

PROJET SDD L2

Calcul Scientifique sur les polynômes

$v_{0.1}$

September 22, 2014

Première version pour questions, commentaires et demandes de précisions. Cette version vous permet de commencer votre travail de documentation et de réflexion. Elle est susceptible de très nombreuses évolutions d'ici quelques semaines, notamment en fonction de nos interactions.

1 Sujet

1.1 But

L'objectif de ce projet est d'arriver à réaliser en langage C, les structures de données ainsi que l'ensemble des fonctionnalités décrites pour effectuer différentes opérations sur des polynômes à coefficients dans $\mathbb{C}$ (et en $F2 = \mathbb{Z}/2\mathbb{Z}$ pour les plus avancés).

Le projet est présenté en deux niveaux : un niveau débutant accessible par tous, puis un niveau intermédiaire permettant d'offrir plus de fonctionnalités.

1.2 Objectifs pédagogiques

Le projet vous place dans une situation d'ingénierie qui vous oblige à :

Réfléchir et vous familiariser très concrètement avec la notion de complexité algorithmique, éventuellement vous mettre en compétition de performances sur de très simples opérations.

Traduire ce modèle général en structures de données et algorithmes réutilisables.

Appliquer et consolider vos acquis.

1.3 Conseil

Ne pas programmer avant d'avoir analysé, modélisé, défini le périmètre de son projet et ses potentielles extensions.

Niveau débutant - Accessible par tous

2 Structures de données

Nous représentons un polynôme par deux structures chaînées qui correspondent à ses formes développées et factorisée. Ces deux structures sont encapsulées dans une troisième qui définit le type *polynomial*.

2.1 Forme développée

La forme développée est représentée par une liste (simplement chaînée LSC ou doublement chaînée LDC, à vous de voir) dont les maillons sont les monômes du polynôme, eux-mêmes représentés par leur coefficient et leur degré. Choisissez de maintenir ordonnés ou non ces monômes, en fonction des algorithmes que vous devez concevoir ensuite. Cela relève de vos propres arbitrages.

Attention - Un monôme nul ne doit jamais exister en mémoire : le polynôme nul est représenté par une liste vide, et le polynôme X^{100} doit être représenté par un monôme unique et non pas par 100.

2.2 Forme factorisée

La forme factorisée est une chaîne dont les maillons représentent les facteurs du polynôme (les polynômes dont le produit donne le polynôme factorisé). Initialement, cette chaîne est vide, puis, une fois le polynôme factorisé, la chaîne est constituée de n polynômes unitaires (n représentant le degré du polynôme).

Rappel - d'après le théorème d'Alembert-Gauss, $\mathbb{C}$ est algébriquement clos, c'est-à-dire que tout polynôme de $\mathbb{C}[X]$ est scindé (il peut se décomposer en un unique produit de polynômes linéaires).

2.3 Coefficients complexes

Les deux composantes du coefficient complexe sont stockées au format flottant *IEEE754* double précision. Vous définissez le type de donnée **complex** et les opérations arithmétiques de base associés (addition, multiplication, soustraction, division, conjugaison, comparaison), ainsi que l'extraction des parties réelle et imaginaire, du module et de l'argument.

3 Fonctionnalités de base

Note - Les fonctionnalités de base doivent être robustes pour servir de socle aux fonctionnalités avancées. Ce n'est pas la partie la plus "amusante" du projet, mais un préalable indispensable.

Conseil Ne perdez pas un temps précieux sur ces implémentations. Avancez méthodiquement et rigoureusement, et restez concis dans la traduction en C de ces algorithmes simples et standards. Ne cherchez pas, en première approche, à optimiser la performance de ces algorithmes, mais veillez à ce qu'ils produisent des résultats conformes à leur spécification.

3.1 Instancier, visualiser

Vous commencez par mettre au point des fonctionnalités de création (depuis un tableau de données qui contient les coefficients, par duplication d'un polynôme existant) et d'affichage de vos polynômes (*pretty print*).

3.2 Arithmétique sur $\mathbb{C}[X]$

Vous concevez et réalisez les opérations d'addition, soustraction, multiplication, division (euclidienne) avec récupération du reste¹, conjugaison, comparaison, extraction des parties réelle et imaginaire de polynômes.

Conseils

- vous pouvez laisser de côté la forme factorisée pendant la réalisation des fonctionnalités de base, puis effectuer une mise à jour de celles-ci, une fois la fonctionnalité avancée de factorisation réalisée.
- **pour les plus avancés** : pour optimiser la division, anticipez sur la méthode de Ruffini-Horner évoquée ci-après. Si le polynôme diviseur est déjà factorisé, vous pouvez exploiter l'une de ses applications : la division rapide d'un polynôme par un polynôme unitaire. Si le numérateur et le dénominateur sont déjà factorisés, la division euclidienne classique serait une contreperformance.

Attention

- vous prendrez garde, pour l'addition et la soustraction, à ne pas produire des polynômes dont certains monômes sont nuls.
- la division par le polynôme nul est une opération interdite : pensez à gérer ce cas.

3.3 Evaluation en un point

Vous implémentez la méthode de Ruffini-Horner² pour évaluer rapidement un polynôme en un point, et l'appliquez, pour test, au changement de base pour l'écriture d'un nombre (en numération de position).

¹Vous devez nécessairement, ici, implémenter la notion de donnée modifiée.

²Point d'entrée pour cette méthode ici : http://fr.wikipedia.org/wiki/Methode_de_Ruffini-Horner

4 Fonctionnalités avancées

4.1 Dérivation et intégration

Vous concevez et réalisez les opérations de dérivation et de recherche de primitive d'un polynôme (d'après un paramètre constant donné).

4.2 Recherche de racines et factorisation

En utilisant la méthode d'Aberth-Ehrlich pour extraire les racines³, en vous appuyant sur la dérivation, l'évaluation, et la division euclidienne, réalisez la fonctionnalité de factorisation.

4.3 Compositions, décomposition (pour changements de variables)

Vous concevez et réalisez la composition de polynômes, i.e $C(X) \leftarrow A(B(X))$, ainsi que la décomposition, i.e. déterminer A , connaissant C et B .

4.4 Recherche du PGCD de deux polynômes

En vous appuyant sur la fonctionnalité de division euclidienne, vous implémentez l'algorithme d'Euclide pour déterminer le PGCD de deux polynômes.

Niveau Intermédiaire - Un peu plus de complexité Les élèves les plus avancés qui souhaitent aller un peu plus loin pourront s'essayer aux compléments suivants (au choix, ou tous si le cœur leur en dit) :

Coefficients de F2 Les coefficients de $F2$ est soumis aux règles de l'arithmétique booléenne (l'addition est représentée par la disjonction logique, la multiplication par la conjonction). Vous adaptez vos algorithmes et concevez une démonstration spectaculaire d'application au contrôle de redondance cyclique.

Interpolation et approximation Interpolation et approximation polynomiales d'un ensemble de données expérimentales par les méthodes de Lagrange, Newton et des moindres carrés (pour jouer à Gauss à la chasse de Cérès).

Fractions rationnelles

- Créer une structure de fraction rationnelle.
- Déterminer racines et pôles.
- Décomposer en éléments simples.
- Appliquer à la recherche de primitives de fractions rationnelles.

³Point d'entrée pour cette méthode ici : http://en.wikipedia.org/wiki/Aberth_method

Euclide étendu Faire évoluer l'algorithme d'Euclide réalisé pour le PGCD vers sa version étendue qui permet d'Extraire un couple de coefficients de Bézout de deux polynômes. Vous pouvez ensuite concevoir et réaliser une démonstration spectaculaire de l'application du lemme Chinois à la résolution d'un système de congruences polynomiales.

5 Utile pour la démonstration de vos fonctionnalités

Pour appliquer et faire la démonstration de vos fonctionnalités, vous produirez le scénario de test suivant, avec des traces d'exécution explicites des opérations effectuées. Si vous ne disposez pas de la factorisation, vous pouvez produire ce qui suit uniquement pour les formes développées.

Faulhaber⁴ [X]

Les formules sommatoires de puissances sont des polynômes de degré $p + 1$ qui vérifient, pour p et X entiers, l'identité suivante :

$$F_p[X] = \sum_{k=1}^X k^p = 1^p + 2^p + \dots + X^p$$

Leur intérêt algorithmique est évident si l'on considère que, même avec l'exponentiation rapide, la complexité de calcul d'une telle somme, laquelle est principalement déterminée par les produits, reste de l'ordre de $O(\ln n)$. Grâce à la formule, et pourvu qu'elle ait été factorisée, on tombe à une complexité en $O(1)$ puisqu'il ne s'agit plus que de réaliser p produits (soit à partir de la forme factorisée, soit en utilisant la méthode de Horner).

En utilisant la relation de récurrence⁵ suivante qui relie ces formules sommatoires, concevoir une fonction qui produit un tableau des n premières formules sommatoires sous forme développée (et sous forme factorisée, si vous avez pu réaliser cette fonctionnalité).

$$\begin{cases} F_0(X) = X + 1 \\ \forall p \in \mathbb{N}^* : F_p(X) = \frac{1}{p+1} (F_0^{p+1}(X) - \sum_{k=0}^{p-1} C_{p+1}^k F_k(X)) \end{cases}$$

En utilisant votre fonctionnalité de PGCD, vérifiez que pour tout p strictement positif, le polynôme $2F_1(X) = X(X + 1)$ divise $F_p(X)$, puisque

$$\begin{cases} \forall p \in \mathbb{N}/p > 1 \wedge p \equiv 1(2) : (2F_1(X))^2 \mid F_p(X) \\ \forall p \in \mathbb{N}/p > 2 \wedge p \equiv 0(2) : (3F_2(X)) \mid F_p(X) \end{cases}$$

⁴Collaborateur de Kepler, et professeur de Descartes, dont le travail publié dans son Academia Algebrae a été quelque peu escamoté par celui de Bernoulli. Le nombre de Bernoulli, se cache dans le polynôme de Faulhaber, et constitue une clé de la théorie des nombres.

Voir cet article de Knuth au sujet de Faulhaber : <http://arxiv.org/abs/math.CA/9207222>

⁵Que l'on doit à Pascal si nos sources ne se trompent pas ☺

En utilisant la fonction de composition qui permet notamment un changement de variable, produire la famille de polynômes $(\tilde{F}_i(X) = F_i(X - \frac{1}{2}))_{i \in \llbracket 0 \ n \rrbracket}$

En utilisant la même fonctionnalité et la comparaison, vérifier que $\tilde{F}_p(X)$ est pair si p est impair, et inversement, i.e. :

$$\begin{cases} p \equiv 0(2) \Leftrightarrow \tilde{F}_p(X) = -\tilde{F}_p(-X) \\ p \equiv 1(2) \Leftrightarrow \tilde{F}_p(X) = \tilde{F}_p(-X) \end{cases}$$

Polynômes de Faulhaber

Au 11^{ème} siècle, le mathématicien musulman, Al-karagi, redécouvrit une remarquable identité remarquable, que des recherches plus récentes attribuent au mathématicien et philosophe néo-pythagoricien Nicomaque de Gêrase (1er siècle) :

$$1^3 + 2^3 + \dots + n^3 = (1 + 2 + \dots + n)^2 = \left(\frac{n(n+1)}{2} \right)^2 = \frac{n^2(n+1)^2}{4}$$

En d'autres termes :

$$F_3(X) = F_1^2(X)$$

En utilisant votre fonctionnalité de décomposition, déterminez les polynômes $f_p(X)$ dits de Faulhaber, tels que $F_p(X) = f_p(F_1(X))$, puis comme vous disposez de la puissance informatique pour explorer, recherchez d'autres relations de la forme $F_p(X) = f_{p,q}(F_{q \leq p}(X))$ en identifiant les plus remarquables d'entre elles (et en les montrant à vos professeurs ☺).

Plus ? L'exploration du lien avec les nombres et les polynômes de Bernoulli n'est pas proposée ☹, mais rien ne vous empêche de prolonger votre parcours dans cette voie. De même, si vous avez abordé certains compléments, il serait dommage d'oublier de les mettre en valeur.

6 Organisation

6.1 Encadrement et calendrier

Depuis 2012, le projet de SDD est encadré sur la durée de son exécution, et non pas seulement évalué en Décembre ou Janvier, au moment de la soutenance.

Vos tuteurs de projet sont :

Groupe A Nora IZRI.

Groupe B Kaïs Klai.

Groupe C Florant DIEDLER.

Groupe PL Nora IZRI.

6.1.1 Mise à disposition du sujet

Une première version du sujet est soumise fin Septembre pour vous permettre de commencer à travailler et pour susciter vos questions et demandes de précisions. Cette première version de cadrage est susceptible de nombreuses évolutions au cours du mois d'Octobre, notamment en fonction des réponses à vos questions.

6.1.2 TP de lancement (TPP1)

Chaque groupe effectue un premier TP (de 4h) le 23 Septembre. Ce sera l'occasion de constituer vos équipes, de prendre connaissance du sujet, effectuer vos premières recherches, initialiser votre travail d'élaboration.

Groupe A le Mardi 23 Septembre après-midi.

Groupe B le Mardi 23 Septembre après-midi.

Groupe C le Mardi 23 Septembre matin.

Groupe PL le Mardi 23 Septembre matin.

6.1.3 Soutenance intermédiaire (S1)

Une soutenance intermédiaire aura lieu vers fin-October pour nous assurer que toutes les équipes ont commencé à travailler. Cette soutenance sera l'occasion d'un point individualisé (par équipe) sur la base d'un échange relativement informel. Il y aura cependant quelques points en jeu sur la note finale, qui seront notamment perdus par les équipes qui n'auraient pas commencé à travailler (sérieusement s'entend):

Groupe A le Mardi

Groupe B le Mardi

Groupe C le Mardi

Groupe PL le Mardi

6.1.4 TP d'appui intermédiaire (TPP2)

Un TP d'appui intermédiaire aura lieu dans la semaine, au pire dans la quinzaine qui suit la soutenance intermédiaire.

Groupe A le Mardi

Groupe B le Mardi

Groupe C le Mardi

Groupe PL le Mardi

6.1.5 Soutenance finale (S2)

Soutenance finale, avant les vacances de Noël.

Groupe A le Mardi

Groupe B le Mardi

Groupe C le Mardi

Groupe PL le Mardi

6.2 Constitution des équipes

Binômes par groupe de TD. Trinômes interdits, sauf un unique pour groupe TD à nombre impair de personnes.

Les délégués de groupe sont chargés de produire une liste des équipes et des ordres de passage sur les plages horaires réservées pour les soutenances.

Il vous faudra avoir indiqué votre équipe à votre délégué avant le **Mercredi 1er Octobre au soir, au plus tard**. Au delà, votre équipe sera constituée d'office.

En cas de dysfonctionnement problématique au sein d'une équipe, n'attendez pas le dernier moment pour l'évoquer : votre tuteur peut vous aider à trouver une solution.

6.3 IHM et technologies tierces parties

Vous pouvez utiliser les technologies de votre choix.

Les technologies tierces parties employées devront être référencées dans votre livrable, et notamment rappelées dans les annexes de votre rapport.

Il est interdit d'utiliser des librairies existantes pour vos SDD et algorithmes clés (en d'autres termes, pas de librairie, sauf librairie graphique).

Notez enfin que l'appréciation de l'IHM n'est qu'un constituant accessoire de la note. Évitez d'y investir trop de temps, au détriment de la recherche de la meilleure performance algorithmique.

Chaque année, malgré cette mise en garde, de nombreux élèves sont frustrés par une piètre rétribution après un investissement significatif sur cette partie.

Cependant, les meilleures réalisations seront mises en compétition en vue de leur présentation aux JPO (Journées Portes Ouvertes) 2015 : cette compétition permettra aux meilleures équipes d'acquérir jusqu'à 2 pts bonus sur leur note de projet.

7 Évaluation

7.1 Comment gagner des points

Critères par ordre décroissant d'importance :

- Le programme fait ce qu'on attend de lui (partiellement, complètement, avec des plus).

- L'interaction utilisateur est ergonomique (facilité d'utilisation) et esthétique.
- Le code source est efficace (SDD, algorithmes efficaces).
- Le code source est propre (indentations, identifiants intelligibles, commentaires bien dosés).

7.2 Comment perdre des points

Critères par ordre décroissant de gravité :

- Même code (même partiel) livré par plusieurs équipes : 0 pour les équipes impliquées.
- Retard de livraison : 2 points par jour au *prorata temporis*.
- Non respect des consignes de packaging⁶ du livrable : jusqu'à -3 pts suivant les cumuls.

7.3 Travail minimum pour espérer avoir 10

Le périmètre fonctionnel minimal consiste à la réalisation des sections 2 et 3 de l'énoncé en réalisant les structures de données nécessaires au projet ainsi que les fonctionnalités de base.

Ce périmètre fonctionnel minimum permet d'être noté sur 12 points : obtenir 12 serait alors la marque d'une réalisation parfaite sur ce périmètre (fonctionnement, ergonomie, etc).

8 Packaging du livrable

Veuillez respecter toutes les consignes de déroulement de ce projet.

8.1 Contenu du livrable

- Vos fichiers sources.
- SURTOUT PAS D'EXÉCUTABLE !
- Votre documentation (Rapport et autres documents) au format PDF.
- Un fichier `README.txt` qui comprend impérativement deux sections :
 - Un état d'avancement honnête et fiable de vos travaux en 10 lignes maximum.
 - Les références aux sources qui ont pu inspirer votre conception, la reprise de code : étant formellement interdite.
 - Et en option, une mise en avant de fonctionnalités supplémentaires que vous aurez choisi de réaliser dans le but d'obtenir une meilleure note.

⁶ voir section suivante.

8.2 Contenu du rapport

Un sommaire détaillé avec pages numérotées.

L'analyse fonctionnelle générale d'environ cinq pages, elle explique quelles sont les principales données traitées et quelles sont les principales fonctionnalités du programme, comment on peut les regrouper, comment elles sont organisées dans le programme. Cette analyse aboutit à la présentation des modules fonctionnels qui sont décrits en détail dans l'analyse détaillée. L'analyse générale doit préciser les choix importants que vous avez faits pour la conception de votre projet.

L'analyse fonctionnelle détaillée d'environ dix pages, elle précise, pour chaque module, son utilité, ses entrées et ses sorties, les données qu'il modifie. Il faut décrire les données utilisées et les algorithmes mis au point (logique de manipulation) avec rigueur, concision et clarté, le tout accompagné de schémas. **Cette description est abstraite⁷ et s'appuie sur le langage algorithmique utilisé en cours et en TD.** C'est un travail de synthèse, de pédagogie et de communication; les extraits de code sont proscrits de vos rapports.

Annexes elles présentent au moins les technologies tierces employées. Et comme option recommandée, bibliographie listent les références qui ont permis d'étayer votre analyse et votre conception.

8.3 Livraison

Une seule remise des travaux par équipe !! (la double remise peut être sanctionnée dans la composante rigueur / discipline de la note). Remise au plus tard le**2014**. Dès le lendemain, application d'une pénalité de 2 points par jour (au *prorata temporis*).

Le livrable se présente sous la forme d'une archive ZIP de nom : **ProjetSDD L2 GP-NOM 1-NOM 2**.ZIP envoyé par email à l'adresse de votre tuteur de projet, ou par tout autre moyen de substitution qu'il vous indiquerait.

Objet ProjetSDD L2 GP-NOM 1-NOM 2

⁷algorithmes indépendants de la représentation interne des données. Surtout pas un pseudo C pour se donner bonne conscience.