TRLIN Moreno **M2 IICAO**
KORAA Galel **04/11/2010**

TP 1 : Enveloppe convexe. Ordre polaire

1) Code

```
// Cette fonction renvoie : - 1 si (angle(Ox,OM)<angle(Ox,OP) ou
//angle(Ox,OM) = angle(Ox,OP) et norme(OM) <= norme(OP))
//                          - 0 sinon.
function [X]=ordre_pol(O,M,P)
// Construction de la matrice qui nous permet de déterminer l'ordre
// polaire
      mat=zeros(3,3);
      mat(1,1)=1
      mat(1,2)=1
      mat(1,3)=1
      mat(2,1)=O(1);
      mat(2,2)=M(1);
      mat(2,3)=P(1);
      mat(3,1)=O(2);
      mat(3,2)=M(2);
      mat(3,3)=P(2);
      determinant=det(mat);
      if(determinant>0) then; X=1;end;
      if(determinant<0) then; X=0;end;
      if(determinant==0) then; X=(norm(M-O)>norm(P-O));end;
endfunction
```

```
// Cette fonction implémente la méthode de Jarvis.
// Paramètres : Entrée : X = ensemble de points
//              Sortie : contour = ensemble des points contours
function [contour]=Jarvis(X)
      n=size(X,2);
      // On récupère l'indice du point qui a la plus petite ordonnée
      ind_pt_Ymin = indiceYmin(X);
      // On crée notre vecteur de booléen qui permet de tester si un
      // point est contour à partir de son indice
      est_contour=zeros(1,n)
      ind_pt=ind_pt_Ymin
      // Création du vecteur contour qui contient le 1er point de plus
      // petite ordonnée
      contour=[X(:,ind_pt_Ymin)]

      // On s'arrête lorsqu'on retombe sur le 1er point
      while(~est_contour(ind_pt_Ymin))
               // On cherche le point M
               for i=1:n
                 if ~est_contour(i) & i<>ind_pt & i<>ind_pt_Ymin then
                   ind_M=i
                   break
                 end
               end
```

```
                // On cherche le point ayant le plus grand ordre polaire
                // avec X(:,int_pt)
            for i=1:n
                if ~est_contour(i)  & i<>ind_pt & i<>ind_M &
(ordre_pol(X(:,ind_pt),X(:,ind_M),X(:,i))) then
                  ind_M=i
                end
              end

        //à la sortie de la boucle, X(:,ind_M) fait partie du contour
              est_contour(ind_M)=1

            //l'algorithme reprend à partir de ce point
            ind_pt=ind_M

            plot(X(1,ind_M),X(2,ind_M),'rv')
          contour=[contour X(:,ind_M)]
 end
endfunction
```

```
// Cette fonction permet de saisir plusieurs points à la souris puis
// trace l'enveloppe convexe.
function trace_enveloppe_cvx()
  P=inputpoly()
  env=Jarvis(P)
  plot(env(1,:),env(2,:))
endfunction
```

```
// Cette fonction permet de renvoyer l'indice du point ayant la plus
// petite ordonnée.
// Paramètre : X = vecteur de points
function ind_pt_Ymin = indiceYmin(X)
     n=size(X,2);
      ind_pt_Ymin=1;
      // on recupere lindice du point dordonnee minimum
      for i=2:n
            if(X(2,ind_pt_Ymin)>X(2,i)) then
                  ind_pt_Ymin=i;
            end
           end
endfunction
```

```
// Entrée d'un ensemble de points à la souris
// en sortie, le tableau X avec p points du plan (dimensions 2 x p)
function [X] = inputpoly()

// création d'une fenêtre pour la saisie des points
f=figure();  // une nouvelle fenetre
set(gca(),"auto_clear","off")
set(gca(),"data_bounds",[0,0;100,100])      // bornes des axes en x et y
```

```
set(gca(),"margins",[0.05,0.05,0.05,0.05])   // marges du repère dans la
                                             // fenêtre
set(gca(),"axes_visible",["on","on","on"])   // afficher les axes
set(gca(),"box","on")
set(gca(),"auto_scale","off")

// boucle de saisie des points
but = 3;
i = 0;
while but==3 | but==0 | but==10 | but==20
  xinfo("Point suivant : bouton gauche - Dernier point : bouton droite");
  i = i+1;
  [but,v0,v1] = xclick();
  X(1,i) = v0;
  X(2,i) = v1;
  plot(X(1,i),X(2,i),"go")
end;
endfunction
```

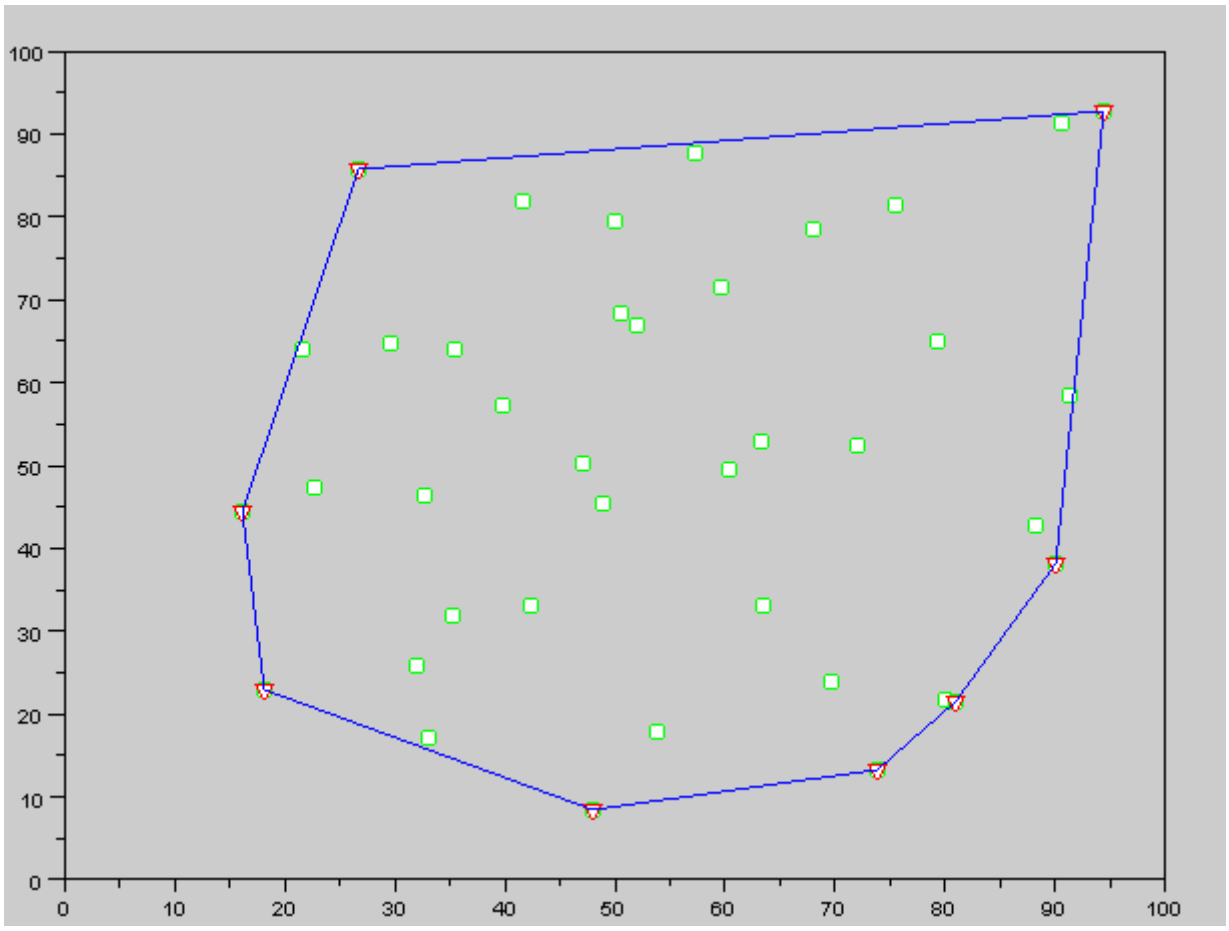
---

2) Exemples

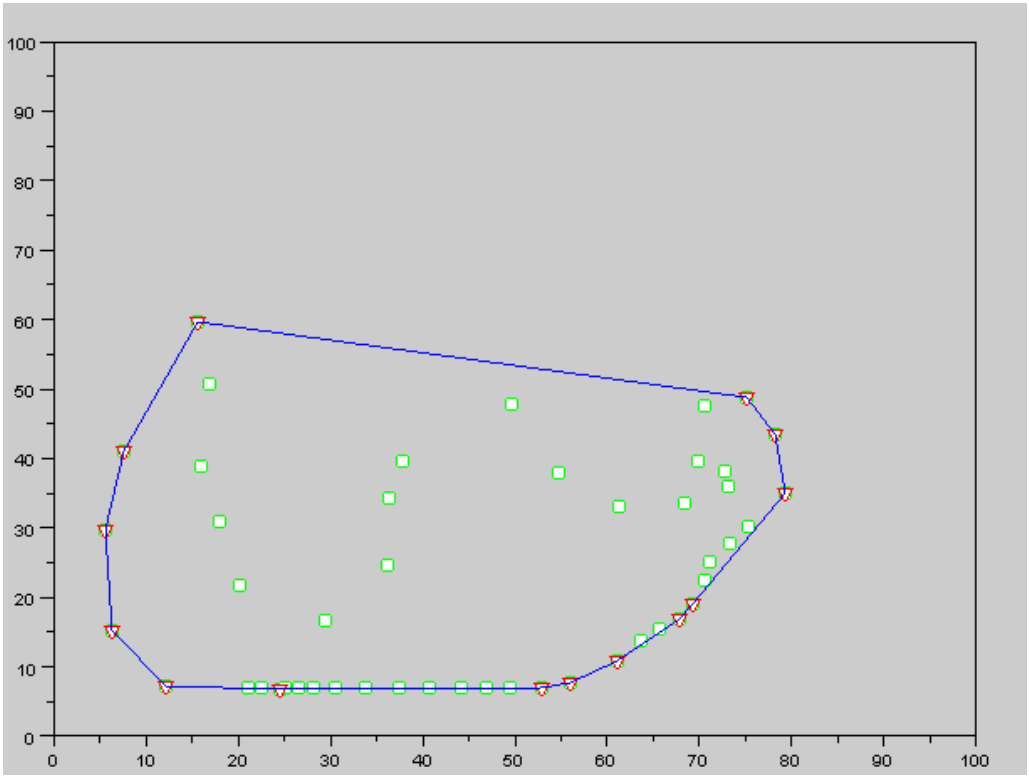
Pour exécuter le programme, il suffit de taper la commande suivante dans la console de Scilab :

```
trace_enveloppe_cvx()
```

- *Exemple 1 :*



- *Exemple 2 :*



- *Exemple 3 :*