TP2 - M43 Analyse numérique

Gloria Faccanoni

On s'intéresse dans ce TP aux diverses méthodes vues en cours et/ou en TD pour le calcul de l'interpolation de Lagrange et d'Hermite d'un ensemble de points.

1 Exercice

On rappelle que le polynôme d'interpolation de Lagrange de degré n d'une fonction f sur l'ensemble des $n+1$ points $\{(x_i, f(x_i))\}_{i=0}^n$ s'écrit

$$p_n(x) = \sum_{i=0}^{n} \big(f(x_i)L_i(x)\big).$$

et que le polynôme d'interpolation de Hermite de degré $N = 2n+1$ d'une fonction f sur l'ensemble des $n+1$ points $\{(x_i, f(x_i))\}_{i=0}^n$ s'écrit

$$Q_N(x) = \sum_{i=0}^{n} \big(f(x_i)A_i(x) + f'(x_i)B_i(x)\big).$$

où

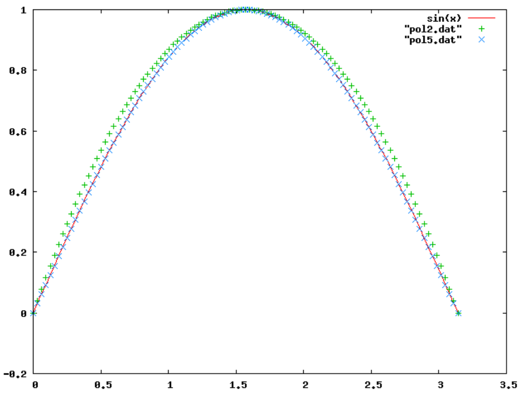
$$\begin{aligned}
L_i(x) &= \prod_{\substack{j=0\\ j\neq i}}^{n} \frac{x-x_j}{x_i-x_j},\\
c_i &= \sum_{\substack{j=0\\ j\neq i}}^{n} \frac{1}{x_i-x_j},\\
A_i(x) &= (1-2(x-x_i)c_i)(L_i(x))^2,\\
B_i(x) &= (x-x_i)(L_i(x))^2.
\end{aligned}$$

1. Soit $f\colon \mathbb{R}\to\mathbb{R}$ la fonction définie par $f(x)=\sin(x)$.
   - 1.1. Calculer (numériquement [1]) les polynômes P_n d'interpolation de Lagrange de degré $n=2$ et $n=5$ sur l'intervalle $[0;\pi]$ de f avec des noeuds équirépartis.
   - 1.2. Dessiner ensuite (par exemple avec gnuplot) dans un même plan les graphes de f et de P_n pour $x\in[0;\pi]$. Pour enregistrer dans un fichier le tableaux des points $(x, P_n(x))$ qu'on utilisera pour comparer f et p_n on pourra utiliser la rédirection ">" lors de l'exécution du programme :

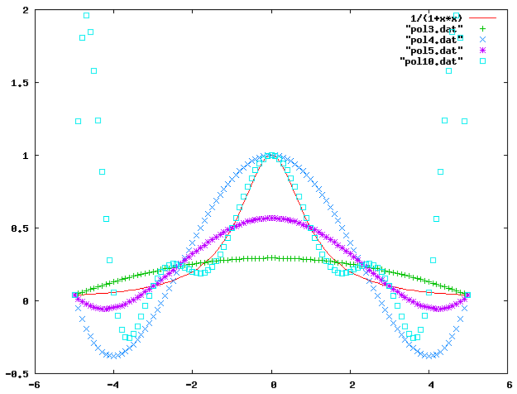
```
gcc -o executable -lm fichier.c
./executable > toto.dat
gnuplot
plot sin(x), 'toto.dat'
```

2. Soit f la fonction de Runge $f\colon \mathbb{R}\to\mathbb{R}$ définie par $f(x)=\frac{1}{1+x^2}$.
   - 2.1. Calculer (numériquement) les polynômes P_n d'interpolation de Lagrange de degré $n=3$, 4, 5 et 10 sur l'intervalle $[-5;5]$ de f avec des noeuds équirépartis.
   - 2.2. Dessiner ensuite dans un même plan les graphes de f et des P_n sur l'intervalle $[-5;5]$.
   - 2.3. Répéter avec les noeuds de Chebychev définis par $x_i = \frac{1}{2}\left(a+b+(b-a)\cos\left(\frac{2i+1}{n+1}\frac{\pi}{2}\right)\right)$, $i=0,\dots,n$.
3. Soit $f\colon \mathbb{R}\to\mathbb{R}$ la fonction définie par $f(x)=\sin(4\pi x)$.
   - 3.1. Calculer (numériquement) les polynômes P_n d'interpolation de Lagrange de degré $n=3$ et Q_N d'interpolation de Hermite sur l'intervalle $[0;1]$ de f avec des noeuds équirépartis.
   - 3.2. Dessiner ensuite dans un même plan les graphes de f, de P_n et de Q_N pour $x\in[0;1]$.
   - 3.3. Répéter avec $n=4$.

1. Conseil : écrire une fonction (informatique) qui reçoit x et n et renvoie $P_n(x)$; dans le `main` on écrira d'abord une boucle qui évalue $P_n(x)$ pour une opportune discrétisation de l'intervalle $[a,b]$ et ensuite on affichera ce tableau qui contient dans la première colonne les valeurs des x et dans la deuxième colonne les correspondants valeurs $P_n(x)$.

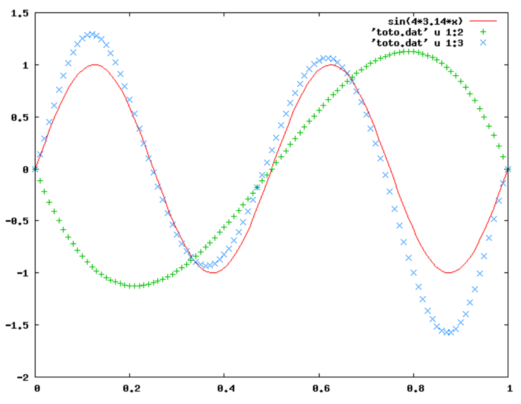


(a) Solution question 1.2

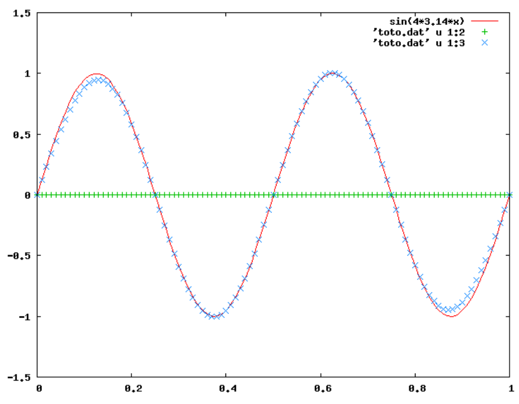


(b) Solution question 1.2 : contre-exemple de Runge concernant l'interpolation de Lagrange sur des noeuds équirépartis.

FIGURE 1: Solutions exercices 1.1 et 1.2.



(a) $n = 3$



(b) $n = 4$

FIGURE 2: Solution question 1.3 : comparaison intérpolations de Lagrange (+) et de Hermite (x) de la fonction $\sin(4\pi x)$.

☎ 0033 (0)4 94 14 23 81
✉ gloria.faccanoni@univ-tln.fr
ℹ http://faccanoni.univ-tln.fr

IMATH Bâtiment U-318
Université du Sud Toulon-Var
Avenue de l'université
83957 LA GARDE - FRANCE

2 Code

lagrange.c

```c
/*
 * File:   lagrange.c
 * Author: Gloria Faccanoni
 *
 * Created on 11 mars 2010, 14:33
 */

#include<stdio.h>
#include<math.h>

/*
 * gcc -o lagrange -lm Lagrange.c && ./lagrange > pol5.dat
 * gnuplot
 * plot sin(x), "pol2.dat" u 1:2, "pol5.dat" u 1:2
 * plot 1/(1+x*x), pol3.dat" u 1:2, "pol4.dat" u 1:2, "pol5.dat" u 1:2
 * plot sin(4*3.14*x), "toto.dat" u 1:2, "toto.dat" u 1:3
 *
 */


int main() {
    int i, j, n, k, m, exo;
    double x[10], y[10], yp[10], t, xv, yvL, yvH, a, b, h;
        double A,B,L,Lp;

    n = 4; // n+1 pts d'interpolation qui fixent le degre du polynome de Lagrange
    exo = 3; // choix de la question

    switch (exo) {
        case 1:
            a = 0.0;
            b = M_PI;
            h = (b - a) / n;
            for (i = 0; i < (n + 1); i++) {
                x[i] = a + i*h;
                y[i] = sin(x[i]);
                yp[i]= cos(x[i]);
            }
            break;
        case 2:
            a = -5.0;
            b = 5.0;
            h = (b - a) / n;
            for (i = 0; i < (n + 1); i++) {
                //x[i] = a + i*h; // noeuds equirepartis
                x[i] = (a+b+(b-a)*cos(0.5*M_PI*(2*i+1.0)/(n+1)))*0.5; // noeuds de Chebychev
                y[i] = 1.0 / (1.0 + x[i] * x[i]); // fct de Runge
                yp[i]= 2.0*x[i]*y[i]*y[i];
            }
            break;
        case 3:
            a = 0.0;
            b = 1.0;
            h = (b - a) / n;
            for (i = 0; i < (n + 1); i++) {
                x[i] = a + i*h;
                y[i] = sin(4.0*M_PI*x[i]);
                yp[i]= 4.0*M_PI*cos(4*M_PI*x[i]);
            }
            break;
    }

    m = 100;
    h = (b - a) / m;
    for (k = 0; k < (m + 1); k++) {
        xv = a + k*h;
        yvL = 0.0;
        yvH = 0.0;
        // Polynomes de Lagrange yvL et de Hermite yvH
        for (i = 0; i < (n + 1); i++) {
            L = 1.0;
            Lp=1.0;
            for (j = 0; j < (n + 1); j++) {
```

```c
                if (i != j) {
                    L *= (xv - x[j]) / (x[i] - x[j]);
                    Lp+= 1.0 / (x[i] - x[j]);
                }
            }
            yvL += L * y[i];
            A = ( 1.0 - 2.0 * (xv-x[i]) * Lp ) * (L * L) ;
            B = (xv-x[i]) * L * L ;
            yvH += A * y[i] + B * yp[i];
        }
        // end polynome de Lagrange et de Hermite
        printf("%lf\t%lf\t%lf\n", xv, yvL, yvH);
    }



    return 0;
}
```