Exercice I: quelques fonctions graphiques

(But : pratique minimale de fonctions graphiques. Ex : Déssiner plusieurs objets géométriques simultanément, couleurs différentes, courbes paramétrées, implicites...)

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) En utilisant `plotparam`,déssiner la courbe $C_1 : t \mapsto M(t)$.

c) Prendre des éléments t_1 et t_2 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. (On peut donner une option pour chaque point ou bien definir la meme option globalement. etudier `couleur` ou `affichage`. On peut aussi dessiner un point à la souris pour connaitre la syntaxe de ces options.)

d) Dessiner la droite $M(t1), M(t2)$ en bleu.

3) On considère la courbe $C_2 : x^3 - 3x.y - 1 = 0$

a) Créer la fonction $f(x, y, z) = x^3 - 3x.y.z - z^3$, et représenter les courbes planes $f(x, y, 1) = 0$ et $f(x, 1, z) = 0$. Ajuster le pas pour que les dessins aient l'air bien lisse. Expliquer le comportement à l'infini de C_2.

Exercice II:

1) Si vous êtes en mode maple, etudier la priorité de `mod`. Par exemple : `3+5 mod 2 ;`. Quelle instruction faut il pour avoir un reste de valeur absolue minimale.

2) Etudier l'intruction `quo(t^2+1,t)`. Qu'a t'il fait ? Comment obtenir le polynôme quotient ?

Exercice III: Mini texte courbes elliptiques

On pourra étudier ces questions dans le livre de B. Perrin-Riou : Algèbre arithmétique et maple.

1 Loi de groupe.

Soit k un corps de caractéristique différente de 2, et $x^3 + ax^2 + bx + c$ un élément de $k[x]$ à racines simples. Dans le plan affine $A = k^2$, on considère la "courbe" C d'équation $y^2 = x^3 + ax^2 + bx + c$. On remarque que dans le complété projectif $\mathbb{P}$ de A, la courbe[1] $\bar{C}$ d'équation $y^2.z = x^3 + ax^2z + bxz^2 + cz^3$ rencontre la droite de l'infini en un unique point Ω, correspondant au point à l'infini de l'axe des y.

Nous allons définir sur $\bar{C} = C \cup \{\Omega\}$ une structure de groupe abélien d'origine Ω. Pour $P, Q \in \bar{C}$, notons d la droite projective (PQ) si $P \neq Q$, et la tangente à P en $\bar{C}$ si $P = Q$. Puisque l'intersection de $\bar{C}$ avec d a déjà 2 points de coordonnées dans k, elle en possède un "troisième" R que l'on peut toujours définir rigoureusement de manière algébrique. (En effet un polynôme de degré 3 à une variable ayant 2 racines dans k a toutes ses racines dans k). On considère alors la droite (ΩR), elle recoupe $\bar{C}$ en un troisième point qui sera par définition $P + Q$.

On obtient alors une loi clairement commutative, mais aussi associative[2]. On remarque de plus que $\Omega + P = P$ pour tout P de $\bar{C}$. De plus les calculs des coordonnées de $P + Q$ en fonction de celles de P et de Q sont explicites. Si P a pour coordonées (x, y), alors $-P$ a pour coordonnées[3] $(x, -y)$

Lorsque n est un entier relatif, on peut calculer $n.P$ en $O(\log(n))$ additions.[4]

2 La méthode de Lenstra

On peut par exemple utiliser cette loi pour tenter de trouver un facteur non trivial d'un entier. Par exemple au lieu de travailler dans $k = \mathbb{Z}/p\mathbb{Z}$, où p est un premier impair, on peut travailler dans $\mathbb{Z}/N.\mathbb{Z}$, et dans la procédure de calcul de $n.P$ tester[5] à chaque addition si un des points est à l'infini[6] modulo un diviseur de N. Si l'on arrive à une impossibilité, on a obtenu un diviseur non trivial de N.

Par exemple, avec N du type `nextprime(20000)*nextprime(40000) ;`, et n un ppcm d'entiers petits, $P = (2, 1)$. On cherche alors

[1] Quel est le lien entre ces 2 courbes ?

[2] On pourra tester ce point sur des exemples une fois la loi programmée

[3] Pourquoi ? interprétation géométrique

[4] On pourra justifier ce point.

[5] Si l'on travaille en affine cela revient à regarder si les divisions que l'on doit faire sont licites en calculant un pgcd avec N

[6] Les points étant sur la courbe, cela revient à tester si un point vaut Ω dans $\mathbb{P}(\mathbb{Z}/m\mathbb{Z})$ où m divise N

des courbes C contenant P telles que le calcul de $n.P$ donne un facteur non trivial de N. Prenons C du type $y^2 = x^3 + bx + c$, pour $b \in \{1, \ldots, 50\}$ on trouve rapidement les c tels que $P \in C$, et pour ces valeurs de C on calcule $n.P$. On pourra essayer d'autres exemples, et estimer le nombres d'additions effectueées sur la courbe elliptique pour briser N.

3 Suggestions de développements

1) Le candidat pourra justifier les notes de bas de page.

2) Illustrer graphiquement l'interprétation géométrique du point Ω via des dessins dans différents repères.

3) a) Créer une procédure point résiduel recevant 2 points de $\bar{C}$ et retournant le "troisième" point d'intersection décrit dans l'énoncé.

b) Programmer l'addition sur $\bar{C}$

4) Créer une procédure de multiplication par un entier n en $O(\log n)$ additions. On pourra le faire d'abord avec l'addition dans $\mathbb{Z}$ puis sur la courbe elliptique.

5) Créer une procédure pour tester s'il existe un diviseur m de N tel que le point de coordonnées projectives (x, y, z) soit égal à Ω dans $\mathbb{P}_2(\mathbb{Z}/m\mathbb{Z})$. Elle pourrait retourner 1 si c'est different de Ω, ou le diviseur m sinon. Sans factoriser N bien sur !

6) Modifier les opérations précédentes pour implémenter la méthode de Lenstra.

7) Quelles méthodes utilise votre logiciel pour factoriser un entier ? (Pour xcas on étudiera la documentation de pari-gp disponible dans le menu aide, car il me semble que xcas utilise la fonction de pari s'il le peut, sinon c'est probablement la méthode de pollar. (Cf le fichier /usr/share/giac/src/ifactor.cc).

4 Vérifications, manipulation

Les opérations sur les courbes elliptiques sont déja implémentées dans le logiciel libre pari aussi appelé gp. La courbe E d'équation $y^2 + a.yx + cy = x^3 + b.x^2 + dx + e$ se definit par le vecteur : `[a,b,c,d,e]`, le point P est noté en affine `[2,1]` et l'on peut faire `elladd(E,P,P)` ou ellpow(E,P,10). Le logiciel xcas peut utiliser ces fonctions après avoir effectue : `pari();` On peut par exemple faire sous xcas : `P :=[1,2];` puis `E :=[0,1,0,2,-15];` puis `pari_ellpow(E,P,10);`

xcas peut fournir aussi des objets modulaire, par exemple si N vaut 1000730021, on voudrait faire cela modulo N. Il suffit que le point P soit un objet modulaire sous xcas. En mode xcas : `P:=[1,2]% N;` puis `pari_ellpow(E,P,10);`. Par exemple `ellpow(E,P,10^20)` prouve que N n'est pas premier. Pour pouvoir lire le message d'erreur, il vallait mieux lancer xcas depuis un terminal, c'est la qu'il apparait :

" *** impossible inverse modulo : Mod(553216596, 1000730021). *** ellpow : impossible inverse modulo : Mod(553216596, 1000730021)."

En mode maple, c'est plus délicat de créer un objet modulaire. Le plus simple est de passer temporairement en mode xcas :

```
maple_mode(0);
P:=[1,2] % N;
maple_mode(1);
```

ou bien d'utiliser la fonction[7] `makemod`, par exemple : `P :=[1,2]*makemod(1,N);`

[7] non documentée