

Projet Geometrie

Youssef Daoud - Anas Mejgari

November 2020

1 Partie 1

1.1 Question 0

Soit $k \in [0, N]$

On a

$$P(u_k) = P_k H_0(0) + P_{k+1} H_2(0) + (u_{k+1} - u_k) m_k H_1(0) + (u_{k+1} - u_k) m_{k+1} H_3(0)$$

Alors

$$P(u_k) = P_k$$

De même on calcule $P(u_{k+1})$

$$P(u_{k+1}) = P_k H_0(1) + P_{k+1} H_2(1) + (u_{k+1} - u_k) m_k H_1(1) + (u_{k+1} - u_k) m_{k+1} H_3(1)$$

$$P(u_{k+1}) = P_{k+1}$$

En plus, on calcule les dérivées de P en u_k et u_{k+1} Pour u_k on trouve:

$$P'(u_k) = \frac{P_k}{u_{k+1} - u_k} H'_0(0) + \frac{P_{k+1}}{u_{k+1} - u_k} H'_2(0) + m_k H'_1(0) + m_{k+1} H'_3(0) = m_k$$

De même pour u_{k+1}

$$P'(u_{k+1}) = \frac{P_k}{u_{k+1} - u_k} H'_0(1) + \frac{P_{k+1}}{u_{k+1} - u_k} H'_2(1) + m_k H'_1(1) + m_{k+1} H'_3(1) = m_{k+1}$$

$$\begin{cases} P(u_k) = P_k \\ P(u_{k+1}) = P_{k+1} \\ P'(u_k) = m_k \\ P'(u_{k+1}) = m_{k+1} \end{cases}$$

Alors, pour tout k de 0 à N . La courbe en (2) interpole les points P_k et les tangentes m_k .

1.2 Question 1

Comme on a choisi une paramétrisation équidistante, avec $u_k = k$ pour $k = 0 \dots n$. On a $u_{k+1} - u_k = 1$. Ainsi par les propriétés vues en cours, on a:

$$\begin{cases} x_k(u_k) = \sum_{i=0}^3 b_{3k+i} B_i^3(u_k) = b_{3k} = P_k \\ x_k(u_{k+1}) = \sum_{i=0}^3 b_{3k+i} B_i^3(u_{k+1}) = b_{3k+3} = P_{k+1} \\ x'_k(u_k) = 3(b_{3k+1} - b_{3k}) = m_k \\ x'_k(u_{k+1}) = b_{3k+3} - b_{3k+2} = m_{k+1} \end{cases}$$

Ainsi, on trouve :

$$\begin{cases} b_{3k} = P_k \\ b_{3k+1} = \frac{1}{3}m_k + P_k \\ b_{3k+2} = -\frac{1}{3}m_{k+1} + P_{k+1} \\ b_{3k+3} = P_{k+1} \end{cases}$$

1.3 Question 2

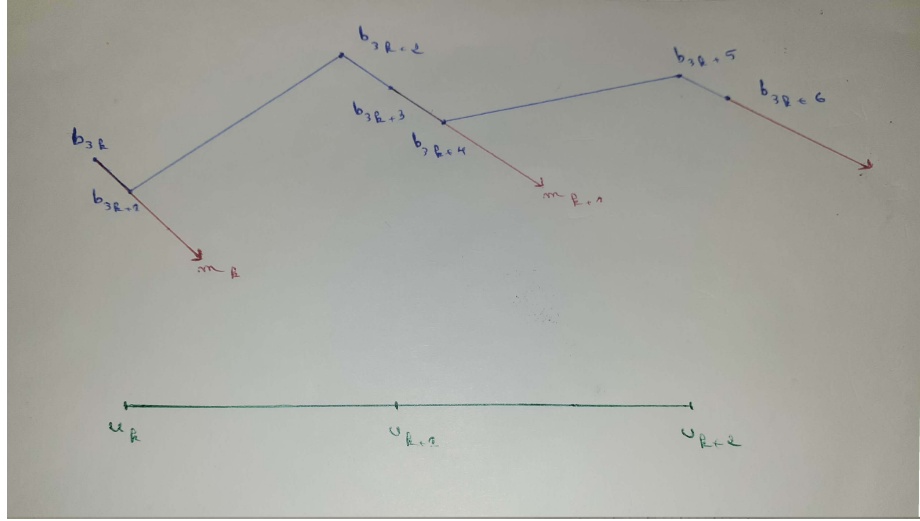


Figure 1: Deux polygones de contrôle consécutifs, x_k et x_{k+1} avec les points de contrôle.

1.4 Question 3

Lors du travail on fait appel à plusieurs méthodes d'approximations des dérivées qu'on a essayé de les mixées avec le principe du paramètre de tension (c) proposé par l'énoncé. Ainsi, on a implémenté les méthodes suivante:

1.4.1 Première Méthode:

On a utilisé l'approximation proposée par l'énoncé pour les m_k avec $k = 1, \dots, N-1$. Concernant, m_0 et m_N , on a proposé un schéma avec un pas à droite pour m_0 et un schéma avec un pas à gauche pour m_N . Donc

$$m_k = \begin{cases} (1-c) \frac{P_1 - P_0}{u_1 - u_0} & k = 0 \\ (1-c) \frac{P_{k-1} - P_{k+1}}{u_{k-1} - u_{k+1}} & k = 1, \dots, N-1 \\ (1-c) \frac{P_{N-1} - P_N}{u_{N-1} - u_N} & k = N \end{cases}$$

1.4.2 Schéma à deuxième ordre:

Il s'agit d'un schéma avec une erreur $O(h^2)$ même pour les extrémités "au contraire du premier schéma où l'erreur en extrémité est $O(h)$ ". On le traduit de la manière suivante:

$$m_k = \begin{cases} (1-c) \frac{-3P_k + 4P_{k+1} - P_{k+2}}{u_{k+2} - u_k} & k = 0, \dots, N-2 \\ (1-c) \frac{-3P_k + 4P_{k-1} - P_{k-2}}{u_{k-2} - u_k} & k = N-1, N \end{cases}$$

1.4.3 Amélioration:

On va essayer d'améliorer la première méthode proposée où on avait une erreur linéaire pour les dérivées en extrémités, en remplaçant ces dernières par leurs approximations calculées lors du schéma précédent:

$$m_k = \begin{cases} (1-c) \frac{-3P_0+4P_1-P_2}{u_2-u_0} & k=0 \\ (1-c) \frac{P_{k-1}-P_{k+1}}{u_{k-1}-u_{k+1}} & k=1, \dots, N-1 \\ (1-c) \frac{-3P_N+4P_{N-1}-P_{N-2}}{u_{N-2}-u_N} & k=N \end{cases}$$

La courbe suivante montre cette différence qui se trouve claire au niveau des représentations de x_1 et x_N :

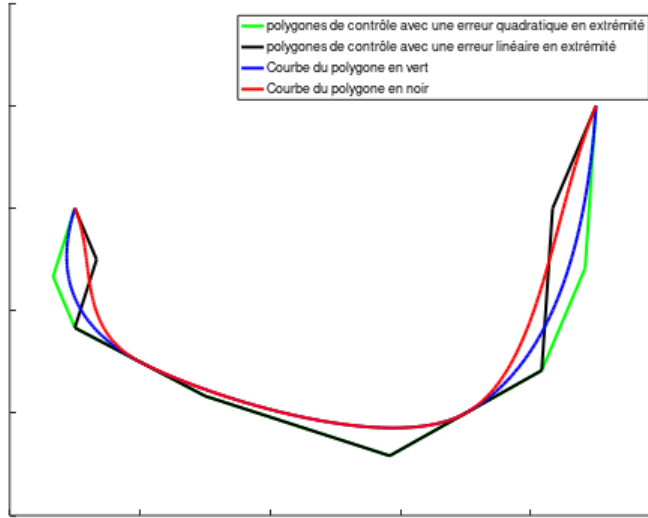


Figure 2: .

1.5 Question 4

1.5.1 Question 4.1

On propose l'exemple de la courbe suivante obtenue pour $P_0 = (1, 6)$, $P_1 = (2, 3)$, $P_2 = (7, 2)$ et $P_3 = (9, 8)$, avec le polygone de contrôle en vert, les polygones des P_k en bleu, et la courbe en rouge.



Figure 3: Schema des courbes pour $c=0$, $c=0.5$ et $c=1$.

On remarque que si c augmente, la courbe se rapproche à être des segments entre chaque P_k et P_{k+1} . C'est bien ce qu'on voit pour le cas de $c = 1$, la courbe est identique avec le polygone de contrôle. En fait, plus c augmente, plus les points b_k se rapprochent des points P_k jusqu'à ce qu'ils coïncident.

1.5.2 Question 4.2

D'après les trois figures ci-dessus, on remarque aucunes ondulations dans les trois courbes ce qui veut dire qu'ils sont lisses et ils suivent bien le chemin décrit par les points P_k . En plus, si on a un polygone convexe entrée, on aura bien une spline convexe en sortie (voir les ci-dessus).

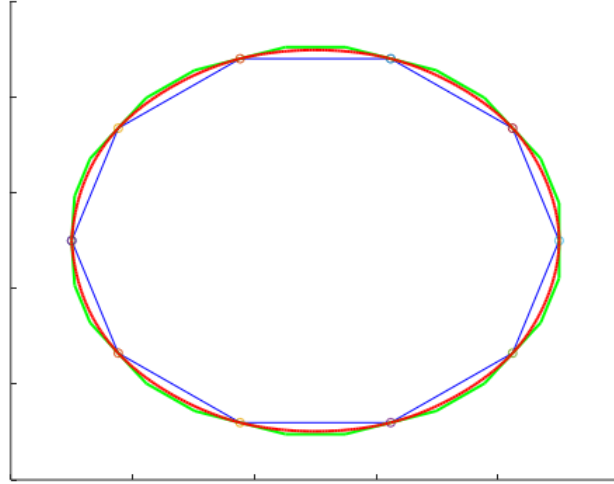


Figure 4: Schema d'un cercle pour $c=0$ et 10 points en entrée.

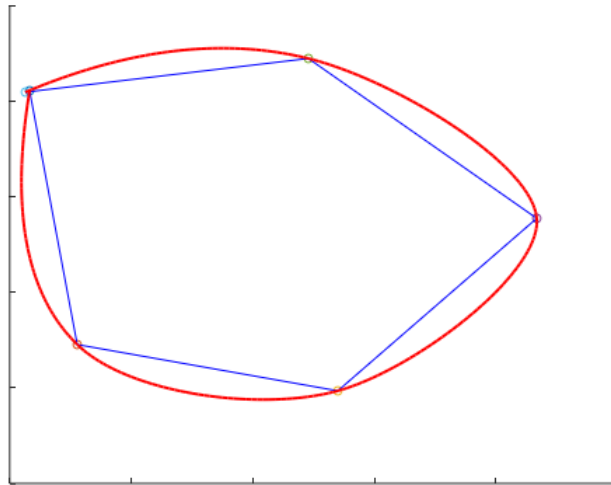


Figure 5: Schema d'un polygone convexe pour $c=0$ et 6 points en entrée.

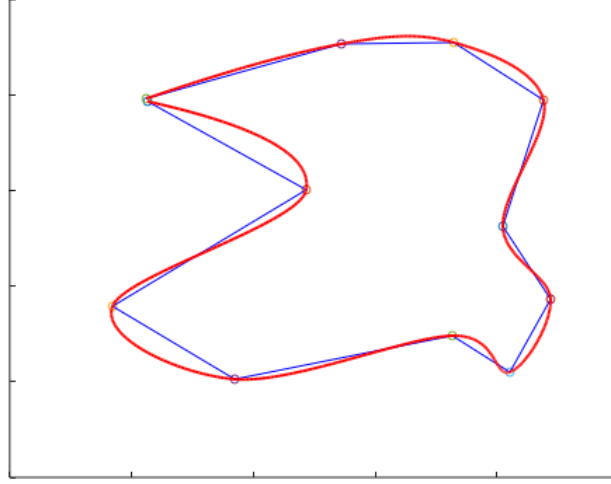


Figure 6: Schema d'un polygone non convexe pour $c=0$ et 6 points en entrée.

1.6 Question 4.3

On voit que les points m_k déterminent la forme de la spline. En fait, on remarque que si on augmente c la spline tend vers un segment sur $[u_k, u_{k+1}]$ au lieu d'être courbée.

En plus, on propose d'implémenter un nouveau choix de m_k par la fonction **tangente_ordre_4**. Ce nouveau choix de m_k est d'ordre 4 sauf les extrémités ce qui donne une courbe plus précise. (Voir documentation ici "Page 9")

$$m_k = \begin{cases} (1-c) \frac{-3P_k + 4P_{k+1} - P_{k+2}}{u_{k+2} - u_k} & k = 0, 1 \\ (1-c) \frac{-P_{k+2} + 8P_{k+1} - 8P_{k-1} + P_{k-2}}{12(u_{k-1} - u_{k+1})} & k = 2, \dots, N-2 \\ (1-c) \frac{-3P_k + 4P_{k-1} - P_{k-2}}{u_{k-2} - u_k} & k = N-1, N \end{cases}$$

Vous trouverez ci-dessous un dessin comparant les deux choix de m_k .



Figure 7: Différence entre les courbe selon les méthodes utilisées pour l'approximation des tangentes ($c=0$).

1.7 Question 5:

On a essayé dans cette question de faire un plot du graphe de la fonction de courbure $k(u)$ définie par:

$$k(u) = \frac{\|\mathbf{x}'(t) \times \mathbf{x}''(t)\|}{\|\mathbf{x}'(t)\|^3}$$

Ainsi, on avait recours à un calcul des dérivées d'une courbe de Bézier. Pour cela, on a implémenté la fonction **derivee**, qui prend comme arguments les points u_0 et u_N ainsi que les points de contrôle b d'une courbe, elle nous renvoie donc les points de contrôle b' de sa dérivée. Après, on fait appel à la fonction courbure qui calcule la fonction de courbure en partant des points de contrôle. Enfin, on dessine la courbure grâce à la fonction *dessin_courbure*.

1.8 Question 5.1:

On propose la courbe paramétrée par $P_0 = (1, 9)$, $P_1 = (3, 1)$, $P_2(5, 5)$, $P_3 = (7, 1)$, $P_4 = (9, 9)$ et les U qui sont les mêmes que précédents. On dessine la courbe avec premièrement l'**approximation des tangentes posée par l'énoncé en rouge**, puis par l'**approximation des tangentes proposée lor de 4.3 en bleu** Ainsi les courbes sont comme qui suit:

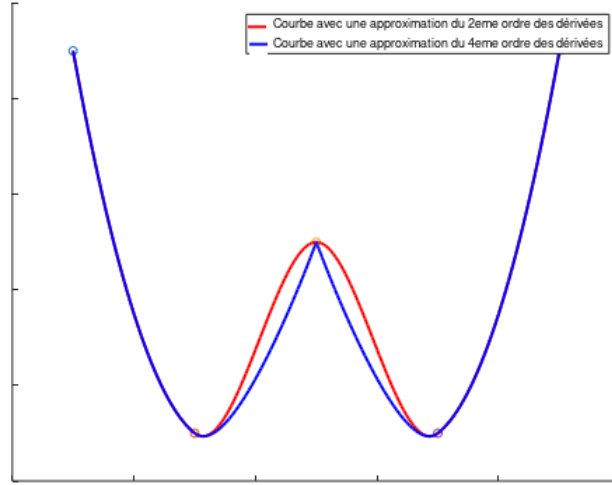


Figure 8: Courbe ($c=0$) pour des différentes approximations .

On obtient donc la fonction de courbure suivante:

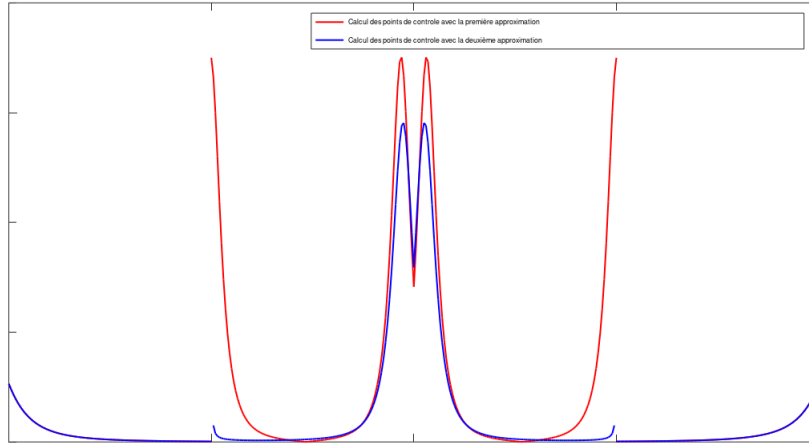


Figure 9: Fonction de courbure pour les 2 courbes précédentes.

On remarque que les 2 courbes sont les mêmes sur $[u_0, u_1]$ et sur $[u_3, u_4]$, et $[u_1, u_3]$ la courbure pour l'approximation du 4ème ordre est clairement inférieure que celle l'approximation proposée au départ. Cette nouvelle approximation nous a permis de réduire clairement la courbure entre la fin de x_1 et le début de x_2 , ainsi qu'entre la fin de x_3 et le début de x_4 . Ainsi, le dessin a une allure plus uniforme (des deux cotés) au voisinage de P_1 et de P_3 . Concernant u_2 , l'approximation a réduit la valeur de la courbure d'une manière significative et donc la vitesse de variation de la tangente est plus réel avec l'approximation d'ordre 4.

1.9 Question 5.2:

On sait bien que la courbure évalue le rapport entre la variation de la direction de la tangente à la courbe et un déplacement d'une longueur infinitésimale sur celle-ci. Pour être significative, la définition de la courbure et ses différentes caractérisations nécessitent que la courbe soit continuellement dérivable près d'un point P.

1.10 Question 6

On propose le dessin suivant qui représente une approximation du classique *courbe de Lissajous*, avec les points de $P_0 = (5, 5), P_1 = (9.75, 0.25), P_2 = (7.9, 7.9), P_3 = (2, 7.9), P_4 = (0.25, 0.25), P_5 = (5, 5), P_6 = (9.75, 9.75), P_7 = (7.9, 2), P_8 = (2, 2), P_9 = (0.25, 9.75), P_{10} = (5, 5)$

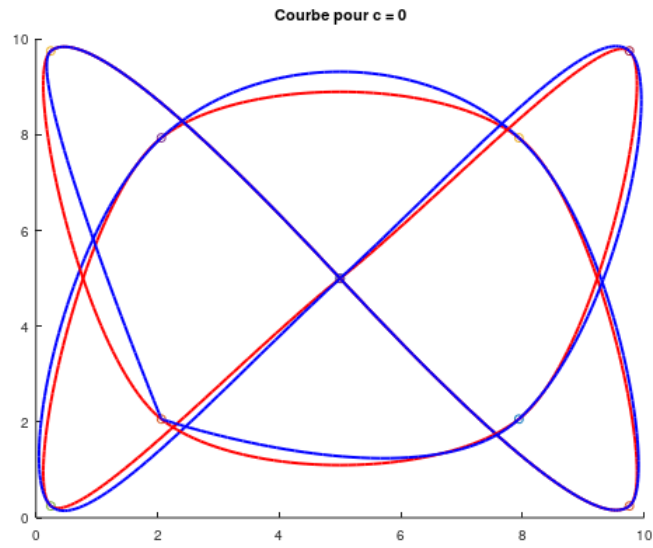


Figure 10: Approximation de la courbe de Lissajous pour $c=0$

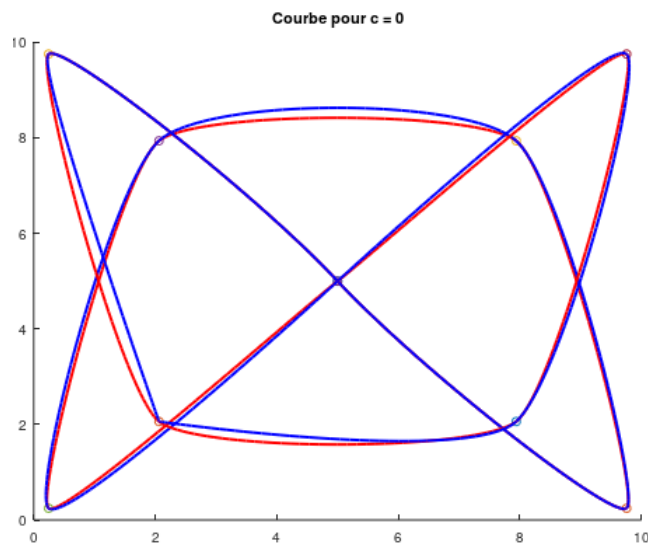


Figure 11: Approximation de la courbe de Lissajous pour $c=0.5$

Ceci est une simple implémentation, on propose la foction **lissajous.m** qui résulte la ourbe suivante

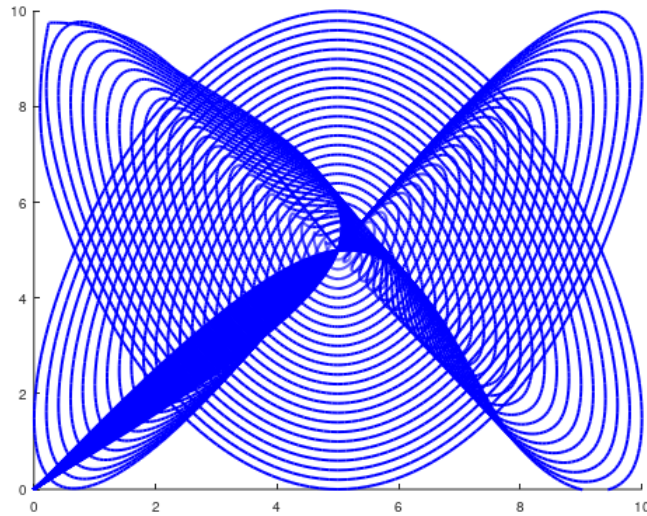


Figure 12: Une figure plus stylée de Lissajous

2 Partie 2

2.1 Question 7

Vous trouverez ci-dessous un exemple de courbe basé sur l'interpolation de Lagrange en utilisant l'algorithme de Aitken-Neville. Vous trouverez l'implémentation de cet algorithme dans le fichier **eval_aitken.m**.

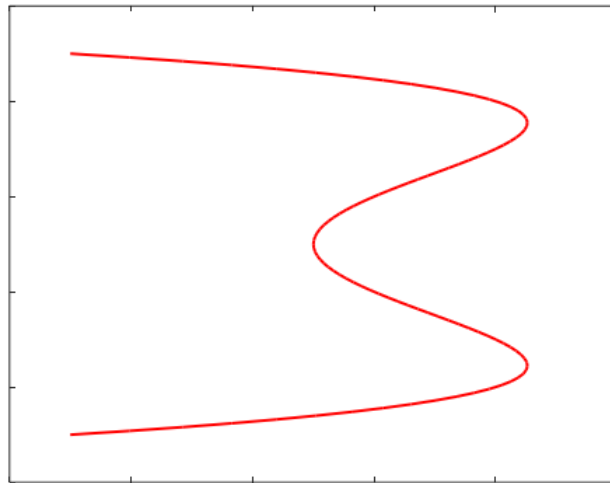


Figure 13: Exmple de courbe en utilisant l'algorithme d'Aitken-Neville

Maintenant, vous trouverez ci-dessous une courbe décrivant la comparaison entre les deux méthodes; **l'interpolation de Lagrange** et **l'interpolation d'Hermite**.

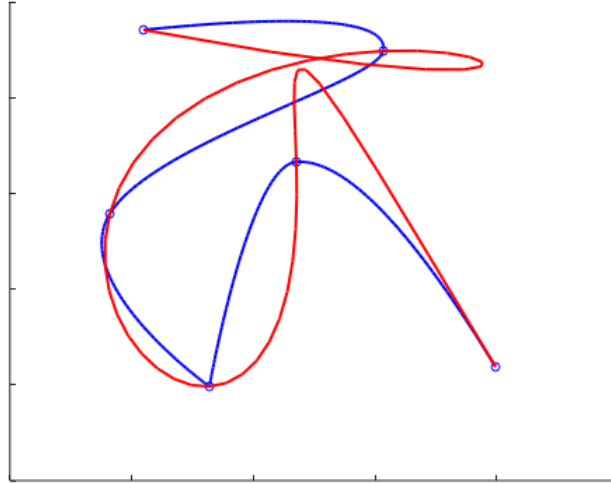


Figure 14: Comparaison entre l'interpolation de Lagrange et l'interpolation d'Hermite

D'après la figure ci-dessus, on remarque que l'interpolation d'Hermite est plus précise par rapport à celle de Lagrange. En plus, on voit moins de courbures dans l'interpolation d'Hermite.

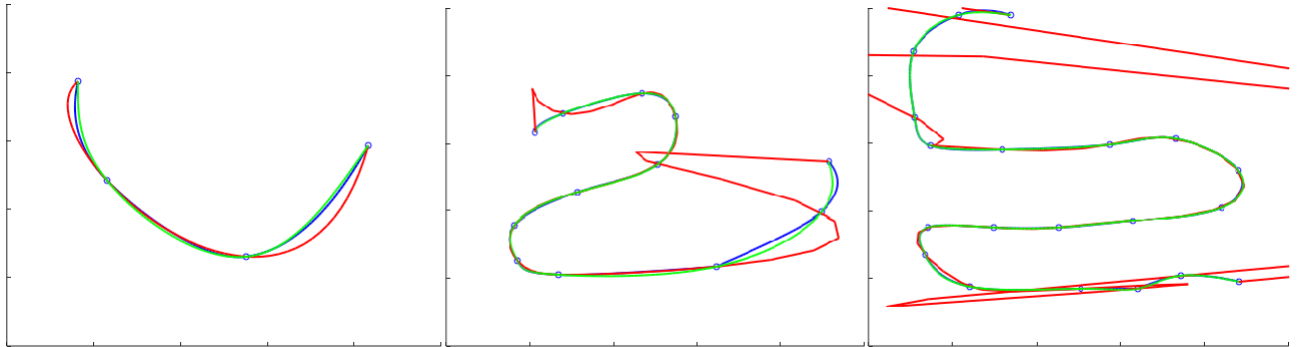


Figure 15: Interpolation de 4, 12 et 20 points par les méthodes de l'interpolation de Lagrange, Spline C^1 et Spline C^2

Après avoir implémenté les splines cubiques C^2 (vous trouverez le code dans le fichier **spline_C2.m**), on remarque que le nombre de points choisis affecte les courbes surtout pour l'implémentation de Lagrange. En fait, on voit que plus le nombre de points augmente plus la courbe de Lagrange oscille beaucoup autour des points.

2.2 Question 8

On a permis à l'utilisateur de choisir s'il veut superposer les deux ou les trois courbes dans le fichier **Courbe_spline.m**.

2.3 Question 9

On voit que la courbe Spline C^2 est la meilleure parmi les trois. En fait, la courbe est C^2 , plus lisse, très robuste et le tracé paraît naturel: il ne présente pas de pointe, de détour, etc.