Clément Collet
Loïc Steunou

Projet d'analyse numérique – GM3 2014/2015

M.Nicolas Forcadel

Présentation Générale

La Méthode de Neville-Aitken :

La méthode de Neville-Aitken est une méthode d'interpolation polynomiale. Cette méthode permet le calcul de la valeur du polynôme d'interpolation en un point donné, avec cette méthode, l'ajout de points d'interpolation est aisé.
Cette méthode permet d'approximer la valeur en ce point donné de certaines fonctions.

Prenons n+1 points donnés (xi,yi) où les xi sont tous différents. Prenons aussi le polynôme d'interpolation P(x) de degré au plus n et vérifiant P(xi) = yi.
La méthode de Neville va nous donner la valeur en x de ce polynôme passant pas les n+1 points donnés.

L'algorithme de Neville :

Soit Pij(x) le polynôme de degré j qui passe par les points (xk,yk)
(avec k=i, i+1, …, i+j).

Les Pij(x) peuvent être déterminé grâce à l'algorithme suivant :

$$\begin{aligned} p_{i,0}(x) &= y_i, & 0 \le i \le n, \\ p_{i,j+1}(x) &= \frac{(x_i - x)p_{i+1,j}(x) + (x - x_{i+j+1})p_{i,j}(x)}{x_i - x_{i+j+1}}, & 0 \le j < n \text{ et } 0 \le i < n-j. \end{aligned}$$

Exemple avec n = 4 :

En utilisant l'algorithme de Neville, on peut remplir le tableau suivant :

$$\begin{array}{ccccc} p_{0,0}(x) = y_0 & & & & \\ & p_{0,1}(x) & & & \\ p_{1,0}(x) = y_1 & & p_{0,2}(x) & & \\ & p_{1,1}(x) & & p_{0,3}(x) & \\ p_{2,0}(x) = y_2 & & p_{1,2}(x) & & \boxed{p_{0,4}(x)} \\ & p_{2,1}(x) & & p_{1,3}(x) & \\ p_{3,0}(x) = y_3 & & p_{2,2}(x) & & \\ & p_{3,1}(x) & & & \\ p_{4,0}(x) = y_4 & & & & \end{array}$$

On obtient alors P(0,4)(x) qui est la valeur en x du polynôme, Notre Projet :

Dans notre projet, nous avons codé un algorithme de Neville-Aitken permettant de nous donner la valeur du polynôme d'interpolation aux points milieux des intervalles [xi, xi+1].

Nous avons donc interpolé 3 fonctions, un polynôme, une somme de cosinus et sinus ainsi que le produit d'un exponentiel et d'un cosinus.

Nous nous sommes ensuite intéressés à la valeur de l'erreur d'approximation qui est :

$$\max_{i} \left| f\left(\alpha_{i}\right) - p\left(\alpha_{i}\right) \right| \text{ lorsque } \alpha_i = \frac{x_i + x_{i+1}}{2}.$$

Nous avons donc codé et calculé cette erreur pour différents échantillons de points :
- Des points équidistants
- Les racines du polynôme de Tchebychev

Nous avons également analysé le comportement de l'erreur d'approximation lorsque le nombre de points d'interpolation augmente ainsi que l'influence de la double précision.

Programmation

Vous trouverez ci-joint plusieurs programmes. Le plus général est DoubleTcheby, on peut y modifier la taille d'échantillon très facilement. Pour les points équidistant (DoubleEqui), toute modification de l'échantillon est plus longue car il faut le faire point par point. Enfin, DoubleTestMultiple nous permet de lancer de nombreux tests en faisant varier le nombre de points, seul l'erreur à chaque fois apparaît du coup à l'écran. Les programmes commençant par Simple ne sont que des traductions de ces programmes en simple précision,

Pour tout le code, tailleechantillon correspond au nombre de valeur de Xi que l'on se donne. Mais par la suite on traite les Alphas, c'est à dire les milieux de ces points. Donc nos itérations se font souvent jusqu’à tailleechantillon-1. L'ensemble du code est détaillé dans celui-ci pas à pas.

Nous allons voir plus en détail l'étape délicate du code. Il s'agit de la boucle récursive permettant d'avancer dans les P(i,j) jusqu’à P(0,n-1) avec n le nombre de points étudiés. Nous aurions pu faire un tableau en 2 dimensions, l'une représentant les i et l'autre les j. Nous avons décidé de tout faire dans le même tableau qui change au fur et à mesure que s’enchaînent les itérations.

Nous commençons avec un tableau initialisé, c'est à dire remplît des Yi (= f (Xi)). La plus petite boucle, celle de i applique la relation de récurrence :

$$p_{i,j+1}(x) = \frac{(x_i - x)p_{i+1,j}(x) + (x - x_{i+j+1})p_{i,j}(x)}{x_i - x_{i+j+1}},$$

qui est codée : tab[i] = ((x[i]-alpha[k]) * tab[i+1] + (alpha[k]-x[i+j+1]) * tab[i]) / (x[i]-x[i+j+1]).

Alpha[k] correspond à la valeur dont on souhaiterais l'approximation. Cela fonctionne car lorsqu'on remplace P(i,j) (tab[i]) par P (i,j+1), on ne sert plus ensuite de P(i,j) (la valeur que l'on a perdu).

j varie de 0 à tailleechantillon-1 et i de 0 à tailleenchantillon-j.

A chaque itération de j, on modifie une valeur de tab en moins, ainsi lors de la dernière itération, la valeur de l'approximation est en tab[0] et correspond à P(0,n-1).

Exploitation des résultats

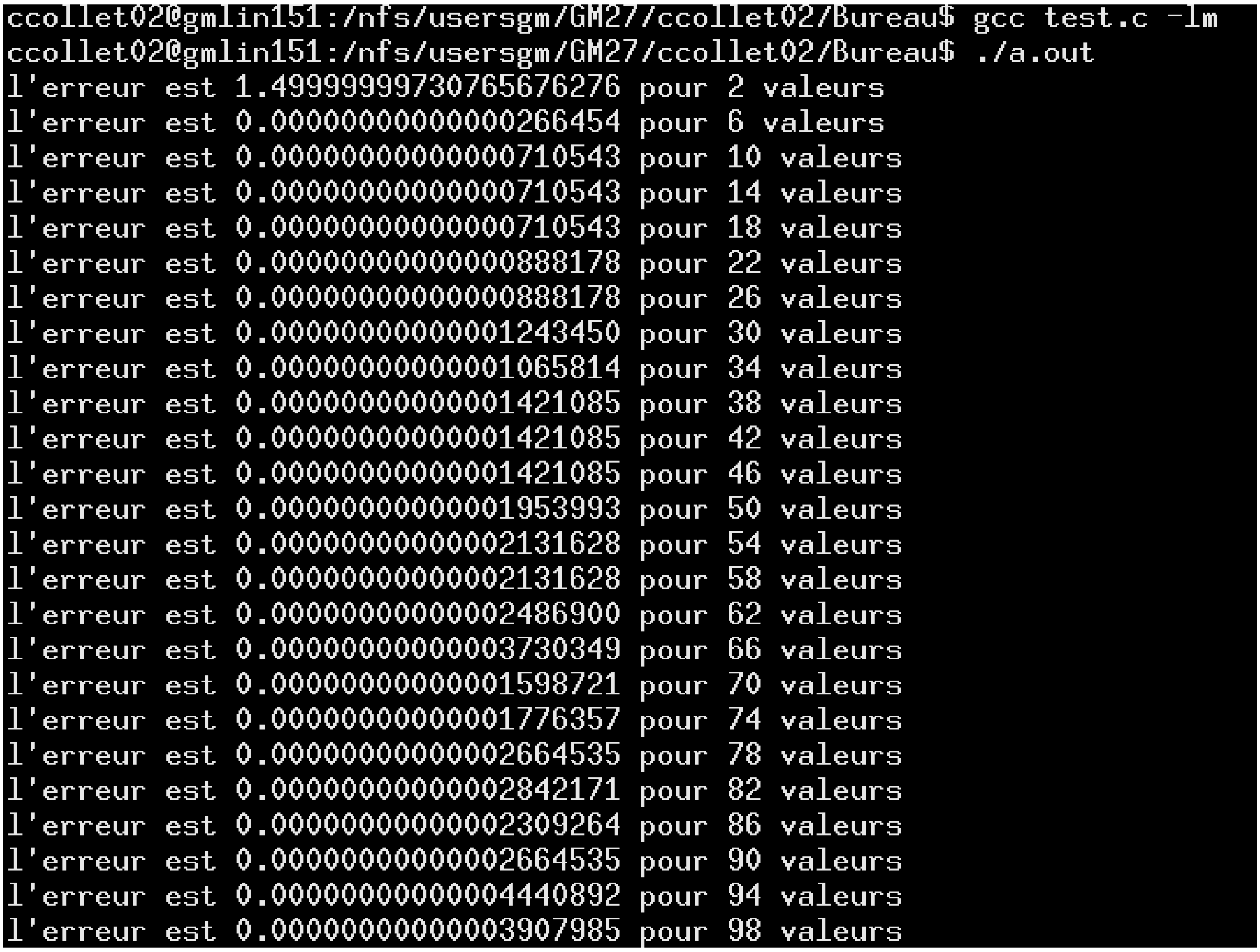
Nous avions 4 paramètres à faire varier au cours de l'exploitation :
- simple ou double précision
- utilisation de Xi racines du polynômes de Tchebychev ou équidistants et issus du découpage de [-1;1]
- différentes fonctions, nous avons pris les trois proposées dans le sujet
- variations du nombres de points

1) Double précision

a/ Racine du polynôme de Tchebychev

Les points Xi sont racines du polynôme de Tchebychev. Autrement dit, x[i] = cos ((2i+1) * Pi / (2n)).

1- Le polynôme

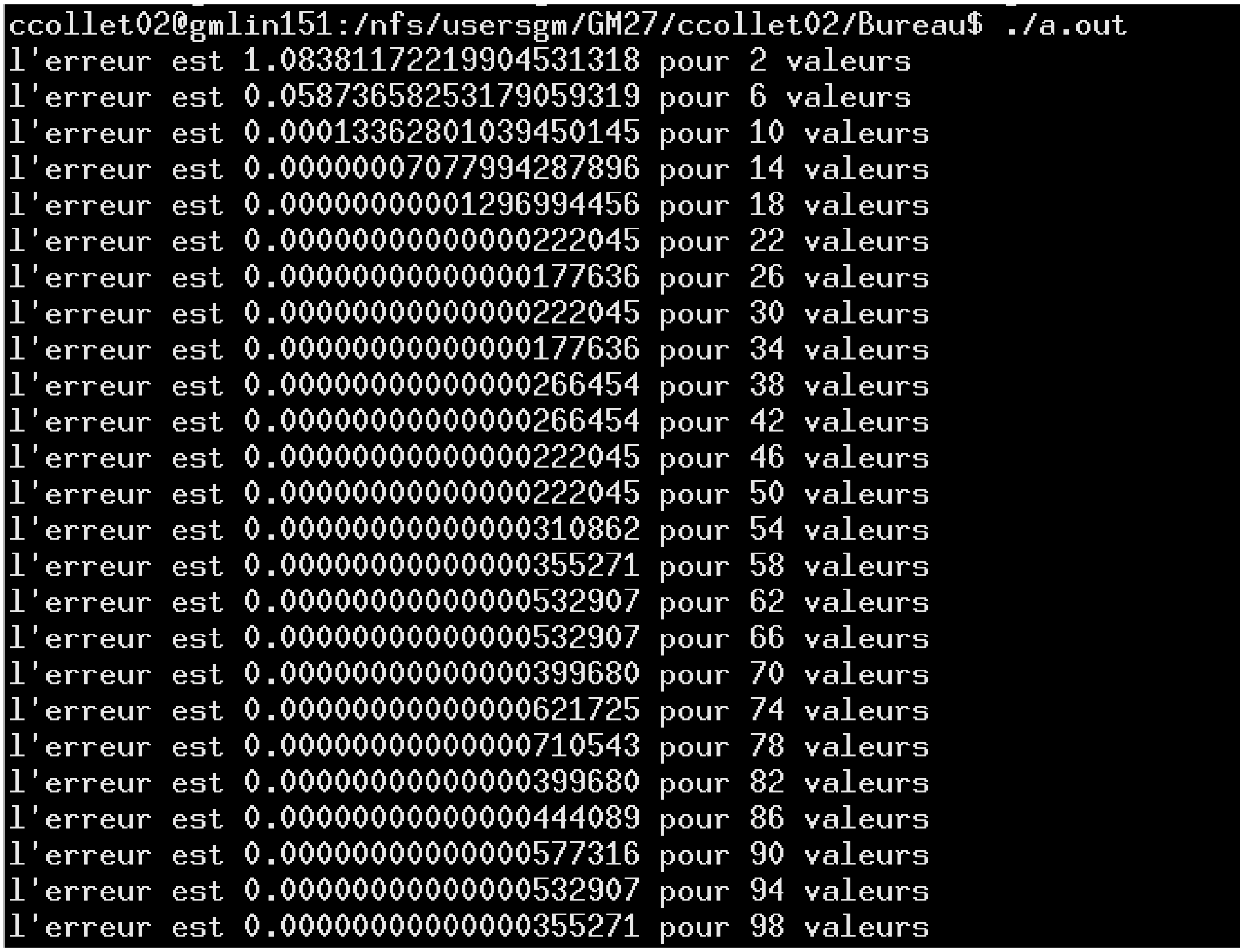


```
ccollet02@gmlin151:/nfs/usersgm/GM27/ccollet02/Bureau$ gcc test.c -lm
ccollet02@gmlin151:/nfs/usersgm/GM27/ccollet02/Bureau$ ./a.out
l'erreur est 1.49999999730765676276 pour 2 valeurs
l'erreur est 0.00000000000000266454 pour 6 valeurs
l'erreur est 0.00000000000000710543 pour 10 valeurs
l'erreur est 0.00000000000000710543 pour 14 valeurs
l'erreur est 0.00000000000000710543 pour 18 valeurs
l'erreur est 0.00000000000000888178 pour 22 valeurs
l'erreur est 0.00000000000000888178 pour 26 valeurs
l'erreur est 0.00000000000001243450 pour 30 valeurs
l'erreur est 0.00000000000001065814 pour 34 valeurs
l'erreur est 0.00000000000001421085 pour 38 valeurs
l'erreur est 0.00000000000001421085 pour 42 valeurs
l'erreur est 0.00000000000001421085 pour 46 valeurs
l'erreur est 0.00000000000001953993 pour 50 valeurs
l'erreur est 0.00000000000002131628 pour 54 valeurs
l'erreur est 0.00000000000002131628 pour 58 valeurs
l'erreur est 0.00000000000002486900 pour 62 valeurs
l'erreur est 0.00000000000003730349 pour 66 valeurs
l'erreur est 0.00000000000001598721 pour 70 valeurs
l'erreur est 0.00000000000001776357 pour 74 valeurs
l'erreur est 0.00000000000002664535 pour 78 valeurs
l'erreur est 0.00000000000002842171 pour 82 valeurs
l'erreur est 0.00000000000002309264 pour 86 valeurs
l'erreur est 0.00000000000002664535 pour 90 valeurs
l'erreur est 0.00000000000004440892 pour 94 valeurs
l'erreur est 0.00000000000003907985 pour 98 valeurs
```

Ceci représente l'erreur de $f(x)=3x^2+2x+7$ (fonction 1). L'erreur pour 2 valeurs donc 3 Xi est importante mais elle diminue très vite pour atteindre un minimum en 6 valeurs.Puis l'erreur augmente avec l'augmentation du nombre de points. Passé 647 points, la méthode ne marche plus car l'erreur en sortie est infini, comme le montre l'image ci-dessous.

```
ccollet02@gmlin151:/nfs/usersgm/GM27/ccollet02/Bureau$ ./a.out
l'erreur est 0.00000000000027355895 pour 645 valeurs
l'erreur est 0.00000000000023625546 pour 646 valeurs
l'erreur est inf pour 647 valeurs
l'erreur est inf pour 648 valeurs
l'erreur est inf pour 649 valeurs
```

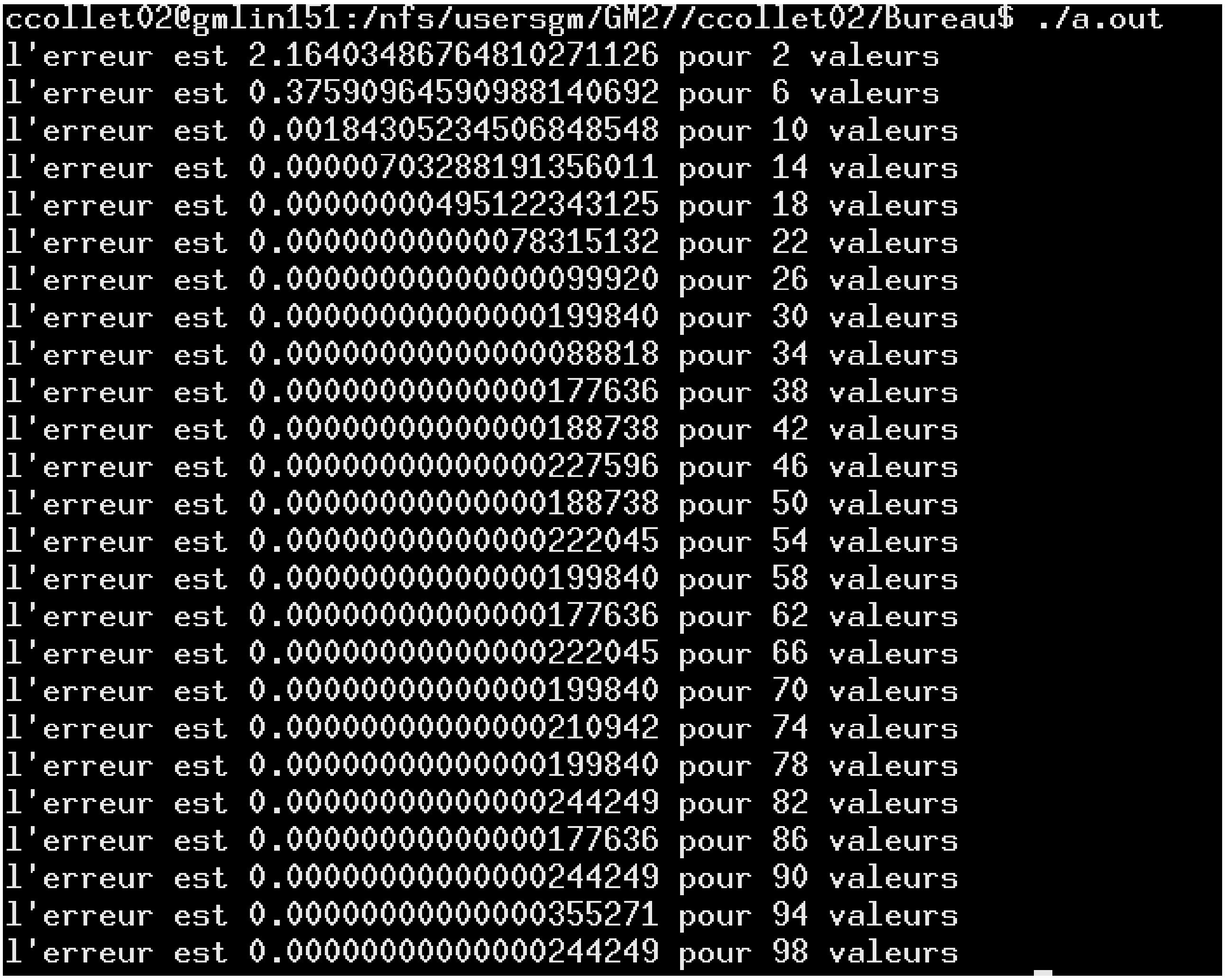
2- La fonction 2



```
ccollet02@gmlin151:/nfs/usersgm/GM27/ccollet02/Bureau$ ./a.out
l'erreur est 1.08381172219904531318 pour 2 valeurs
l'erreur est 0.05873658253179059319 pour 6 valeurs
l'erreur est 0.00013362801039450145 pour 10 valeurs
l'erreur est 0.00000007077994287896 pour 14 valeurs
l'erreur est 0.00000000001296994456 pour 18 valeurs
l'erreur est 0.00000000000000222045 pour 22 valeurs
l'erreur est 0.00000000000000177636 pour 26 valeurs
l'erreur est 0.00000000000000222045 pour 30 valeurs
l'erreur est 0.00000000000000177636 pour 34 valeurs
l'erreur est 0.00000000000000266454 pour 38 valeurs
l'erreur est 0.00000000000000266454 pour 42 valeurs
l'erreur est 0.00000000000000222045 pour 46 valeurs
l'erreur est 0.00000000000000222045 pour 50 valeurs
l'erreur est 0.00000000000000310862 pour 54 valeurs
l'erreur est 0.00000000000000355271 pour 58 valeurs
l'erreur est 0.00000000000000532907 pour 62 valeurs
l'erreur est 0.00000000000000532907 pour 66 valeurs
l'erreur est 0.00000000000000399680 pour 70 valeurs
l'erreur est 0.00000000000000621725 pour 74 valeurs
l'erreur est 0.00000000000000710543 pour 78 valeurs
l'erreur est 0.00000000000000399680 pour 82 valeurs
l'erreur est 0.00000000000000444089 pour 86 valeurs
l'erreur est 0.00000000000000577316 pour 90 valeurs
l'erreur est 0.00000000000000532907 pour 94 valeurs
l'erreur est 0.00000000000000355271 pour 98 valeurs
```

Ceci représente l'erreur de f(x)=cos(x)+cos(2*x)+sin(3*x)+sin(4*x) (fonction 2). On constate que l'erreur diminue plus lentement au début, jusqu’à un minimum vers 22 valeurs. L'erreur obtenu est d'ailleurs plus faible que celle pour le polynôme. Ensuite l'erreur stagne puis augmente à partir de 54 valeurs. La méthode ne fonctionne plus après plus de 648 points.

2- La fonction 2



```
ccollet02@gmlin151:/nfs/usersgm/GM27/ccollet02/Bureau$ ./a.out
l'erreur est 2.16403486764810271126 pour 2 valeurs
l'erreur est 0.37590964590988140692 pour 6 valeurs
l'erreur est 0.00184305234506848548 pour 10 valeurs
l'erreur est 0.00000703288191356011 pour 14 valeurs
l'erreur est 0.00000000495122343125 pour 18 valeurs
l'erreur est 0.00000000000078315132 pour 22 valeurs
l'erreur est 0.00000000000000099920 pour 26 valeurs
l'erreur est 0.00000000000000199840 pour 30 valeurs
l'erreur est 0.00000000000000088818 pour 34 valeurs
l'erreur est 0.00000000000000177636 pour 38 valeurs
l'erreur est 0.00000000000000188738 pour 42 valeurs
l'erreur est 0.00000000000000227596 pour 46 valeurs
l'erreur est 0.00000000000000188738 pour 50 valeurs
l'erreur est 0.00000000000000222045 pour 54 valeurs
l'erreur est 0.00000000000000199840 pour 58 valeurs
l'erreur est 0.00000000000000177636 pour 62 valeurs
l'erreur est 0.00000000000000222045 pour 66 valeurs
l'erreur est 0.00000000000000199840 pour 70 valeurs
l'erreur est 0.00000000000000210942 pour 74 valeurs
l'erreur est 0.00000000000000199840 pour 78 valeurs
l'erreur est 0.00000000000000244249 pour 82 valeurs
l'erreur est 0.00000000000000177636 pour 86 valeurs
l'erreur est 0.00000000000000244249 pour 90 valeurs
l'erreur est 0.00000000000000355271 pour 94 valeurs
l'erreur est 0.00000000000000244249 pour 98 valeurs
```

Ceci représente l'erreur de f(x) = exp(x)*cos(5*x) (fonction 3). On constate que l'erreur est aussi lente à atteindre son minimum pour 26 valeurs. Ce minimum est encore plus faible que pour la fonction 2 et est conservé durant un grand intervalle de taille d'échantillon. Comme le montre l'image ci-dessous, la limite est encore 648 valeurs.

```
ccollet02@gmlin151:/nfs/usersgm/GM27/ccollet02/Bureau$ ./a.out
l'erreur est 0.00000000000001476597 pour 645 valeurs
l'erreur est 0.00000000000001598721 pour 646 valeurs
l'erreur est 0.00000000000001065814 pour 647 valeurs
l'erreur est inf pour 648 valeurs
l'erreur est inf pour 649 valeurs
```

b/ Points équidistants

On choisit l'intervalle [-1;1]. On extrait un certain nombre de points équidistant de cette intervalle. Pour faciliter notre exploitation, nous avions choisi soit trois points (-1,0 et 1) soit 5 points (-1,-0.5,0,0.5 et 1) soit 9 points (-1,-0.75,-0.5,-0.25,0,0.25,0.5,0.75 et 1) et enfin soit 21 points (-1, ; -0,9 ; -0,8 …).

nombre de points	3	5	9	21
erreur	0.00000000000000000000	0.00000000000000000000	0.00000000000000000000	8,135*10^-13

Pour le polynôme, nous trouvons une erreur infime, si bien qu'elle n’apparaît qu'à partir de 21 points alors que nous sommes pourtant en double précision. Elle est supérieur à celle obtenu avec les racines du polynôme de Tchebychev.

nombre de points	3	5	9	21
erreur	2.32778765662550357618	0.56295262302457893533	0.00803749333929815180	1,233*10^-11

Pour la fonction 2, nous observons encore que l'erreur n'est pas atteinte tout de suite, qu'il y a une diminution de l'erreur. Elle est très largement supérieur à l'erreur obtenue avec les racines. Cette dernière était de l'ordre de 10^-15 pour 22 valeurs.

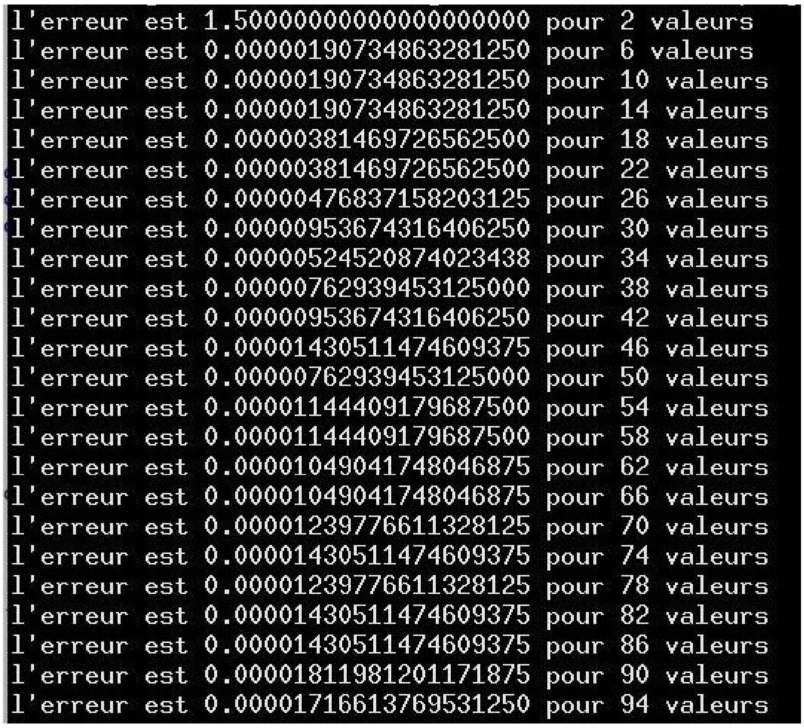
nombre de points	3	5	9	21
erreur	2.34697099561625055486	0.98126778747259257862	0.05888035312417105871	1,52*10-8

Pour la fonction 3, la diminution de l'erreur est encore plus lente et la différence avec l'ancienne erreur est très importante puisque l'erreur n'était que de l'ordre de 10^-13,

2) Simple précision

a/ Racine du polynôme de Tchebychev

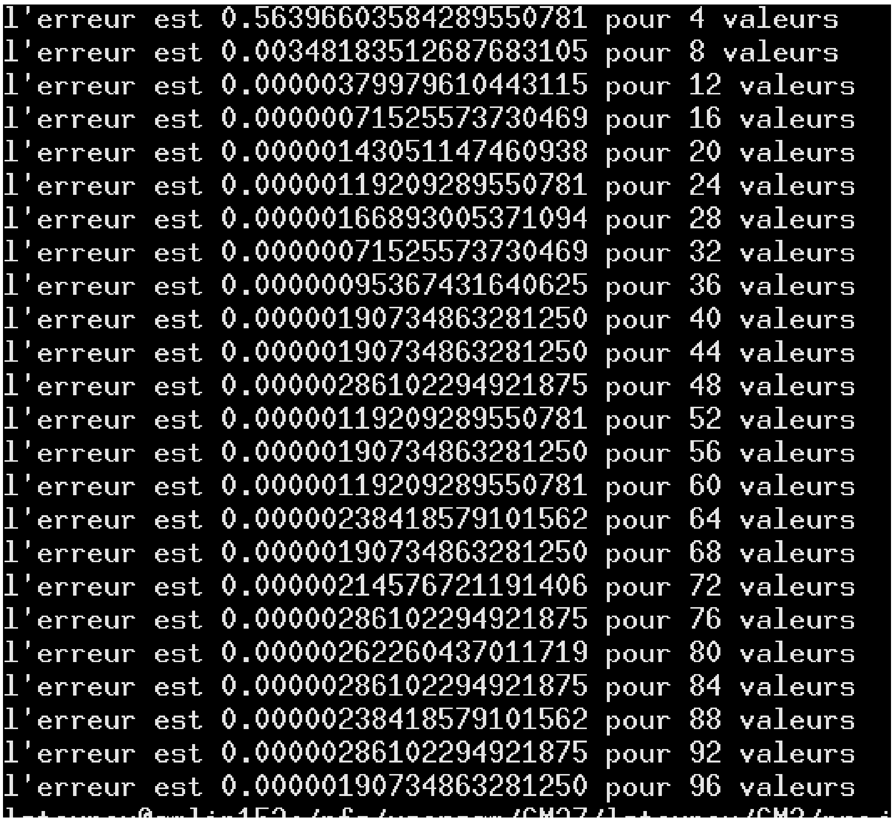
1- Le polynôme



```
l'erreur est 1.50000000000000000000 pour 2 valeurs
l'erreur est 0.00000190734863281250 pour 6 valeurs
l'erreur est 0.00000190734863281250 pour 10 valeurs
l'erreur est 0.00000190734863281250 pour 14 valeurs
l'erreur est 0.00000381469726562500 pour 18 valeurs
l'erreur est 0.00000381469726562500 pour 22 valeurs
l'erreur est 0.00000476837158203125 pour 26 valeurs
l'erreur est 0.00000953674316406250 pour 30 valeurs
l'erreur est 0.00000524520874023438 pour 34 valeurs
l'erreur est 0.00000762939453125000 pour 38 valeurs
l'erreur est 0.00000953674316406250 pour 42 valeurs
l'erreur est 0.00001430511474609375 pour 46 valeurs
l'erreur est 0.00000762939453125000 pour 50 valeurs
l'erreur est 0.00001144409179687500 pour 54 valeurs
l'erreur est 0.00001144409179687500 pour 58 valeurs
l'erreur est 0.00001049041748046875 pour 62 valeurs
l'erreur est 0.00001049041748046875 pour 66 valeurs
l'erreur est 0.00001239776611328125 pour 70 valeurs
l'erreur est 0.00001430511474609375 pour 74 valeurs
l'erreur est 0.00001239776611328125 pour 78 valeurs
l'erreur est 0.00001430511474609375 pour 82 valeurs
l'erreur est 0.00001430511474609375 pour 86 valeurs
l'erreur est 0.00001811981201171875 pour 90 valeurs
l'erreur est 0.00001716613769531250 pour 94 valeurs
```

Ceci représente l'erreur en simple précision de la fonction $f(x)=3x^2+2x+7$. Nous pouvons voir que cette erreur décroit rapidement et stagne rapidement autour de 10^(-5). De plus, lorsque que nous voulons calculer l'erreur pour 95 valeurs, l'ordinateur nous indique que l'erreur est infinie. Pour comparer la double et la simple précision dans ce cas, nous pouvons constater qu'en double précision l'approximation est plus précise, et donc l'erreur est moindre. De plus, le nombre de valeur avant d'obtenir une erreur « infinie » est de 94 en simple précision contre 646 en double précision.

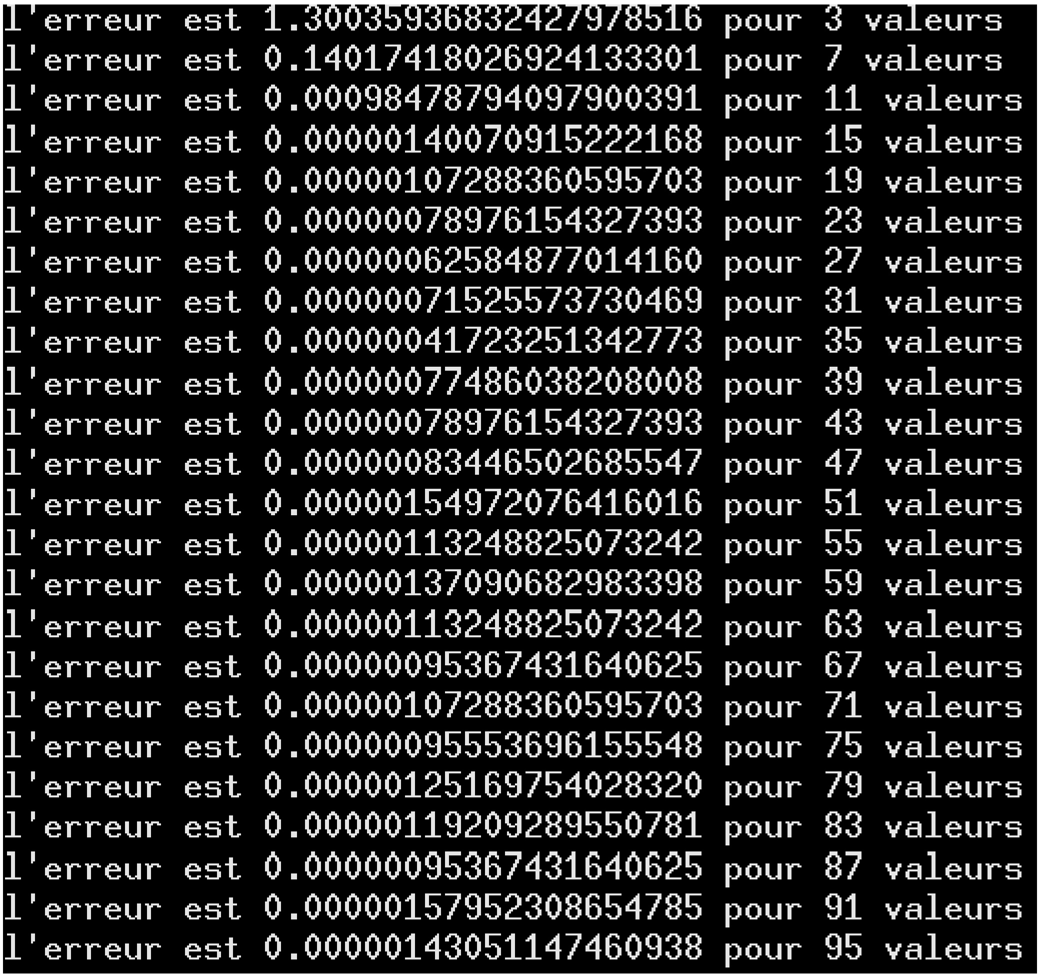
2- La fonction 2



```
l'erreur est 0.56396603584289550781 pour 4 valeurs
l'erreur est 0.00348183512687683105 pour 8 valeurs
l'erreur est 0.00000379979610443115 pour 12 valeurs
l'erreur est 0.00000071525573730469 pour 16 valeurs
l'erreur est 0.00000143051147460938 pour 20 valeurs
l'erreur est 0.00000119209289550781 pour 24 valeurs
l'erreur est 0.00000166893005371094 pour 28 valeurs
l'erreur est 0.00000071525573730469 pour 32 valeurs
l'erreur est 0.00000095367431640625 pour 36 valeurs
l'erreur est 0.00000190734863281250 pour 40 valeurs
l'erreur est 0.00000190734863281250 pour 44 valeurs
l'erreur est 0.00000286102294921875 pour 48 valeurs
l'erreur est 0.00000119209289550781 pour 52 valeurs
l'erreur est 0.00000190734863281250 pour 56 valeurs
l'erreur est 0.00000119209289550781 pour 60 valeurs
l'erreur est 0.00000238418579101562 pour 64 valeurs
l'erreur est 0.00000190734863281250 pour 68 valeurs
l'erreur est 0.00000214576721191406 pour 72 valeurs
l'erreur est 0.00000286102294921875 pour 76 valeurs
l'erreur est 0.00000262260437011719 pour 80 valeurs
l'erreur est 0.00000286102294921875 pour 84 valeurs
l'erreur est 0.00000238418579101562 pour 88 valeurs
l'erreur est 0.00000286102294921875 pour 92 valeurs
l'erreur est 0.00000190734863281250 pour 96 valeurs
```

Ces résultats d'erreurs sont ceux obtenus en simple précisons et en utilisant la fonction f(x)=cos(x)+cos(2*x)+sin(3*x)+sin(4*x). On peut voir ici que l'erreur décroit un peu moins vite, elle stagne autour 2*10^(-6).
Nous pouvons donc voir que cette fonction s'approxime mieux que le polynôme lorsque l'on utilise les racines du polynôme de Tchebytchev.
Il est également possible d'obtenir l'erreur pour 96 valeurs, au delà l'ordinateur nous renverra inf pour le calcul de cette dernière. C'est ici encore, moins précis qu'en double précision (erreur plus élevée), de plus le nombre de valeur avant d'obtenir inf est aussi beaucoup moins élevé.

3- Le fonction 3



```
l'erreur est 1.300359368324279785l6 pour 3 valeurs
l'erreur est 0.14017418026924133301 pour 7 valeurs
l'erreur est 0.000984787940979003 91 pour 11 valeurs
```

b) Points équidistants

1- Le polynôme

```
l'erreur est 0.00000000000000000000 pour 2 valeur
l'erreur est 0.00000095367431640625 pour 3 valeur
l'erreur est 0.00000000000000000000 pour 4 valeur
l'erreur est 0.00000095367431640625 pour 5 valeur
l'erreur est 0.00000190734863281250 pour 6 valeur
l'erreur est 0.00000143051147460938 pour 7 valeur
l'erreur est 0.00000000000000000000 pour 8 valeur
l'erreur est 0.00000286102294921875 pour 9 valeur
l'erreur est 0.00000476837158203125 pour 10 valeur
l'erreur est 0.00000572204589843750 pour 11 valeur
l'erreur est 0.00001335144042968750 pour 12 valeur
l'erreur est 0.00001096725463867188 pour 13 valeur
l'erreur est 0.00001239776611328125 pour 14 valeur
l'erreur est 0.00006341934204101562 pour 15 valeur
l'erreur est 0.00000000000000000000 pour 16 valeur
l'erreur est 0.00013828277587890625 pour 17 valeur
l'erreur est 0.00028896331787109375 pour 18 valeur
l'erreur est 0.00060272216796875000 pour 19 valeur
l'erreur est 0.00134134292602539062 pour 20 valeur
l'erreur est 0.00168561935424804688 pour 21 valeur
l'erreur est 0.00085735321044921875 pour 22 valeur
l'erreur est 0.00585174560546875000 pour 23 valeur
l'erreur est 0.01607322692871093750 pour 24 valeur
l'erreur est 0.00816154479980468750 pour 25 valeur
l'erreur est 0.04246520996093750000 pour 26 valeur
l'erreur est 0.03420400619506835938 pour 27 valeur
l'erreur est 0.08667659759521484375 pour 28 valeur
l'erreur est 0.25674247741699218750 pour 29 valeur
l'erreur est 0.46495056152343750000 pour 30 valeur
l'erreur est 0.82248687744140625000 pour 31 valeur
l'erreur est 0.00000000000000000000 pour 32 valeur
l'erreur est 1.25451087951660156250 pour 33 valeur
l'erreur est 1.64943885803222656250 pour 34 valeur
l'erreur est 4.59261608123779296875 pour 35 valeur
l'erreur est 31.76859664916992187500 pour 36 valeur
l'erreur est 18.25057411193847656250 pour 37 valeur
l'erreur est 60.57248687744140625000 pour 38 valeur
l'erreur est 60.89247131347656250000 pour 39 valeur
```

Voici maintenant l'erreur pour des points pris dans l'intervalle [-1;1]. Pour comparer avec les racines du polynôme de Tchebychev, on peut voir que l'erreur est globalement plus élevé et donc que l'approximation est moins précise. De plus, avec un peu plus de 30 valeurs, l'approximation devient totalement imprécise et l'erreur est très élevé.

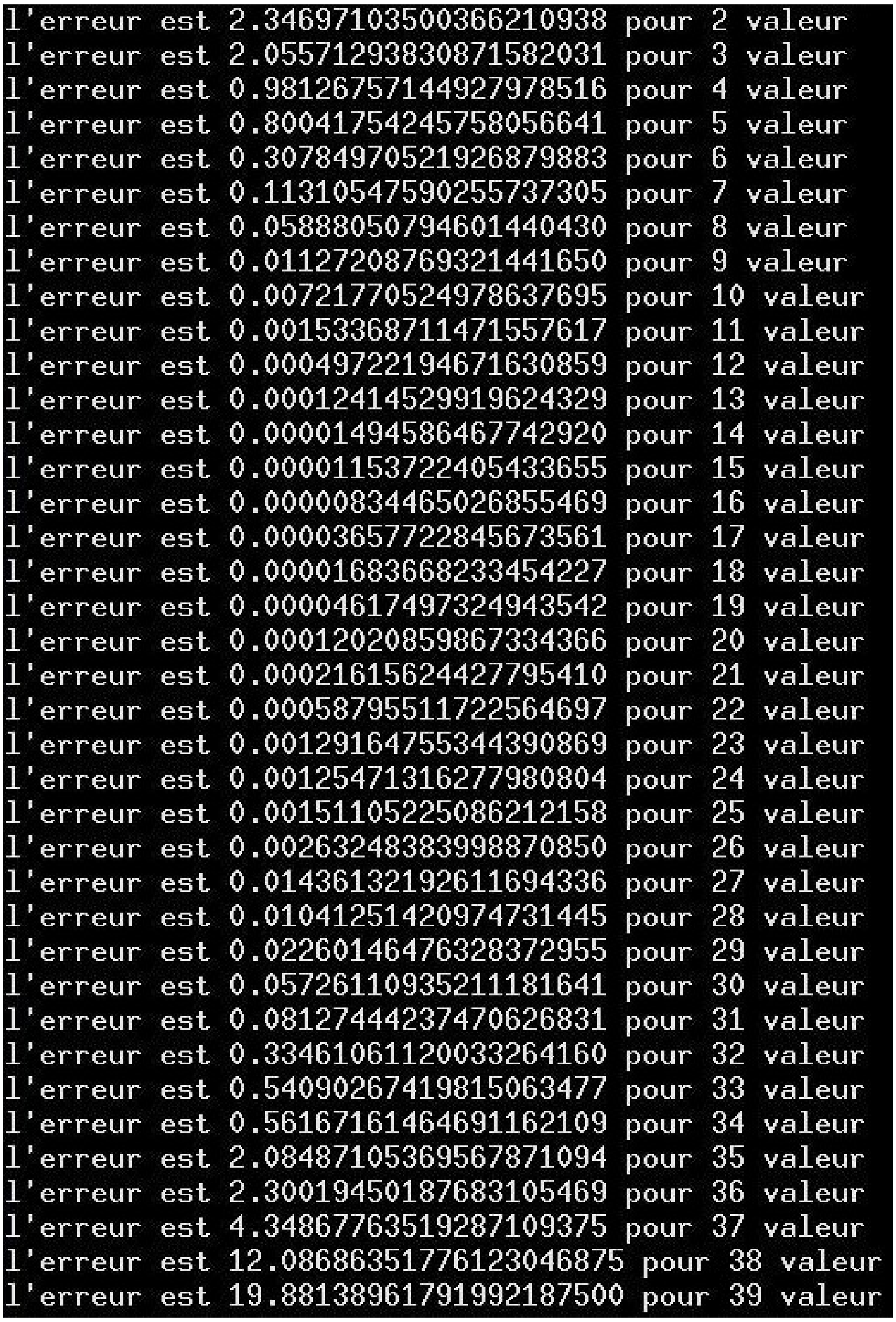
Pour comparer la simple avec la double précision, nous pouvons tirer les mêmes conclusions qu'avec le polynôme de Tchebychev.

2- La fonction2

```
l'erreur est 2.3277876377105712890б pour 2 valeur
l'erreur est 0.85348153114318847656 pour 3 valeur
l'erreur est 0.56295263767242431641 pour 4 valeur
l'erreur est 0.11976981163024902344 pour 5 valeur
l'erreur est 0.08424252271652221680 pour 6 valeur
l'erreur est 0.01113712787628173828 pour 7 valeur
l'erreur est 0.00803759694099426270 pour 8 valeur
l'erreur est 0.00072500109672546387 pour 9 valeur
l'erreur est 0.00052765011787414551 pour 10 valeur
l'erreur est 0.00003400444984436035 pour 11 valeur
l'erreur est 0.00002379715442657471 pour 12 valeur
l'erreur est 0.00000169873237609863 pour 13 valeur
l'erreur est 0.00000703334808349609 pour 14 valeur
l'erreur est 0.00000540912151336670 pour 15 valeur
l'erreur est 0.00000935792922973633 pour 16 valeur
l'erreur est 0.00002036616206169128 pour 17 valeur
l'erreur est 0.00026803836226463318 pour 18 valeur
l'erreur est 0.00016641616821289062 pour 19 valeur
l'erreur est 0.00093439221382141113 pour 20 valeur
l'erreur est 0.00026851892471313477 pour 21 valeur
l'erreur est 0.00091314315795898438 pour 22 valeur
l'erreur est 0.00179284811019897461 pour 23 valeur
l'erreur est 0.00272984802722930908 pour 24 valeur
l'erreur est 0.00632679462432861328 pour 25 valeur
l'erreur est 0.01091654598712921143 pour 26 valeur
l'erreur est 0.01437675952911376953 pour 27 valeur
l'erreur est 0.03592902421951293945 pour 28 valeur
l'erreur est 0.13331437110900878906 pour 29 valeur
l'erreur est 0.27970686554908752441 pour 30 valeur
l'erreur est 0.20518195629119873047 pour 31 valeur
l'erreur est 0.79740476608276367188 pour 32 valeur
l'erreur est 1.29852962493896484375 pour 33 valeur
l'erreur est 2.10754752159118652344 pour 34 valeur
l'erreur est 6.51163101196289062500 pour 35 valeur
l'erreur est 9.54286670684814453125 pour 36 valeur
l'erreur est 12.00980186462402343750 pour 37 valeur
l'erreur est 46.15468597412109375000 pour 38 valeur
l'erreur est 38.58087539672851562500 pour 39 valeur
```

2 Pour cette fonction, nous pouvons également dire qu'avec des points équidistants, l'approximation est moins précise.
Avec un peu plus de 30 valeurs, l'erreur commence également à être très élevée.
Cette fonction s'approxime moins bien que le polynôme avec des points équidistants.
En effet l'erreur est globalement plus élevée ici que l'erreur avec le polynôme.

3- La fonction 3



```
l'erreur est 2.34697103500366210938 pour 2 valeur
l'erreur est 2.05571293830871582031 pour 3 valeur
l'erreur est 0.98126757144927978516 pour 4 valeur
l'erreur est 0.80041754245758056641 pour 5 valeur
l'erreur est 0.30784970521926879883 pour 6 valeur
l'erreur est 0.11310547590255737305 pour 7 valeur
l'erreur est 0.05888050794601440430 pour 8 valeur
l'erreur est 0.01127208769321441650 pour 9 valeur
l'erreur est 0.00721770524978637695 pour 10 valeur
l'erreur est 0.00153368711471557617 pour 11 valeur
l'erreur est 0.00049722194671630859 pour 12 valeur
l'erreur est 0.00012414529919624329 pour 13 valeur
l'erreur est 0.00001494586467742920 pour 14 valeur
l'erreur est 0.00001153722405433655 pour 15 valeur
l'erreur est 0.00000834465026855469 pour 16 valeur
l'erreur est 0.00003657722845673561 pour 17 valeur
l'erreur est 0.00001683668233454227 pour 18 valeur
l'erreur est 0.00004617497324943542 pour 19 valeur
l'erreur est 0.00012020859867334366 pour 20 valeur
l'erreur est 0.00021615624427795410 pour 21 valeur
l'erreur est 0.00058795511722564697 pour 22 valeur
l'erreur est 0.00129164755344390869 pour 23 valeur
l'erreur est 0.00125471316277980804 pour 24 valeur
l'erreur est 0.00151105225086212158 pour 25 valeur
l'erreur est 0.00263248383998870850 pour 26 valeur
l'erreur est 0.01436132192611694336 pour 27 valeur
l'erreur est 0.01041251420974731445 pour 28 valeur
l'erreur est 0.02260146476328372955 pour 29 valeur
l'erreur est 0.05726110935211181641 pour 30 valeur
l'erreur est 0.08127444237470626831 pour 31 valeur
l'erreur est 0.33461061120033264160 pour 32 valeur
l'erreur est 0.54090267419815063477 pour 33 valeur
l'erreur est 0.56167161464691162109 pour 34 valeur
l'erreur est 2.08487105369567871094 pour 35 valeur
l'erreur est 2.30019450187683105469 pour 36 valeur
l'erreur est 4.34867763519287109375 pour 37 valeur
l'erreur est 12.08686351776123046875 pour 38 valeur
l'erreur est 19.88138961791992187500 pour 39 valeur
```

Globalement, les conclusions que nous pouvons tirer avec l'étude de l'erreur sont les mêmes que pour les deux études précédentes.

De plus, l'approximation de la fonction 3 est celle qui, avec des points équidistants, se fait le moins bien.

Conclusion

Comme on pouvait s'y attendre, l'erreur est plus faible en double précision qu'en simple précision. Elle est également plus faible lorsque les points utilisés sont issus du polynôme de Tchebychev que lorsqu'il s'agit de points équidistants. Cependant il est intéressant de noter que toutes les fonctions ne réagissent pas pareil en fonction que les points soient issus ou non des racines du polynôme de Tchebychev. En effet quelque soit la précision, le polynôme s'approxime mieux que les deux autres avec des points équidistants alors que la fonction 3 est celle qui s'approxime le moins bien. C'est le contraire lorsque nous utilisons les points issus des racines du polynôme de Tchebychev.

Quand à l'influence du nombre de points, on remarque que globalement, plus le nombre de points augmente plus l'erreur diminue. Elle stagne ensuite puis augmente très rapidement à partir d'un certain nombre de points. Cette valeur varie en fonction des paramètres (fonction, précision, choix des points).

Ce projet nous a permis de coder une méthode mathématique vu en cours d'analyse numérique et de pouvoir observer le fait qu'avec un petit nombre de points, notre approximation est excellente.

Nous avons également pris du plaisir à travailler sur ce sujet.