

TP n°3 Corrigé : Intégration numérique (2)

1. Régularité vs. vitesse de convergence

- (a) La fonction f définie sur l'intervalle $[0, 1]$ par $f(x) = |2x^4 - 1|$ est continue sur $[0, 1]$, de classe C^∞ sur chacun des intervalles $]0, 2^{-1/4}[$ et $]2^{-1/4}, 1[$ mais sa dérivée présente une discontinuité au point $x = 2^{-1/4}$ elle n'est donc pas de classe C^1 sur $[0, 1]$.
- (b) Le calcul exact donne

$$\begin{aligned}\int_0^1 f(t) dt &= \int_0^{2^{-1/4}} (1 - 2t^4) dt + \int_{2^{-1/4}}^1 (2t^4 - 1) dt \\ &= \left[t - \frac{2}{5}t^5 \right]_0^{2^{-1/4}} + \left[\frac{2}{5}t^5 - t \right]_{2^{-1/4}}^1 \\ &= -\frac{3}{5} + \frac{4}{5}2^{3/4}.\end{aligned}$$

- (c) Méthode de quadrature de Simpson :

```
function I=Simpson(g,N,a,b)
dx=(b-a)/N;
X=(a:dx:b-dx);
I=sum((g(X)+4*g(X+dx/2)+g(X+dx)))*dx/6;
endfunction
```

- (d) Pour la fonction f considérée :

```
function y=f(x)
y=abs(2*x^4-1);
endfunction
IE=-3/5+4/5*2^(3/4); // integrale exacte
```

```
N=2.^[5:1:14];
IA=[];
for n=N
IA=[IA,Simpson(f,n,0,1)];
end
errIA=abs(IA-IE); // erreur de calcul

scf(0);
plot2d([N',N',N'],[N'.^(-2),N'.^(-4),errIA'],...
logflag="ll",leg="1/N^2@1/N^4@Simpson")
```

Le résultat graphique obtenu est présenté sur la figure 1. La vitesse de convergence s'apparente à une décroissance de l'erreur proportionnelle à $1/N^2$.

- (e) Habituellement (cf. TP n°2) la méthode de Simpson permet d'obtenir une valeur approchée à $O(1/N^4)$ près. Seulement ce résultat nécessite des hypothèses fortes de régularité sur la fonction intégrée (f de classe C^4), hypothèses qui font ici défaut.
- (f) Pour remédier à ce défaut de régularité, on propose donc d'effectuer l'intégration numérique en deux parties, en isolant le point de C^1 discontinuité.

$$\int_0^1 f(t) dt = \int_0^{2^{-1/4}} f(t) dt + \int_{2^{-1/4}}^1 f(t) dt$$

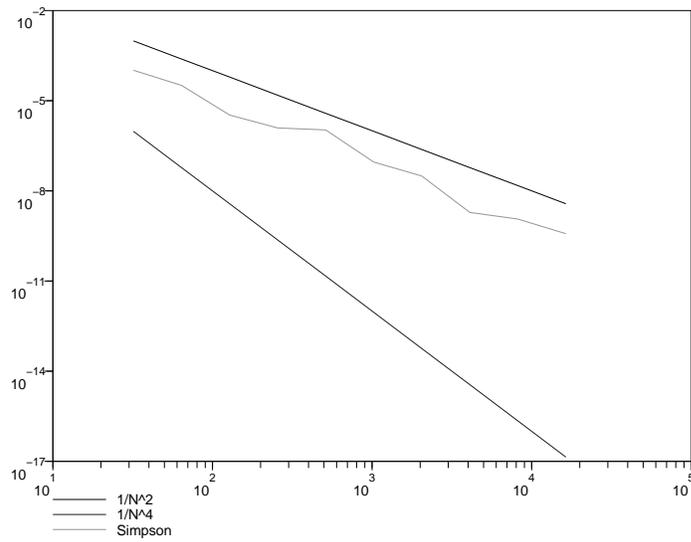


FIG. 1 – $\int_0^1 f(t) dt$ par la méthode de Simpson

```

alpha=2^(-1/4);
IAb=[];
for n=N
IAb=[IAb,Simpson(f,n,0,alpha)+Simpson(f,n,alpha,1)];
end
errIAb=abs(IAb-IE);

scf(1);
plot2d([N',N',N',N'],[N'.^(-2),N'.^(-4),errIA',errIAb'],..
logflag="ll",leg="1/N^2@1/N^4@Simpson@Simpson modifie")

```

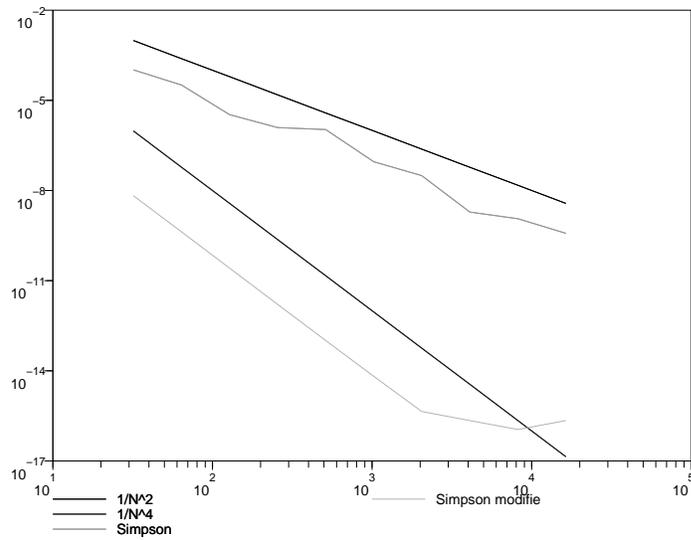


FIG. 2 – $\int_0^1 f(t) dt$ par deux méthodes de Simpson

On observe à présent, sur la figure 2, une réelle décroissance de l'erreur en $1/N^4$. On remarque toujours que lorsque l'erreur de calcul approche la précision machine (de l'ordre de 10^{-15} , alors la décroissance cesse.

2. Méthodes de Gauss

La méthode de Gauss pour le poids ω est de la forme

$$\int_{-1}^1 f(t)\omega(t) dt \simeq \sum_{i=0}^l \lambda_i f(x_i).$$

S'il s'agit plus généralement du calcul de l'intégrale d'une fonction f sur un intervalle $[a, b]$, elle s'écrit

$$\int_a^b f(t)\omega\left(\frac{a+b}{2} + t\frac{b-a}{2}\right) dt \simeq \frac{b-a}{2} \sum_{i=0}^l \lambda_i f\left(\frac{a+b}{2} + x_i\frac{b-a}{2}\right).$$

(a) On implémente ici les méthodes de Gauss-Legendre GL1 pour $l = 1$ et GL2 pour $l = 2$.

```

function I = GL1(g,a,b)
m=(a+b)/2;
dx=(b-a)/2;
I=dx*(g(m-dx/sqrt(3))+g(m+dx/sqrt(3)));
endfunction

function I = GL2(g,a,b)
m=(a+b)/2;
dx=(b-a)/2;
I=dx*(5/9*g(m-dx*sqrt(3/5))+8/9*g(m)+5/9*g(m+sqrt(3/5)));
endfunction

deff('y=f1(x)', 'y=1/(1+x.*x)');
deff('y=f2(x)', 'y=sqrt(x)');

function I=trapeze(g,a,b)
I=(g(a)+g(b))*(b-a)/2;
endfunction

function I=simpson(g,a,b)
dx=(b-a);

```

```

X=(a:dx:b-dx);
I=(g(a)+4*g(a+dx/2)+g(a+dx))*dx/6;
endfunction

disp('Premiere fonction')
disp('Gauss-Legendre 1')
GL1(f1,-1,1)

disp('Gauss-Legendre 2')
GL2(f1,-1,1)

disp('Trapezes')
trapeze(f1,-1,1)

disp('Simpson')
simpson(f1,-1,1)

disp('Deuxieme fonction')
disp('Gauss-Legendre 1')
GL1(f2,0,1)

disp('Gauss-Legendre 2')
GL2(f2,0,1)

disp('Trapezes')
trapeze(f2,0,1)

disp('Simpson')
simpson(f2,0,1)

```

Les résultats obtenus sont :

Premiere fonction

```

Gauss-Legendre 1
ans =
    1.5

```

```

Trapezes
ans =
    1.

```

```

Gauss-Legendre 2
ans =
    1.5833333

```

```

Simpson
ans =
    1.6666667

```

Deuxieme fonction

Gauss-Legendre 1

ans =
0.6738873

Trapezes

ans =
0.5

Gauss-Legendre 2

ans =
0.7211283

Simpson

ans =
0.6380712

(b) Pour la méthode de Gauss-Tchebyshev

```
function I=GT(g,N)
Lambda=%pi/(N+1);
X=cos((1:2:2*N+2)/(2*N+2)*%pi);
I=sum(Lambda.*g(X));
endfunction

deff('y=g1(x)', 'y=sqrt(1+x.*x)');
deff('y=g2(x)', 'y=sqrt(1+x)');

N=int(2.^[1:0.2:6]);
sol1=[]; sol2=[];
for n=N
    sol1=[sol1,GT(g1,n)]; sol2=[sol2,GT(g2,n)];
end

I1=3.8201977890274194;
I2=2.8284271;
err1=abs(sol1-I1);
err2=abs(sol2-I2);
scf(2);
plot2d([N',N'],[err1',err2'],logflag="ll",leg="g1@g2")
```

La convergence est bien plus rapide pour la première des deux fonctions.

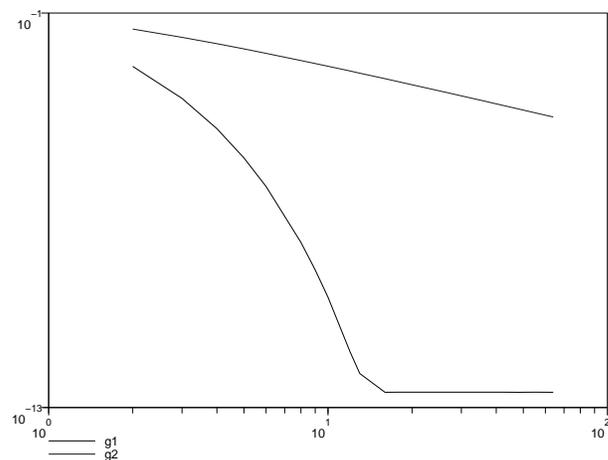


FIG. 3 – Convergence de Gauss-Tchebyshev