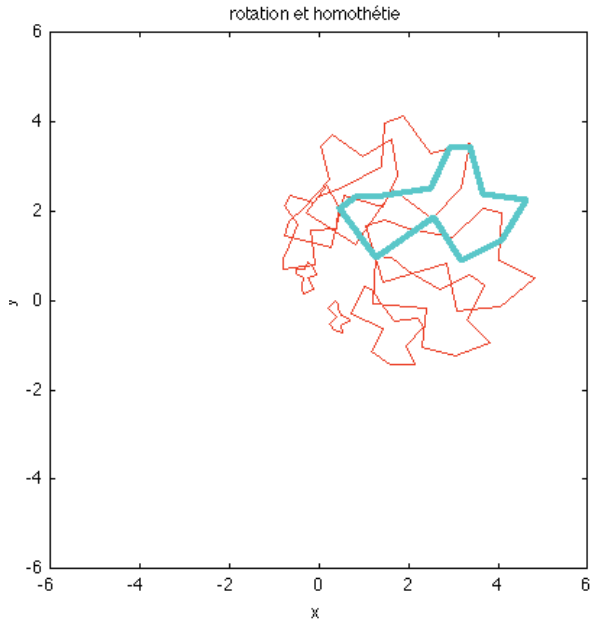
TP2: Pour les enseignants:

Pour ce TP, tout est axé sur les graphiques, et la manipulation des tableaux des coordonnées de points mesurés sur une image. Ce TP est assez long, et ce n'est pas très grave si les étudiants ne font pas la partie "Pour aller plus loin".

Ici une représentation de la rotation combinée à l'homothétie, figure originale en bleu.



Voici le script qui fait la plupart des questions du TP2: J'ai mis beaucoup de commentaires, donc chaque commande devrait être lisible dans l'ensemble. J'ai utilisé des fonctionnalités vectorielles plutôt que des boucles partout ou c'était possible.

```
% script TP2: manipulations d'une forme géométrique
clf

% lecture et affichage de l'image
a=imread('tp2.jpg');
image(a);

% les points
po=[99, 261]; % le point o
p=[130,121; 186 197; 273 134; 316 200; 378 170; ...
    416 107; 350 100; 330 26; 298 29; ...
    269 91; 196 102; 156 103];
```



```
% on soustrait les coordonnées de l'origine du repère et on inverse
% l'orientation de y
x=p(:,1)-po(1);
y=-p(:,2)+po(2);

% on rajoute le premier point à la fin du
% vecteur pour boucler la boucle
x=[x;x(1)];
y=[y;y(1)];

% on mesure la taille d'un pixel à l'aide de l'echelle
p1=[278, 79]; p2=[377 173];
% longueur de l'echelle en pixels
l=sqrt((p1(1)-p2(1))^2+(p1(2)-p2(2))^2);
d=2/l; % taille d'un pixel en mètres

% on met à l'echelle en mètres
x=x*d;
y=y*d;

% on trace le contour
plot(x,y,'*-')
xlim([-6,6]); ylim([-6,6]);
grid on

% mesure de la longueur du contour
% (à mettre dans une fonction contlong.m)
n=length(x);
l=0;
for ind=1:n-1
l=l+sqrt((x(ind+1)-x(ind))^2+(y(ind+1)-y(ind))^2);
end

% on mesure l'angle et le rayon de chaque point
ang=atan(y./x);
r=sqrt(x.^2+y.^2);

% animation de la rotation combinée à l'homothétie
hold off
n=100; % nombre d'itérations
alphavec=linspace(0,2*pi,n); % vecteur des angles de rotation
lvec=zeros(n,1); % pour mémoriser la longueur du contour
subplot(2,1,1)
for ind=1:length(alphavec)
```

```
    alpha=alphavec(ind);

  % la rotation
  xx=r.*cos(ang+alpha);
  yy=r.*sin(ang+alpha);

  % l'homothétie
  k=cos(alpha);
  xx=k*xx; yy=k*yy;

  % on mesure la longueur
  lvec(ind)=contlong(xx,yy);

  % on trace le contour original et le contour transformé
  plot(x,y,'b',xx,yy,'r');
  axis equal
  xlim([-6,6]); ylim([-6,6]);
  drawnow
end
% on trace la dépendance de la longueur avec alpha
subplot(2,1,2)
plot(alphavec,lvec)

% On mesure la taille du plus petit
% rectangle qui contient la figure
lx=max(x)-min(x);
ly=max(y)-min(y);
Srec=lx*ly

% mesure de la surface avec les pixels
s=sum(sum(a(:,:,1)~=255|a(:,:,2)~=255|a(:,:,3)~=255));
Spix=s*d^2
```

Et voici la fonction qui mesure la longueur du contour:

```
function l=contlong(x,y);
% mesure de la longueur du contour
% on somme les longueurs des segments successifs
n=length(x);
l=0;
for ind=1:n-1
l=l+sqrt((x(ind+1)-x(ind))^2+(y(ind+1)-y(ind))^2);
end
```


Pour la rotation, on commence par calculer le rayon et l'angle de chaque point par rapport à l'axe des x: `ang=atan(y./x); r=sqrt(x.^2+y.^2);`. On n'a pas besoin de boucle, en utilisant les opérations élément par élément `.^ ./`. Pour l'homothétie, il suffit de multiplier `x` et `y` par le rapport `k`.

Dans "pour aller plus loin": Effectivement, la longueur du contour est invariante par rotation, le graph est une droite de pente nulle.

La longueur dépend linéairement du rapport d'homothétie, la pente de cette droite, c'est la longueur du contour original, puisque $l(k = 0) = 0$ et $l(k = 1)$ est égale à la longueur originale du contour.

Pour une extension s selon x: pour s grand, le contour tend vers un segment horizontal, donc la pente de $l = l(s), s \to \infty$ tend vers deux fois la *largeur* originale du contour.

Pour la surface, on peut obtenir les dimensions du plus petit rectangle (horizontal) avec les fonctions `min` et `max`. Pour obtenir la surface de la forme géométrique, on compte les pixels dont le RGB est (255, 255, 255). L'image est stockée sous la forme d'un tableau à trois dimensions, les lignes et les colonnes correspondent aux pixels, et la troisième dimension a trois éléments, correspondant au RGB. Dans le script, je fais cela en une seule ligne avec la fonction `sum` et un test logique, mais les étudiants peuvent aussi le faire avec des boucles.