX,-y0_new-deltY,'k')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

figure(2)

hold on

y0=sqrt((-2*D1^2+D1^4+1+(4*D1^2-4).*x+(-4*D1^2+4).*x.^2)/(4*D1^2));
  plot(x,y0, 'b')
  plot(x,-y0,'b')
  plot(x+deltX,y0+deltY,'k')        %x+deltX
  plot(x-deltX,y0-deltY,'k')
  plot(x+deltX,-y0+deltY,'k')        %x+deltX
  plot(x-deltX,-y0-deltY,'k')

y2=(0.5*(1/(b^2-D2^2))).*(-b*D2^2+b^3+b*a^2-2*a*b.*x+sqrt(2*b^2*D2^2*a^2-
4*D2^2.*x*a^3+4*D2^4.*x*a+4*D2^2.*x.^2*a^2+4*b^2.*x.^2*D2^2-
4*b^2*D2^2*a.*x-2*D2^4*a^2-4*D2^4.*x.^2+D2^2*a^4+b^4*D2^2+D2^6-
2*b^2*D2^4));
  plot(x,y2,'g')
  plot(x+deltX,y2+deltY,'k')        %x+deltX
  plot(x-deltX,y2-deltY,'k')

y3=(0.5*(1/(b^2-D2^2))).*(-b*D2^2+b^3+b*a^2-2*a*b.*x-sqrt(2*b^2*D2^2*a^2-
4*D2^2.*x*a^3+4*D2^4.*x*a+4*D2^2.*x.^2*a^2+4*b^2.*x.^2*D2^2-
4*b^2*D2^2*a.*x-2*D2^4*a^2-4*D2^4.*x.^2+D2^2*a^4+b^4*D2^2+D2^6-
2*b^2*D2^4));
  plot(x,y3,'r')
  plot(x+deltX,y3+deltY,'k')        %x+deltX
  plot(x-deltX,y3-deltY,'k')

title('localisation du canon et incertitudes')
xlabel('coordonnee x du canon');
ylabel('coordonnee y du canon');
axis([-1 1.5 -1 2]);


plot(0,0,'*')
plot(1,0,'*')
plot(a,b,'*')
```

```
text(0.03,-0.04,'A')
text(1.03,-0.04,'B')
text(a+0.03,b-0.04,'D')

grid on
hold off

disp(' ')
disp('Maintenant nous allons étudier les incertitudes que nous donnent
les points que nous choisissons :')
disp(' ')

C = input('nouvelle distance A-B (en centimètre) : ');
a2 = (input('valeur du nouveau alpha (en centimètre) : '))/C;
%coordonnées du point D dans le nouveau repère en unité
b2 = (input('valeur du nouveau beta (en centimètre) : '))/C;
D1new = (input('CB-CA (en centimètre) : '))/C;
%CB-CA en unité
D2new = (input('CD-CA (en centimètre) : '))/C;
%CD-CA en unité

%%%%%%%%%%%%%%%%incertitudes liées aux points choisis%%%%%%%%%%%%%%%%

  function k=coool(xy)
  global D1new;
  global D2new;
  global a2;
  global b2;

k = [(4*D1new^2-4)*xy(1)^2+4*D1new^2*xy(2)^2-(4*D1new^2-4)*xy(1)-
D1new^4+2*D1new^2-1; (4*a2^2-4*D2new^2)*xy(1)^2+(4*b2^2-
4*D2new^2)*xy(2)^2+(4*a2*D2new^2-4*a2^3-4*a2*b2^2)*xy(1)+(4*b2*D2new^2-
4*b2^3-4*b2*a2^2)*xy(2)+8*a2*b2*xy(1)*xy(2)+a2^4+b2^4+2*a2^2*b2^2-
2*a2^2*D2new^2-2*b2^2*D2new^2+D2new^4];

endfunction

  xy0=[0.5 1.5];
  [Xnew,valeur_en_X]=fsolve('coool',xy0)              %%%le programme

nous redonne les coordonnées x et y connues%%%

%%%%%%%%%%itération pour trouver si le système a d'autres
solution%%%%%%%%%%

%    for x0=-1:2
%     for y0=-2:3
%           xy0=[x0 y0];
%           [Xnew,valeur_en_X]=fsolve('coool',xy0);
%           if (abs(valeur_en_X) < 1e-10)
%           Xnew,
%           else
%           0,
%           endif
%     endfor
%     endfor

    function h=projet(st)
```

```
    global X;
    global a2;
    global b2;
    global Xnew;
    global D1new;
    global D2new;

%%%%le programme doit nous renvoyer s et t%%%%
    w = 0.02;             %calculé: variation de D1
    z = 0.02;             %calculé: variation de D2

 h = [((-8*D1new^2+8).*Xnew(1)+(4*D1new^2-4))*st(1)-
(8*D1new^2.*Xnew(2))*st(2)+(-8.*Xnew(1)^2*D1new-
8.*Xnew(2)^2*D1new+8*D1new.*Xnew(1)+4*D1new^3-4*D1new)*w;((8*a2^2-
8*D2new^2).*Xnew(1)+(4*a2*D2new^2-4*a2^3-
4*a2*b2^2)+8*a2*b2.*Xnew(2))*st(1)+((8*b2^2-
8*D2new^2).*Xnew(2)+(4*b2*D2new^2-4*b2^3-
4*b2*a2^2)+8*a2*b2.*Xnew(1))*st(2)+(-8*D2new.*Xnew(1)^2-
8*D2new.*Xnew(2)^2+8*a2.*Xnew(1)*D2new+8*b2*D2new.*Xnew(2)-
(4*a2^2+4*b2^2)*D2new+4*D2new^3)*z];
    endfunction

      st0=[0.5 1];
      [delX,valeur_en_delX]=fsolve('projet',st0);

      deX = abs(delX(1));
      deY = abs(delX(2));

      dX = (deX*(C/8))*1000;        %%donne le nouveau delta x en mètre%%
      dY = (deY*(C/8))*1000;        %%donne le nouveau delta y en mètre%%

      disp('new_deltaX =');
      disp(dX),                          %affiche delta x
      disp('new_deltaY =');              %affiche delta y
      disp(dY),

figure(3)

hold on

y0=sqrt((-2*D1new^2+D1new^4+1+(4*D1new^2-4).*x+(-
4*D1new^2+4).*x.^2)/(4*D1new^2));
  plot(x,y0, 'b')
  plot(x,-y0,'b')
  plot(x+deX,y0+deY,'k')       %x+deltX
  plot(x-deX,y0-deY,'k')
  plot(x+deX,-y0+deY,'k')       %x+deltX
  plot(x-deX,-y0-deY,'k')

y2=(0.5*(1/(b2^2-D2new^2))).*(-b2*D2new^2+b2^3+b2*a2^2-
2*a2*b2.*x+sqrt(2*b2^2*D2new^2*a2^2-
4*D2new^2.*x*a2^3+4*D2new^4.*x*a2+4*D2new^2.*x.^2*a2^2+4*b2^2.*x.^2*D2new
^2-4*b2^2*D2new^2*a2.*x-2*D2new^4*a2^2-
4*D2new^4.*x.^2+D2new^2*a2^4+b2^4*D2new^2+D2new^6-2*b2^2*D2new^4));
  plot(x,y2,'g')
  plot(x+deX,y2+deY,'k')       %x+deltX
  plot(x-deX,y2-deY,'k')
```

```
y3=(0.5*(1/(b2^2-D2new^2))).*(-b2*D2new^2+b2^3+b2*a2^2-2*a2*b2.*x-
sqrt(2*b2^2*D2new^2*a2^2-
4*D2new^2.*x*a2^3+4*D2new^4.*x*a2+4*D2new^2.*x.^2*a2^2+4*b2^2.*x.^2*D2new
^2-4*b2^2*D2new^2*a2.*x-2*D2new^4*a2^2-
4*D2new^4.*x.^2+D2new^2*a2^4+b2^4*D2new^2+D2new^6-2*b2^2*D2new^4));
  plot(x,y3,'r')
  plot(x+deX,y3+deY,'k')       %x+deltX
  plot(x-deX,y3-deY,'k')

title('localisation du canon et incertitudes')
xlabel('coordonnee x du canon');
ylabel('coordonnee y du canon');
axis([-3 3 -1 4.5]);

grid on

plot(0,0,'*')
plot(1,0,'*')
plot(a2,b2,'*')
text(0.03,-0.04,'A')
text(1.03,-0.04,'B')
text(a2+0.03,b2-0.04,'D')
```

END