X - Quadratures de Gauss. Exemples.

1. Premier exemple.

On définit la fonction `approx_int`, qui donne la valeur approchée de l'intégrale de $\frac{f(x)}{\sqrt{1-x^2}}$ sur $[-1,1]$ et la fonction `exact_int` qui donne la valeur exacte de cette intégrale.

Les deux fonctions `approx_int` et `exact_int` donnent un résultat non évalué car on utilise les formes inertes `Sum` et `Int`. Cela donne ensuite plus de latitude à l'utilisateur :

```
> restart :
  approx_int:=(f,n)->Pi/n*Sum(f(cos((2*j+1)*Pi/(2*n))),j=0..n-1) :
  exact_int:=f->Int(f(x)/sqrt(1-x^2),x=-1..1) :
```

L'égalité suivante représente donc l'approximation que nous allons étudier :

```
> exact_int(f)=approx_int(f,n);
```

$$\int_{-1}^{1} \frac{\mathrm{f}(x)}{\sqrt{1-x^2}}\,dx = \frac{\pi \sum_{j=0}^{n-1} \mathrm{f}(\cos(\frac{1}{2}\,\frac{(2\,j+1)\,\pi}{n}))}{n}$$

Nous allons tout d'abord vérifier l'exactitude de la formule sur les polynômes.

On sait que la formule d'approximation précédente est réellement une égalité pour tous les polynômes de degré inférieur ou égal à $2n-1$. On va le vérifier en considérant un polynôme P quelconque de degré inférieur à 11, c'est-à-dire $2n-1$ avec $n=6$:

```
> P:=x->Sum(a[k]*x^k,k=0..11), 'exact_int(P)'=sort(value(exact_int(P)));
```

$$P := x \to \sum_{k=0}^{11} a_k\,x^k, \mathit{string}; \mathrm{exact\_int}(P) = \frac{3}{8}\,\pi\,a_4 + \frac{5}{16}\,\pi\,a_6 + \frac{35}{128}\,\pi\,a_8 + \frac{63}{256}\,\pi\,a_{10} + \pi\,a_0 + \frac{1}{2}\,\pi\,a_2$$

Dans le calcul ci-dessous, on a écrit les formules d'approximation pour P, à l'ordre $3,4,5,6$.

On note que c'est bien pour $n=6$ que la formule d'approximation de l'intégrale devient en fait une égalité :

```
> for n from 3 to 6 do
    'approx_int'(P,n)=sort(expand(value(approx_int(P,n))));
  od;
```

$$\mathrm{approx\_int}(P,\,3) = \frac{3}{8}\,\pi\,a_4 + \frac{9}{32}\,\pi\,a_6 + \frac{27}{128}\,\pi\,a_8 + \frac{81}{512}\,\pi\,a_{10} + \pi\,a_0 + \frac{1}{2}\,\pi\,a_2$$

$$\mathrm{approx\_int}(P,\,4) = \frac{3}{8}\,\pi\,a_4 + \frac{5}{16}\,\pi\,a_6 + \frac{17}{64}\,\pi\,a_8 + \frac{29}{128}\,\pi\,a_{10} + \pi\,a_0 + \frac{1}{2}\,\pi\,a_2$$

$$\mathrm{approx\_int}(P,\,5) = \frac{3}{8}\,\pi\,a_4 + \frac{5}{16}\,\pi\,a_6 + \frac{35}{128}\,\pi\,a_8 + \frac{125}{512}\,\pi\,a_{10} + \pi\,a_0 + \frac{1}{2}\,\pi\,a_2$$

$$\mathrm{approx\_int}(P,\,6) = \frac{3}{8}\,\pi\,a_4 + \frac{5}{16}\,\pi\,a_6 + \frac{35}{128}\,\pi\,a_8 + \frac{63}{256}\,\pi\,a_{10} + \pi\,a_0 + \frac{1}{2}\,\pi\,a_2$$

Nous allons maintenant traiter un premier exemple non polynômial, et étudier la qualité de l'approximation pour une application simple, $x \mapsto \cos x$.

Voici tout d'abord la valeur exacte de l'intégrale, calculée par Maple avec 15 chiffres significatifs (NB: Maple est incapable de calculer cette intégrale de manière symbolique) :

```
> f:=cos: exact_int(f): r:=evalf(%,15): %%=r;
```

$$\int_{-1}^{1} \frac{\cos(x)}{\sqrt{1-x^2}}\,dx = 2.40393943063441$$

Voici maintenant les valeurs approchées de la même intégrale, suivies de l'erreur commise, pour un nombre de points compris entre 3 et 6. On voit que la formule d'approximation est très précise, puisqu'elle donne 14 chiffres exacts dès $n = 6$:

```
> for n from 3 to 6 do
    'approx_int'(f,n)=evalf(approx_int(f,n),15);evalf(rhs(%)-r,15);
  od;
```

$$\text{approx\_int}(\cos,\ 3) = 2.40407099009524$$

$$.00013155946083$$

$$\text{approx\_int}(\cos,\ 4) = 2.40393883861107$$

$$-.5920233410^{-6}$$

$$\text{approx\_int}(\cos,\ 5) = 2.40393943228728$$

$$.16528710^{-8}$$

$$\text{approx\_int}(\cos,\ 6) = 2.40393943063127$$

$$-.31410^{-11}$$

La théorie montre que la quantité suivante majore en valeur absolue l'erreur commise dans la formule d'approximation (on a majoré par 1 la dérivée n-ième de $f(x) = \cos x$) :

```
> Delta:=n->Pi/2^(2*n-1)/(2*n)!;
```

$$\Delta := n \to \frac{\pi}{2^{(2\,n-1)}\,(2\,n)!}$$

Voici donc un majorant de l'erreur commise dans la formule à 6 points.

Le résultat est tout à fait conforme à ce qui a été observé dans les calculs précédents :

```
> evalf(Delta(6));
```

$$.320245441410^{-11}$$

2. Deuxième exemple.

L'erreur commise dans la formule d'approximation à n points est majorée par une quantité faisant directement intervenir les valeurs de la dérivée $2n$-ième de l'application f.

Nous allons illustrer cette corrélation avec les applications f définies par $f(x) = \frac{1}{1+\lambda^2 x^2}$, où λ est un paramètre réel.

On commence par l'application $x \mapsto \dfrac{1}{1+x^2}$, c'est-à-dire par $\lambda = 1$:

```
> f:=x->1/(1+x^2);
```

$$f := x \to \frac{1}{1+x^2}$$

Maple sait calculer la valeur exacte, et la valeur numérique, de l'intégrale associée à f :

```
> J:=exact_int(f) :
  %=value(%);
  K:=evalf(rhs(%),15) :
  lhs(%%)=K;
```

$$\int_{-1}^{1} \frac{1}{(1+x^2)\sqrt{1-x^2}}\,dx = \frac{1}{2}\,\pi\,\sqrt{2}$$

$$\int_{-1}^{1} \frac{1}{(1+x^2)\sqrt{1-x^2}}\,dx = 2.22144146907919$$

On affiche maintenant les approximations de l'intégrale associée à f dans la formule à n points, avec $n = 5$, $n = 10$, $n = 15$ puis $n = 20$.
On voit que la formule est quasiment exacte sur quinze chiffres significatifs dès que $n = 20$:

```
> for n from 5 to 20 by 5 do
    'approx_int'(f,n)=evalf(approx_int(f,n),15);evalf(rhs(%)-K,15);
  od;
```

$$\text{approx\_int}(f,\ 5) = 2.22210212083180$$

$$.00066065175261$$

$$\text{approx\_int}(f,\ 10) = 2.22144137087021$$

$$-.9820898\,10^{-7}$$

$$\text{approx\_int}(f,\ 15) = 2.22144146909377$$

$$.1458\,10^{-10}$$

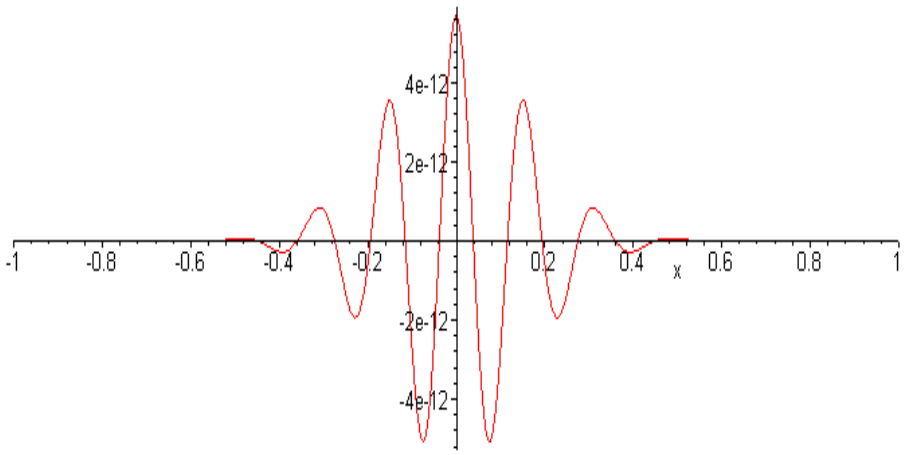
$$\text{approx\_int}(f,\ 20) = 2.22144146907918$$

$$-.1\,10^{-13}$$

On sait qu'un majorant de l'erreur dans la formule à n points est $\dfrac{\pi M}{2^{2n-1}(2n)!}$, où M est un majorant en valeur absolue de la dérivée $2n$-ième de f.

La courbe représentative suivante montre qu'un majorant de l'erreur, pour la formule à 20 points, est de l'ordre de $5 \cdot 10^{-12}$: c'est conforme à ce que nous avons observé, à savoir une erreur réelle de l'ordre de 10^{-14} :

```
> plot(Pi/2^39/(40)!*diff(f(x),x$40),x=-1..1);
```



Nous allons généraliser l'exemple précédent en introduisant la fonction $x \mapsto g(x) = f(\lambda x)$:

```
> lambda:='lambda': g:=x->1/(1+lambda^2*x^2);
```

$$g := x \to \frac{1}{1+\lambda^2\,x^2}$$

On commence par calculer l'intégrale de g "pas à pas" :

```
> J:=exact_int(g);
```

$$J := \int_{-1}^{1} \frac{1}{(1+\lambda^2\,x^2)\,\sqrt{1-x^2}}\,dx$$

Tout d'abord on réduit l'intervalle d'intégration en tenant compte de la parité :

```
> J:=2*subs(-1..1=0..1,J);
```

$$J := 2\int_{0}^{1} \frac{1}{(1+\lambda^2\,x^2)\,\sqrt{1-x^2}}\,dx$$

Puis on utilise le changement de variable $t = \arccos x$:

```
> J:=student[changevar](t=arccos(x),J,t);
```

$$J := 2\int_{0}^{1/2\,\pi} \frac{1}{1+\lambda^2\cos(t)^2}\,dt$$

Puis le changement de variable $u = \tan t$:

```
> J:=simplify(student[changevar](u=tan(t),J,u));
```

$$J := 2\int_0^{\infty} \frac{1}{1+u^2+\lambda^2}\,du$$

Et enfin, pour aller vers une intégrale évidente, le changement de variable $u = v\sqrt{\lambda^2+1}$.

Il faut préciser que *lambda* est réel pour régler le problème de la borne infinie :

```
> assume(lambda,real);
  expand(factor(student[changevar](u=sqrt(lambda^2+1)*v,J,v)));
```

$$2\frac{\displaystyle\int_0^{\infty} \frac{1}{1+v^2}\,dv}{\sqrt{\lambda^2+1}}$$

On sait que l'intégrale qui apparait dans le résultat précédent vaut $\dfrac{\pi}{2}$.

Maple confirme évidemment et on obtient la valeur exacte de l'intégrale J.

Pour $\lambda = 1$, on retrouve la valeur qui avait été obtenue précédemment :

```
> J:=value(J): J:=unapply(J,lambda);
```

$$J := \lambda \to \frac{\pi}{\sqrt{1+\lambda^2}}$$

On calcule maintenant, pour $\lambda = 10$, les valeurs approchées de l'intégrale associée à g, dans la formule à n points, avec $n = 5, 10, 15, 20$.

On constate que notre formule d'approximation est beaucoup moins précise qu'avant, c'est-à-dire pour $\lambda = 1$:

```
> lambda:=10; for n from 5 to 20 by 5 do
    'approx_int'(g,n)=evalf(approx_int(g,n),15); evalf(rhs(%)-J(lambda),15);
  od;
```

$$\lambda := 10$$

$$\text{approx\_int}(g,\ 5) = .677408935763284$$

$$.364808783082051$$

$$\text{approx\_int}(g,\ 10) = .237856346097948$$

$$-.074743806583285$$

$$\text{approx\_int}(g,\ 15) = .345530030691021$$

$$.032929878009788$$

$$\text{approx\_int}(g,\ 20) = .301281644911966$$

$$-.011318507769267$$

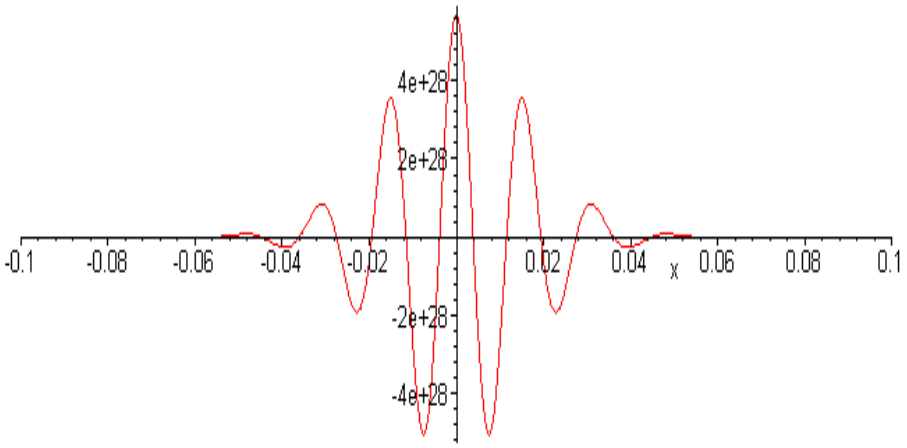
Le problème observé ici a au moins deux explications, qui se recoupent d'ailleurs.

Première explication :

On sait que précision de la formule d'approximation à n points est conditionnée par les valeurs de la dérivée $2n$-ième de l'application. Or nous avons défini l'application g par $g(x) = f(10x)$. Chaque dérivation de g provoque l'apparition d'un facteur 10. Les valeurs de la dérivée 40-ième de g s'obtiennent donc en multipliant par 10^{40} celle de f. On le voit en traçant la courbe représentative de la fonction suivante, dont le maximum en valeur absolue devrait nous donner un majorant de l'erreur commise dans la formule à 20 points.

Le tracé a été centré sur l'intervalle $[-0.1, 0.1]$. On voit que les variations de la dérivée 40-ième de g sont extrêmement brutales sur un petit intervalle de centre l'origine, à tel point qu'on ne peut les utiliser valablement dans le calcul d'un majorant de l'erreur :

```
> plot(Pi/2^39/(40)!*diff(g(x),x$40),x=-0.1..0.1);
```



Deuxième explication :

Voici le tracé simultané de $f(x) = \frac{1}{1+x^2}$ et $x \mapsto g(x) = f(10x)$ sur $[-1, 1]$ (f est au dessus).

On voit comment les valeurs les plus "lourdes" de g sont concentrées au voisinage de l'origine.

On comprend que notre formule d'approximation, qui utilise un échantillon très dispersé de valeurs de g (les abscisses de Chebyshev sont d'ailleurs plus "denses" aux voisinage de -1 et de 1 qu'au voisinage de 0) soit peu adaptée à l'application g :

```
> T1:=plot([f,g],-1..1,thickness=[0,3]) :T1;
```