Correction du devoir à la maison

Exercice 1 : une solution possible

```
> restart;
> P:=z^10+z^2+1;
```

$$P := z^{10} + z^2 + 1$$

```
> S:=[solve(P,z)];
```

$$S := \Bigg[-\frac{1}{2}+\frac{1}{2} I\sqrt{3},\ -\frac{1}{2}-\frac{1}{2} I\sqrt{3},\ \frac{1}{2}+\frac{1}{2} I\sqrt{3},\ \frac{1}{2}-\frac{1}{2} I\sqrt{3},$$

$$\frac{\frac{1}{6} I\sqrt{6}\sqrt{(100+12\sqrt{69})^{(1/3)}\left((100+12\sqrt{69})^{(2/3)}+4-2\,(100+12\sqrt{69})^{(1/3)}\right)}}{(100+12\sqrt{69})^{(1/3)}},$$

$$\frac{\frac{-1}{6} I\sqrt{6}\sqrt{(100+12\sqrt{69})^{(1/3)}\left((100+12\sqrt{69})^{(2/3)}+4-2\,(100+12\sqrt{69})^{(1/3)}\right)}}{(100+12\sqrt{69})^{(1/3)}},$$

$$\left(-3\,(100+12\sqrt{69})^{(1/3)}\left(-(100+12\sqrt{69})^{(2/3)}-4-4\,(100+12\sqrt{69})^{(1/3)}+\sqrt{3}\,(100+12\sqrt{69})^{(2/3)} I-4 I\sqrt{3}\right)\right)^{(1/2)} \Big/ \left(6\,(100+12\sqrt{69})^{(1/3)}\right),$$

$$-\left(-3\,(100+12\sqrt{69})^{(1/3)}\left(-(100+12\sqrt{69})^{(2/3)}-4-4\,(100+12\sqrt{69})^{(1/3)}+\sqrt{3}\,(100+12\sqrt{69})^{(2/3)} I-4 I\sqrt{3}\right)\right)^{(1/2)} \Big/ \left(6\,(100+12\sqrt{69})^{(1/3)}\right),$$

$$\sqrt{3}\left((100+12\sqrt{69})^{(1/3)}\left((100+12\sqrt{69})^{(2/3)}+4+4\,(100+12\sqrt{69})^{(1/3)}+\sqrt{3}\,(100+12\sqrt{69})^{(2/3)} I-4 I\sqrt{3}\right)\right)^{(1/2)} \Big/ \left(6\,(100+12\sqrt{69})^{(1/3)}\right),$$

$$-\sqrt{3}\left((100+12\sqrt{69})^{(1/3)}\left((100+12\sqrt{69})^{(2/3)}+4+4\,(100+12\sqrt{69})^{(1/3)}+\sqrt{3}\,(100+12\sqrt{69})^{(2/3)} I-4 I\sqrt{3}\right)\right)^{(1/2)} \Big/ \left(6\,(100+12\sqrt{69})^{(1/3)}\right)\Bigg]$$

Liste des valeurs approchées des modules des racines :

```
> [ seq( evalf(abs(S[i])) ,i=1..nops(S) ) ];
```

[1., 1., 1., 1., 0.8688369618, 0.8688369618, 1.072829869, 1.072829869, 1.072829869, 1.072829869]

Liste des valeurs approchées des arguments des racines :

```
> [ seq( evalf(argument(S[i])) ,i=1..nops(S) ) ];
```

[2.094395103, -2.094395103, 1.047197551, -1.047197551, 1.570796327, -1.570796327, -0.3519288607, 2.789663793, 0.3519288608, -2.789663793]

Attention, on demandait comme résultats des **listes** (et non pas des séquences, ou un affichage en colonne comme ce qu'on peut obtenir avec une boucle *for*).

Exercice 2 : une solution possible

```
> restart;
> racines:=proc(P)
  local d,a,b,c,delta;
  d:=degree(P,X);
  if d<>2 then RETURN("Erreur")
  else a:=coeff(P,X,2);
       b:=coeff(P,X,1);
       c:=coeff(P,X,0);
       delta:=b^2-4*a*c;
       if delta<0 then RETURN("Pas de racines réelles")
       elif delta=0 then RETURN(-b/(2*a))
       else
  RETURN((-b-sqrt(delta))/(2*a),(-b+sqrt(delta))/(2*a));
       fi;
  fi;
  end:
> racines(X^3+4);
```

"Erreur"

```
> racines(X^2+1);
```

"Pas de racines réelles"

```
> racines(X^2-2*X+1);
```

$$1$$

```
> racines(X^2+5*X+2);
```

$$-\frac{5}{2}-\frac{\sqrt{17}}{2},\ -\frac{5}{2}+\frac{\sqrt{17}}{2}$$

Plusieurs remarques :

- bien penser à traiter tous les cas ($0<\Delta$, $\Delta=0$, $\Delta<0$).
- attention à la règle de priorité entre la multiplication et la division :

```
> a:=3; b:=7;
  -b/2*a;
  -b/(2*a);
```

$$a := 3$$
$$b := 7$$
$$\frac{-21}{2}$$
$$\frac{-7}{6}$$

Dans le doute, mieux vaut mettre des parenthèses, même si elles sont superflues.

- enfin, notez bien que le test "si degré(P)<>2 alors erreur sinon suite d'instructions qui calculent les racines" se termine en fin de programme (*fi*). Il vaut penser que si degré(P)<>2, le programme n'est pas censé effectuer les instructions qui calculent les racines !

Exercice 3

1a) Un calcul et un dessin (que je ne reproduirai pas ici mais qui étaient exigés) montrent que :

$$\mathrm{Jrect}(f, a, b, n) = \sum_{k=0}^{n-1} \frac{(b-a)\, f\left(a+\frac{k\,(b-a)}{n}\right)}{n}.$$

1b) Voici comment on peut écrire *rectangle* comme une fonction de plusieurs variables, en utilisant la commande *sum*.

```
> restart;
  rectangle:=(f,a,b,n)->sum((b-a)/n * f(a+k*(b-a)/n) , k=0..n-1);
```

$$rectangle := (f, a, b, n) \to \sum_{k=0}^{n-1} \frac{(b-a)\, f\left(a + \frac{k\,(b-a)}{n}\right)}{n}$$

Un exemple :

```
> rectangle(exp,0,5,15);
  evalf(%);
```

$$\frac{1}{3}+\frac{1}{3}\mathbf{e}^{(1/3)}+\frac{1}{3}\mathbf{e}^{(2/3)}+\frac{1}{3}\mathbf{e}+\frac{1}{3}\mathbf{e}^{(4/3)}+\frac{1}{3}\mathbf{e}^{(5/3)}+\frac{1}{3}\mathbf{e}^{2}+\frac{1}{3}\mathbf{e}^{(7/3)}+\frac{1}{3}\mathbf{e}^{(8/3)}+\frac{1}{3}\mathbf{e}^{3}+\frac{1}{3}\mathbf{e}^{(10/3)} + \frac{1}{3}\mathbf{e}^{(11/3)}+\frac{1}{3}\mathbf{e}^{4}+\frac{1}{3}\mathbf{e}^{(13/3)}+\frac{1}{3}\mathbf{e}^{(14/3)}$$

$$124.2067149$$

1c)

```
> f:=x->x^2*ln(x+1);
  a:=0;
  b:=5;
```

$$f := x \to x^2 \ln(x+1)$$
$$a := 0$$
$$b := 5$$

Remarque : on rappelle qu'une *fonction* se définit dans Maple par la flèche (par exemple) : $x \to x^2 \ln(x+1)$. Si on souhaite lui donner un nom (c'est-à-dire la "stocker" dans une variable), on fait une affectation par $:=$. Par exemple, si on veut l'appeler *f*, on fait $f := x\text{->}\, x^2 \ln(x+1)$.

```
> evalf(rectangle(f,a,b,10));
  evalf(rectangle(f,a,b,50));
  evalf(rectangle(f,a,b,100));
```

$$53.12706907$$
$$61.64371382$$
$$62.74976012$$

```
> ?student
> with(student);
```

[D, *Diff*, *Doubleint*, *Int*, *Limit*, *Lineint*, *Product*, *Sum*, *Tripleint*, *changevar*, *completesquare*, *distance*, *equate*, *integrand*, *intercept*, *intparts*, *leftbox*, *leftsum*, *makeproc*, *middlebox*, *middlesum*, *midpoint*, *powsubs*, *rightbox*, *rightsum*, *showtangent*, *simpson*, *slope*, *summand*, *trapezoid*]

```
> evalf(leftsum(f(x),x=a..b,10));
  evalf(leftsum(f(x),x=a..b,50));
  evalf(leftsum(f(x),x=a..b,100));
```

$$53.12706905$$
$$61.64371381$$
$$62.74976010$$

On obtient quasiment les mêmes valeurs (ouf !). Les petits écarts sont dûs à des erreurs d'approximations.

Pour info, l'intégrale $\int_a^b f(x)\,dx$ vaut (en valeur approchée) :

```
> evalf(int(f(x),x=a..b));
```

$$63.86500884$$

1d)

```
> limit(rectangle(f,a,b,n),n=infinity) - int(f(x),x=a..b);
```

$$-\frac{1}{3}\ln(1)+42\ln(6)-42\ln(2)-42\ln(3)$$

```
> simplify(%);
```

$$0$$

Remarque. On demande de montrer que la limite vaut *J*. On ne demande pas de montrer que la limite et *J* ont mêmes valeurs approchées (ce n'est pas suffisant). C'est pour cela qu'on garde des expressions **exactes** pour la limite et pour *J*, et qu'on n'utilise pas la commande *evalf*.

1e) Par définition de la limite, pour tout ε >0, il existe *N* tel que pour tout *n*>=*N* , | Jrect(f,a,b,n) - $\int_a^b f(x)\,dx$ | <= ε. Pour ε = 0.3, il existe *N* tel que pour tout *n* >= *N*, | Jrect(f,a,b,n) - $\int_a^b f(x)\,dx$ | <= 0.3. Il existe donc (au moins un) entier *m* tel que | Jrect(f,a,b,m) - $\int_a^b f(x)\,dx$ | <= 0.3

Pour trouver ce plus petit entier *m*, on procède avec une boucle *while*.

```
> J:=int(f(x),x=a..b):
> m:=1:
  while abs(evalf(J-rectangle(f,a,b,m)))>0.3 do m:=m+1 od:
> m;
```

$$373$$

La réponse est donc m=373.

Remarques :

- le contraire de | Jrect(f,a,b,m) - $\int_a^b f(x)\,dx$ | <= 0.3 est | Jrect(f,a,b,m) - $\int_a^b f(x)\,dx$ | >0.3 (et non pas >=). C'est cette condition qui doit figurer dans le *while*.
- pour que l'expression booléenne | Jrect(f,a,b,m) - $\int_a^b f(x)\,dx$ | >0.3 puisse être évaluée, on doit avoir recours à une expression approchée par *evalf*.
- enfin, je rappelle une dernière fois que la "virgule" des nombres décimaux (notation française) est un point dans Maple (notation anglo-saxone).

2a) Un calcul et un dessin (que je ne reproduirai pas ici mais qui étaient exigés) montrent que :

$$\text{Jtrap}(f, a, b, n) = \sum_{k=0}^{n-1} \frac{(b-a)\left(f\left(a+\frac{k\,(b-a)}{n}\right)+f\left(a+\frac{(k+1)\,(b-a)}{n}\right)\right)}{2\,n}.$$

2b) Comme une procédure (pour changer) :

```
> trapeze:=proc(f,a,b,n)
  local h;
  h:=(b-a)/n;
  RETURN(sum(h/2 * (f(a+k*h)+f(a+(k+1)*h)), k=0..n-1) );
  end:
```

```
> trapeze(exp,0,5,15);
  evalf(%);
```

$$\frac{1}{3}\mathbf{e}^{(2/3)}+\frac{1}{3}\mathbf{e}^{(1/3)}+\frac{1}{3}\mathbf{e}^{(4/3)}+\frac{1}{3}\mathbf{e}^{2}+\frac{1}{3}\mathbf{e}^{(5/3)}+\frac{1}{3}\mathbf{e}^{(8/3)}+\frac{1}{3}\mathbf{e}^{(7/3)}+\frac{1}{3}\mathbf{e}^{3}+\frac{1}{3}\mathbf{e}^{(10/3)}+\frac{1}{3}\mathbf{e}^{(13/3)}$$
$$+\frac{1}{3}\mathbf{e}^{4}+\frac{1}{3}\mathbf{e}^{(11/3)}+\frac{1}{3}\mathbf{e}^{(14/3)}+\frac{1}{6}\mathbf{e}^{5}+\frac{1}{6}+\frac{1}{3}\mathbf{e}$$

$$148.7755747$$

2c)

```
> f(x) ; a ; b ;
```

$$x^2\ln(x+1)$$
$$0$$
$$5$$

```
> evalf(trapeze(f,a,b,10));
  evalf(trapeze(f,a,b,50));
  evalf(trapeze(f,a,b,100));
```

$$64.32556575$$
$$63.88341317$$
$$63.86960978$$

```
> evalf( trapezoid(f(x),x=a..b,10) );
  evalf( trapezoid(f(x),x=a..b,50) );
  evalf( trapezoid(f(x),x=a..b,100) );
```

$$64.32556573$$
$$63.88341315$$
$$63.86960977$$

On obtient quasiment les mêmes valeurs (ouf !). Les petits écarts sont dûs à des erreurs d'approximations.

2d) Mêmes remarques qu'en 1d)

```
> limit(trapeze(f,a,b,n),n=infinity)-int(f(x),x=a..b);
```

$$42\ln(6)-42\ln(2)-42\ln(3)$$

```
> simplify(%);
```

$$0$$

2e) Le même raisonnement qu'à la question 1e donne l'existence de *p*.

```
> p:=1:
  while abs(evalf(J-trapeze(f,a,b,p)))>0.3 do p:=p+1 od:
> p;
```

$$13$$

La réponse est *p=13*. On remarque que ce nombre est bien meilleur que celui obtenu par la méthode des rectangles. Cela signifie que pour cette fonction *f*, pour obtenir une erreur <0.3 sur le calcul de l'intégrale, on doit utiliser moins de subdivisions *n* avec les trapèzes qu'avec les rectangles. Cela laisse penser que quand n-> ∞, la méthode des trapèzes converge plus vite vers l'intégrale $\int_a^b \mathrm{f}(x)\,dx$ que la méthode des rectangles.