

MEC6212: GENERATION de MAILLAGES

Exercice 2: Interpolation de Lagrange

10 janvier 2025

Énoncé

À partir d'un nombre de points distincts représentant la discrétisation d'une courbe quelconque, construire un interpolant de Lagrange qui approxime cette courbe :

1. Soit un échantillonnage (x_i, y_i) de cette courbe en un nombre nbCNTRL de points ;
2. Écrire une fonction Matlab, `geoLAGRANGE.m` pour approximer ces points, utilisant l'interpolation de Lagrange ;
3. Le protocole d'appel :

$[P, PTS] = \text{geoLAGRANGE}(\text{CNTRL}, \text{nbCNTRL})$

où,

CNTRL : tableau qui contient les coordonnées (x,y) des nbCNTRL points de collocation, structurés de la façon suivante :

$x = \text{CNTRL}(1 : \text{nbCNTRL}, 1)$ et $y = \text{CNTRL}(1 : \text{nbCNTRL}, 2)$;

nbCNTRL : nombre de points de collocation

P : tableau qui contient les coordonnées (x,y) des PTS points interpolés :

$x = P(1 : \text{PTS}, 1)$, et $y = P(1 : \text{PTS}, 2)$

PTS : nombre de points interpolés

On note que les tableaux CNTRL et P sont distincts.

4. Cette fonction sera appelée d'un programme montré à la Figure 1.

```

switch SEG(iSEG,1)%typeSEG
case 10%----- Bezier
    [P,ptsSEG] = geoBEZIER(PLN(PLN1:PLN2,1:2),SEG(iSEG,3));
    SEG(iSEG,4) = iPT+1;
    SEG(iSEG,5) = ptsSEG;%----- nombre de PT sur le segment
    for j=1:ptsSEG-1
        iPT = iPT + 1;
        GEO(iPT,1) = P(j,1);
        GEO(iPT,2) = P(j,2);
    end
case 20%----- spline
    [P,ptsSEG] = geoSPLINE(PLN(PLN1:PLN2,1:2),SEG(iSEG,3));
    SEG(iSEG,4) = iPT+1;
    SEG(iSEG,5) = ptsSEG;%----- nombre de PT sur le segment
    for j=1:ptsSEG-1
        iPT = iPT + 1;
        GEO(iPT,1) = P(j,1);
        GEO(iPT,2) = P(j,2);
    end
case 30%----- polynome de Lagrange
    nbCNTSEG = finCNTSEG -iniCNTSEG +1;
    [P,ptsSEG] = geoLAGRANGE(PLN(iniCNTSEG:finCNTSEG,1:2),SEG(iSEG,3));
    for j=1:ptsSEG-1
        GEO(iPT,1) = P(j,1);
        GEO(iPT,2) = P(j,2);
        iPT = iPT + 1;
    end
otherwise
end

```

FIGURE 1 – Programme appelant pour différents types d'interpolants

1 Interpolation de Lagrange

Soit un ensemble de $n + 1$ points,

$$(x_0, y_0), (x_1, y_1), \dots, (x_n, y_n).$$

On cherche un polynôme $p_n(x)$, de degré n interpolant ces $n + 1$ points (x_i, y_i) , de la forme,

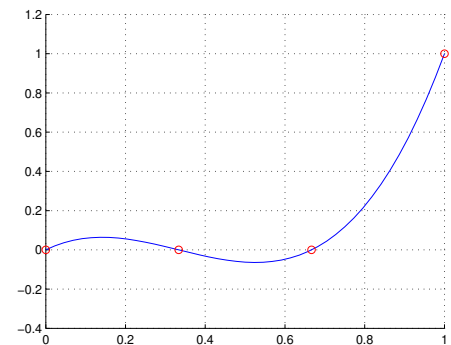
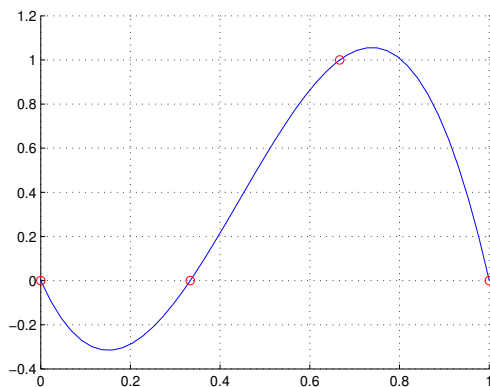
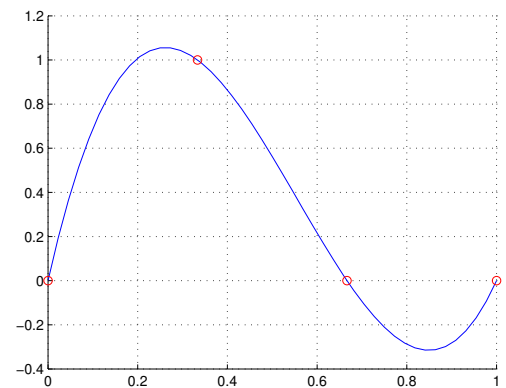
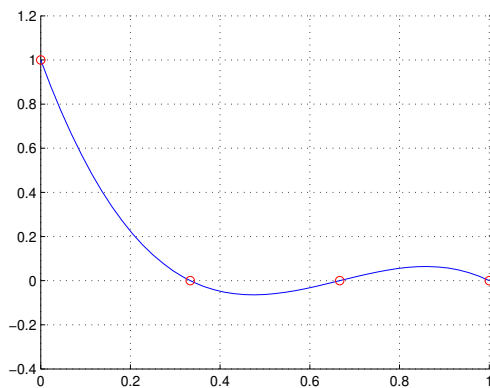
$$p(x) = y_0 L_0(x) + y_1 L_1(x) + \dots + y_n L_n(x).$$

Cette approximation s'interprète comme une combinaison linéaire les polynômes de Lagrange L_k , pondérés par les valeurs des données y_k .

Le k -ème polynôme de Lagrange L_k est l'**unique** polynôme qui vérifie,

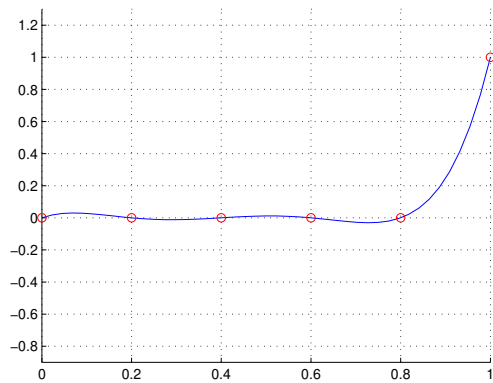
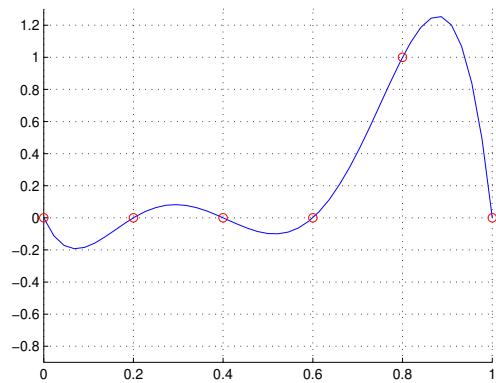
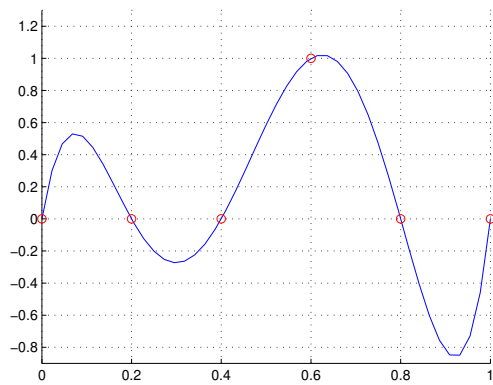
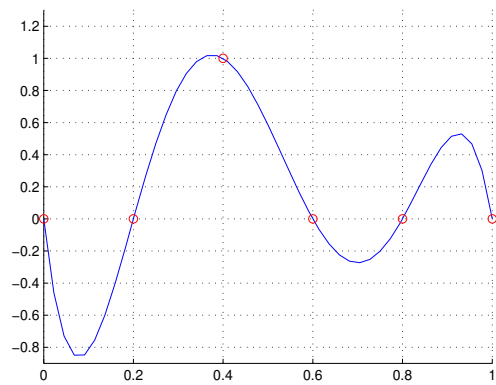
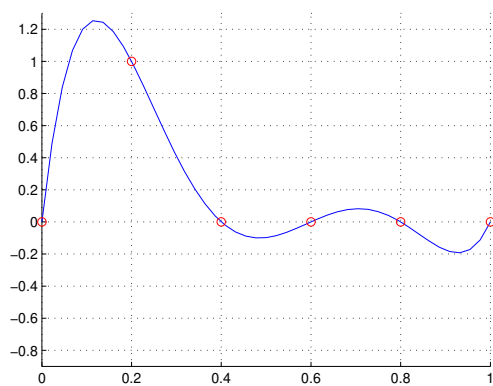
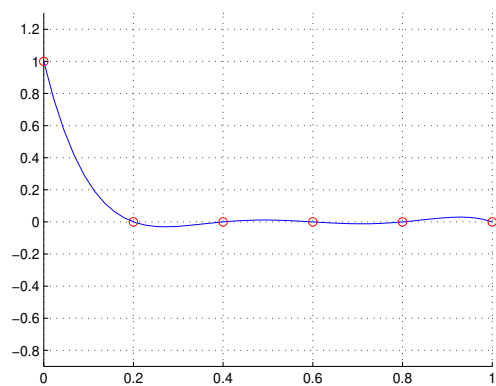
x_k	x_0	x_1	$\dots$	x_k	$\dots$	x_n
$L_k(x)$	0	0	$\dots$	1	$\dots$	0

c-à-d, le polynôme L_k s'annule à tous les points de collocation, sauf au point x_k , où il vaut 1.
Par exemple, pour les 4 polynômes de degré 3, L_k $k = 1\dots 4$,



$$\begin{aligned}
 & f(x_0) \frac{(x-x_1)(x-x_2)(x-x_3)}{(x_0-x_1)(x_0-x_2)(x_0-x_3)} & + & & f(x_1) \frac{(x-x_0)(x-x_2)(x-x_3)}{(x_1-x_0)(x_1-x_2)(x_1-x_3)} \\
 & + f(x_2) \frac{(x-x_0)(x-x_1)(x-x_3)}{(x_2-x_0)(x_2-x_1)(x_2-x_3)} & + & & f(x_3) \frac{(x-x_0)(x-x_1)(x-x_2)}{(x_3-x_0)(x_3-x_1)(x_3-x_2)}
 \end{aligned}$$

ou bien, les 6 polynômes de degré 5, L_k $k = 1 \dots 6$,



On construit les $n + 1$ polynômes de Lagrange

$$L_0(x), L_1(x), \dots, L_n(x).$$

à partir de l'écriture

$$\frac{(x - x_0)(x - x_1)(x - x_2) \dots (x - x_k) \dots (x - x_{n-1})(x - x_n)}{(x_k - x_0)(x_k - x_1)(x_k - x_2) \dots (x_k - x_k) \dots (x_k - x_{n-1})(x_k - x_n)}$$

On note que le terme

$$\frac{(x - x_k)}{(x_k - x_k)}$$

est indéterminé. En le retirant du produit, $i \neq k$, on obtient,

$$L_k(x) = \prod_{i \neq k} \frac{(x - x_i)}{(x_k - x_i)}$$

Ce qui donne,

$$L_k(x_k) = 1 \text{ et pour } i \neq k, L_k(x_i) = 0$$

Finalement, le polynôme d'interpolation devient,

$$p(x) = \sum_{k=0}^n y_k * L_k(x)$$

2 Paramétrisation

Dans le développement précédent, l'interpolant construit est basé sur une représentation explicite, $y = f(x)$, c-à-d que le paramètre est x .

Dans ce travail, ce qui recherché, en fait, est une représentation paramétrique plutôt qu'explicite. Le paramètre u est arbitraire, et $0 \leq u \leq nbCTRL - 1$ donne une paramétrisation en fonction du numéro k du point, qui est proche de la paramétrisation intrinsèque basée sur la longueur de la courbe.

La fonction geoLAGRANGE doit construire une approximation des (x_i, y_i) distribués en fonction d'un paramètre $0 \leq u \leq uFin$. Alors $P(x(u), y(u))$ s'exprime,

$$\begin{aligned} x(u) &= \sum_{k=0}^n x_k * L_k(u) \\ y(u) &= \sum_{k=0}^n y_k * L_k(u) \end{aligned}$$

où les polynômes $L_k(u)$ sont semblables aux $L_k(x)$, mais où le paramètre x est remplacé par u .

```

function [P,PTS] = geoLAGRANGE(CNTRL,nbCNTRL)
%
%   Interpolant de Lagrange de degre nbCNTRL-1, echantillonne sur
%   U(1)= 0 <= u <= nbCNTRL-1=U(nbCNTRL) avec les nbCNTRL points de
%   collocation CNTRL ==>(x,y), et dU uniforme.
%
PTS=20;%----- nombre de points echantillonnees
nbINTRVL=nbCNTRL-1;%----- parametrisation de la courbe
dU= double(nbINTRVL)/(PTS-1);
U=(0:double(nbINTRVL));
P(1,1:2)=CNTRL(1,1:2);%----- points en extremités
P(1,3)=0;%----- parametre u debut
P(PTS,1:2)=CNTRL(nbCNTRL,1:2);
P(PTS,3)=U(nbCNTRL);%----- parametre u fin
u=0;
%----- subdivise en un nombre specifie de points avec repartition uniforme
for iPTS=uint16(2):PTS-1
    u=u+dU;
    P(iPTS,1:3)= [0 0 u];%----- (x,y) du point correspondant au parametre u
    for i=1:nbCNTRL
        L=1.;
        for j=1:nbCNTRL
            if (j~=i)
                L= L*(u-U(j))/(U(i)-U(j));
            end
        end
        P(iPTS,1:2) = P(iPTS,1:2) + L*CNTRL(i,1:2);
    end
end
end

```

FIGURE 2 – Construction de l’interpolant sous forme paramétrique uniforme

La Fig. 2, montre un exemple de fonction qui calcule une approximation de Lagrange qui prend en arguments,

CNTRL : tableau qui contient les coordonnées (x,y) des nbCNTRL points de collocation, structuré :

$x = \text{CNTRL}(1:\text{nbCNTRL},1)$ et $y = \text{CNTRL}(1:\text{nbCNTRL},2)$;

nbCNTRL : nombre de points de collocation

et retourne, les *PTS* points de l’interpolant, sous forme du tableau *P*,

P : tableau qui contient les coordonnées (x,y) des PTS points interpolés, structuré :

$x=P(1:PTS,1)$, $y=P(1:PTS,2)$ et $u=P(1:PTS,3)$

PTS : nombre de points interpolés

3 Paramétrisation quelconque

Dans l'exemple de la Section précédente, la fonction,

$$[P,PTS] = \text{geoLAGRANGE}(\text{CNTRL},\text{nbCNTRL})$$

retourne $PTS = 20$ points uniformément répartis sur l'intervalle.

L'objectif de cet exercice est de construire une approximation polygonale $P(1 : \text{nbPT}, 1 : 3)$ de nbPT points pour la courbe $(x(t), y(t), t)$ où la paramétrisation t est fonction de la courbure, tel qu'illustré à la Fig. 3.

On utilisera comme critère, la distance entre le point milieu d'un segment $PT_1 - PT_2$ du polygone et le point milieu du segment curviligne approximé (en pointillé). C'est-à-dire, pour chaque segment du polygone, la distance entre les points A et B sera inférieure à une valeur cible donnée. Ceci est équivalent à la courbure.

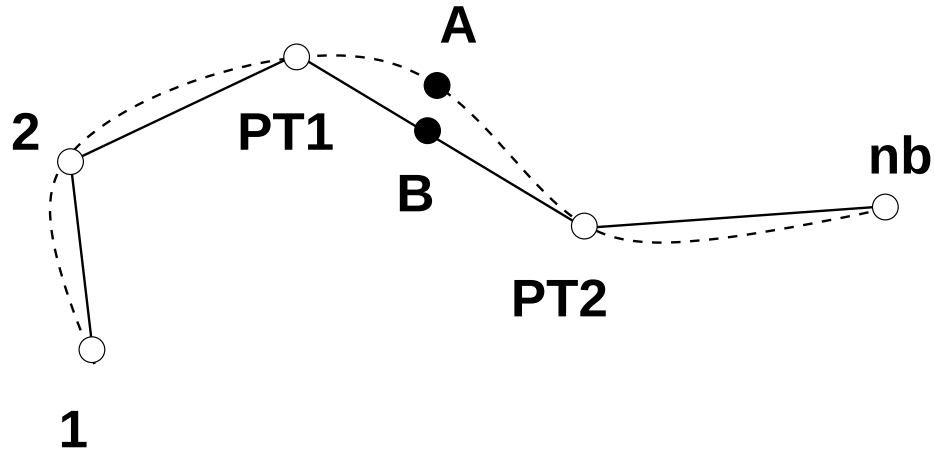


FIGURE 3 – Critère de la paramétrisation

$$L_{AB} = \sqrt{(x_A - x_B)^2 + (y_A - y_B)^2}$$

où la position P_A est obtenue par :

$$\begin{aligned} x_A &= f(t_A) \\ y_A &= g(t_A) \end{aligned}$$

et la valeur du paramètre $t_A = .5 * (t_1 + t_2)$. Les coordonnées du point P_B à mi-chemin du segment sont obtenues par,

$$\begin{aligned} x_B &= \frac{1}{2}(x_{pt1} + x_{pt2}) \\ y_B &= \frac{1}{2}(y_{pt1} + y_{pt2}) \end{aligned}$$

Le point de départ de cet algorithme de subdivision binaire est le segment 1 – 2 reliant les extrémités de la courbe comme illustré à la Fig. 3.

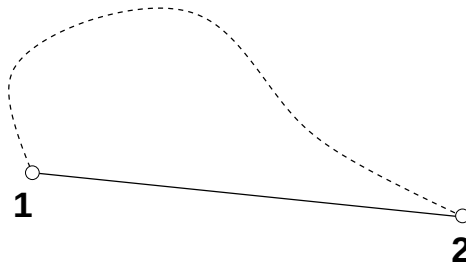
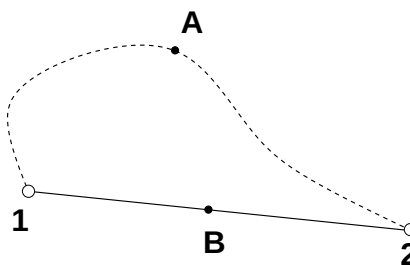
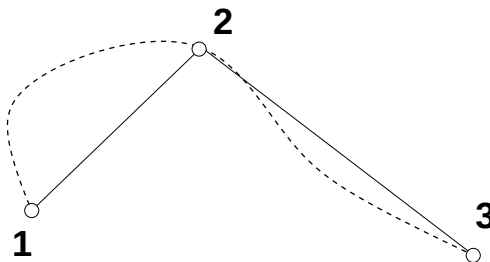


FIGURE 4 – Intervalle de départ

Visuellement, on constate que c'est une approximation grossière. Alors, on subdivise l'intervalle en introduisant le point B dans le polygone, et le point A correspondant sur l'approximation de la courbe, Fig. 5(a).



(a) Test 1er Seg



(b) 1ère subdivision

FIGURE 5 – Premi'ere itération sur l'intervalle de départ

Le résultat, 1 – 2 – 3, est une meilleure approximation montrée à la Fig. 5(b)

On répète ce procédé sur le segment 1 – 2 montré à la Fig. 6(a), donnant 1 – 2 – 3 – 4 à la Fig. 6(a).

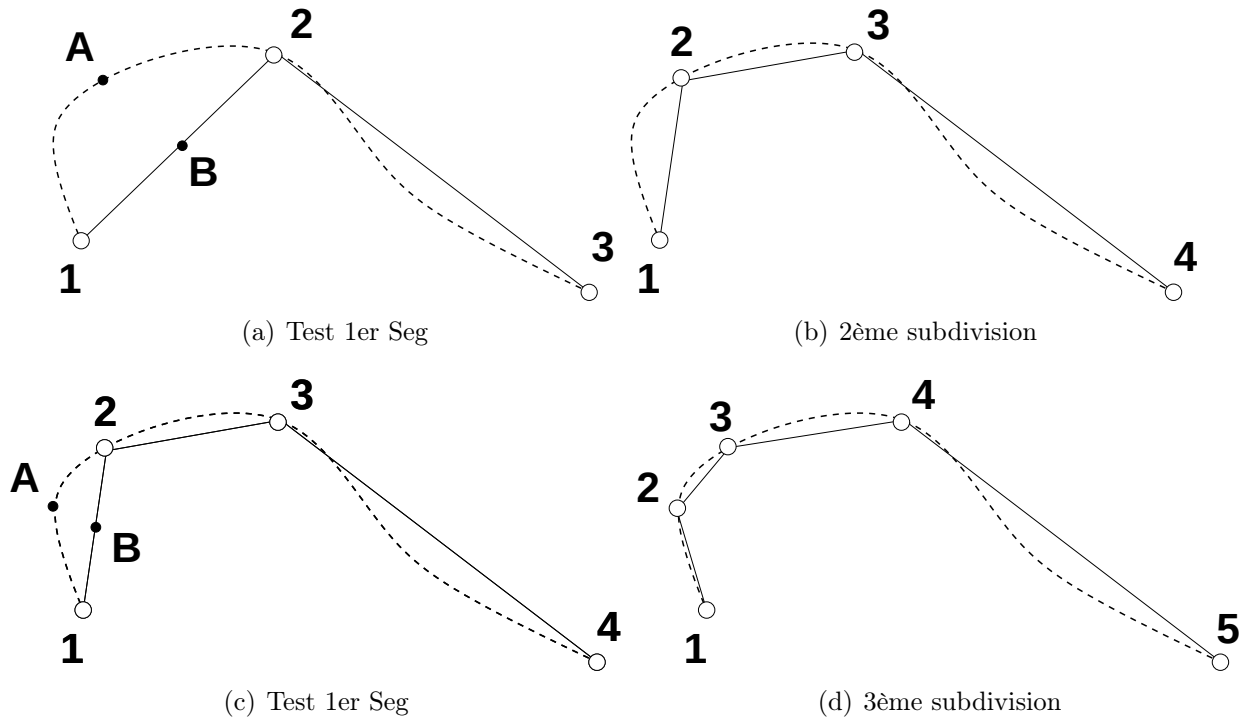


FIGURE 6 – Algorithme de subdivision binaire appliqué au 1er segment, 1 – 2

Le segment 1 – 2, jusqu'à ce que le critère soit vérifié, comme illustré, par exemple, à la Fig. 6(d). Alors, le test est appliqué au segment suivant, 2, Fig. 7(a), donnant le polygone 1 – 2 – 3 – 4 – 5 – 6

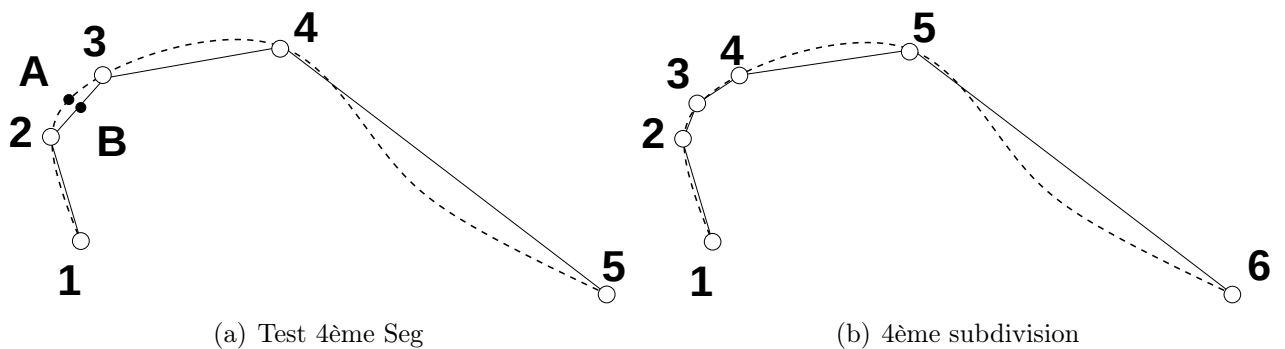


FIGURE 7 – Algorithme de subdivision binaire appliqué au segment 2 – 3

Après qu'un segment vérifie le critère, on passe au suivant, et ainsi de suite jusqu'au dernier segment, comme décrit dans le programme à la Fig. 8.

```
function [P,PTS] = geoLAGRANGE(CNTRL,nbCNTRL)
%   Interpolant de Lagrange de degre nbCNTRL-1, echantillonne sur
%   U(1)= 0 <= u <= nbCNTRL-1=U(nbCNTRL)
%   avec les nbCNTRL points de collocation CNTRL ==>(x,y)
%----- Intervalle initial-----
PTS=2; PT1=1; PT2=2;
U=(0:double(nbCNTRL-1));%----- parametrisation de la courbe
P(1,1:2)=CNTRL(1,1:2);
P(1,3)=0;%----- parametre u debut
P(2,1:2)=CNTRL(nbCNTRL,1:2);
P(2,3)=double(nbCNTRL-1);%----- parametre u fin
%----- subdivise tant que le critere geometrique n'est pas verifie
while PT1 < PTS
    Q(1:3) = (P(PT1,1:3)+P(PT2,1:3))/2;%----- Point milieu de l'intervalle
    PTS=PTS+1;
    for iPT=PTS:-1:PT2+1%----- decale la discretisation d'un PT
        P(iPT,1:3) = P(iPT-1,1:3);
    end
    %On evalue le point sur le SEG type Lagrange -----
    u=Q(3);
    P(PT2,1:3)= [0 0 u];%----- Calcule les coordonnees (x,y)
    %----- du point correspondant au parametre u sur le SEG
    for i=1:nbCNTRL
        L=1.;
        for j=1:nbCNTRL
            if (j~=i)
                L= L*(u-U(j))/(U(i)-U(j));
            end
        end
        P(PT2,1:2) = P(PT2,1:2) + L*CNTRL(i,1:2);
    end
    ecart = (P(PT2,1) - Q(1))^2 +(P(PT2,2) - Q(2))^2;
    if ecart < .00001
        PT1 = PT1+2;
        PT2 = PT1+1;
    end
end
end
```

FIGURE 8 – Construction de l'interpolant avec répartition adaptée à la courbure.

4 Exercice

Faire cet exercice après la lecture du document `exercice_MARSgeom`

1. Étudier l'algorithme de la Fig. 8
2. Transcrire dans un fichier matlab, `geoLAGRANGE.m` et remplacer son équivalent `geoLAGRANGE.p` dans **MARS**.
3. Exécuter à partir de l'interface :
 - 3.1. Construire une polygline dans le panneau **Construction générale** .
 - 3.2. Construire une approximation de Lagrange avec le bouton **Lagrange** du panneau **Segments**.
 - 3.3. Vérifier que le résultat est bien celui attendu.
4. Modifier le critère *ecart*, et analyser visuellement le comportement.
5. Vérifier que le résultat donne bien une répartition concentrée selon la courbure. Par exemple : Avec un cercle, la courbure devrait être constante. Tandis qu'avec une ellipse, on s'attend à une concentration sur les extrémités de l'axe majeure.
6. Peut-on prévoir le nombre de points du polygone final ?
7. Comment modifier l'algorithme de sorte que l'on puisse spécifier le nombre de points du polygone final ?