Université de Paris 7 | **TP01 Découverte logiciel** | **agreg 2020 (F.Han [1])**

Minimum à retenir/réviser avant l'épreuve, et donc à archiver cette annnée :

1) Etre habitué à utiliser l'aide.

2) opérations classiques en mode exact, approché, algèbre linéaire, calcul modulaire et polynomial. Notamment, pgcd, couples de Bezout, produit de matrices, noyau (y compris pour un élément de $M_n(\mathbb{Z}/p\mathbb{Z})$,

3) Quelques instructions de programmation : boucles `for`, `while` , suite : `seq`. Créer une fonction, créer une fonction à partir d'un symbole.

4) Quelques instructions graphique de base, et surtout savoir obtenir ces fonctions et leurs options via les menus déroulant. Ex fonctions (paramétriques, implicites, statistiques ; ex histogrammes) et géométrie (points, droites...) et options d'affichage via le menu déroulant.

1 Présentation Xcas et Sage

1.1 Xcas

- Types de base : entiers, flottants multiprécision, symboles, fonctions, nombres modulaires, corps finis non premiers.
- Listes : deux types : `()` et `[]` . Le niveaux de parenthèses s'applatissent et ne comptent pas, en revanche les niveaux de crochets comptent.
- ensembles

1.2 Sage

- Une multitude de types, il faut déclarer chaque variable, créer les anneaux dans lesquels on souhaite travailler et savoir régulièrement changer d'anneaux. Le symbolic ring `SR` a une philosophie similaire à celle d'Xcas, mais il ne suffira pas sous sage. (`PolynomialRing` ...)
- Idem pour les Listes : beaucoup de types différents : t-uples, listes, vecteurs, matrices.... Les niveaux de parenthèses et de crochets comptent. Pour les applatir on utilise `flatten`

Exemple : Un calcul nous amène à une variable `F` contenant `(x^4-4)/(x-sqrt(2))`. Nous savons que `F` est en fait un polynôme en `x` et souhaitons calculer son pgcd avec par exemple `x^2-2`.

(giac/xcas)

```
0>> F:=(x^4-4)/(x-sqrt(2))
1>> gcd(F,x^2-2);
x+sqrt(2)
2>> normal(F);
x^3+sqrt(2)*x^2+2*x+2*sqrt(2)
```

(sage)

```
# sage 7.4 (cf. clefagreg 9.0, paquet sage disponible en 2019)
sage: F=(x^4-4)/(x-sqrt(2))
sage: F.gcd(x^2-2)  # calcul dans SR impossible ds 7.4
...
ValueError: gcd: arguments must be polynomials over the rationals
sage: R.<x>=PolynomialRing(QQ[sqrt(2)])
sage: R(F)    # pas de coercion disponible pour sqrt(2) en sage 7.4
...
TypeError: <type 'sage.symbolic.expression.Expression'>

sage: Q2.<sqrt2>=QQ[sqrt(2)]
sage: F=(x^4-4)/(x-sqrt2)  # il faut donc travailler des le debut dans l'anneau R
sage: F.gcd(x^2-2)
x + sqrt2
```

1. `http://webusers.imj-prg/frederic.han/agreg`

```
# avec sage 8.1
sage: F=(x^4-4)/(x-sqrt(2))
sage: F.gcd(x^2-2)   # PB: il n'y a pas eu simplification
x^2 - 2
sage: R.<x>=PolynomialRing(QQ[sqrt(2)])
sage: R(F)
x^3 + sqrt2*x^2 + 2*x + 2*sqrt2
sage: R(F).gcd(x^2-2)
x + sqrt2
```

1.3 Choisir sa configuration Xcas

⚠ Lors du premier lancement, les valeurs par défaut ne coïncident pas forcément avec vos préférences. Au premier lancement on répond à quelques questions. Si vous n'avez pas de préférences, choisissez le mode xcas car c'est celui de référence par défaut dans la documentation. Si vous connaissez python, vous pouvez tenter le mode Python qui permet d'utiliser quelques syntaxes python (Soit dans le menu `Cfg>configuration du cas` ou en tapant `python_compat(1)`). Le mode maple est pour les experts en maple, et le mode autre est pour le mode tortue logo...

On peut ensuite modifier/affiner ses choix dans le menu `Cfg`. Ensuite, sauvez vos préférences grâce au menu. Il faut donc choisir le mode xcas (éventuellement python) et étudier la configuration du "cas" pour ne pas mal interpréter les réponses. (On peut l'obtenir rapidement en cliquant sur la barre grise entre les boutons SAUVER et STOP)

Pour la configuration du "cas", je conseille de :
COCHER `radians`, et de DÉCOCHER : `approx`, `complex`, `Cmpx_var`, `Sqrt`. On travaillera ainsi en mode exact dans le corps engendré par les coefficients de l'expression, alors que si l'on coche complex ou Sqrt, on rajoute i et des racines carrées ce qui ne permet pas par exemple une factorisation de polynôme sur un corps prescrit.

2 Calcul symbolique, calcul approché

Observez dans le Tutoriel :
Dans le tutoriel :

- Pour Xcas MENU AIDE>DÉBUTER EN CALCUL FORMEL
- MENU HELP/ TUTORIEL SAGE

on trouve facilement comment manier les objets du calcul formel.

Ex : comment affecter une variable, définir une fonction, substituer

C'est aussi un endroit où l'on trouve facilement la syntaxe complète de programmation.

2.1 Découverte du logiciel

Exercice I:

1) a) Comment obtenir de l'aide sur un mot donné que l'on connait à peu près (par exemple sqrt) on fait : ?sqrt. Par exemple, comment fait on $\sqrt[3]{23}$? Rangez le dans la variable `z`

b) Donner une valeur approchée de `z` à la précision par défaut. (Soit en forçant le logiciel a être en flottants (Ex 1.)

c) Trouver la syntaxe des constantes réelles ou complexes π, e, i (où $i^2 = -1$). Faire afficher (sans que la précision par défaut soit modifiée pour la suite.) les 1000 premières décimales de π.

2) On trouve aussi dans le tutoriel des explications sur les différentes notions de développer et simplifier.

L'interface : Apprenez à supprimer, copier, déplacer une ligne, un dessin, un programme...(ie l'entrée et la réponse)

3) Aide bulle sur un mot connu : Tapez `seq(` et étudiez les paramètres de cette fonction avec l'aide bulle [2]. Cette méthode est utile lorsque l'on connait le mot mais pas très bien les arguments ou les options.

4) Etudier le menu aide. Référence [3] calcul formel. Cette méthode est la plus adaptée (avec les menus déroulants, ou bien lorque ces derniers sont trop succincts) pour trouver une instruction dont on ne connait pas le mot clef.

Par exemple trouver la fonction pgcd de deux entiers en utilisant ce menu.

a) Comment ajouter un élément à une liste ? Ex ajoutez l'élément 55 à la liste `l:=[1,33,4]`. La liste `l` a t'elle été modifiée ?

(giac/xcas)

```
30>> l:=[1,33,4];
[1,33,4]
31>> augment(l,55);
[1,33,4,55]
32>> l;
[1,33,4]
33>> l:=append(l,55);
[1,33,4,55]
34>> l;
[1,33,4,55]
```

(sage)

```
l=[1,33,4]
l.append(55)  # ici l est modifiee
l+l           # concatenation des listes
a=11111
del(a)        # pour liberer a
```

b) `a:=1111;`. Comment libérer la variable `a` ?

(giac/xcas)

```
a:=11111;
purge(a); // cf menu deroulant Prg memoire
a;
a:=11111; del(a) ; // fonctionne aussi TB
```

(sage)

```
a=11111
del(a)        # pour liberer a
```

c) On fait avec la liste précédente : `l2:=l ; l2[0]:=66`. La liste `l` a t'elle été modifiée ? (Dans Xcas, si l'on souhaite une affectation par adresse mémoire on pourra utiliser `=<`)

(giac/xcas)

```
48>>  l2:=l;
[1,33,4,55]
49>>  l2[0]:=66;
[66,33,4,55]
50>>  l;
[1,33,4,55]
51>>  l3=<l;
[1,33,4,55]
52>>  l3[0]=<77;
[77,33,4,55]
53>>  l;
[77,33,4,55]
```

2. Laisser la souris sur la ligne avec la parenthèse ouverte, et attendre que la bulle apparaisse.
3. C'est le plus utile. à retenir

(sage)

```
sage: l=[1,33,4,55]
sage: l
[1, 33, 4, 55]
sage: l2=l
sage: l2[0]=66
sage: l
[66, 33, 4, 55]
sage: l2
[66, 33, 4, 55]
```

5) Affectez à `a` la valeur $e^{2i\pi/7}$. Calculez/Simplifiez $\sum_{i=10}^{16} a^i$. Essayez de simplifier $\sqrt[3]{7}+\sqrt{5}$ expliquez la notion de polynôme pour xcas et en déduire la signification d'un `rootof`. Quel est l'intérêt d'un "rootof" pour des calculs du type $(\sqrt[3]{7}+\sqrt{5})^{100}$?

(giac/xcas)

```
54>>  a:=exp(2*i*pi/7);
exp(2*i*pi/7)
55>>  S:=sum(a**k,k,10,16);
exp(2*i*pi/7)^16+exp(2*i*pi/7)^15+exp(2*i*pi/7)^13+exp(2*i*pi/7)^12+exp(2*i*pi/7)^11+exp(2*i*p
56>>  simplify(S);
0
57>>  simplify(a);
rootof([[1,0,0],[1,-1,1,-1,1,-1,1]])

/* Pour trouver 0 pour S on a eu de la chance que xcas transforme a en une classe
dans un corps de rupture. Les calculs sont alors bien plus simple puisqu'il
suffit de faire une division Euclidienne de la somme de la somme en a^k par
 le polynome minimal de a */
```

(sage)

```
sage: a=exp(2*i*pi/7)
sage: S=sum([a**k for k in range(10,17)])
sage: S.simplify()  # ne donne rien (sage 8.1) car a n'est pas defini algebriquement.
e^(6/7*I*pi) + e^(4/7*I*pi) + e^(2/7*I*pi) + e^(-2/7*I*pi) + e^(-4/7*I*pi) + e^(-6/7*I*pi) + 1
sage: # il faudra creer un corps de rupture plutot que le SR
```

6) Rangez la matrice $\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}$ dans la variable `M1`. Calculez le produit de `M1` avec sa transposée, calculez le noyau de `M1`. Rangez le vecteur $(1,2,2)$ dans la variable `v1` et calculez `M1*v1` et le produit scalaire de `v1` avec lui même.

(giac/xcas)

```
58>>  M1:=[[1,2,3],[4,5,6]];
[[1,2,3],[4,5,6]]
59>>  M1*transpose(M1);
matrix[[14,32],[32,77]]
60>>  v1:=[1,2,2];
[1,2,2]
61>>  M1*v1;
[11,26]
62>>  v1*v1;
9
63>>  ker(M1);
[[-1,2,-1]]
```

(sage)

```
sage: M1=matrix([[1,2,3],[4,5,6]]);
sage: M1.kernel()    # attention c'est un noyau a gauche !!
```

```
Free module of degree 2 and rank 0 over Integer Ring
Echelon basis matrix:
[]
sage: M1.right_kernel()

Free module of degree 3 and rank 1 over Integer Ring
Echelon basis matrix:
[ 1 -2  1]
```

L'index (ou complétion par tabulation). Utile pour obtenir un mot clef et ses synonymes, mais aussi des intructions connexes non équivalentes.

On peut aussi obtenir l'index par le menu aide.

2.2 Symboles, fonctions, programmes

Dans vos logiciels de calculs formel, il ne faut pas confondre les symboles, les fonctions, et même les petits programmes que l'on fait. Un progamme du type :

```
f(x):={
   if(x<0){
     return 0
   };
   else{
     return x;
   }
}
```

ou

```
def f(x):
   if(x<0):
     return 0
   else:
     return x
```

ne pourra pas être utilisé pour dériver f formellement.

Il faut maintenant maitriser la différence entre fonctions et symboles, car certaines instructions attendent des symboles et d'autres des fonctions.

Exercice II:

Par exemple on définit le symbole `P:=x/(x^2+1)` et la fonction `Q(x):=sin(1/x)` (sous xcas on peut aussi faire de manière équivalente : `Q:=x->sin(1/x)`)

a) Etudiez les types de `P Q Q(x) Q(5) Q(6/pi)` (on pourra simplifier). D'une manière générale, il vaut mieux effectuer les calculs compliqués avec des symboles plutot qu'avec des fonctions. Les résultats seront le plus souvent bien plus lisibles et rapides.

b) Calculer les dérivées de P et Q.

c) (A retenir !) Comment [4] obtenir à partir du symbole P défini ci dessus, la fonction $x \mapsto \dfrac{x}{x^2+1}$?

En déduire la dérivée de $\dfrac{\sin(1/x)}{(\sin(1/x))^2+1}$

(giac/xcas)

```
64>>  purge(x);
"No such variable x"
65>>  x=='x';
vrai
66>>  P:=x/(x^2+1);// un symbole
x/(x^2+1)
67>>  Q:=x->sin(1/x);// une fonction
 (x)->sin(1/x)
68>>  type(P);
expression
69>>  type(Q);
func
70>>  type(Q(x));
expression
71>>  type(Q(5));//un symbole car sin(1/5) est laisse tel quel
```

4. Cf tutoriel débuter en calcul formel

```
expression
72>>  type(normal(Q(6/pi)));// un rationel car c'est une valeur remarquable (si l'on a simplif
rational
81>>  diff(P,x);
(x^2+1-x*2*x)/(x^2+1)^2
82>>  function_diff(Q);
 (` x`)->-1/` x`^2*cos(1/` x`)
83>>  Q';
 (x)->-1/x^2*cos(1/x)
84>>  unapply(P,x);// cr\'ee la fonction P (cf tuto debuter en calcul formel)
 (x)->x/(x^2+1)
85>>  simplify(function_diff(unapply(P,x)(Q))); // cr\'ee la fonction P et la compose avec Q
(` x`)->(-1/` x`^2*cos(1/` x`)*(sin(1/` x`)^2+1)+sin(1/` x`)*2/` x`^2*cos(1/` x`)*sin(1/` x`))/
86>>  simplify(diff(sin(1/x)/(sin(1/x)^2+1),x));
-cos(1/x)^3/(x^2*cos(1/x)^4-4*x^2*cos(1/x)^2+4*x^2)
```

(sage)

```
sage: x=var('x')
sage: P=x^2+1
sage: Q(x)=sin(1/x)
sage: type(P)
<type 'sage.symbolic.expression.Expression'>
sage: type(Q)
<type 'sage.symbolic.expression.Expression'>
sage: Q
x |--> sin(1/x)
sage: P
x^2 + 1
sage: diff(P,x)
2*x
sage: Q'
  File "<ipython-input-47-f13e18ce214c>", line 1
    Q'
     ^
SyntaxError: EOL while scanning string literal

sage: diff(Q)
x |--> -cos(1/x)/x^2
sage: fP(y)=P.subs(x=y)
sage: fP
y |--> y^2 + 1
```

Exercice III:

1) Exprimer $\cos 5a$ en fonction de $\cos a$ (où a est une variable formelle).

2) Calculer $\int_0^{\frac{\pi}{2}} \frac{1}{2+\sin(5x)} dx$. Dans quels cas de fractions trigonométriques pensez vous que votre logiciel ne puisse pas vous donner la réponse ?

3) Calculez $\int \frac{1}{2+\sin(5x)}$. Affichez un dessin de la réponse pour vérifier la continuité.

2.3 Dessins, calcul formel

Exercice IV: (cf Ecrit 2007)

On accèdera aux fonctions graphiques facilement via le menu déroulant.

1) Créer une fonction : $M : t \mapsto (\frac{\cos t}{\sin^3 t}, \frac{\sin t}{\sin^3 t})$

2) a) En mode géométrie, dessiner le point $M(\pi/2)$

b) Déssiner la courbe $C_1 : t \mapsto M(t)$. (Cf menu déroulant pour trouver l'instruction)

c) En utilisant la commande `assume(t1=[0.7,0,Pi]);` en géométrie intéractive [5], utilisez le bouton apparu à droite pour expliquez ce qu'est une courbe paramétrée en affichant le point `point(M(t1))`. Sous Xcas, on peut insérer des `attributs` graphiques via le menu déroulant.

d) Ajoutez aussi l'élément `t2` de $]0..\pi[$ et dessiner $M(t_1), M(t_2), M(-t_1-t_2)$. Afficher ces points avec une grosse croix en taille un peu plus grande.

e) Dessiner la droite $M(t1), M(t2)$ en bleu. Que conjecturez vous ?

3) Démontrez votre conjecture par un calcul formel.

4) a) Exprimez les coordonnées de $M(t)$ en fonction de $u = \tan(t/2)$. On pourra utiliser une instruction logicielle pour passer en $\tan(t/2)$ puis subtituer les `tan(t/2)` par des `u` ou trouver une méthode directe. Stockez les dans `xu` et `yu`.

b) On considère l'équation (un des buts de l'option C sera de bien comprendre ce que font ces commandes) :
`mystere:=factors(resultant(denom(xu)*x-numer(xu),denom(yu)*y-numer(yu),u))[2];` ou bien celle ci : `mystere2:=eliminate([x=xu,y=yu],u)[0];` Dessinez la.

c) Démontrez que $M(t)$ appartient à cette courbe.

2.4 Petit programme

Exercice V: Illustration Newton

Dans cet exercice nous allons illustrer ce que représente une convergence quadratique : ie ce que représente une majoration du type :

$$|l - u_{n+1}| \leq C.|l - u_n|^2$$

On rencontre par exemple ce genre de majoration dans la méthode de Newton pour trouver un zéro non multiple d'une fonction f assez régulière.

Nous allons ici prendre $f : x \mapsto x^2 - 2$ et trouver une valeur approchée de $\sqrt{2}$.

1) Comparez à la main la méthode de Newton $x_{n+1} = x_n - \dfrac{f(x_n)}{f'(x_n)}$ avec celle de Heron qui calcule $\sqrt{d}$ en définissant u_{n+1} comme la moyenne de u_n et $\dfrac{d}{u_n}$.

2) On considère la suite récurrente :

$$x_{n+1} = \frac{x_n}{2} + \frac{1}{x_n},\ x_0 = 2$$

Créez une fonction `newt(n,d)` qui calcule de manière itérative x_n avec d chiffres significatifs. ⚠ On prendra soin de faire tous les calculs en valeur approchée avec d chiffres significatifs. Evitez de modifier la variable `Digits` car il faut bien comprendre sa portée, et préferez une fonction qui convertisse un entier en un flottant de précision d.

3) Pour $0 \leq n \leq 12$ donnez la valeur exacte de

$$E(\log_{10}(|x_n - \sqrt{2}|))$$

et expliquez grâce à cette illustration la notion de convergence quadratique en language commun.

(giac/xcas)

```
sq2:=approx(sqrt(2),10000);

newt(n,d):={
  local u,j;
  u:=approx(2,d);
  for(j:=0;j<n;j++){
    u:=u/2+1/u;
  }
  return u;
}
```

5. Sous sage on peut regarder *interact*

```
54>> newt(5,10);
1.41421356237
/* Il faut augmenter la precision pour que la difference avec sqrt(2) soit non
nulle. Asymptotiquement le nombre de chiffres exacts est multiplie par 2 d'une
iteration sur l'autre. */
55>> seq(floor(log10(abs(newt(k,5000)-sqrt(2)))),k=0..12);
-1,-2,-3,-6,-12,-25,-49,-98,-196,-392,-784,-1568,-3136
// Time 0.43
56>> [ floor(log10(abs(newt(k,5000)-sqrt(2))))  for k in range(13) ];
[-1,-2,-3,-6,-12,-25,-49,-98,-196,-392,-784,-1568,-3136]
```

2.5 Développer/factoriser

Exercice VI:

1) Trouver le coefficient de $a^3b^2cd^2$ dans $(a+b+c+d)^8$. Vérifier votre réponse théorique avec la forme développée (utiliser `normal` pour développer un calcul lourd). (On pourra étudier l'aide de coeff pour recupérer le coefficient d'un monôme).

Exercice VII:

1) Il existe dans xcas plusieurs façons de développer [6] : `ratnormal`, `normal`, `simplify`, `expand`. Lors de calculs lourds sur les polynômes il faut vraiment préférer "ratnormal" ou "normal" qui optimiseront. "simplify" sera plus lent car il tentera des simplifications plus sophistiquées. On pourra l'utiliser après avoir fait un "normal" non satisfaisant. En fait, "expand" travaillera avec les objets les plus généraux et pourra être très lourd car il ne regroupe pas les dénominateurs. Essayez ces intructions sur, `P1:=(x^2-1)/(x-1)`, `P2:=-cos(5*x)+16*cos(x)*sin(x)^4-12*cos(x)*sin(x)^2+cos(x)` et `P3:=(a*sqrt(3)+b/sqrt(6))^4`

2) Factoriser $x^{12}-1$ dans $\mathbb{Z}[x]$. En factorisant des polynômes judicieusement choisis, obtenir la valeur du polynôme cyclotomique Φ_{12} ?

3) Factorisez $P=(2x+1)^2(x^5-1)/(x-1)$ dans $\mathbb{R}[x]$ et dans $\mathbb{C}[x]$. Peut t'on espérer un résultat exact ?

a) Factoriser $X^{12}-1$ sur $\mathbb{Q}[\sqrt{3}]$ et sur $\mathbb{Q}[\sqrt{3},i]$ et sur $\mathbb{Q}[e^{2i\pi/9}]$. (On peut tenter de rajouter une liste d'éléments en option de factor)

6. sans compter les synonymes