Mathématiques

Composition n° 1

13 octobre 2014

Durée : 1h30

- **L'usage de la calculatrice est interdit durant l'épreuve.**
- **Le candidat attachera la plus grande importance à la clarté, à la précision et à la concision de la rédaction.**
- **Si le candidat découvre en cours d'épreuve ce qu'il croît être une erreur d'énoncé, il le précisera dans sa copie.**
- **L'épreuve comporte deux exercices**

Bon succès.

Nom : Prénom :

Sauvagardez votre script sous le nom NOM_PRENOM_NO-EXO.sci et envoyez-le par email à l'adresse bernhard.elsner@math.uvsq.fr. Sujet de l'email : Côntrole 1 MA350.

Exercice 1 — *La tour infinie*

Calculer $a^{a^{a^{\cdot^{\cdot^{\cdot}}}}}$ pour $a = \frac{1}{2}$.

Exercice 2 — *Les Simpsons*

1. Rappel : Soient a, b, c trois réels distincts deux à deux et α, β, γ trois réels. On rappelle qu'il existe exactement une fonction polynomiale p de degré au plus deux dont la courbe passe par les trois points (a, α), (b, β) et (c, γ). Elle est donnée par le polynôme d'interpolation de Lagrange :

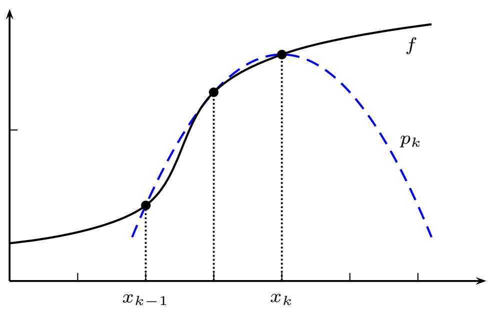
$$p(x) = \alpha\frac{(x-b)(x-c)}{(a-b)(a-c)} + \beta\frac{(x-a)(x-c)}{(b-a)(b-c)} + \gamma\frac{(x-a)(x-b)}{(c-a)(c-b)}.$$

Sur papier et sans ordinateur :

1.a. Vérifiez la formule de Lagrange ci-dessus sur un brouillon.

1.b. Ecrivez ci-dessous une primitive de p. (Il n'est pas nécessaire de regrouper les termes.)

2. En TD on a vu deux méthodes pour calculer une valeur approchée d'une intégrale $\int_a^b f$, à savoir l'approximation par rectangles, puis par trapèzes. La première revient à intégrer une fonction constante par morceaux et la deuxième une fonction affine par morceaux. Sans surpise, on obtient un résultat encore plus précis lorsqu'on utilise des fonctions de second degré par morceaux – c'est la méthode de Thomas Simpson (1710-1761) :
Soit $a = x_0 < x_1 < \ldots < x_n = b$ une subdivision équidistante de l'intervalle $[a, b]$. Pour $k \in [\![1, n]\!]$ on note p_k le polynôme de degré au plus deux qui coïncide avec f aux trois points x_{k-1}, x_k et $(x_{k-1} + x_k)/2$.



On détermine la valeur exacte de $\int_{x_{k-1}}^{x_k} p_k$, puis on pose

$$S_n = \sum_{k=1}^{n} \int_{x_{k-1}}^{x_k} p_k.$$

Si f est une fonction suffisament régulière la suite (S_n) converge vers $\int_a^b f$.

2.a. Programmer avec SciLab la fonction `Primitive(a,b,c,alpha,beta,gamma,x)` vue dans la question précédente.

2.b. Ecrire une fonction `Approx(f,a,b,n)` qui donne la valeur approchée S_n.

2.c. On considère l'exemple $\int_0^\pi \sin$. Déterminer la valeur exacte I de cette intégrale par un calcul à la main, puis tester la méthode de Simpson en vérifiant que l'erreur $|S_n - I|$ tend vers 0 lorsque $n \to \infty$.
L'erreur est un $O(n^{-k})$ avec un certain entier k ; conjecturez lequel !

Corrigé de l'exercice 1 — *La tour infinie*

On trouve environ ≈ 0.6411857.

```
a=.5
y=a
for k=1:25
    y=a^y
    disp(y)
end

deff('y=f(x)','y=2^x*x-1')
l=fsolve(1,f)
disp(l)
```

Corrigé de l'exercice 2 — *Les Simpsons*

```
function y=Primitive(a,b,c,aa,bb,cc,x)
  y=aa*(a-b)^(-1)*(a-c)^(-1)*(x^3/3-(b+c)*x^2/2+b*c*x)+bb*(b-a)^(-1)*(b-c)^(-1)*(x^3/3-(a+c)*x^2/2+a*c*x)
     +cc*(c-b)^(-1)*(c-a)^(-1)*(x^3/3-(b+a)*x^2/2+b*a*x)
endfunction

function va=Approx(f,a,b,n)
  h=(b-a)/n
  x=a
  va=0
  for k=1:n
     va=va+Primitive(x,x+h/2,x+h,f(x),f(x+h/2),f(x+h),x+h)-Primitive(x,x+h/2,x+h,f(x),f(x+h/2),f(x+h),x)
     x=x+h
  end
endfunction
```

On a $\int_0^\pi \sin = 2$. Pour trouver l'ordre de l'erreur on affiche la liste $n^k|S_n - 2|$, $n = 1, \ldots, 20$. Pour $k = 4$ on voit une convergence vers une constante. Donc l'erreur est un $O(n^{-k})$.

```
for n=1:20
  disp(n^4*abs(Approx(sin,0,\%pi,n)-2))
end
```

Alternative : On peut remarquer que $r = Cn^{-k}$ équivaut à $\log(r) = -k\log(n) + \log(C)$; dans une échelle logarithmique où $n' = \log(n)$ et $r' = \log(r)$ cela équivaut à l'équation $r' = -kn' + C'$ qui décrit une droite de coefficient directeur $-k$. En traçant l'erreur dans un système de coordonnées à l'échelle logarithmique (option `logflag='ll'`) on retrouve bien que le coefficient directeur est -4.

```
E=[]
for n=1:20
  E=[E,abs(Approx(sin,0,\%pi,n)-2)]
end
clf(0)
plot2d(E,logflag='ll')
```