SALVI Titouan
TAILHANDIER Eulalie

Compte rendu TP n°3 : Interpolation polynomiale

<u>But du TP :</u> création de fonctions m-file et à travers l'étude de la convergence de l'interpolation de Lagrange

Exercice n°1 :

- La première question demande de définir deux vecteurs, nous avons rentré les coordonnées des vecteurs lignes X et Y dans deux matrices distinctes.

```
% QUESTION 1 : Enregistrement des vecteur X et Y :
 X = [-10 -8 -6 -4 -2 0 2 4 6 8 10]
 Y =[3.1623 2.8284 2.4495 2.0000 1.4142 0 1.4142 2.0000 2.4495 2.8284 3.1623]
```

- Les trois questions suivantes nécessitent la création de trois fonctions m-file :

   - La **fonction de Lagrange** retourne pour un x et un i donnés la valeur du polynôme Li. Les polynômes Li seront ensuite utilisés pour i allant de 0 à n comme coefficients dans l'interpolation de Lagrange. Le x lui, définit la valeur en laquelle on identifie la fonction.
   Ces polynômes correspondent à des produits. La première étape est celle de l'initialisation du polynôme à 1 et se poursuit par des itérations pour obtenir le produit final. Pour cela mise en place d'une boucle for balayant les indices de 0 à j. J est le nombre de valeurs connues pour la fonction qui correspond à la taille de la matrice : au nombre d'éléments. Nous excluons la valeur i=j grâce à une condition imposée par un if à chaque itération.

```
Editor - C:\Users\Admin\Documents\PERSO Titouan\ECOLE\INSIC\TA\matlab\TP2 3105T
tp3.m | Discretisation.m | Interpole.m | Lagrange.m | +
function L=Lagrange(X,x,i)
L=1
 for j =1: length(X)
     if i~=j
         L =L*((x-X(j))/(X(i)-X(j)))
     end
 end
```

On appelle la fonction créée dans notre programme et on l'évalue par exemple ici pour (X,x,i) =(X,2,3)

```
%QUESTION 2 : Fonction Lagrange
%l'utilisateur doit rentrer i et x

i=3
x=2
L=Lagrange(X,x,i);
```

SALVI Titouan
TAILHANDIER Eulalie

- La **fonction Interpole** permet pour un x donné d'exécuter la fonction d'interpolation de Lagrange à partir des coefficients Li de Lagrange connus et des matrices X et Y correspondant aux valeurs réelles connues de notre fonction.
  Pour effectuer la somme, nous initialisons la somme à 0 puis nous mettons en place une boucle for balayant toutes les valeurs (lenght(Y)=lenght(X)).A chaque itération, nous ajoutons un terme à la somme en appelant le produit du le polynôme de Lagrange Li et du yi correspondant.

Editor - C:\Users\Admin\Documents\PERSO Titouan\ECOLE\INSIC\1A\mathlab\T

tp3.m | Discretisation.m | Interpole.m | Lagrange.m

```
function Z=Interpole(X,Y,x)
  Z=0
  for k=1:length(X)
      Z=Lagrange(X,x,k)*Y(k)+Z
  end
```

Nous appelons la fonction créée dans notre programme et on l'évalue en (X,Y,x) =(X,Y,3)

```
%QUESTION 3 : Fonction interpol
%l'utilisateur doit rentrer x

x=3
Z=Interpole(X,Y,x)
```

- La **fonction discrétisation** discrétise un intervalle [a ;b] en fonction d'un nombre de sous-intervalles n décidé par l'utilisateur.

Nous utilisons une variable h pour trouver la taille de chaque intervalle. Ensuite, il suffit de mettre en place une boucle for qui à chaque itération complète le vecteur ligne Xthéo. Les différentes coordonnées du vecteur sont trouvées grâce à une fonction récursive. Nous initialisons la fonction avec la valeur a et à chaque itération d'indice i on ajoute le pas. Cela revient à utiliser a à chaque itération et faire la somme de a avec le produit de l'indice par le pas.

Editor - C:\Users\Admin\Documents\PERSO Titouan\ECOLE\INSIC\1A\mathlab\TP

tp3.m | Discretisation.m* | Interpole.m | Lagrange.m

```
function Xtheo=Discretisation(a,b,n)
  h=(b-a)/n
  for m=0:n
      Xtheo(m+1)=a+h*m
  end
```

Nous appelons la fonction créée dans notre programme et l'évaluons en (a,b,n)=(-10,10,20) avec-10 et 10 nos bornes d'intervalle et 20 le nombre de sous-intervalles.

Cette fonction nous permet de créer un nouveau tableau, plus complet que le précédent. A chaque xi trouvé, nous associons la valeur de l'interpolation de Lagrange connue grâce au premier tableau. Nous enregistrons les données dans deux nouveaux vecteurs lignes : Xthéo et Ythéo.

```
% QUESTION 4 :Fonction discrétisation
%l'utilisateur doit rentrer n

n=20
a=-10
b=10

Xtheo=Discretisation(a,b,n)
% Création de la matrice Y théorique
for k=1:n+1
    Ytheo(k)=Interpole(X,Y,Xtheo(k))
end
```

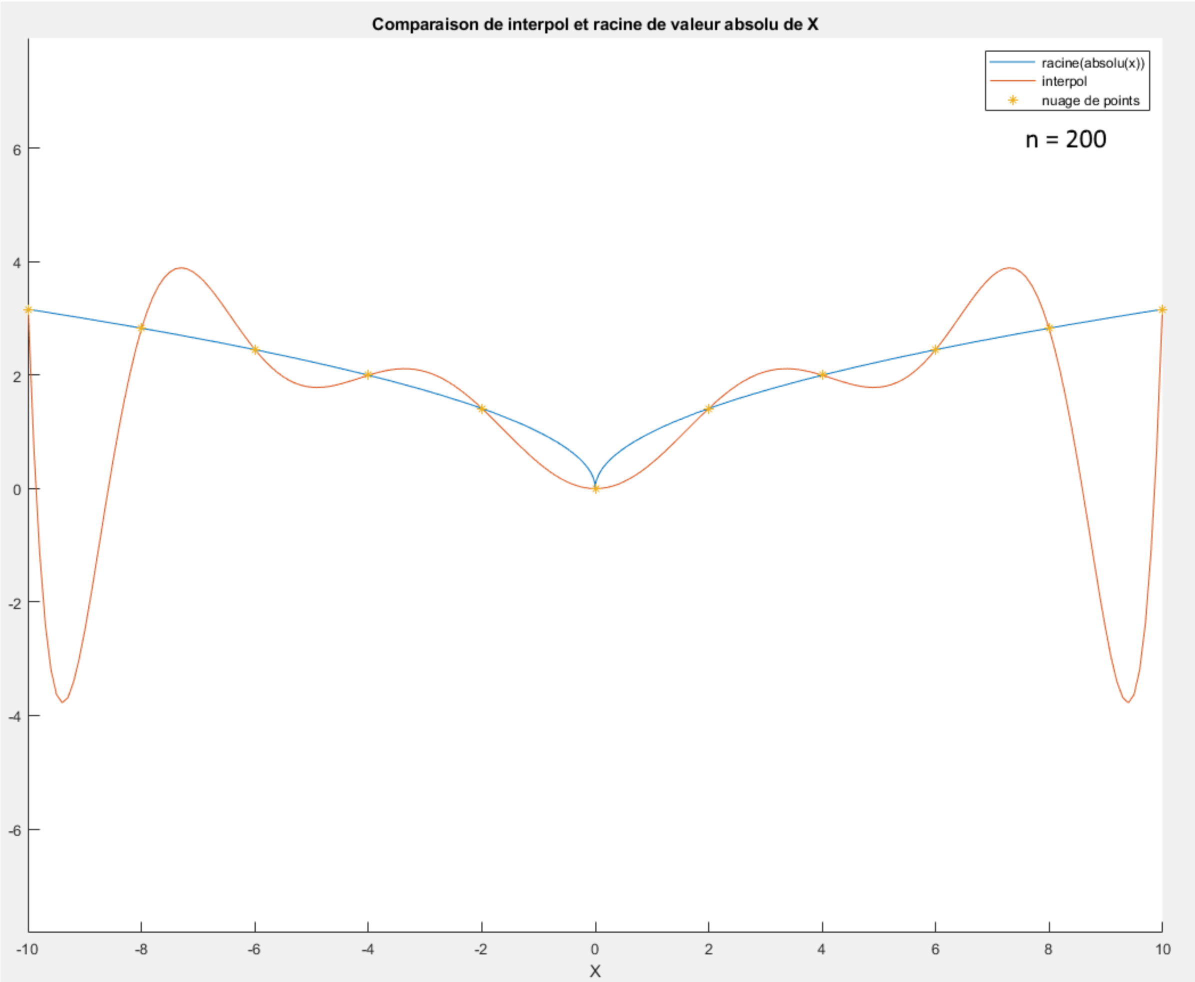
- Tracer les courbes des différentes fonctions :

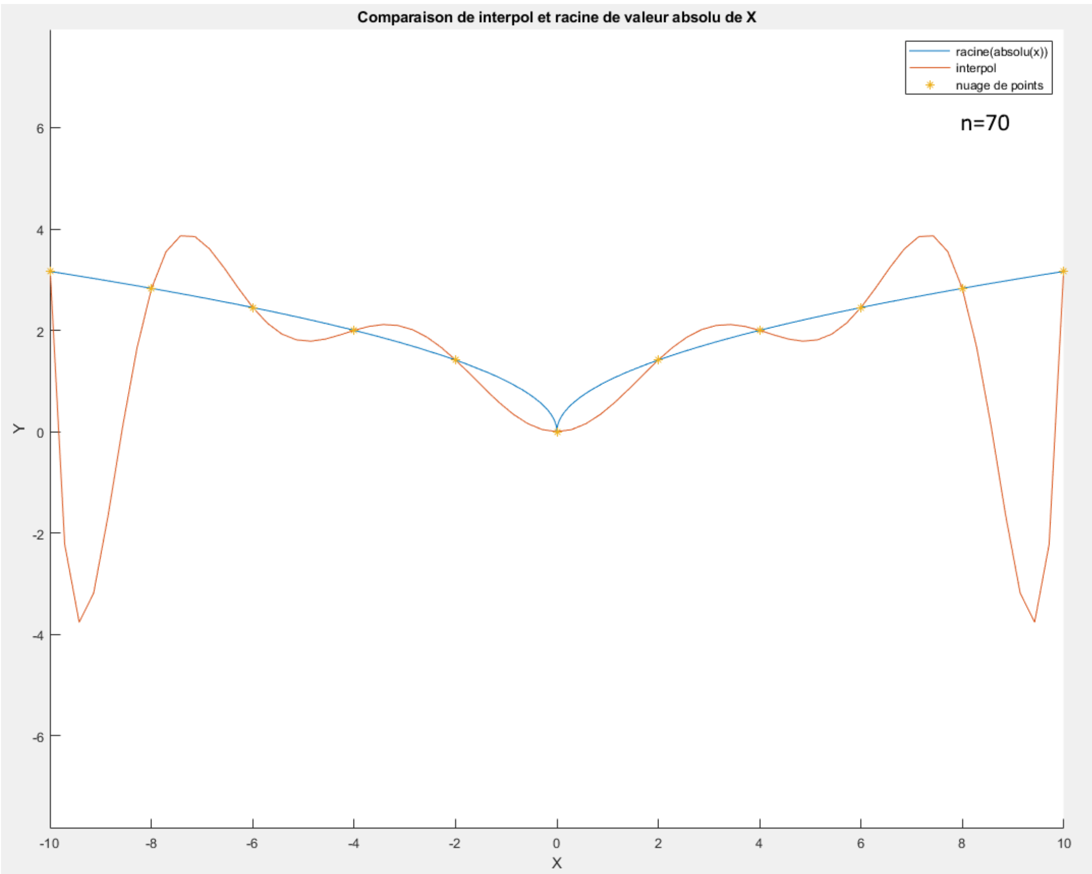
L'étape suivante est la comparaison des graphes de la fonction dont on connait seulement les coordonnées exactes (xi,yi) de 11 points d'après l'énoncé, de celui de la fonction interpolation de Lagrange trouvée à partir de ce même énoncé et de la fonction continue $[-10;10] \to R;\ \ x \to \sqrt{|x|}$.

```
% QUESTION 5 : Graphique des fonction interpol et racine de x sur l'interval [-10;10]

hold on,
fplot('sqrt(abs(x))',[-10 10])
hold on
plot(Xtheo,Ytheo)
hold on
plot(X,Y,'*')
axis equal
title('Comparaison de interpol et racine de valeur absolu de X')
xlabel('X')
ylabel('Y')
legend(' racine(absolu(x))',' interpol',' nuage de points')
```

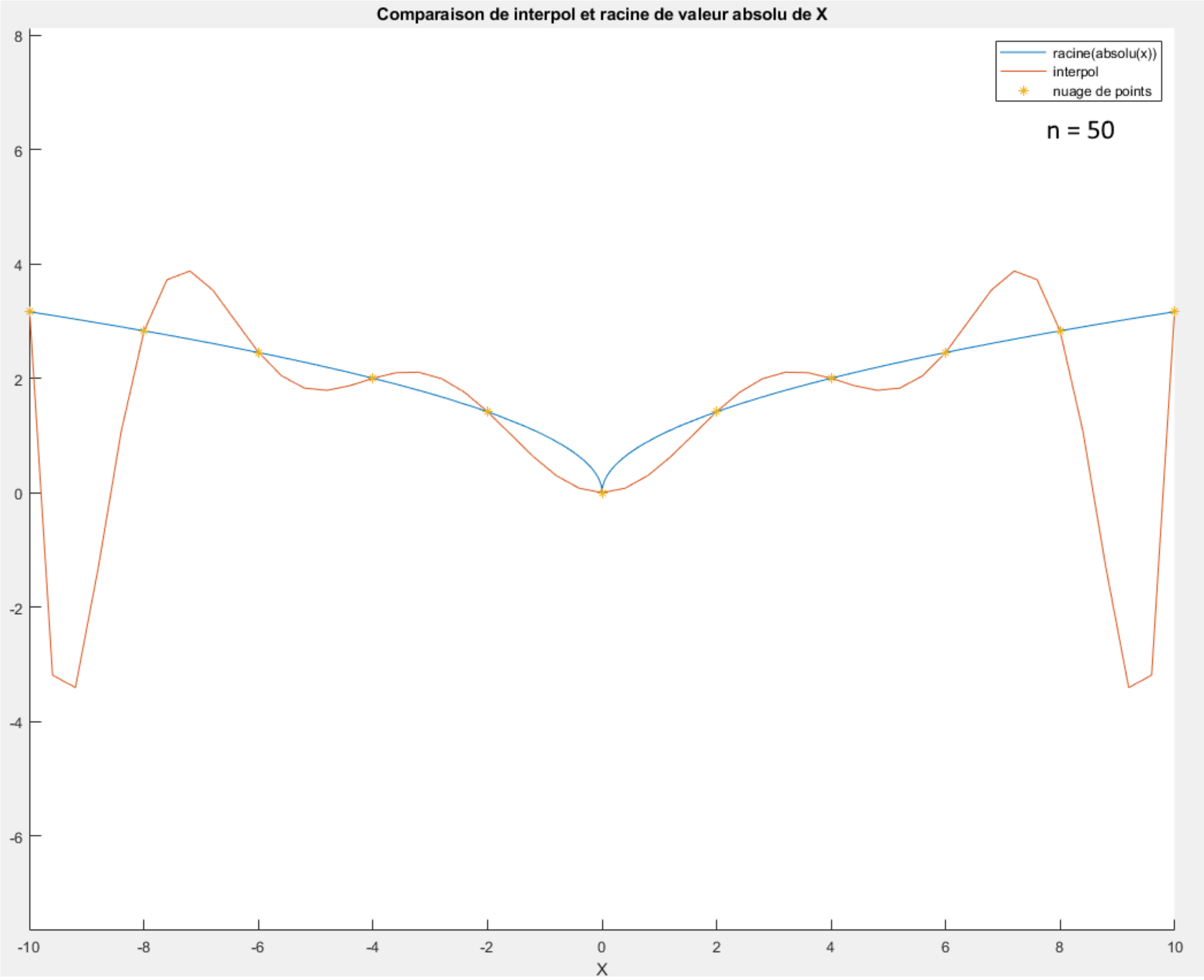
Ci-dessous, les différents graphiques illustrant la variation de n :

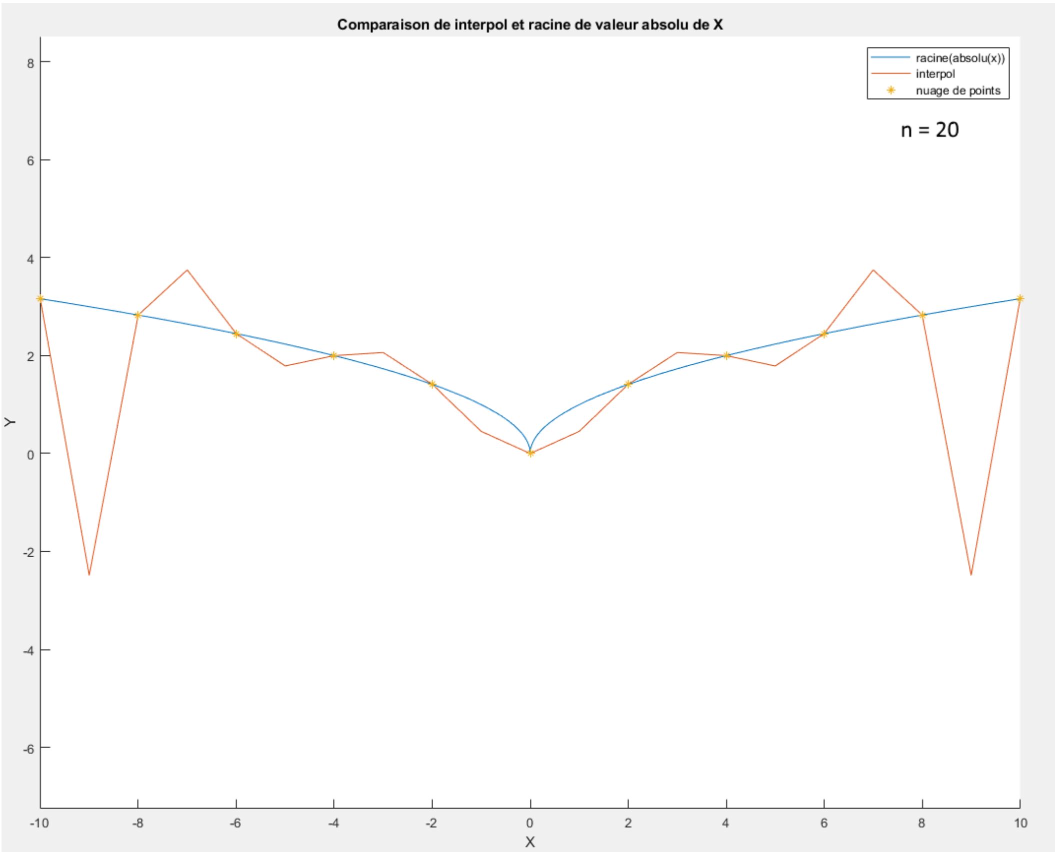
SALVI Titouan
TAILHANDIER Eulalie





SALVI Titouan
TAILHANDIER Eulalie





Nous nous apercevons que nous avons onze points de collocation, ce qui est cohérent puisque le tableau initial possède 11 valeurs, le degré du polynôme de Lagrange. Plus le n augmente, plus on discrétise l'intervalle plus on applique la fonction interpole à un grand nombre d'antécédents. Graphiquement plus n augmente plus la courbe d'interpole traduit le graphe d'une fonction continue et dérivable. Elle possède de moins en moins de points anguleux.

- La dernière question repose sur les calculs d'erreurs quadratiques et absolues.

Nous appliquons les formules des deux erreurs l'une après l'autre. Pour l'erreur quadratique une boucle for nous permet de sommer chaque terme à la somme globale à chaque itération.

L'utilisation des fonctions sqrt, abs et max nous permet de répondre à la question.

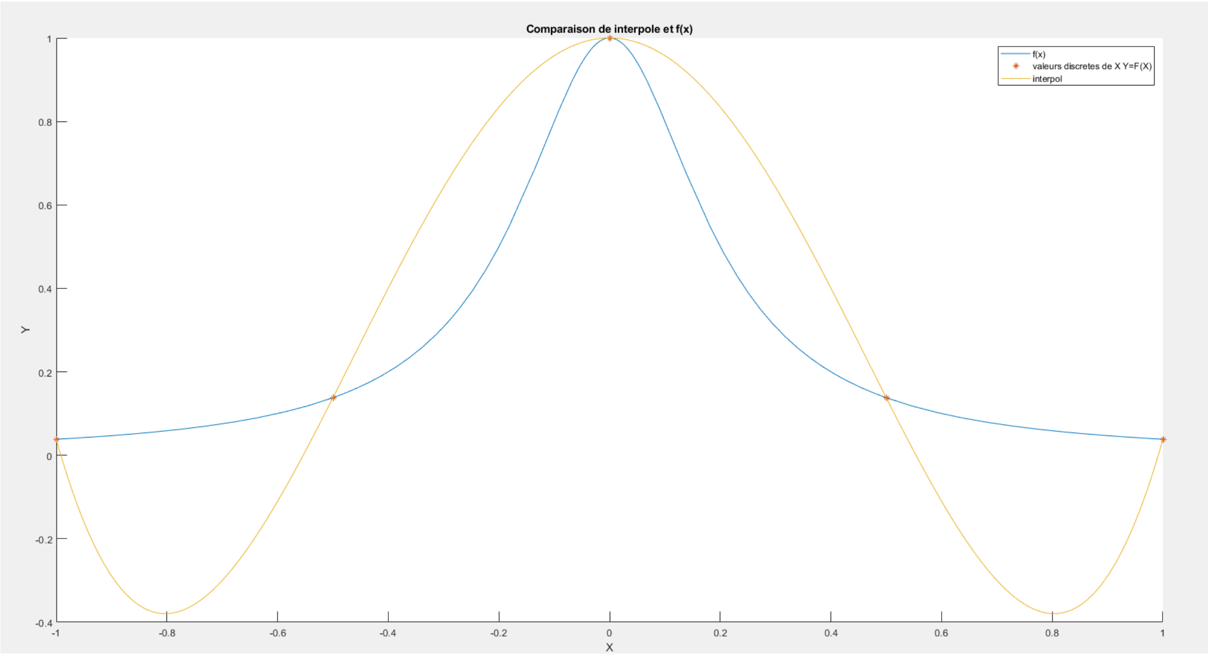
```
%QUESTION 7 : Erreur quadratique de l'approximation : E
E=0
for w=1:length(Ytheo)
E=E+(Ytheo(w)-sqrt(abs(Xtheo(w))))^2
end
%QUESTION 6 : Erreur absolue de l'approximation : e

e=max(abs(Ytheo-sqrt(abs(Xtheo))))
```

Exercice n°2 :

Nous rappelons les 3 fonctions m-file pour interpoler avec Lagrange une nouvelle fonction f :

L'intervalle de travail de [-1 ;1] est discrétisé par la fonction discrétisation pour créer un tableau à 5 valeurs, dans X3. Ensuite le programme évalue la valeur exacte de la fonction en ces points xi. La fonction interpole permet d'évaluer l'interpolation de Lagrange de ces mêmes xi. Puis, nous utilisons l'expression demandée pour créer un nouveau nuage de points donc une matrice X et re-appliquer la fonction à ces antécédents.

```
%  exercice 2
%--------------

close all

%FONCTION F(x) :

%Déclaration des bornes de l'intervalle de travail
a=-1;
b=1;

%Creation des vecteurs X2 et Y2 comme ceux de l'exercice 1 à l'aide de la
%fonction discrétisation

%Il faut que l'utilisateur rentre le n
n=4;

X2=Discretisation(a,b,n);
 for q=1:length(X2)
     Y2(q)=1/(1+25*(X2(q))^2)
 end

%Creation des vecteurs X3 et Y3 (comme Xtheo et  Ytheo de l'exo 1)
 n2=500
 X3=Discretisation(a,b,n2);
 for s=1:length(X3)
    Y3(s)=Interpole(X2,Y2,X3(s))
 end


%Graphique des fonctions interpol et f(x) sur l'intervalle [-1;1]

hold on
fplot('1/(1+25*x^2)',[-1 1])  %graph F(x) continue
plot(X2,Y2,'*')               %graph des valeurs discretes de X Y=F(X)
plot(X3,Y3)                   %graph de l'interpol
title('Comparaison de interpole et f(x)')
xlabel('X')
ylabel('Y')
legend('f(x)','valeurs discretes de X Y=F(X)','interpol')
```

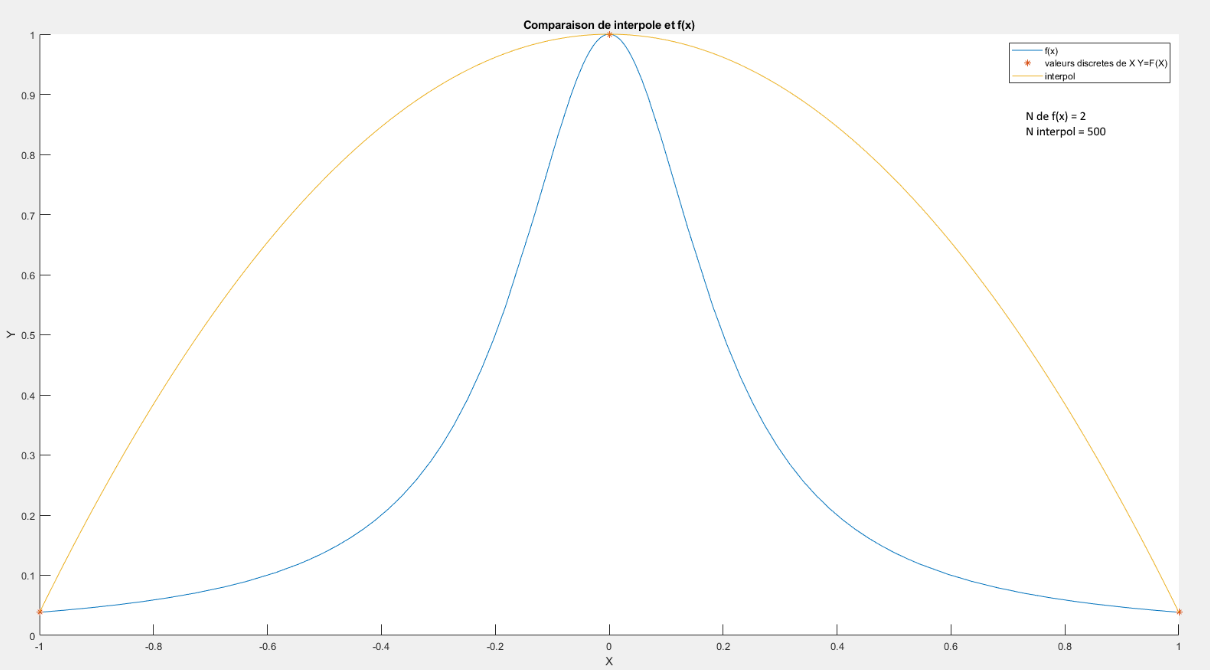


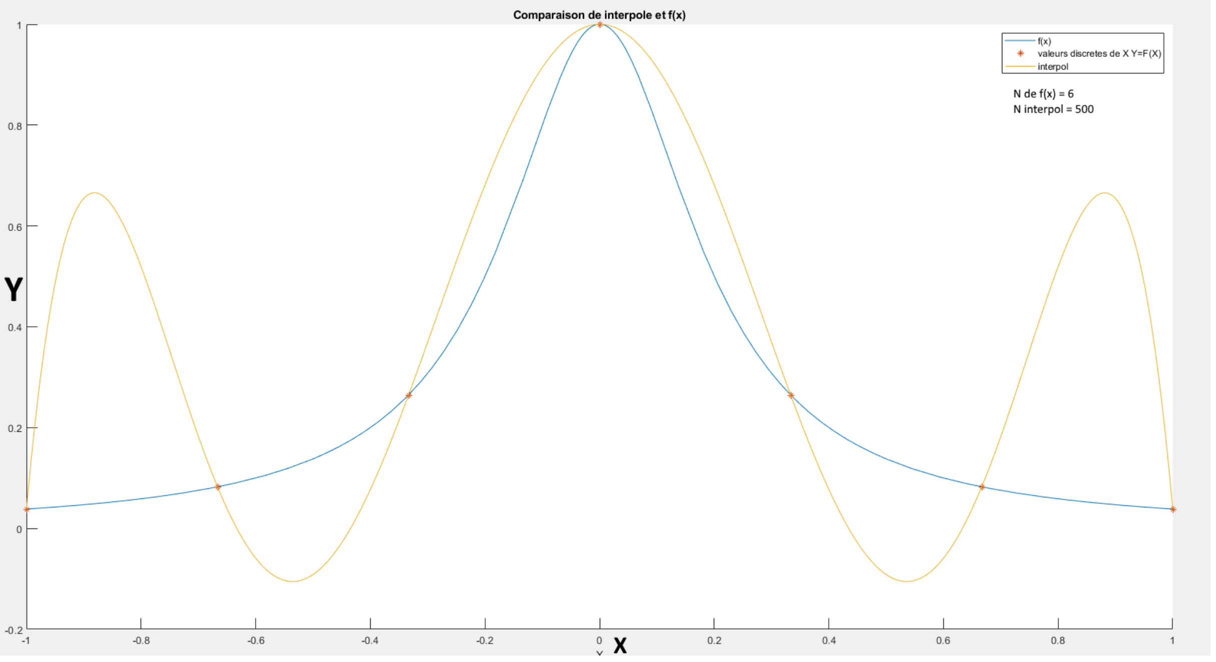

```
%FONCTION Nuage de points :

n=500
for i =1:n
    X4(i)=((b-a)/2)*cos(((2*i+1)/(2*(n+1)))*pi)+(b+a)/2
end

 for s=1:length(X4)
    Y4(s)=Interpole(X2,Y2,X4(s))
 end

 %Graphique des fonctions interpol et f(x)avec le nouveau nuage de points sur l'intervalle [-1;1]
close all
hold on
fplot('1/(1+25*x^2)',[-1 1])  %graph F(x) continue
plot(X2,Y2,'*')                %graph des valeurs discretes de X Y=F(X)
plot(X4,Y4)                    %graph de l'interpol avec le nuage de points
title('Comparaison de interpole et f(x)')
xlabel('X')
ylabel('Y')
legend('f(x)','valeurs discretes de X Y=F(X)','interpol par le nuage de points')
```

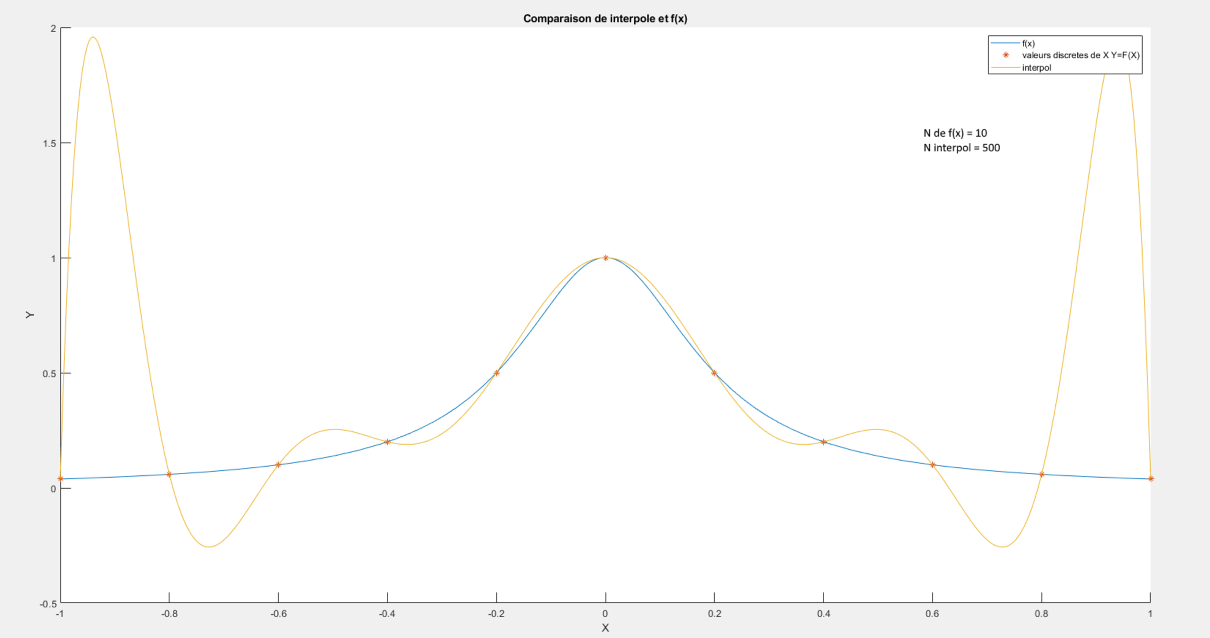
Plus n augmente plus le tableau initial possède de valeur, plus le degré du polynôme de Lagrange augmente, plus nous obtenons de points de collocations entre la courbe réelle et la courbe d'interpolation ; ainsi plus la courbe devient précise. Cependant plus n augmente plus la courbe d'interpolation oscille et s'éloigne de la valeur réelle. Cela correspond avec le constat fait en cours.

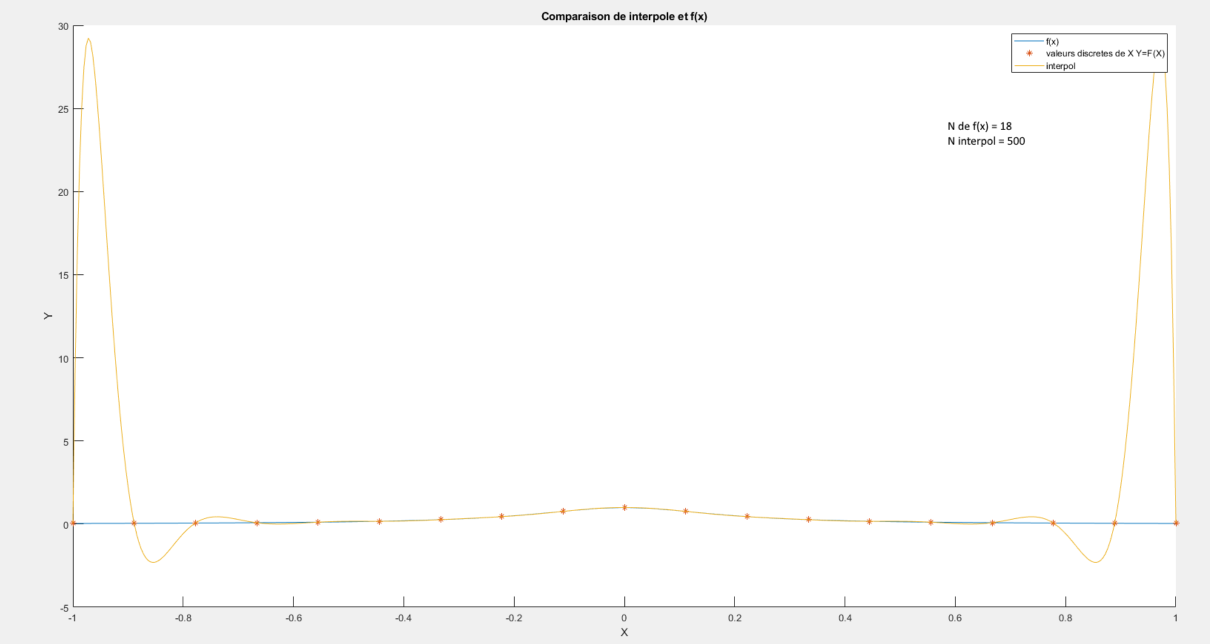
Ci-dessous les graphiques montrant cela :

SALVI Titouan
TAILHANDIER Eulalie







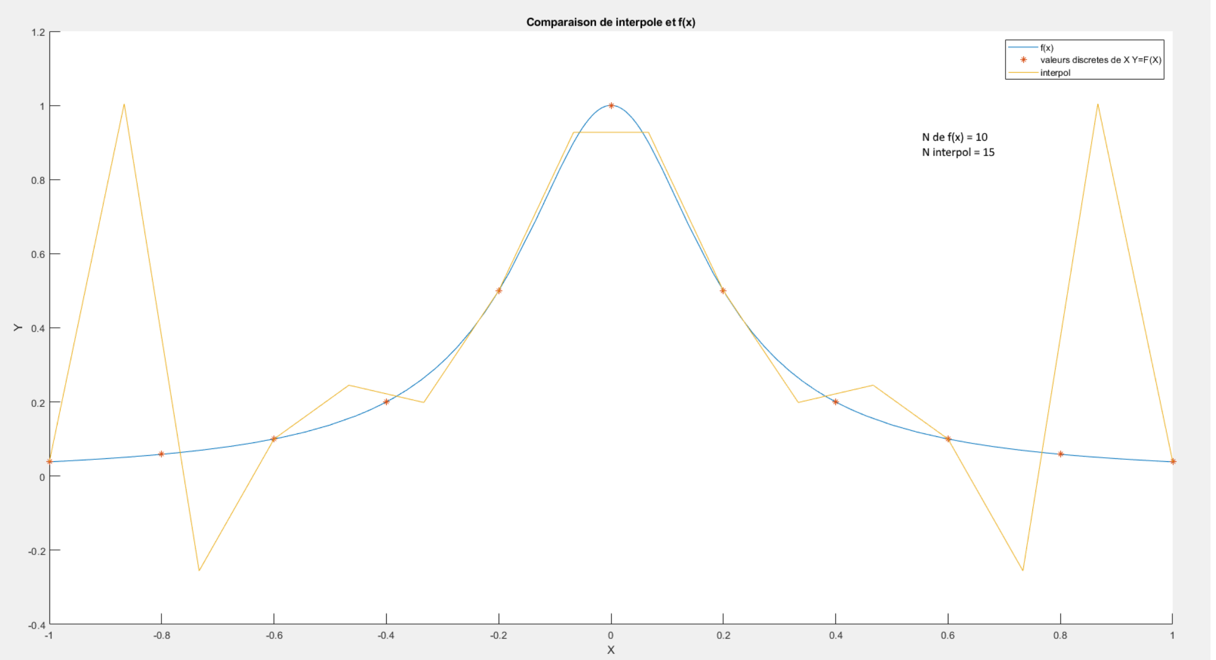


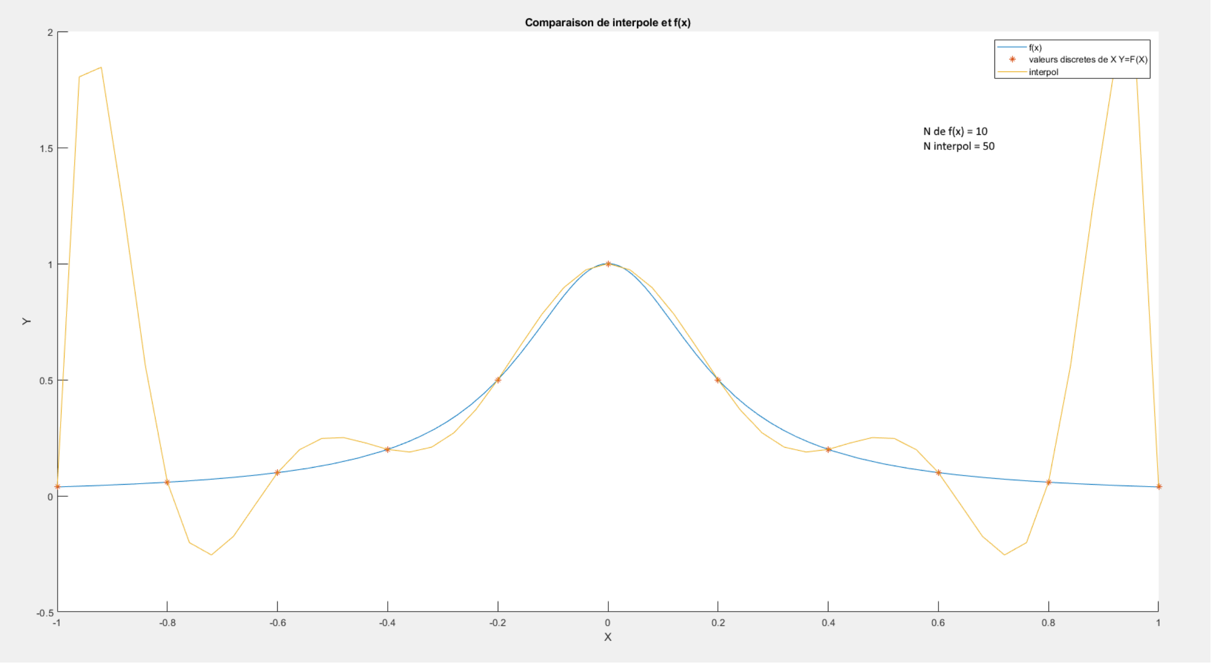
On constate aussi le même phénomène de continuité et de dérivabilité vu dans l'exercice quand on fait varier le nombre d'antécédents auxquels on applique la fonction.

Après avoir utilisé le nuage de points proposé dans l'exercice plutôt que la fonction discrétisation, on constate que plus n est grand plus on se rapproche du graphique vu avec la fonction interpole à partir de l'intervalle discrétisé. Concernant les points de collocations, la même observation est remarquée.

Ci-dessous les graphiques montrant cela :

SALVI Titouan
TAILHANDIER Eulalie





SALVI Titouan
TAILHANDIER Eulalie