

SRT3 - Travaux pratiques Méthodes Numérique

TP3 Méthodes numériques : Interpolation

RÉSUMÉ
TP3 Méthodes numériques : Interpolation

SOMMAIRE

	1
SRT3 - Travaux pratiques Méthodes Numérique	1
TP3 Méthodes numériques : Interpolation	1
TP3 Méthodes numériques : Interpolation	1
SOMMAIRE	2
Introduction	3
Un premier problème d'interpolation	3
Interpolation spline	13
Splines cubiques modifiées	25
4. Régression	30
Conclusion	43

Introduction

Un premier problème d'interpolation

Nous allons commencer par convertir le fichier binaire ex1_4.bin en un fichier ASCII pour visualiser les points d'interpolation.

```
gcc -o bin2ascii bin2ascii.c -lm -Ofast
```

```
./bin2ascii ex1_4.bin
```

```
(athenista@kali-athena)-[~/.../Tp/tpMethodesNumériques/tpInterpolation/Sources]
$ gcc -o bin2ascii bin2ascii.c -lm -Ofast

(athenista@kali-athena)-[~/.../Tp/tpMethodesNumériques/tpInterpolation/Sources]
$ ./bin2ascii ex1_4.bin
nombre de couples lu: 4
```

Figure 1 : Création du fichier ex1_4.txt

Nous allons ensuite utiliser Gnuplot pour visualiser les points d'interpolation.

```
gnuplot
plot "ex1_4.txt"
```

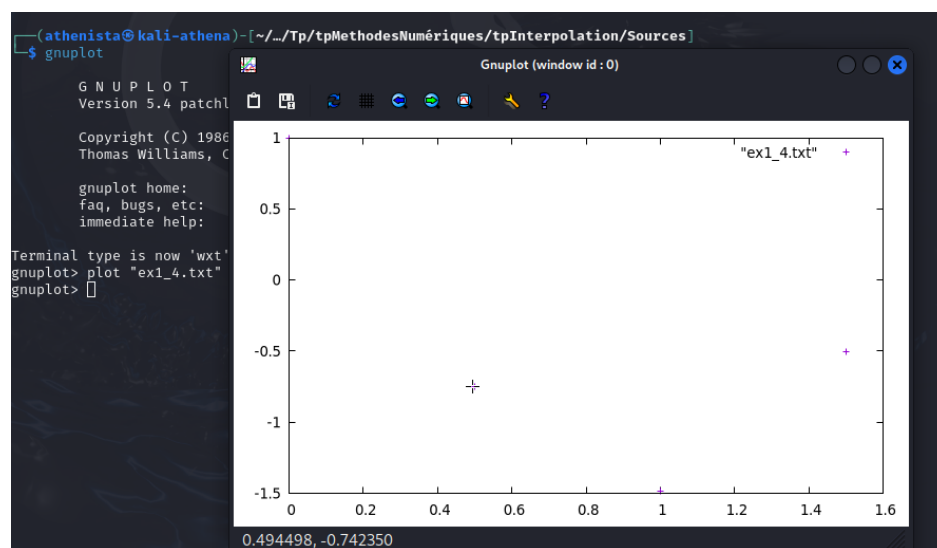


Figure 2 : Graphique du fichier ex1_4.txt

Nous pouvons améliorer le graphique en ajoutant with lines

```
gnuplot> plot "ex1_4.txt" with lines
```

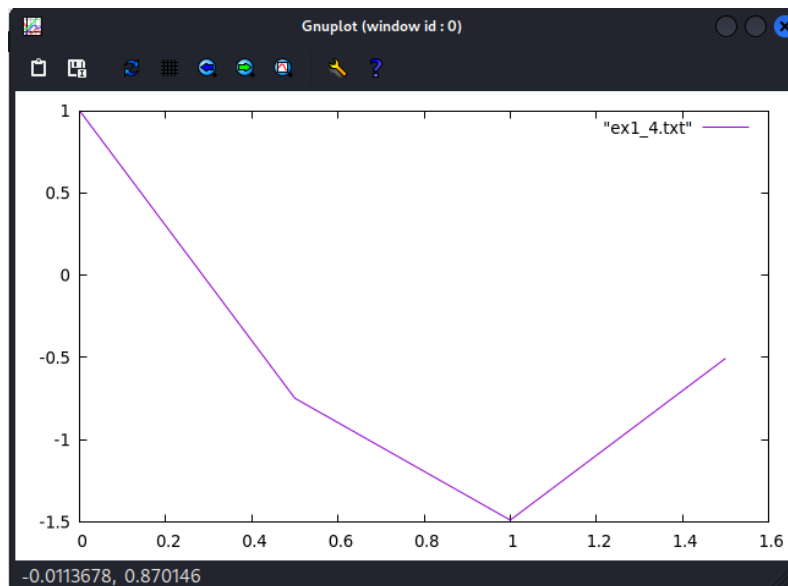


Figure 3 : Graphique du fichier ex1_4.txt avec les lignes

Nous allons désormais compiler le programme d'interpolation de Lagrange.

```
gcc -o inter_lag inter_lag.c -lm -Ofast
```

```
(athenista@kali-athena)-[~/.../Tp/tpMethodesNumériques/tpInterpolation/Sources]
$ gcc -o inter_lag inter_lag.c -lm -Ofast
```

Figure 4 : Compilation du fichier inter_lag

Nous allons ensuite pouvoir effectuer l'interpolation et générer le fichier binaire de résultats.

Avant de rajouter les différents paramètres au programme compilé, nous avons commencé par l'exécuter sans paramètres.

```
(athenista@kali-athena)-[~/.../Tp/tpMethodesNumériques/tpInterpolation/Sources]
$ ./inter_lag
Usage : ./inter_lag a b nb_pts_calcul fich_sortie fich_pts_construction
```

Figure 5 : Exécution du fichier inter_lag

Pour effectuer l'interpolation sur l'intervalle $[0, 1.5]$ avec 2000 points équi-répartis, nous allons utiliser les paramètres suivants :

```
./inter_lag 0 1.5 2000 int_lag_4_2000.bin ex1_4.bin
```

```
(athenista@kali-athena)-[~/.../Tp/tpMethodesNumériques/tpInterpolation/Sources]
$ ./inter_lag 0 1.5 2000 int_lag_4_2000.bin ex1_4.bin
Nombre de points de construction : 4
Nombre de points d'interpolation : 2000
```

Figure 6 : Création du fichier `int\_lag\_4\_2000.bin` contenant les résultats de l'interpolation.

Nous allons ensuite convertir le fichier binaire de résultats en fichier ASCII

```
./bin2ascii int_lag_4_2000.bin
```

```
(athenista@kali-athena)-[~/.../Tp/tpMethodesNumériques/tpInterpolation/Sources]
$ ./bin2ascii int_lag_4_2000.bin
nombre de couples lu: 2000
```

Figure 7 : Génération du binaire int_lag_4_2000.bin

Nous allons ensuite visualiser le résultat de l'interpolation avec Gnuplot

```
gnuplot> plot "int_lag_4_2000.txt"
```

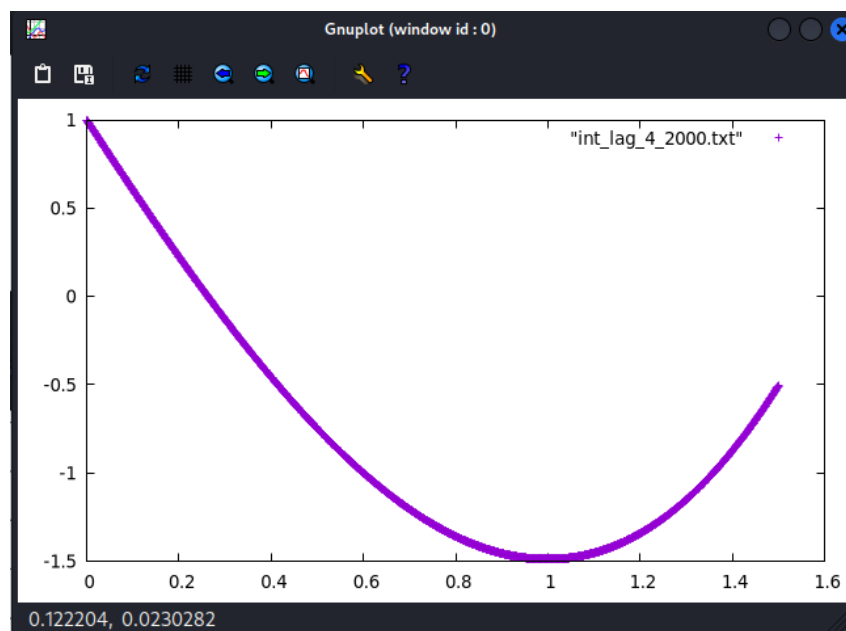


Figure 8 : Génération du graphique pour le fichier int_lag_4_2000.txt

```
gnuplot
```

```
plot "int_lag_4_2000.txt" with lines
```

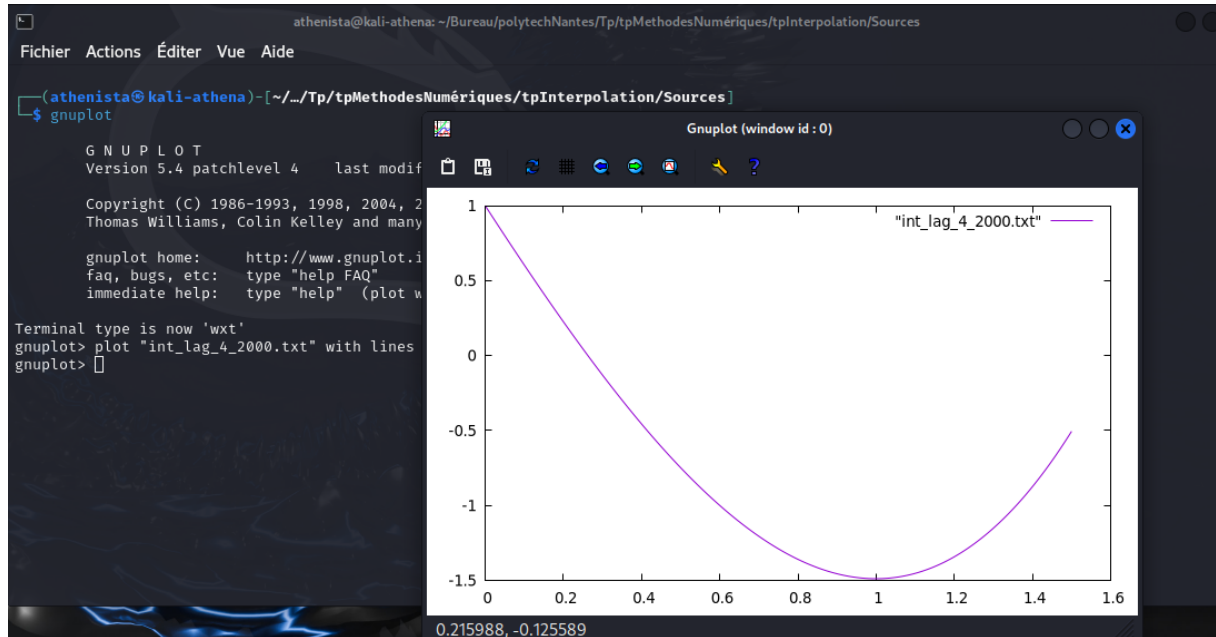


Figure 9 : Génération du graphique pour le fichier int_lag_4_2000.txt avec des lignes

Nous allons ensuite calculer et afficher l'erreur d'interpolation. Nous allons commencer par compiler le programme diffbin

```
gcc -o diffbin diffbin.c -lm -Ofast
```

```

(athenista@kali-athena)-[~/.../Tp/tpMethodesNumeriques/tpInterpolation/Sources]
$ gcc -o diffbin diffbin.c -lm -Ofast
  
```

Figure 10 : Compilation du fichier diffbin

Puis, nous allons utiliser diffbin pour obtenir l'erreur d'interpolation

```
./diffbin ex1_2000.bin int_lag_4_2000.bin diff_lag_4_2000.bin

./bin2ascii diff_lag_4_2000.bin
```

```

(root@kali-athena)-[/home/athenista/Bureau/polytechNantes/Tp/tpMethodesNumeriques/tpInterpolation/Sources]
# ./diffbin ex2_2000.bin int_lag_4_2000.bin diff_lag_4_2000.bin
Erreur : valeurs de x différentes entre ex2_2000.bin et int_lag_4_2000.bin

(root@kali-athena)-[/home/athenista/Bureau/polytechNantes/Tp/tpMethodesNumeriques/tpInterpolation/Sources]
# ./diffbin ex1_2000.bin int_lag_4_2000.bin diff_lag_4_2000.bin
./bin2ascii diff_lag_4_2000.bin
Erreur d'ouverture du fichier ex1_2000.bin
nombre de couples lu: 2000
Erreur de lecture de x
  
```

Figure 11 : Exécution du fichier diffbin pour visualiser l'erreur réelle d'interpolation

Petit problème, le fichier ex1_2000.bin n'existe pas, il faut le générer avec le code my_val_fonc.c fourni dans le dossier Sources pour ce TP.

```
gcc -o my_val_fonc my_val_fonc.c -lm -Ofast
```

```
./my_val_fonc 0 1.5 2000 ex1_2000.bin
```

```
(root@kali-athena)-[/home/athenista/Bureau/polytechNantes/Tp/tpMethodesNumeriques/tpInterpolation/Sources]
# gcc -o my_val_fonc my_val_fonc.c -lm -Ofast
./my_val_fonc 0 1.5 2000 ex1_2000.bin
```

Figure 12 : Création du fichier ex1_2000.bin

```
./bin2ascii ex1_2000.bin
```

```
(root@kali-athena)-[/home/athenista/Bureau/polytechNantes/Tp/tpMethodesNumeriques/tpInterpolation/Sources]
# ./bin2ascii ex1_2000.bin
nombre de couples lu: 2000
```

Figure 13 : Génération du fichier ex1_2000.txt

```
gnuplot
```

```
plot "ex1_2000.txt" with lines
```

```
(root@kali-athena)-[/home/athenista/Bureau/polytechNantes/Tp/tpMethodesNumeriques/tpInterpolation/Sources]
# ./bin2ascii ex1_2000.bin
nombre de couples lu: 2000

(root@kali-athena)-[/home/athenista/Bureau/polytechNantes/Tp/tpMethodesNumeriques/tpInterpolation/Sources]
# gnuplot

G N U P L O T
Version 5.4 patchlevel 4    last modified 2022-07-10

Copyright (c) 1986-1993, 1998, 2004, 2007-2022
Thomas Williams, Colin Kelley and many others

gnuplot home:      http://www.gnuplot.info
faq, bugs, etc:   type "help FAQ"
immediate help:   type "help" (plot window: hit 'h')

Terminal type is now 'wxt'
```

```
gnuplot> plot "ex1_2000.txt" with lines
```

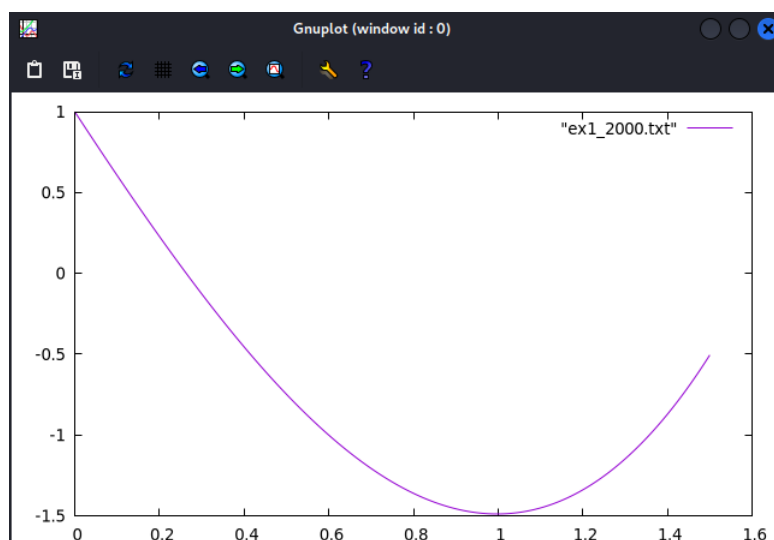


Figure 14 : Génération du fichier ex1_2000.txt

```
./diffbin ex1_2000.bin int_lag_4_2000.bin diff_lag_4_2000.bin  
  
./bin2ascii diff_lag_4_2000.bin
```

```
(root@kali-athena)-[/home/athenista/Bureau/polytechNantes/Tp/tpMethodesNumeriques/tpInterpolation/Sources]  
# ./diffbin ex1_2000.bin int_lag_4_2000.bin diff_lag_4_2000.bin  
./bin2ascii diff_lag_4_2000.bin  
nombre de couples lu: 2000
```

Figure 15 : Création du fichier diff_lag_4_2000.txt` contenant les erreurs d'interpolation.

```
gnuplot  
  
plot "diff_lag_4_2000.txt" w l
```

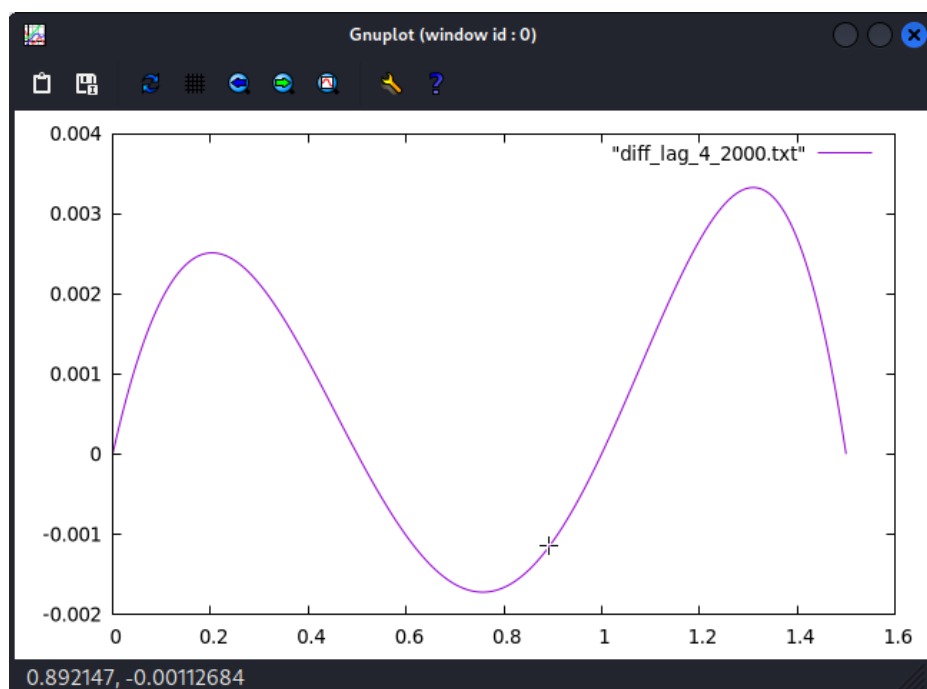


Figure 16 : Visualisation de l'erreur en valeur absolue avec Gnuplot

Nous allons commencer par générer le fichier ex1_10.bin pour effectuer l'interpolation sur 10 points

```
gcc -o my_val_fonc my_val_fonc.c -lm -Ofast  
  
./my_val_fonc 0 1.5 10 ex1_10.bin
```

```
(root@kali-athena)-[/home/athenista/Bureau/polytechNantes/Tp/tpMethodesNumeriques/tpInterpolation/Sources]  
# gcc -o my_val_fonc my_val_fonc.c -lm -Ofast  
./my_val_fonc 0 1.5 10 ex1_10.bin
```

Figure 17 : Compilation du fichier my_val_fonc.c

Nous allons désormais effectuer l'interpolation sur 10 points :

```
./inter_lag 0 1.5 2000 int_lag_10_2000.bin ex1_10.bin
```

```
./bin2ascii int_lag_10_2000.bin
```

```
(root@kali-athena)-[/home/athenista/Bureau/polytechNan
# ./inter_lag 0 1.5 2000 int_lag_10_2000.bin ex1_10.bin
./bin2ascii int_lag_10_2000.bin
Nombre de points de construction : 10
Nombre de points d'interpolation : 2000
nombre de couples lu: 2000
```

Figure 18 : Création de l'interpolation sur 10 points

```
gnuplot
```

```
plot "int_lag_10_2000.txt" with lines
```

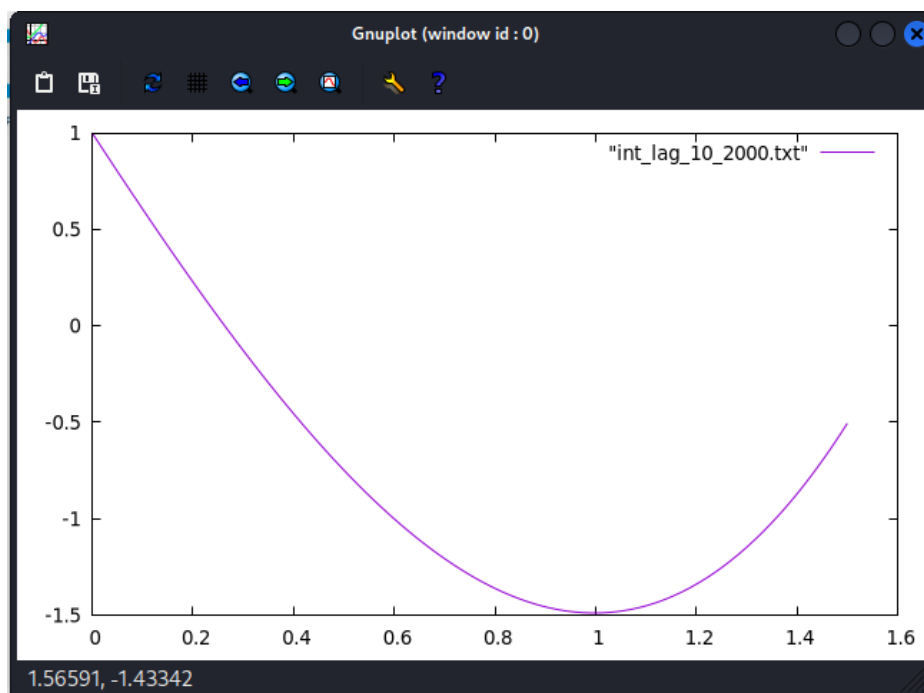


Figure 19 : Visualisation du résultat de l'interpolation sur 10 points

Pour comparer les erreurs

```
./diffbin ex1_2000.bin int_lag_10_2000.bin diff_lag_10_2000.bin
```

```
./bin2ascii diff_lag_10_2000.bin
```

```
(root@kali-athena)-[/home/athenista/Bureau/polytechNantes/Tp/tpMethodesNumeriques/tpInterpolation/Sources]
# ./diffbin ex1_2000.bin int_lag_10_2000.bin diff_lag_10_2000.bin
./bin2ascii diff_lag_10_2000.bin
nombre de couples lu: 2000
```

Figure 20 : Création du fichier pour générer l'erreur réelle d'interpolation pour les 2000 points de la courbe de $f(x) = e^x - 5 \sin(x)$

gnuplot

plot "diff_lag_10_2000.txt" with lines

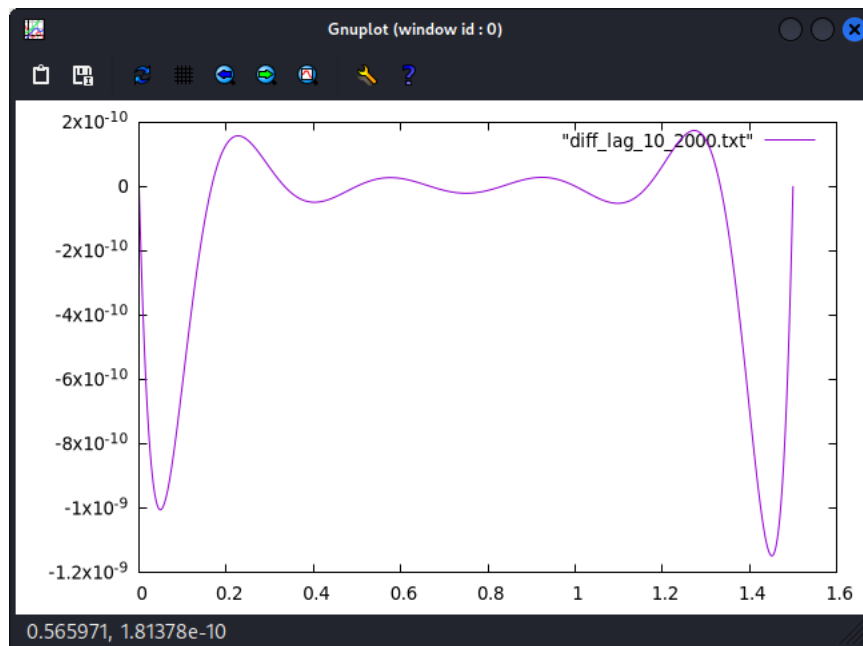


Figure 21 : Visualisation du fichier diff_lag_10_2000.txt

5) Utilisez `inter_fin_diff` permettant d'effectuer l'interpolation en utilisant les différences finies, afin de comparer les temps d'exécution des 2 programmes sur une interpolation sur 50 points évaluée en 100000 points à l'aide du fichier binaire `ex1_50.bin`.

Nous devons commencer par générer le fichier `ex1_50.bin`, cela se fait avec la commande

```
gcc -o inter_fin_diff inter_fin_diff.c -lm -Ofast
```

```
(root@kali-athena)-[/home/athenista/Bureau/polytechNantes/Tp/tpMethodesNumeriques/tpInterpolation/Sources]
# gcc -o inter_fin_diff inter_fin_diff.c -lm -Ofast
```

Figure 22 : Compilation du fichier `inter_fin_diff.c`

```
./my_val_fonc 0 1.5 50 ex1_50.bin
```

```
(root@kali-athena)-[/home/athenista]
# ./my_val_fonc 0 1.5 50 ex1_50.bin
```

Figure 22 : Création du fichier binaire d'une fonction sur une subdivision de points équirépartis

```
./inter_fin_diff 0 1.5 100000 int_fin_diff_50_100000.bin ex1_50.bin
```

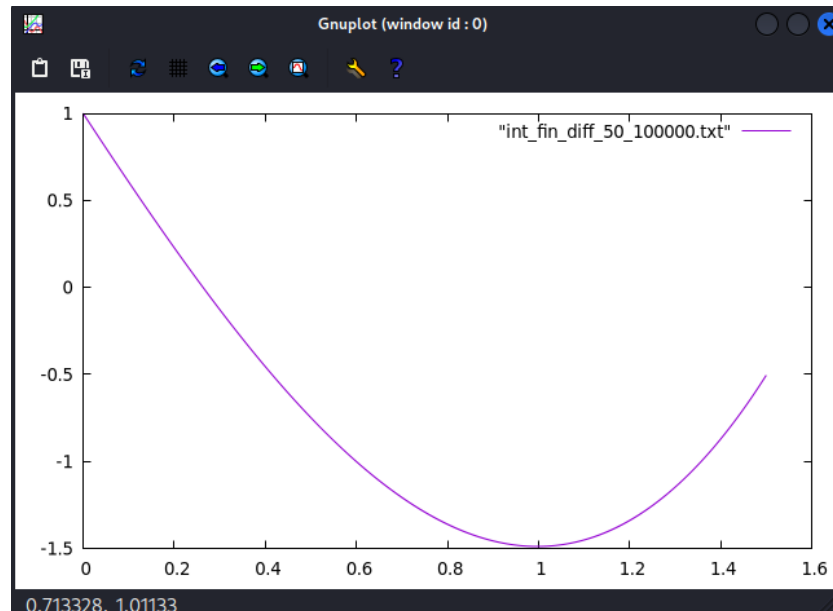
```
(root@kali-athena)-[/home/athenista/Bureau/polytechNantes/Tp/tpMethode]
# ./inter_fin_diff 0 1.5 100000 int_fin_diff_50_100000.bin ex1_50.bin
Nombre de points de construction : 50
Nombre de points d'interpolation : 100000
```

```
./bin2ascii int_fin_diff_50_100000.bin
```

```
(root@kali-athena)-[/home/athenista/Bureau]
# ./bin2ascii int_fin_diff_50_100000.bin
nombre de couples lu: 100000
```

gnuplot

```
plot "int_fin_diff_50_100000.txt" with lines
```



Nous allons désormais mesurer le temps d'exécution des deux programmes (inter_lag et inter_fin_diff) pour 50 points évalués en 100000 points, puis comparez les résultats.

```
time ./inter_lag 0 1.5 100000 int_lag_50_100000.bin ex1_50.bin
```

```
(root@kali-athena)-[/home/athenista/Bureau/polytechNantes/Tp/tpMethodes]
# time ./inter_lag 0 1.5 100000 int_lag_50_100000.bin ex1_50.bin
Nombre de points de construction : 50
Nombre de points d'interpolation : 100000

real    0m0,301s
user    0m0,297s
sys     0m0,004s
```

```
time ./inter_fin_diff 0 1.5 100000 int_fin_diff_50_100000.bin ex1_50.bin
```

```
(root@kali-athena)-[/home/athenista/Bureau/polytechNantes/Tp/tpMethodes]
# time ./inter_fin_diff 0 1.5 100000 int_
Nombre de points de construction : 50
Nombre de points d'interpolation : 100000

real    0m0,029s
user    0m0,025s
sys     0m0,004s
```

Les mesures des temps d'exécution montrent que l'interpolation par différences finies est nettement plus rapide que l'interpolation de Lagrange. L'interpolation de Lagrange a pris 0,301 secondes, tandis que les différences finies ont pris seulement 0,029 secondes. Cette différence s'explique par l'efficacité algorithmique des différences finies, qui minimisent les opérations en utilisant des relations de récurrence, ayant une complexité de $O(n^2)$ contre $O(n^3)$ pour Lagrange. Pour des applications nécessitant une interpolation rapide sur un grand nombre de points, il est recommandé d'utiliser les différences finies. En conclusion, cette méthode offre une performance supérieure tout en maintenant la précision nécessaire pour des calculs intensifs.

6) Utilisez les formules du cours pour estimer le nombre d'opérations nécessaires pour chacun des deux programmes (interpolation de Lagrange et interpolation par différences finies) et confirmez le ratio entre les temps d'exécutions mesurés précédemment.

En se basant sur le cours,

Interpolation de Lagrange

Le nombre d'additions et de multiplications nécessaires pour une interpolation de Lagrange est donné par les formules suivantes :

- Nombre d'additions, $N_{L,a}(m) : m \cdot n \cdot (2n + 3)$
- Nombre de multiplications, $N_{L,m}(m) : 2 \cdot m \cdot n \cdot (n + 1)$

Pour $n = 50$ et $m = 100000$:

$$N_{L,a}(100000) = 100000 \cdot 50 \cdot (2 \cdot 50 + 3) = 100000 \cdot 50 \cdot 103 = 515000000$$

$$N_{L,m}(100000) = 2 \cdot 100000 \cdot 50 \cdot 51 = 510000000$$

Interpolation par différences finies

Le nombre d'additions et de multiplications nécessaires pour une interpolation par différences finies est donné par les formules suivantes :

- Nombre d'additions, $N_{D,a}(m) : n \cdot (n + 1) + m \cdot n$
- Nombre de multiplications, $N_{D,m}(m) : n \cdot (n + 1)/2 + m \cdot n$

Pour $n = 50$ et $m = 100000$:

$$N_{D,a}(100000) = 50 \cdot (50 + 1) + 100000 \cdot 50 = 50 \cdot 51 + 5000000 = 2550 + 5000000 = 5002550$$

$$N_{D,m}(100000) = 50 \cdot (50 + 1)/2 + 100000 \cdot 50 = 50 \cdot 25.5 + 5000000 = 1275 + 5000000 = 5001275$$

Le programme d'interpolation par différences finies est beaucoup plus efficace en termes de complexité temporelle par rapport au programme d'interpolation de Lagrange. Les résultats expérimentaux de temps d'exécution confirment ces estimations : l'interpolation de Lagrange a pris 0.301 secondes, tandis que l'interpolation par différences finies n'a pris que 0.029 secondes, ce qui correspond à la différence de complexité entre $O(n^2)$ et $O(n^3)$.

Interpolation spline

4.1. Splines cubiques naturelles

Générez les fichiers binaires correspondants à des subdivisions comportant $n=3, 5, 9, 17, 33$ et 65 points équirépartis sur l'intervalle $[a = 0, b = 1]$, ainsi qu'un fichier pour une subdivision de $n=2000$ points sur le même intervalle à des fins d'affichage par Gnuplot.

Nous allons commencer par modifier le fichier `my_val_fonc.c` pour travailler avec la fonction $g(x) = \sin(2\pi x)$.

Nous commençons donc par utiliser le `#define DEUXPI (8 * atan(1))` et pour le double fonction(double x) nous modifions le return en `sin(2 * DEUXPI * x)`;

```
10 #include <stdio.h>
11 #include <stdlib.h>
12 #include <math.h>
13 #define DEUXPI (8 * atan(1))
14
15 double fonction(double);
16
```

```
58 double fonction(double x) {
59     double result;
60
61     //result = exp(x)-5*sin(x);
62     return sin(2 * DEUXPI * x);
63
64     return(result);
65 }
66
```

`gcc -o my_val_fonc my_val_fonc.c -lm -Ofast`

```
./my_val_fonc 0 1 3 sin_3.bin

./my_val_fonc 0 1 5 sin_5.bin

./my_val_fonc 0 1 17 sin_17.bin

./my_val_fonc 0 1 33 sin_33.bin

./my_val_fonc 0 1 65 sin_65.bin

./my_val_fonc 0 1 2000 sin_2000.bin
```

```
(root@kali-athena)-[/home/athenista/Bureau/polytechNantes/Tp/tpMethodesNumériques/tpInterpolation/Sources]
# ./my_val_fonc 0 1 3 sin_3.bin

(root@kali-athena)-[/home/athenista/Bureau/polytechNantes/Tp/tpMethodesNumériques/tpInterpolation/Sources]
# ./my_val_fonc 0 1 5 sin_5.bin

(root@kali-athena)-[/home/athenista/Bureau/polytechNantes/Tp/tpMethodesNumériques/tpInterpolation/Sources]
# ./my_val_fonc 0 1 17 sin_17.bin

(root@kali-athena)-[/home/athenista/Bureau/polytechNantes/Tp/tpMethodesNumériques/tpInterpolation/Sources]
# ./my_val_fonc 0 1 33 sin_33.bin

(root@kali-athena)-[/home/athenista/Bureau/polytechNantes/Tp/tpMethodesNumériques/tpInterpolation/Sources]
# ./my_val_fonc 0 1 65 sin_65.bin

(root@kali-athena)-[/home/athenista/Bureau/polytechNantes/Tp/tpMethodesNumériques/tpInterpolation/Sources]
# ./my_val_fonc 0 1 2000 sin_2000.bin
```

2)

En utilisant spline_nat puis diffabsbin, visualisez l'erreur d'interpolation obtenue pour les 6 subdivisions considérées.

```
gcc -o spline_nat spline_nat.c -lm -Ofast
```

```
gcc -o diffabsbin diffabsbin.c -lm -Ofast
```

```
# Pour n = 3
```

```
./spline_nat 0 1 2000 int_spl_n_sin_3_2000.bin < sin_3.bin
```

```
./diffabsbin sin_2000.bin int_spl_n_sin_3_2000.bin diff_spl_n_sin_3.bin
```

```
./bin2ascii diff_spl_n_sin_3.bin
```

```
(root@kali-athena)-[/home/athenista/Bureau/polytechNantes/Tp/tpMethodesNumériques/tpInterpolation/Sources]
# ./spline_nat 0 1 2000 int_spl_n_sin_3_2000.bin < sin_3.bin

(root@kali-athena)-[/home/athenista/Bureau/polytechNantes/Tp/tpMethodesNumériques/tpInterpolation/Sources]
# ./diffabsbin sin_2000.bin int_spl_n_sin_3_2000.bin diff_spl_n_sin_3.bin
./bin2ascii diff_spl_n_sin_3.bin
nombre de couples lu: 2000
```

```
awk '{print $2}' diff_spl_n_sin_3.txt | sort -n | tail -1
```

Erreur maximale pour n = 3: 9.999996912662e-01

```
(root@kali-athena)-[/home/athenista/Bureau/polytechNantes/Tp/tpMethodes]
# awk '{print $2}' diff_spl_n_sin_3.txt | sort -n | tail -1
# Erreur maximale pour n = 3: 9.999996912662e-01
9.999996912662e-01
```

Pour n = 5

```
./spline_nat 0 1 2000 int_spl_n_sin_5_2000.bin < sin_5.bin
```

```
./diffabsbin sin_2000.bin int_spl_n_sin_5_2000.bin diff_spl_n_sin_5.bin
```

```
./bin2ascii diff_spl_n_sin_5.bin
```

```
(root@kali-athena)-[/home/athenista/Bureau/polytechNantes/Tp/tpMethodes]
# ./spline_nat 0 1 2000 int_spl_n_sin_5_2000.bin < sin_5.bin
./diffabsbin sin_2000.bin int_spl_n_sin_5_2000.bin diff_spl_n_sin_5.bin
./bin2ascii diff_spl_n_sin_5.bin
nombre de couples lu: 2000
```

```
awk '{print $2}' diff_spl_n_sin_5.txt | sort -n | tail -1
```

Erreur maximale pour n = 5: 9.999996912662e-01

```
(root@kali-athena)-[/home/athenista/Bureau/polytechNantes/Tp/tpMethodes]
# awk '{print $2}' diff_spl_n_sin_5.txt | sort -n | tail -1
# Erreur maximale pour n = 5: 9.999996912662e-01
9.999996912662e-01
```

Pour n = 17

```
./spline_nat 0 1 2000 int_spl_n_sin_17_2000.bin < sin_17.bin
```

```
./diffabsbin sin_2000.bin int_spl_n_sin_17_2000.bin diff_spl_n_sin_17.bin
```

```
./bin2ascii diff_spl_n_sin_17.bin
```

```
(root@kali-athena)-[/home/athenista/Bureau/polytechNantes/Tp/tpMethodes]
# ./spline_nat 0 1 2000 int_spl_n_sin_17_2000.bin < sin_17.bin
./diffabsbin sin_2000.bin int_spl_n_sin_17_2000.bin diff_spl_n_sin_17.bin
./bin2ascii diff_spl_n_sin_17.bin
nombre de couples lu: 2000
```

```
awk '{print $2}' diff_spl_n_sin_17.txt | sort -n | tail -1
```

Erreur maximale pour n = 17: 9.987323266394e-04

```
(root@kali-athena)-[/home/athenista/Bureau/polytechNantes/Tp/
# awk '{print $2}' diff_spl_n_sin_17.txt | sort -n | tail -1
# Erreur maximale pour n = 17: 9.987323266394e-04
9.987323266394e-04
```

Pour n = 33

```
./spline_nat 0 1 2000 int_spl_n_sin_33_2000.bin < sin_33.bin
```

```
./diffabsbin sin_2000.bin int_spl_n_sin_33_2000.bin diff_spl_n_sin_33.bin
```

```
./bin2ascii diff_spl_n_sin_33.bin
```

```
(root@kali-athena)-[/home/athenista/Bureau/polytechNantes/Tp/tpMethodes
# ./spline_nat 0 1 2000 int_spl_n_sin_33_2000.bin < sin_33.bin
./diffabsbin sin_2000.bin int_spl_n_sin_33_2000.bin diff_spl_n_sin_33.bin
./bin2ascii diff_spl_n_sin_33.bin
nombre de couples lu: 2000
```

```
awk '{print $2}' diff_spl_n_sin_33.txt | sort -n | tail -1
```

Erreur maximale pour n = 33: 9.997816941429e-06

```
(root@kali-athena)-[/home/athenista/Bureau/polytechNantes/Tp
# awk '{print $2}' diff_spl_n_sin_33.txt | sort -n | tail -1
# Erreur maximale pour n = 33: 9.997816941429e-06
9.997816941429e-06
```

Pour n = 65

```
./spline_nat 0 1 2000 int_spl_n_sin_65_2000.bin < sin_65.bin
```

```
./diffabsbin sin_2000.bin int_spl_n_sin_65_2000.bin diff_spl_n_sin_65.bin
```

```
./bin2ascii diff_spl_n_sin_65.bin
```

```
(root@kali-athena)-[/home/athenista/Bureau/polytechNantes/Tp/tpMethodes
# ./spline_nat 0 1 2000 int_spl_n_sin_65_2000.bin < sin_65.bin
./diffabsbin sin_2000.bin int_spl_n_sin_65_2000.bin diff_spl_n_sin_65.bin
./bin2ascii diff_spl_n_sin_65.bin
nombre de couples lu: 2000
```

```
awk '{print $2}' diff_spl_n_sin_65.txt | sort -n | tail -1
```

Erreur maximale pour n = 65: 9.963040976668e-07

```
(root@kali-athena)-[/home/athenista/Bureau/polytechNantes/Tp
# awk '{print $2}' diff_spl_n_sin_65.txt | sort -n | tail -1
# Erreur maximale pour n = 65: 9.963040976668e-07
9.963040976668e-07
```

Pour n = 2000

./spline_nat 0 1 2000 int_spl_n_sin_2000_2000.bin < sin_2000.bin

./diffabsbin sin_2000.bin int_spl_n_sin_2000_2000.bin diff_spl_n_sin_2000.bin

./bin2ascii diff_spl_n_sin_2000.bin

```
(root@kali-athena)-[/home/athenista/Bureau/polytechNantes/Tp/tpMethodesNumeriques]
# # Pour n = 2000
./spline_nat 0 1 2000 int_spl_n_sin_2000_2000.bin < sin_2000.bin
./diffabsbin sin_2000.bin int_spl_n_sin_2000_2000.bin diff_spl_n_sin_2000.bin
./bin2ascii diff_spl_n_sin_2000.bin
nombre de couples lu: 2000
```

awk '{print \$2}' diff_spl_n_sin_2000.txt | sort -n | tail -1

Erreur maximale pour n = 2000: 0.000000000000e+00

```
(root@kali-athena)-[/home/athenista/Bureau/polytechNantes/Tp/tpMethodesNumeriques]
# awk '{print $2}' diff_spl_n_sin_2000.txt | sort -n | tail -1
0.000000000000e+00
```

Une fois les erreurs max obtenues, nous allons créer le fichier table.txt contenant les couples (h, erreur_max) :

echo "h erreur_max" > table.txt

echo "0.5 9.999996912662e-01" >> table.txt # n=3

echo "0.25 9.999996912662e-01" >> table.txt # n=5

echo "0.125 9.999996912662e-01" >> table.txt # n=9

echo "0.0625 9.987323266394e-04" >> table.txt # n=17

echo "0.03125 9.997816941429e-06" >> table.txt # n=33

echo "0.015625 9.963040976668e-07" >> table.txt # n=65

echo "0.00050025 0.000000000000e+00" >> table.txt # n=2000

Passons aux commandes gnuplot

gnuplot

$$f(x) = a * x^{**4}$$

fit f(x) "table.txt" using 1:2 via a

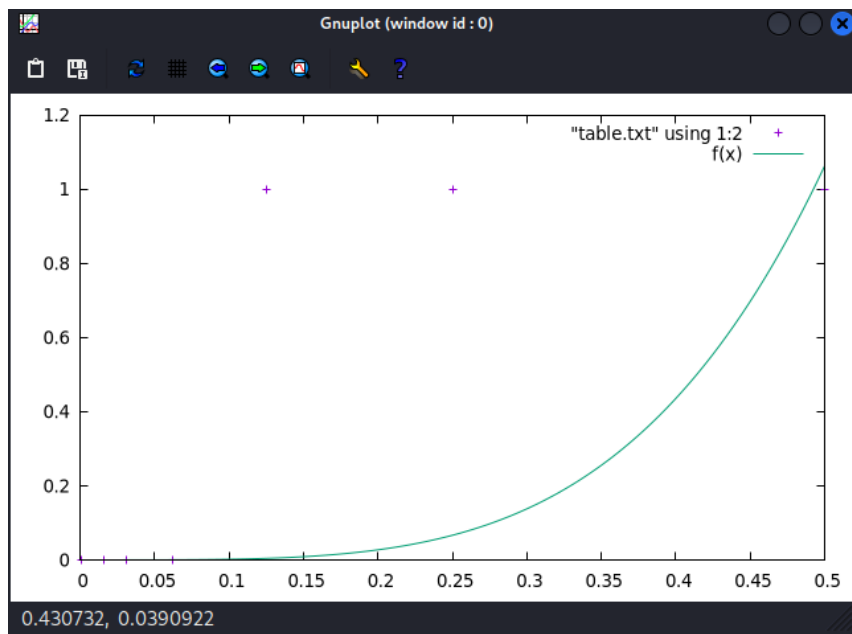
plot "table.txt" using 1:2, f(x)

```
Terminal type is now 'wxt'
gnuplot> f(x) = a * x**4
gnuplot> fit f(x) "table.txt" using 1:2 via a
iter   chisq      delta/lim  lambda  a
  0  2.8706199432e+00   0.00e+00  2.37e-02  1.000000e+00
  1  1.8828974457e+00  -5.25e+04  2.37e-03  1.499637e+01
  2  1.8672193427e+00  -8.40e+02  2.37e-04  1.699300e+01
  3  1.8672193108e+00  -1.71e-03  2.37e-05  1.699585e+01
iter   chisq      delta/lim  lambda  a

After 3 iterations the fit converged.
final sum of squares of residuals : 1.86722
rel. change during last iteration : -1.70869e-08

degrees of freedom (FIT_NDF) : 6
rms of residuals (FIT_STDFIT) = sqrt(WSSR/ndf) : 0.557856
variance of residuals (reduced chisquare) = WSSR/ndf : 0.311203

Final set of parameters          Asymptotic Standard Error
-----
a = 16.9958                      +/- 8.908 (52.41%)
gnuplot> plot "table.txt" using 1:2, f(x)
```



3) On travaille maintenant sur la fonction $f(x) = e^x - 5 \sin(x)$ sur l'intervalle $[0, 1.5]$. Générez les fichiers binaires correspondant aux subdivisions comportant 3, 5, 9, 17, 33 et 65 points, équirépartis sur l'intervalle $[0, 1.5]$. Remarque 4 (ex1_2000.bin). Le fichier ex1_2000.bin comportant 2000 points existe déjà puisque vous l'avez utilisé pour l'interpolation polynomiale.

Nous commençons par changer la fonction double fonction(double x) pour qu'elle retourne $\exp(x) - 5 \sin(x)$.

```
58 double fonction(double x) {
59
60     double result;
61
62     result = exp(x)-5*sin(x);
63     //return sin(2 * DEUXPI * x);
64
65     return(result);
66 }
```

Nous allons ensuite compiler le fichier my_val_fonc.c :

```
gcc -o my_val_fonc my_val_fonc.c -lm -Ofast
```

Générer les fichiers binaires pour la nouvelle subdivision :

```
./my_val_fonc 0 1.5 3 ex1_3.bin
```

```
./my_val_fonc 0 1.5 5 ex1_5.bin
```

```
./my_val_fonc 0 1.5 9 ex1_9.bin
```

```
./my_val_fonc 0 1.5 17 ex1_17.bin
```

```
./my_val_fonc 0 1.5 33 ex1_33.bin
```

```
./my_val_fonc 0 1.5 65 ex1_65.bin
```

```
./my_val_fonc 0 1.5 2000 ex1_2000.bin
```

```
(root@kali-athena)-[/home/athenista/Bureau]
# ./my_val_fonc 0 1.5 3 ex1_3.bin
./my_val_fonc 0 1.5 5 ex1_5.bin
./my_val_fonc 0 1.5 9 ex1_9.bin
./my_val_fonc 0 1.5 17 ex1_17.bin
./my_val_fonc 0 1.5 33 ex1_33.bin
./my_val_fonc 0 1.5 65 ex1_65.bin
./my_val_fonc 0 1.5 2000 ex1_2000.bin
```

Nous pouvons ensuite compiler les fichiers spline_nat et diffabsbin

```
gcc -o spline_nat spline_nat.c -lm -Ofast
```

```
gcc -o diffabsbin diffabsbin.c -lm -Ofast
```

```
(root@kali-athena)-[/home/athenista/Bureau]
# gcc -o spline_nat spline_nat.c -lm -Ofast
gcc -o diffabsbin diffabsbin.c -lm -Ofast
```

4) En utilisant spline_nat puis diffabsbin, visualisez l'erreur d'interpolation obtenue pour les 6 subdivisions considérées. Montrez que l'erreur maximale d'interpolation sur $[0, 1.5]$ croît comme le carré du pas h .

#pour n=3

```
./spline_nat 0 1.5 2000 int_spl_n_ex1_3_2000.bin < ex1_3.bin
```

```
./diffabsbin ex1_2000.bin int_spl_n_ex1_3_2000.bin diff_spl_n_ex1_3.bin
```

```
./bin2ascii diff_spl_n_ex1_3.bin
```

```
awk '{print $2}' diff_spl_n_ex1_3.txt | sort -n | tail -1
```

Erreur maximale pour n = 3 : 9.999996912662e-01

```
(root@kali-athena)-[/home/athenista/Bureau/polytechNantes/Tp/tpMethodes]
# #pour n=3
./spline_nat 0 1.5 2000 int_spl_n_ex1_3_2000.bin < ex1_3.bin
./diffabsbin ex1_2000.bin int_spl_n_ex1_3_2000.bin diff_spl_n_ex1_3.bin
./bin2ascii diff_spl_n_ex1_3.bin
awk '{print $2}' diff_spl_n_ex1_3.txt | sort -n | tail -1
nombre de couples lu: 2000
9.999996912662e-01
```

#pour n=5

```
./spline_nat 0 1.5 2000 int_spl_n_ex1_5_2000.bin < ex1_5.bin
```

```
./diffabsbin ex1_2000.bin int_spl_n_ex1_5_2000.bin diff_spl_n_ex1_5.bin
```

```
./bin2ascii diff_spl_n_ex1_5.bin
```

```
awk '{print $2}' diff_spl_n_ex1_5.txt | sort -n | tail -1
```

#Erreur maximale pour n = 5 : 9.991953329958e-01

```
(root@kali-athena)-[/home/athenista/Bureau/polytechNantes/Tp/tpMethodes]
# #pour n=5
./spline_nat 0 1.5 2000 int_spl_n_ex1_5_2000.bin < ex1_5.bin
./diffabsbin ex1_2000.bin int_spl_n_ex1_5_2000.bin diff_spl_n_ex1_5.bin
./bin2ascii diff_spl_n_ex1_5.bin
awk '{print $2}' diff_spl_n_ex1_5.txt | sort -n | tail -1
nombre de couples lu: 2000
9.991953329958e-01
```

#pour n=9

```
./spline_nat 0 1.5 2000 int_spl_n_ex1_9_2000.bin < ex1_9.bin
```

```
./diffabsbin ex1_2000.bin int_spl_n_ex1_9_2000.bin diff_spl_n_ex1_9.bin
```

./bin2ascii diff_spl_n_ex1_9.bin

awk '{print \$2}' diff_spl_n_ex1_9.txt | sort -n | tail -1

Erreur maximale pour n = 9 : 9.982207812719e-02

```
(root@kali-athena)-[/home/athenista/Bureau/polytechNantes/Tp/tpMethodes]
# #pour n=9
./spline_nat 0 1.5 2000 int_spl_n_ex1_9_2000.bin < ex1_9.bin
./diffabsbin ex1_2000.bin int_spl_n_ex1_9_2000.bin diff_spl_n_ex1_9.bin
./bin2ascii diff_spl_n_ex1_9.bin
awk '{print $2}' diff_spl_n_ex1_9.txt | sort -n | tail -1
nombre de couples lu: 2000
9.982207812719e-02
```

#pour n=17

./spline_nat 0 1.5 2000 int_spl_n_ex1_17_2000.bin < ex1_17.bin

./diffabsbin ex1_2000.bin int_spl_n_ex1_17_2000.bin diff_spl_n_ex1_17.bin

./bin2ascii diff_spl_n_ex1_17.bin

awk '{print \$2}' diff_spl_n_ex1_17.txt | sort -n | tail -1

Erreur maximale pour n = 17 : 9.999037075610e-04

```
(root@kali-athena)-[/home/athenista/Bureau/polytechNantes/Tp/tpMethodesNun]
# #pour n=17
./spline_nat 0 1.5 2000 int_spl_n_ex1_17_2000.bin < ex1_17.bin
./diffabsbin ex1_2000.bin int_spl_n_ex1_17_2000.bin diff_spl_n_ex1_17.bin
./bin2ascii diff_spl_n_ex1_17.bin
awk '{print $2}' diff_spl_n_ex1_17.txt | sort -n | tail -1
nombre de couples lu: 2000
9.999037075610e-04
```

#pour n=33

./spline_nat 0 1.5 2000 int_spl_n_ex1_33_2000.bin < ex1_33.bin

./diffabsbin ex1_2000.bin int_spl_n_ex1_33_2000.bin diff_spl_n_ex1_33.bin

./bin2ascii diff_spl_n_ex1_33.bin

awk '{print \$2}' diff_spl_n_ex1_33.txt | sort -n | tail -1

#Erreur maximale pour n = 33: 9.999102371372e-06

```
(root@kali-athena)-[/home/athenista/Bureau/polytechNantes/Tp/tpMethode]
# #pour n=33
./spline_nat 0 1.5 2000 int_spl_n_ex1_33_2000.bin < ex1_33.bin
./diffabsbin ex1_2000.bin int_spl_n_ex1_33_2000.bin diff_spl_n_ex1_33.bin
./bin2ascii diff_spl_n_ex1_33.bin
awk '{print $2}' diff_spl_n_ex1_33.txt | sort -n | tail -1
# Notez l'erreur maximale pour n = 33
nombre de couples lu: 2000
9.999102371372e-06
```

#pour n=65

```
./spline_nat 0 1.5 2000 int_spl_n_ex1_65_2000.bin < ex1_65.bin
```

```
./diffabsbin ex1_2000.bin int_spl_n_ex1_65_2000.bin diff_spl_n_ex1_65.bin
```

```
./bin2ascii diff_spl_n_ex1_65.bin
```

```
awk '{print $2}' diff_spl_n_ex1_65.txt | sort -n | tail -1
```

```
# Erreur maximale pour n = 65 : 9.981247120372e-06
```

```
(root@kali-athena)-[/home/athenista/Bureau/polytechNantes/Tp/tpMethode]
# #pour n=65
./spline_nat 0 1.5 2000 int_spl_n_ex1_65_2000.bin < ex1_65.bin
./diffabsbin ex1_2000.bin int_spl_n_ex1_65_2000.bin diff_spl_n_ex1_65.bin
./bin2ascii diff_spl_n_ex1_65.bin
awk '{print $2}' diff_spl_n_ex1_65.txt | sort -n | tail -1
# Notez l'erreur maximale pour n = 65
nombre de couples lu: 2000
9.981247120372e-06
```

#pour n=2000

```
./spline_nat 0 1.5 2000 int_spl_n_ex1_2000_2000.bin < ex1_2000.bin
```

```
./diffabsbin ex1_2000.bin int_spl_n_ex1_2000_2000.bin diff_spl_n_ex1_2000.bin
```

```
./bin2ascii diff_spl_n_ex1_2000.bin
```

```
awk '{print $2}' diff_spl_n_ex1_2000.txt | sort -n | tail -1
```

```
# Erreur maximale pour n = 2000: 0.000000000000e+00
```

```
(root@kali-athena)-[/home/athenista/Bureau/polytechNantes/Tp/tpMethodesNumé]
# #pour n=2000
./spline_nat 0 1.5 2000 int_spl_n_ex1_2000_2000.bin < ex1_2000.bin
./diffabsbin ex1_2000.bin int_spl_n_ex1_2000_2000.bin diff_spl_n_ex1_2000.bin
./bin2ascii diff_spl_n_ex1_2000.bin
awk '{print $2}' diff_spl_n_ex1_2000.txt | sort -n | tail -1
nombre de couples lu: 2000
0.000000000000e+00
```

Pour compléter cette analyse, nous allons créer le fichier table_ex1.txt avec les couples (h, erreur_max) et valider le modèle en h sous Gnuplot.

```
echo "h erreur_max" > table_ex1.txt

echo "0.5 9.999996912662e-01" >> table_ex1.txt # n=3

echo "0.25 9.991953329958e-01" >> table_ex1.txt # n=5

echo "0.125 9.982207812719e-02" >> table_ex1.txt # n=9

echo "0.0625 9.999037075610e-04" >> table_ex1.txt # n=17

echo "0.03125 9.999102371372e-06" >> table_ex1.txt # n=33

echo "0.015625 9.981247120372e-06" >> table_ex1.txt # n=65

echo "0.00050025 0.000000000000e+00" >> table_ex1.txt # n=2000
```

2. Valider le modèle en h sous Gnuplot

Ensuite, vous allez utiliser Gnuplot pour valider le modèle en h. Ouvrez Gnuplot et entrez les commandes suivantes :

```
gnuplot

f(x) = a*x**2

fit f(x) "table_ex1.txt" using 1:2 via a

plot "table_ex1.txt" using 1:2, f(x)
```

```
(root@kali-athena)-[/home/athenista/Bureau/polytechNantes/Tp/tpMethodesNumériques/tpInterpolation/Sources]
# gnuplot

G N U P L O T
Version 5.4 patchlevel 4    last modified 2022-07-10

Copyright (C) 1986-1993, 1998, 2004, 2007-2022
Thomas Williams, Colin Kelley and many others

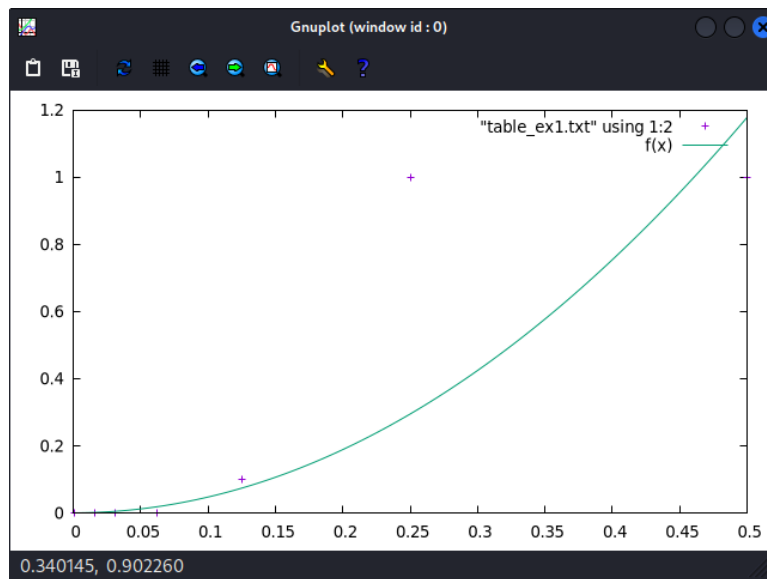
gnuplot home:      http://www.gnuplot.info
faq, bugs, etc:    type "help FAQ"
immediate help:    type "help" (plot window: hit 'h')

Terminal type is now 'wxt'
gnuplot> f(x) = a*x**2
gnuplot> fit f(x) "table_ex1.txt" using 1:2 via a
iter   chisq      delta/lim  lambda   a
  0  1.4469962676e+00   0.00e+00  9.76e-02  1.000000e+00
  1  5.4363020079e-01  -1.66e+05  9.76e-03  4.246424e+00
  2  5.2929108605e-01  -2.71e+03  9.76e-04  4.709538e+00
  3  5.2929105687e-01  -5.51e-03  9.76e-05  4.710199e+00
iter   chisq      delta/lim  lambda   a

After 3 iterations the fit converged.
final sum of squares of residuals : 0.529291
rel. change during last iteration : -5.51306e-08

degrees of freedom (FIT_NDF)          : 6
rms of residuals (FIT_STDFIT) = sqrt(WSSR/ndf) : 0.29701
variance of residuals (reduced chisquare) = WSSR/ndf : 0.0882152

Final set of parameters          Asymptotic Standard Error
=====
a = 4.7102          +/- 1.15          (24.42%)
gnuplot> plot "table_ex1.txt" using 1:2, f(x)
```



Splines cubiques modifiées

Le source `spline_mod.c` permet d'effectuer une interpolation par des splines cubiques modifiées, c'est à dire utilisant certains points aux extrémités de l'intervalle pour estimer la courbure de la fonction aux extrémités.

En utilisant les fichiers binaires déjà présents pour $f(x)$ sur l'intervalle $[0, 1.5]$, montrez que l'erreur maximale d'interpolation par splines cubiques modifiées varie comme le cube de h .

Compiler `spline_mod` et `diffabsbin`

Commencez par compiler le fichier source `spline_mod.c`.

```
sh
```

```
gcc -o spline_mod spline_mod.c -lm -Ofast
```

```
gcc -o diffabsbin diffabsbin.c -lm -Ofast
```

Interpolation et calcul de l'erreur

#Pour $n = 3$

```
./spline_mod 0 1.5 2000 int_spl_m_ex1_3_2000.bin < ex1_3.bin
```

```
./diffabsbin ex1_2000.bin int_spl_m_ex1_3_2000.bin diff_spl_m_ex1_3.bin
```

```
./bin2ascii diff_spl_m_ex1_3.bin
```

```
awk '{print $2}' diff_spl_m_ex1_3.txt | sort -n | tail -1
```

Erreur maximale pour $n = 3$: 9.999996912662e-01

```
(root@kali-athena)-[/home/athenista/Bureau/polytechNantes/Tp/tpMethodes]
# #Pour n = 3
./spline_mod 0 1.5 2000 int_spl_m_ex1_3_2000.bin < ex1_3.bin
./diffabsbin ex1_2000.bin int_spl_m_ex1_3_2000.bin diff_spl_m_ex1_3.bin
./bin2ascii diff_spl_m_ex1_3.bin
awk '{print $2}' diff_spl_m_ex1_3.txt | sort -n | tail -1
nombre de couples lu: 2000
9.999996912662e-01
```

#Pour n = 5

./spline_mod 0 1.5 2000 int_spl_m_ex1_5_2000.bin < ex1_5.bin

./diffabsbin ex1_2000.bin int_spl_m_ex1_5_2000.bin diff_spl_m_ex1_5.bin

./bin2ascii diff_spl_m_ex1_5.bin

awk '{print \$2}' diff_spl_m_ex1_5.txt | sort -n | tail -1

Erreur maximale pour n = 5: 9.997002621242e-01

```
(root@kali-athena)-[/home/athenista/Bureau/polytechNantes/Tp/tpMethodes]
# #Pour n = 5
./spline_mod 0 1.5 2000 int_spl_m_ex1_5_2000.bin < ex1_5.bin
./diffabsbin ex1_2000.bin int_spl_m_ex1_5_2000.bin diff_spl_m_ex1_5.bin
./bin2ascii diff_spl_m_ex1_5.bin
awk '{print $2}' diff_spl_m_ex1_5.txt | sort -n | tail -1
nombre de couples lu: 2000
9.997002621242e-01
```

#Pour n=9

./spline_mod 0 1.5 2000 int_spl_m_ex1_9_2000.bin < ex1_9.bin

./diffabsbin ex1_2000.bin int_spl_m_ex1_9_2000.bin diff_spl_m_ex1_9.bin

./bin2ascii diff_spl_m_ex1_9.bin

awk '{print \$2}' diff_spl_m_ex1_9.txt | sort -n | tail -1

Erreur maximale pour n = 9 : 9.983092000654e-02

```
(root@kali-athena)-[/home/athenista/Bureau/polytechNantes/Tp/tpMethodes]
# #Pour n=9
./spline_mod 0 1.5 2000 int_spl_m_ex1_9_2000.bin < ex1_9.bin
./diffabsbin ex1_2000.bin int_spl_m_ex1_9_2000.bin diff_spl_m_ex1_9.bin
./bin2ascii diff_spl_m_ex1_9.bin
awk '{print $2}' diff_spl_m_ex1_9.txt | sort -n | tail -1
nombre de couples lu: 2000
9.983092000654e-02
```

#Pour n = 17

./spline_mod 0 1.5 2000 int_spl_m_ex1_17_2000.bin < ex1_17.bin

```
./diffabsbin ex1_2000.bin int_spl_m_ex1_17_2000.bin diff_spl_m_ex1_17.bin
```

```
./bin2ascii diff_spl_m_ex1_17.bin
```

```
awk '{print $2}' diff_spl_m_ex1_17.txt | sort -n | tail -1
```

Erreur maximale pour n = 17 : 9.999741349676e-05

```
(root@kali-athena)-[/home/athenista/Bureau/polytechNantes/Tp/tpMethodesN]
# #Pour n = 17
./spline_mod 0 1.5 2000 int_spl_m_ex1_17_2000.bin < ex1_17.bin
./diffabsbin ex1_2000.bin int_spl_m_ex1_17_2000.bin diff_spl_m_ex1_17.bin
./bin2ascii diff_spl_m_ex1_17.bin
awk '{print $2}' diff_spl_m_ex1_17.txt | sort -n | tail -1
nombre de couples lu: 2000
9.999741349676e-05
```

#Pour n = 33

```
./spline_mod 0 1.5 2000 int_spl_m_ex1_33_2000.bin < ex1_33.bin
```

```
./diffabsbin ex1_2000.bin int_spl_m_ex1_33_2000.bin diff_spl_m_ex1_33.bin
```

```
./bin2ascii diff_spl_m_ex1_33.bin
```

```
awk '{print $2}' diff_spl_m_ex1_33.txt | sort -n | tail -1
```

Erreur maximale pour n = 33 : 9.985872296869e-05

```
(root@kali-athena)-[/home/athenista/Bureau/polytechNantes/Tp/tpMethodesN]
# #Pour n = 33
./spline_mod 0 1.5 2000 int_spl_m_ex1_33_2000.bin < ex1_33.bin
./diffabsbin ex1_2000.bin int_spl_m_ex1_33_2000.bin diff_spl_m_ex1_33.bin
./bin2ascii diff_spl_m_ex1_33.bin
awk '{print $2}' diff_spl_m_ex1_33.txt | sort -n | tail -1
nombre de couples lu: 2000
9.985872296869e-05
```

Pour n= 65

```
./spline_mod 0 1.5 2000 int_spl_m_ex1_65_2000.bin < ex1_65.bin
```

```
./diffabsbin ex1_2000.bin int_spl_m_ex1_65_2000.bin diff_spl_m_ex1_65.bin
```

```
./bin2ascii diff_spl_m_ex1_65.bin
```

```
awk '{print $2}' diff_spl_m_ex1_65.txt | sort -n | tail -1
```

Erreur maximale pour n = 65 : 9.981247106994e-06

```
(root@kali-athena)-[/home/athenista/Bureau/polytechNantes/Tp/tpMethodesNum]
# # Pour n= 65
./spline_mod 0 1.5 2000 int_spl_m_ex1_65_2000.bin < ex1_65.bin
./diffabsbin ex1_2000.bin int_spl_m_ex1_65_2000.bin diff_spl_m_ex1_65.bin
./bin2ascii diff_spl_m_ex1_65.bin
awk '{print $2}' diff_spl_m_ex1_65.txt | sort -n | tail -1
nombre de couples lu: 2000
9.981247106994e-06
```

Pour n = 2000

```
./spline_mod 0 1.5 2000 int_spl_m_ex1_2000_2000.bin < ex1_2000.bin
```

```
./diffabsbin ex1_2000.bin int_spl_m_ex1_2000_2000.bin diff_spl_m_ex1_2000.bin
```

```
./bin2ascii diff_spl_m_ex1_2000.bin
```

```
awk '{print $2}' diff_spl_m_ex1_2000.txt | sort -n | tail -1
```

Erreur maximale pour n = 2000 : 0.000000000000e+00

```
(root@kali-athena)-[/home/athenista/Bureau/polytechNantes/Tp/tpMethodesNum]
# # Pour n = 2000
./spline_mod 0 1.5 2000 int_spl_m_ex1_2000_2000.bin < ex1_2000.bin
./diffabsbin ex1_2000.bin int_spl_m_ex1_2000_2000.bin diff_spl_m_ex1_2000.bin
./bin2ascii diff_spl_m_ex1_2000.bin
awk '{print $2}' diff_spl_m_ex1_2000.txt | sort -n | tail -1
nombre de couples lu: 2000
0.000000000000e+00
```

Créez le fichier `table_ex1_mod.txt` en ajoutant les couples (h, erreur_max) :

```
echo "h erreur_max" > table_ex1_mod.txt
```

```
echo "0.5 9.999996912662e-01" >> table_ex1_mod.txt
```

```
echo "0.25 9.997002621242e-01" >> table_ex1_mod.txt
```

```
echo "0.125 9.983092000654e-02" >> table_ex1_mod.txt
```

```
echo "0.0625 9.999741349676e-05" >> table_ex1_mod.txt
```

```
echo "0.03125 9.985872296869e-05" >> table_ex1_mod.txt
```

```
echo "0.015625 9.981247106994e-06" >> table_ex1_mod.txt
```

```
echo "0.00050025 0.000000000000e+00" >> table_ex1_mod.txt
```

Valider le modèle en h sous Gnuplot

$$f(x) = a \cdot x^3$$

fit $f(x)$ "table_ex1_mod.txt" using 1:2 via a

plot "table_ex1_mod.txt" using 1:2, $f(x)$

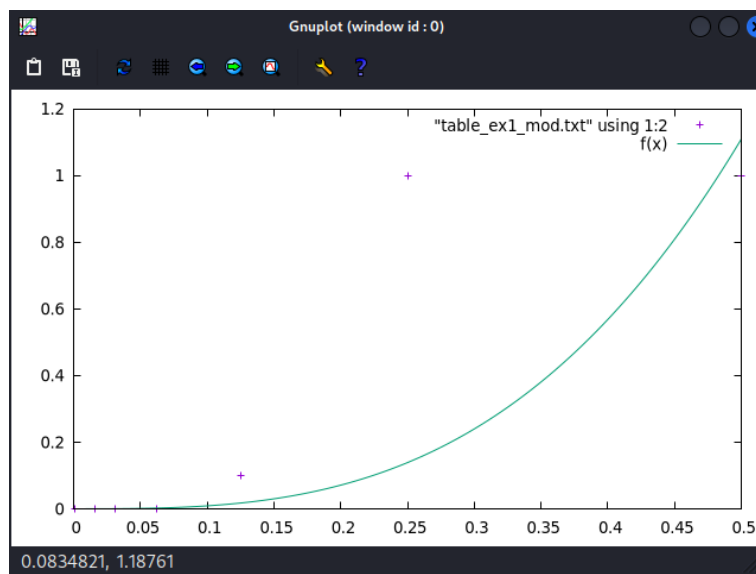
```

Terminal type is now 'wxt'
gnuplot> f(x) = a*x**3
gnuplot> fit f(x) "table_ex1_mod.txt" using 1:2 via a
iter   chisq      delta/lim  lambda  a
0  1.7436086696e+00  0.00e+00  4.76e-02  1.000000e+00
1  7.7550933343e-01 -1.25e+05  4.76e-03  7.887443e+00
2  7.6014270857e-01 -2.02e+03  4.76e-04  8.869960e+00
3  7.6014267730e-01 -4.11e-03  4.76e-05  8.871363e+00
iter   chisq      delta/lim  lambda  a

After 3 iterations the fit converged.
final sum of squares of residuals : 0.760143
rel. change during last iteration : -4.11385e-08

degrees of freedom (FIT_NDF) : 6
rms of residuals (FIT_STDFIT) = sqrt(WSSR/ndf) : 0.355936
variance of residuals (reduced chisquare) = WSSR/ndf : 0.12669

Final set of parameters          Asymptotic Standard Error
=====
a = 8.87136 +/- 2.825 (31.85%)
gnuplot> plot "table_ex1_mod.txt" using 1:2, f(x)
gnuplot>
  
```



4. Régression

On a relevé la caractéristique entrée/sortie d'un amplificateur électronique non-linéaire en faisant varier sa tension d'entrée entre $-10V$ et $10V$ par pas de $100mV$ et en relevant la tension de sortie pour chacune des valeurs d'entrée.

L'appareil de mesure de la tension de sortie présente un certain bruit. Après une mise à l'échelle des tensions par une division par $V_u = 10V$, on a obtenu un fichier binaire de mesures nommé

ampli_201.bin

- bin2ascii ampli_201.bin

Remarque 5 (Nombre de points de mesure).

Le nombre 201 est le nombre de points de mesure.

On désire modéliser la relation entrée/sortie de l'amplificateur à l'aide d'un polynôme en effectuant une régression de sorte que l'erreur absolue sur la tension de sortie ne dépasse pas $160mV$ et que le degré du polynôme soit le plus faible possible.

- 1) En visualisant le fichier de mesures à l'aide de Gnuplot (après sa conversion en fichier ASCII), donnez une valeur approximative de la tension de saturation de l'amplificateur.

Convertir le fichier binaire en ASCII

Utilisez la commande `bin2ascii` pour convertir le fichier binaire `ampli_201.bin` en un fichier ASCII.

`./bin2ascii ampli_201.bin`

```
(root@kali-athena)-[/home/a]
# ./bin2ascii ampli_201.bin
nombre de couples lu: 201
```

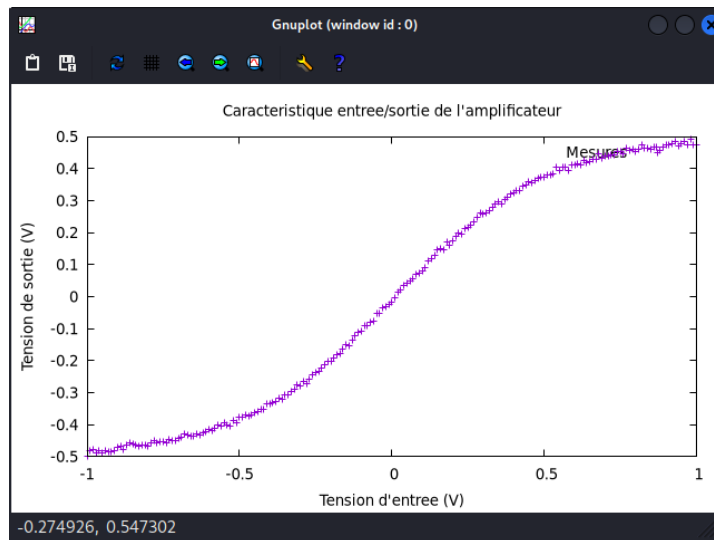
gnuplot

`set title "Caracteristique entree/sortie de l'amplificateur"`

`set xlabel "Tension d'entree (V)"`

```
set ylabel "Tension de sortie (V)"
```

```
plot "ampli_201.txt" using 1:2 with points title "Mesures"
```



2) La caractéristique entrée/sortie que vous visualisez est bruitée mais semble vérifier une certaine symétrie, laquelle ? La fonction gain est-elle plutôt paire ou impaire ? Qu'en déduisez-vous pour

le degré des polynômes qui seront probablement de bons candidats ?

La caractéristique entrée/sortie de l'amplificateur visualisée est bruitée mais montre une symétrie autour de l'origine (0,0), suggérant que la fonction gain est impaire, puisque $f(-x) \approx -f(x)$. Cela signifie que les polynômes candidats pour modéliser cette relation devraient être de degré impair, comportant des termes tels que x, x^3, x^5 , etc., pour capturer correctement la nature impaire de la fonction.

3) Effectuez le calcul de la régression pour des degrés 3, 5, 7, 9 et 11 à l'aide du source

```
regression.c.
```

- `gcc -o regression regression.c -lm -Ofast`

Visualisez pour chacun de ces cas les courbes obtenues sous Gnuplot.

- `regression -1 1 3 2000 reg_3_ampli_201.bin ampli_201.bin`

```
gcc -o regression regression.c -lm -Ofast
```

Effectuer la régression pour chaque degré

#Pour le degré 3

```
./regression -1 1 3 2000 reg_3_ampli_201.bin ampli_201.bin
```

#Pour le degré 5

```
./regression -1 1 5 2000 reg_5_ampli_201.bin ampli_201.bin
```

#Pour le degré 7

```
./regression -1 1 7 2000 reg_7_ampli_201.bin ampli_201.bin
```

#Pour le degré 9

```
./regression -1 1 9 2000 reg_9_ampli_201.bin ampli_201.bin
```

#Pour le degré 11

```
./regression -1 1 11 2000 reg_11_ampli_201.bin ampli_201.bin
```

```
./bin2ascii reg_3_ampli_201.bin > reg_3_ampli_201.txt
```

```
./bin2ascii reg_5_ampli_201.bin > reg_5_ampli_201.txt
```

```
./bin2ascii reg_7_ampli_201.bin > reg_7_ampli_201.txt
```

```
./bin2ascii reg_9_ampli_201.bin > reg_9_ampli_201.txt
```

```
./bin2ascii reg_11_ampli_201.bin > reg_11_ampli_201.txt
```

gnuplot

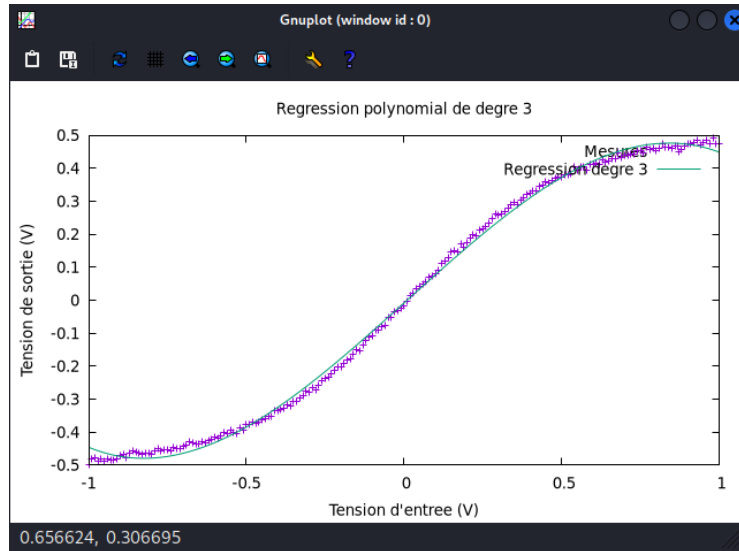
```
set title "Regression polynomial de degre 3"
```

```
set xlabel "Tension d'entree (V)"
```

```
set ylabel "Tension de sortie (V)"
```

```
plot "ampli_201.txt" using 1:2 with points title "Mesures", \
```

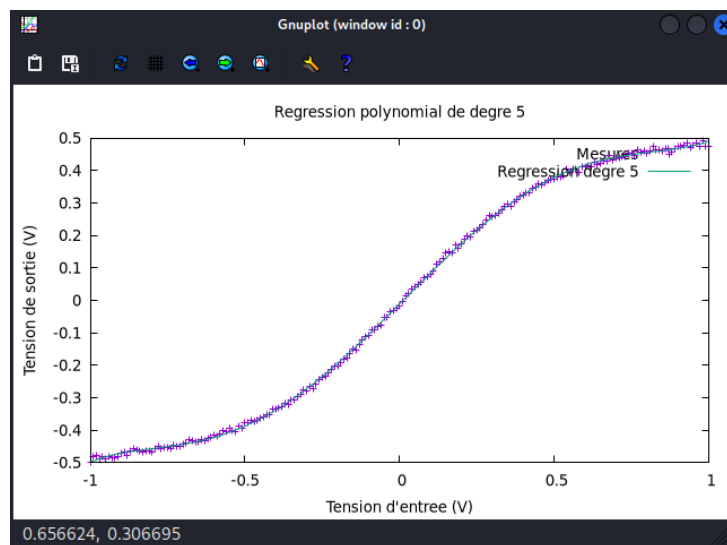
```
"reg_3_ampli_201.txt" using 1:2 with lines title "Regression degre 3"
```



set title "Regression polynomial de degre 5"

**plot "ampli_201.txt" using 1:2 with points title "Mesures", **

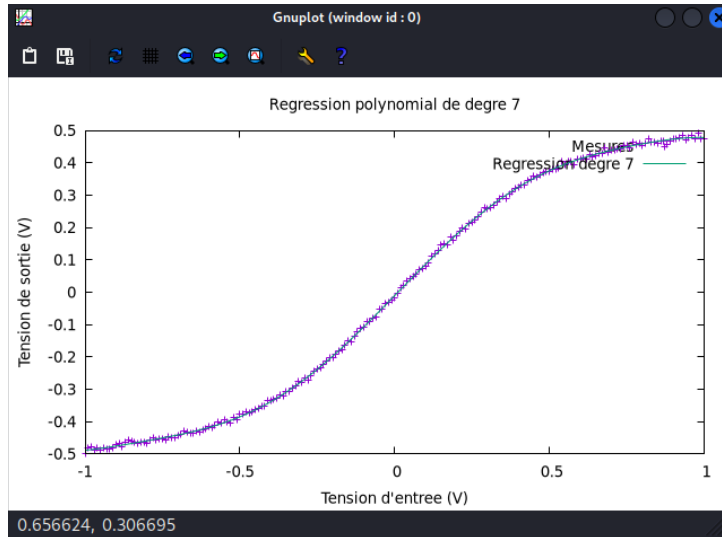
"reg_5_ampli_201.txt" using 1:2 with lines title "Regression degre 5"



set title "Regression polynomial de degre 7"

**plot "ampli_201.txt" using 1:2 with points title "Mesures", **

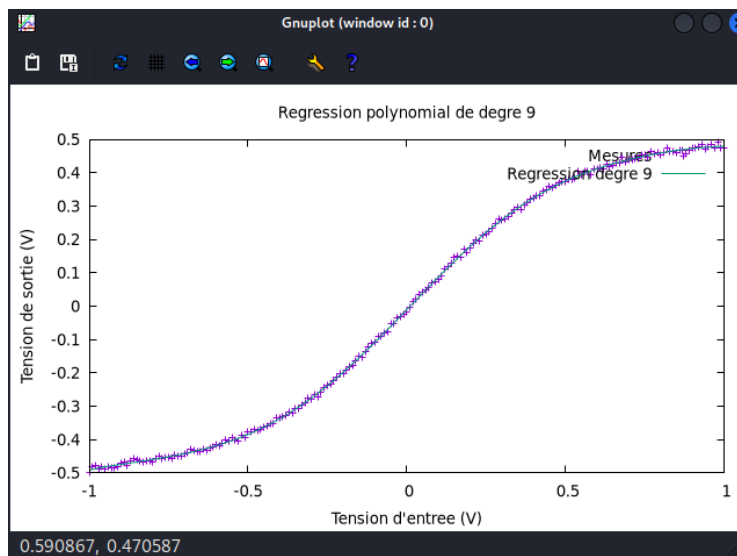
"reg_7_ampli_201.txt" using 1:2 with lines title "Regression degre 7"



set title "Regression polynomial de degre 9"

**plot "ampli_201.txt" using 1:2 with points title "Mesures", **

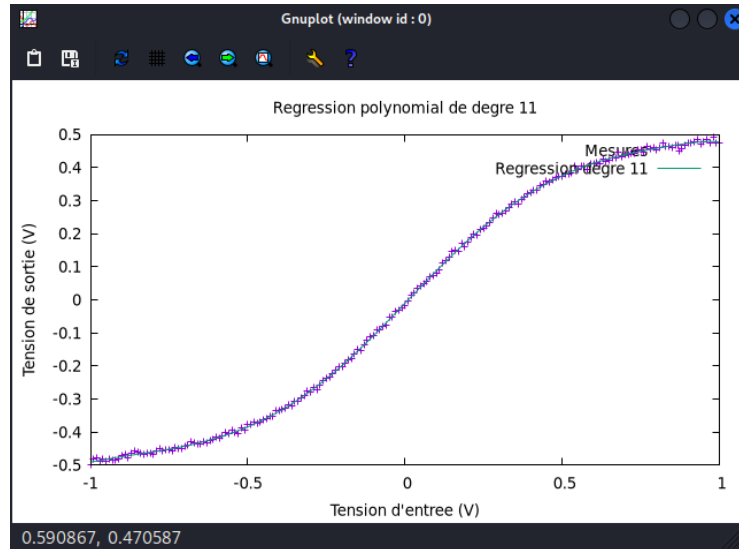
"reg_9_ampli_201.txt" using 1:2 with lines title "Regression degre 9"



set title "Regression polynomial de degre 11"

**plot "ampli_201.txt" using 1:2 with points title "Mesures", **

"reg_11_ampli_201.txt" using 1:2 with lines title "Regression degre 11"



- 4) En utilisant le programme diffbin, tracez sous Gnuplot les courbes des résidus pour les différents degrés. Quel serait le meilleur candidat pour le problème posé ?

```
gcc -o diffbin diffbin.c -lm -Ofast
```

Nous allons ensuite calculer les résidus pour chaque degré

#Pour le degré 3

```
./regression -1 1 3 201 reg_3_ampli_201.bin ampli_201.bin
```

```
./diffbin ampli_201.bin reg_3_ampli_201.bin diff_3_ampli_201.bin
```

```
./bin2ascii diff_3_ampli_201.bin > diff_3_ampli_201.txt
```

```
(root@kali-athena)-[/home/athenista/Bureau/polytechNantes/Tp/
# #Pour le degré 3
./regression -1 1 3 201 reg_3_ampli_201.bin ampli_201.bin
./diffbin ampli_201.bin reg_3_ampli_201.bin diff_3_ampli_201.bin
./bin2ascii diff_3_ampli_201.bin > diff_3_ampli_201.txt
Nombre de points: 201
```

#Pour le degré 5

```
./regression -1 1 5 201 reg_5_ampli_201.bin ampli_201.bin
```

```
./diffbin ampli_201.bin reg_5_ampli_201.bin diff_5_ampli_201.bin
```

```
./bin2ascii diff_5_ampli_201.bin > diff_5_ampli_201.txt
```

```
(root@kali-athena)-[/home/athenista/Bureau/polytechNantes/Tp/tp1]
# #Pour le degré 5
./regression -1 1 5 201 reg_5_ampli_201.bin ampli_201.bin
./diffbin ampli_201.bin reg_5_ampli_201.bin diff_5_ampli_201.bin
./bin2ascii diff_5_ampli_201.bin > diff_5_ampli_201.txt
Nombre de points: 201
```

#Pour le degré 7

```
./regression -1 1 7 201 reg_7_ampli_201.bin ampli_201.bin
```

```
./diffbin ampli_201.bin reg_7_ampli_201.bin diff_7_ampli_201.bin
```

```
./bin2ascii diff_7_ampli_201.bin > diff_7_ampli_201.txt
```

```
(root@kali-athena)-[/home/athenista/Bureau/polytechNantes/Tp/tp1]
# #Pour le degré 7
./regression -1 1 7 201 reg_7_ampli_201.bin ampli_201.bin
./diffbin ampli_201.bin reg_7_ampli_201.bin diff_7_ampli_201.bin
./bin2ascii diff_7_ampli_201.bin > diff_7_ampli_201.txt
Nombre de points: 201
```

#Pour le degré 9

```
./regression -1 1 9 201 reg_9_ampli_201.bin ampli_201.bin
```

```
./diffbin ampli_201.bin reg_9_ampli_201.bin diff_9_ampli_201.bin
```

```
./bin2ascii diff_9_ampli_201.bin > diff_9_ampli_201.txt
```

```
(root@kali-athena)-[/home/athenista/Bureau/polytechNantes/Tp/tp1]
# #Pour le degré 9
./regression -1 1 9 201 reg_9_ampli_201.bin ampli_201.bin
./diffbin ampli_201.bin reg_9_ampli_201.bin diff_9_ampli_201.bin
./bin2ascii diff_9_ampli_201.bin > diff_9_ampli_201.txt
Nombre de points: 201
```

#Pour le degré 11

```
./regression -1 1 11 201 reg_11_ampli_201.bin ampli_201.bin
```

```
./diffbin ampli_201.bin reg_11_ampli_201.bin diff_11_ampli_201.bin
```

```
./bin2ascii diff_11_ampli_201.bin > diff_11_ampli_201.txt
```

```
(root@kali-athena)-[/home/athenista/Bureau/polytechNantes/Tp/tp1]
# #Pour le degré 11
./regression -1 1 11 201 reg_11_ampli_201.bin ampli_201.bin
./diffbin ampli_201.bin reg_11_ampli_201.bin diff_11_ampli_201.bin
./bin2ascii diff_11_ampli_201.bin > diff_11_ampli_201.txt
Nombre de points: 201
```

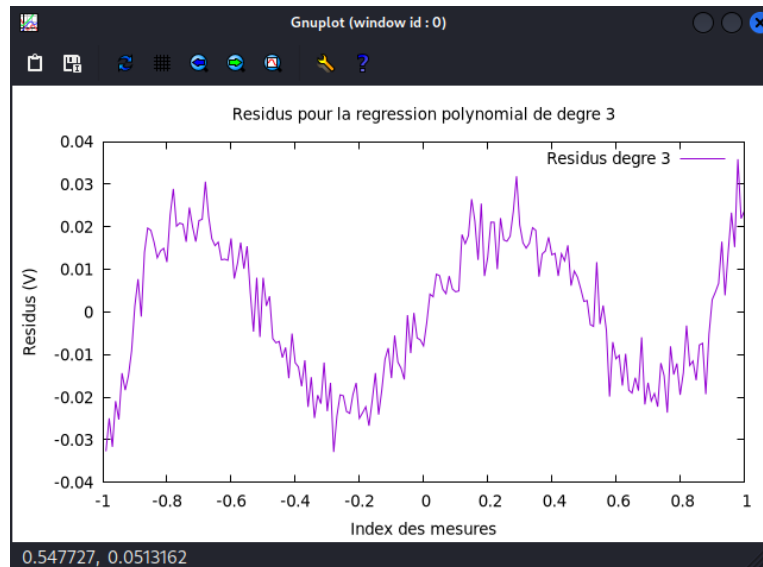
gnuplot

set title "Résidus pour la régression polynomial de degré 3"

set xlabel "Index des mesures"

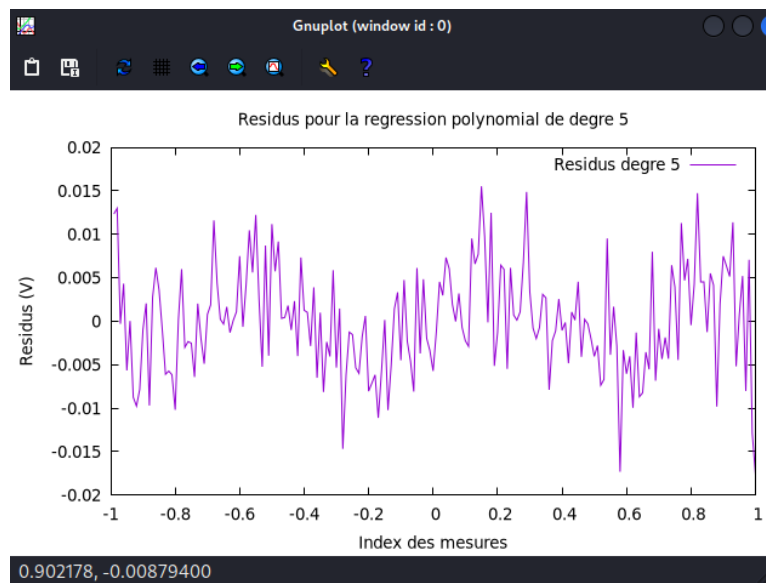
set ylabel "Residus (V)"

plot "diff_3_ampli_201.txt" using 1:2 with lines title "Residus degree 3"



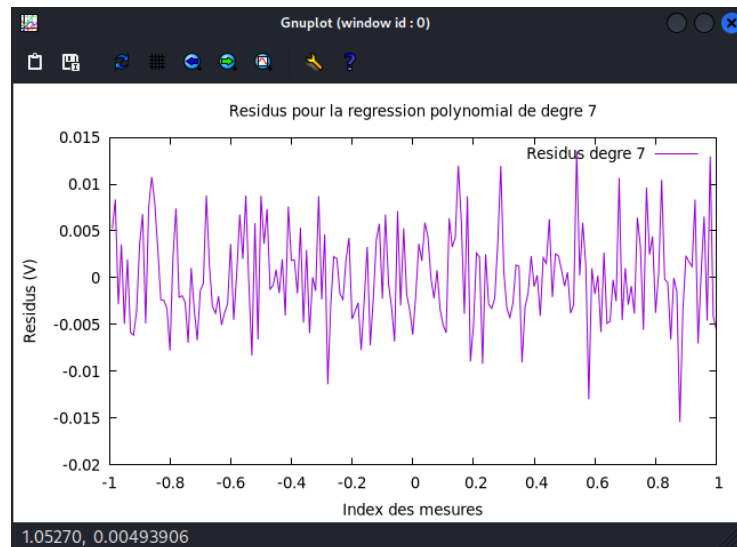
set title "Résidus pour la régression polynomial de degré 5"

plot "diff_5_ampli_201.txt" using 1:2 with lines title "Residus degree 5"



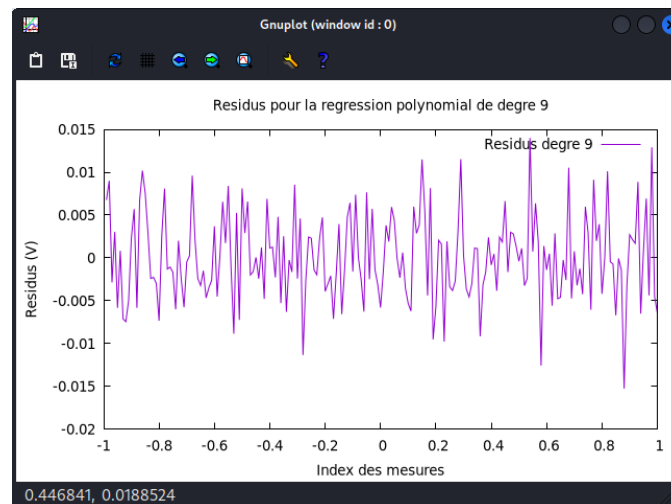
```
set title "Résidus pour la régression polynomial de degré 7"
```

```
plot "diff_7_ampli_201.txt" using 1:2 with lines title "Residus degree 7"
```



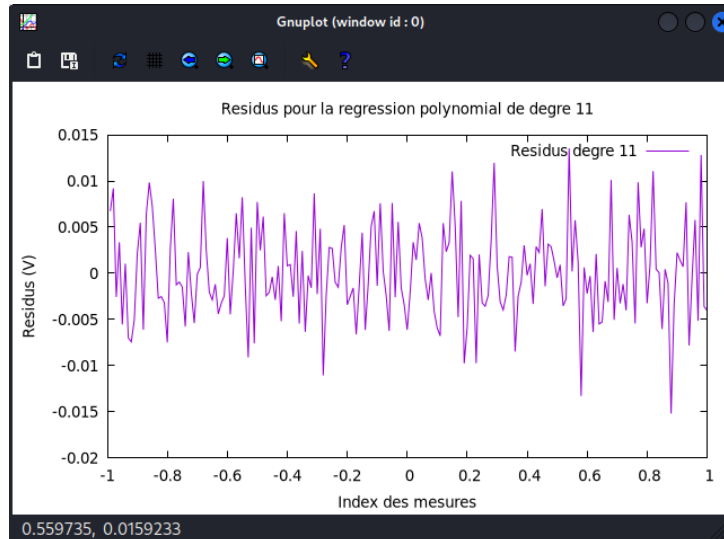
```
set title "Résidus pour la régression polynomial de degré 9"
```

```
plot "diff_9_ampli_201.txt" using 1:2 with lines title "Residus degree 9"
```



```
set title "Résidus pour la régression polynomial de degré 11"
```

```
plot "diff_11_ampli_201.txt" using 1:2 with lines title "Residus degree 11"
```



Le meilleur candidat pour modéliser la relation entrée/sortie de l'amplificateur est le polynôme de degré 9. En analysant les courbes des résidus, nous constatons que les résidus pour le degré 9 sont uniformément répartis autour de zéro, avec des variations limitées entre -0.015 V et 0.015 V, indiquant une meilleure capture des variations non linéaires. Les degrés 7 et 11 montrent également une bonne performance, mais le degré 9 offre un meilleur équilibre entre complexité du modèle et précision, sans surajustement visible. Ainsi, pour minimiser l'erreur absolue sur la tension de sortie et maintenir un degré de polynôme aussi bas que possible, le polynôme de degré 9 est recommandé.

5) L'amplificateur présente une certaine tension de décalage en entrée ainsi qu'un certain gain

dans la partie centrale de sa caractéristique. Donnez une estimation de ces 2 paramètres à

l'aide du modèle obtenu.

- $f(x) = a + b \cdot x + c \cdot x^2 + d \cdot x^3$

- fit $f(x)$ "reg_3_ampli_201.txt" via a,b,c,d

#Pour le degré 3

./regression -1 1 3 201 reg_3_ampli_201.bin ampli_201.bin

./bin2ascii reg_3_ampli_201.bin > reg_3_ampli_201.txt

```
Sources > ❏ reg_3_ampli_201.txt
1  nombre de couples lu: 201
2  232396078e-01
3  -9.900000000000e-01 -4.496778475032e-01
4  -9.800000000000e-01 -4.532855287586e-01
5  -9.700000000000e-01 -4.566487566854e-01
6  -9.600000000000e-01 -4.597700045946e-01
```

il faut donc penser à supprimer les 2 premières lignes sinon

gnuplot

- $f(x)=a+b \cdot x+c \cdot x^2+d \cdot x^3$
- fit f(x) "reg_3_ampli_201.txt" via a,b,c,d

```
Terminal type is now 'wxt'
gnuplot> f(x)=a+b*x+c*x**2+d*x**3
gnuplot> fit f(x) "reg_3_ampli_201.txt" via a,b,c,d
bad data on line 1 of datafile
```

```
gnuplot> f(x)=a+b*x+c*x**2+d*x**3
gnuplot> fit f(x) "reg_3_ampli_201.txt" via a,b,c,d
iter   chisq      delta/lim  lambda  a          b          c          d
0 4.5631762493e+02  0.00e+00  6.47e-01  1.000000e+00  1.000000e+00  1.000000e+00  1.000000e+00
1 5.7704373197e-02 -7.91e+08  6.47e-02 -1.150227e-02  7.961379e-01  2.375301e-02 -3.048389e-01
2 8.2040495224e-08 -7.03e+10  6.47e-03 -8.932384e-03  8.593706e-01  1.034569e-02 -4.120856e-01
3 1.3078646725e-17 -6.27e+14  6.47e-04 -8.931578e-03  8.594544e-01  1.034417e-02 -4.122185e-01
4 1.3634760762e-25 -9.59e+12  6.47e-05 -8.931578e-03  8.594544e-01  1.034417e-02 -4.122185e-01
* 1.3637852720e-25  2.27e+01  6.47e-04 -8.931578e-03  8.594544e-01  1.034417e-02 -4.122185e-01
* 1.3637852720e-25  2.27e+01  6.47e-03 -8.931578e-03  8.594544e-01  1.034417e-02 -4.122185e-01
* 1.3637852720e-25  2.27e+01  6.47e-02 -8.931578e-03  8.594544e-01  1.034417e-02 -4.122185e-01
* 1.3637852720e-25  2.27e+01  6.47e-01 -8.931578e-03  8.594544e-01  1.034417e-02 -4.122185e-01
* 1.3635987598e-25  9.00e+00  6.47e+00 -8.931578e-03  8.594544e-01  1.034417e-02 -4.122185e-01
* 1.3635332474e-25  4.19e+00  6.47e+01 -8.931578e-03  8.594544e-01  1.034417e-02 -4.122185e-01
5 1.3634760762e-25  0.00e+00  6.47e+00 -8.931578e-03  8.594544e-01  1.034417e-02 -4.122185e-01
iter   chisq      delta/lim  lambda  a          b          c          d
After 5 iterations the fit converged.
final sum of squares of residuals : 1.36348e-25
rel. change during last iteration : 0
degrees of freedom (FIT_NDF) : 196
rms of residuals (FIT_STDFIT) = sqrt(WSSR/ndf) : 2.63752e-14
variance of residuals (reduced chisquare) = WSSR/ndf : 6.95651e-28
Final set of parameters      Asymptotic Standard Error
a = -0.00893158      +/- 2.798e-15 (3.132e-11%)
b = 0.859454      +/- 8.076e-15 (9.397e-13%)
c = 0.0103442      +/- 6.259e-15 (6.05e-11%)
d = -0.412219      +/- 1.234e-14 (2.993e-12%)
correlation matrix of the fit parameters:
a      1.000
b     -0.009  1.000
c     -0.745  0.019  1.000
d      0.013 -0.916 -0.030  1.000
```

L'amplificateur présente une tension de décalage en entrée estimée à -0.00893158 V et un gain dans la partie centrale de sa caractéristique de 0.859454. Ces valeurs ont été obtenues en ajustant un modèle polynomial $f(x)=a+b \cdot x+c \cdot x^2+d \cdot x^3$ aux données de l'amplificateur, où les paramètres a et b représentent respectivement la tension de décalage et le gain.

#Voici les commandes Gnuplot pour ajuster et visualiser les modèles polynomiaux pour les degrés #5, 7, 9 et 11 :

#Pour le degré 5

```
./regression -1 1 5 201 reg_5_ampli_201.bin ampli_201.bin
```

```
./bin2ascii reg_5_ampli_201.bin > reg_5_ampli_201.txt
```

gnuplot

$$f(x) = a + b*x + c*x**2 + d*x**3 + e*x**4 + f*x**5$$

fit f(x) "reg_5_ampli_201.txt" via a, b, c, d, e, f

set title "Régression polynomial de degré 5"

plot "ampli_201.txt" using 1:2 with points title "Mesures", \ "reg_5_ampli_201.txt" using 1:2 with
lines title "Régression degré 5"

#Pour le degré 7

./regression -1 1 7 201 reg_7_ampli_201.bin ampli_201.bin

./bin2ascii reg_7_ampli_201.bin > reg_7_ampli_201.txt

gnuplot

$$f(x) = a + b*x + c*x**2 + d*x**3 + e*x**4 + f*x**5 + g*x**6 + h*x**7$$

fit f(x) "reg_7_ampli_201.txt" via a, b, c, d, e, f, g, h

set title "Régression polynomial de degré 7"

plot "ampli_201.txt" using 1:2 with points title "Mesures", \ "reg_7_ampli_201.txt" using 1:2 with
lines title "Régression degré 7"

#Pour le degré 9

./regression -1 1 9 201 reg_9_ampli_201.bin ampli_201.bin

./bin2ascii reg_9_ampli_201.bin > reg_9_ampli_201.txt

gnuplot

$$f(x) = a + b*x + c*x**2 + d*x**3 + e*x**4 + f*x**5 + g*x**6 + h*x**7 + i*x**8 + j*x**9$$

fit f(x) "reg_9_ampli_201.txt" via a, b, c, d, e, f, g, h, i, j

set title "Régression polynomial de degré 9"

plot "ampli_201.txt" using 1:2 with points title "Mesures", \ "reg_9_ampli_201.txt" using 1:2 with
lines title "Régression degré 9"

#Pour le degré 11

```
./regression -1 1 11 201 reg_11_ampli_201.bin ampli_201.bin
```

```
./bin2ascii reg_11_ampli_201.bin > reg_11_ampli_201.txt
```

```
gnuplot
```

$$f(x) = a + b*x + c*x^{**2} + d*x^{**3} + e*x^{**4} + f*x^{**5} + g*x^{**6} + h*x^{**7} + i*x^{**8} + j*x^{**9} + k*x^{**10} + l*x^{**11}$$

```
fit f(x) "reg_11_ampli_201.txt" via a, b, c, d, e, f, g, h, i, j, k, l
```

```
set title "Régression polynomial de degré 11"
```

```
plot "ampli_201.txt" using 1:2 with points title "Mesures", \ "reg_11_ampli_201.txt" using 1:2 with  
lines title "Régression degré 11"
```

Conclusion