

Rapport de stage de 2^{ème} année

Evolution de la toolbox CRONE sous Matlab

Juin, juillet et septembre 2008

Tuteur : OUSTALOUP Alain		Qualité : Professeur	
Maitre de stage : MALTI Rachid		Qualité : Maître de conférences	
SCAMPS Maxime	Département : Electronique		18 Septembre 2008

Table des matières

I. Le laboratoire IMS et le LAPS.....	4
1. Le laboratoire IMS	4
2. Le LAPS	7
II. Fonction de transfert de système à dérivées non entières	12
1. Introduction	12
2. Intégration et dérivation réelles	12
3. Représentation des systèmes non entiers	15
III. La Toolbox CRONE.....	18
1. Qu'est-ce que la toolbox CRONE ?	18
2. Analyse de l'existant	18
3. Synthèse des modifications à apporter	20
IV. Réalisation.....	23
1. Traitement des polynômes remarquables	23
2. Hiérarchisation et encapsulation des fonctions	23
3. Mise en place de la fonction « multi »	26
4. Modification de l'algorithme de la fonction « commensurate »	30
V. Procédures de test.....	33
1. Test et démo des différentes fonctions	33
2. Rapport de test par classe	35
VI. Documentation : le User's Guide de la toolbox Matlab.....	38
VII. Conclusion	40
1. Organisation du travail	40
2. Bilan du stage	42

Introduction

Ce stage s'est déroulé du 1^{er} juin au 19 septembre 2008 au LAPS sous la direction de Mr Malti.

L'équipe CRONE de l'IMS développe depuis 1998 la Toolbox CRONE, spécialisée dans la dérivation non entière appliquée en Automatique, en collaboration avec le groupe PSA. L'évolution de la Toolbox CRONE vers une version orientée objet a été entamée mais pas entièrement finalisée. Le stage effectué s'inscrit dans la continuité de ces développements. L'objectif principal était d'avoir, à l'issue du stage, un module mathématique, qui constitue le cœur de la toolbox, entièrement opérationnel et une documentation technique à jour.

Le stage a comporté une approche théorique du sujet à traiter, une prise de connaissance de l'existant et des besoins d'améliorations à apporter, puis la réalisation de ces améliorations, la mise en œuvre des procédures de test et de documentation en vigueur.

Ce stage a mis à l'épreuve la sagacité et la patience de Mr Malti, qu'il en soit ici remercié.

I. Le laboratoire IMS et le LAPS

1. Le laboratoire IMS

La fusion des unités de recherche PIOM, IXL et LAPS a donné naissance au 1er janvier 2007 au laboratoire de l'Intégration du Matériau au Système (IMS). Cette nouvelle unité du CNRS, rattachée au département ST2I (Sciences et Technologie de l'Information et de l'Ingénierie) rassemble près de 300 personnes dont plus de 100 chercheurs et enseignants-chercheurs rattachés principalement aux établissements ENSEIRB, ENSCPB et Université Bordeaux 1.

En collaboration étroite avec le LaBRI et INRIA futurs, le laboratoire aquitain IMS est un partenaire actif dans l'implantation sur le campus de l'Université Bordelaise d'un nouveau pôle de recherche fort, avec à court terme une concentration de plus de 1000 personnes œuvrant dans le domaine porteur des Sciences et Technologies de l'Information et de la Communication (STIC). Dans ce cadre, l'IMS offre un positionnement scientifique original avec un ensemble cohérent de compétences allant de l'assemblage des matériaux jusqu'à l'ingénierie des systèmes. Il est un acteur majeur dans l'étude et le développement des systèmes embarqués complexes et hétérogènes.

a. Fiche signalétique

- Statut : Unité Mixte de Recherche (UMR) 5218 CNRS
- Rattachement : ENSEIRB, ENSCPB, Université Bordeaux 1
- Directeur : Pascal FOUILLAT
- Directeurs adjoints : Valérie VIGNERAS (MCM), Claude PELLET (COFI), Jean-Paul Bourrières (LAPS)
- Adresse : 351 cours de la libération, F-33405 TALENCE Cedex
- Téléphones : +33 5 40 00 65 70 (MCM), +33 5 40 00 65 40 (COFI), +33 5 40 00 24 00 (LAPS)
- Télécopie : +33 5 56 37 15 45
- Email : info@ims-bordeaux.fr // Site Web : <http://www.ims-bordeaux.fr>

b. Personnel

- Enseignants-chercheurs: 101
- Chercheurs: 11
- Agents techniques: 38
- Agents administratifs: 14
- Post-doctorat : 12
- Doctorants : 128

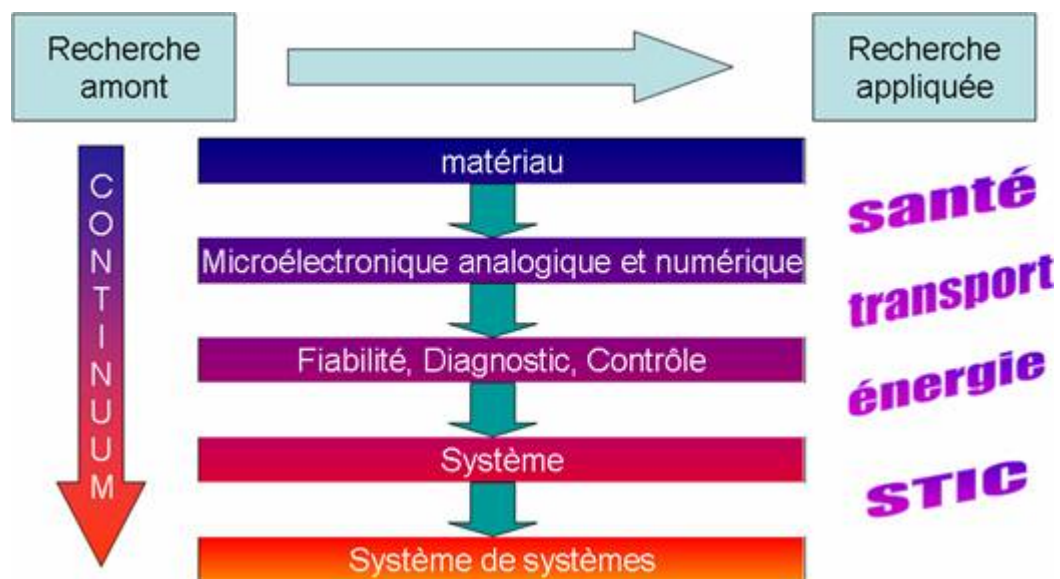
Total : 304 personnes

c. Activités

Le laboratoire IMS développe des actions de recherche originales et cohérentes dans les domaines de :

- la modélisation et l'élaboration des matériaux, des capteurs et des microsystèmes pour les dispositifs électroniques
- la modélisation, la conception, l'intégration et l'analyse de fiabilité des composants, circuits et assemblages
- l'identification, la commande, le diagnostic, le traitement du signal et des images, la supervision et la conduite des processus complexes et hétérogènes.

Les domaines d'application concernent principalement les télécommunications, les transports, la santé, l'énergie. Ils sont traités au travers de nombreux projets européens, nationaux ou régionaux.



d. Organisation

Les activités de recherche sont organisées en 9 groupes de recherche répartis entre les 3 départements : MCM, COFI et LAPS. On va décrire les 2 premiers départements cités; on consacre une partie entière pour la présentation du LAPS.

Description du département MCM

Le département MCM (Matériaux, Capteurs et Microsystèmes) provient de la fusion des activités de l'ex-laboratoire PIOM et du groupe Microsystèmes/MicroCapteurs de l'ancien laboratoire IXL. Les activités menées dans ce département sont partagées entre les 3 domaines que sont les Matériaux, les MEMS (Micro-Electro-Mechanical Systems) et le Bio-Electromagnétisme. On décrit les activités faites dans chacun de ces groupes.

-
- Groupe Matériaux
 - Electronique Organique
 - Matériaux fonctionnalisés pour les télécommunications
 - Matériaux naturel et télédétection
- Groupe MEMS
 - Matériaux Microassemblés
 - Capteurs à ondes acoustiques et détection (bio) chimique
 - Micropoutres, micro-cavités et humidité
- Groupe Bio-électromagnétisme
 - Effets bio-électromagnétiques des champs ELF et RF
 - Neuro-immunologie

Description du département COFI

Le département COFI (COncption et Fiabilité) est issu de l'ancien laboratoire IXL excepté l'ancien groupe MicroSystèmes/MicroCapteurs rattaché aujourd'hui au département MCM. Les activités sont là aussi organisées autour des 3 groupes de recherche Conception, Puissance et Fiabilité. On décrit les activités à l'intérieur de ces 3 groupes :

- Groupe Conception
 - Circuits Radiofréquences et Hyperfréquences, Circuits Mixtes Ultra Rapides
 - Systèmes Neuromorphiques
 - Test par faisceau laser
 - Caractérisation et modélisation des technologies
 - Systèmes numériques embarqués
- Groupe Puissance
 - Technologie des assemblages et optimisation des composants
 - Convertisseurs et gestion des sources hybrides
- Groupe Fiabilité
 - Prévision de la fiabilité
 - Analyse et modélisation des dégradations, CEM
 - Technologies optoélectroniques et hyperfréquences
 - Packaging et micro-assemblages

e. Partenariat

Les chercheurs du laboratoire IMS entretiennent et structurent de nombreuses collaborations avec le monde industriel. Le laboratoire affiche la volonté de jouer un rôle actif dans le monde socio-économique par la valorisation des travaux de recherche, le transfert technologique et la création d'entreprise. En 2006, 10 nouveaux dépôts de brevets sont venus renforcer son potentiel d'innovation.

Parmi toutes les actions en cours, on peut citer :

- Le laboratoire commun avec STMicroelectronics pour les recherches autour des circuits radiofréquences, circuits mixtes ultra fréquences ainsi que pour la caractérisation et la modélisation des technologies d'intégration
- Le laboratoire commun avec Total-Fina-Elf dans le domaine de l'imagerie sismique
- Le partenariat avec PSA et la labellisation ERT (Equipe de Recherche Technologique) dans les recherche sur le contrôle global du châssis et l'identification dynamique du conducteur.
- Le partenariat avec THALES au sein du pôle STIC de l'Université Bordeaux 1 (IMS, LaBRI, INRIA) dont les travaux traitent des systèmes numériques embarqués.

f. Recherche et formation

- Ecoles doctorales de l'Université Bordeaux 1 :
 - Sciences Physiques et de l'Ingénieur
 - Sciences Chimiques
 - Sciences de l'Environnement
- Masters de l'Université Bordeaux 1 et de l'Ecole Pratique des Hautes Etudes (EPHE)
- Ecole Nationale Supérieure d'Electronique, d'Informatique et Radiocommunications de Bordeaux (ENSEIRB)
- Ecole Nationale Supérieure de Chimie et de Physique de Bordeaux (ENSCPB)

2. Le LAPS

a. Présentation générale

Le LAPS (Laboratoire d'Automatique, Productique et de Signal et image) a été créé en 1991 par la fusion du LARFRA (Laboratoire d'Automatique, de Reconnaissances des Formes et de Robotique Agricole), créé en 1981, et du GRAI (Groupe de Recherche en Automatisation Intégrée), créé en 1971. Le laboratoire s'appelait alors LAP. L'équipe ESI (Equipe Signal et Image) de l'ENSEIRB a ensuite intégré celui-ci en 2001 pour donner le

LAPS. Depuis 2007 et la création du laboratoire IMS, le LAPS est devenu un département de ce nouveau laboratoire.

Les activités du département sont organisées autour des 3 groupes de recherche que sont le groupe Automatique, le groupe GRAI-Productique et le groupe Signal et Image. Le groupe Automatique sera plus présenté plus en détail étant donné que c'est celui auquel je suis intégré dans le cadre du stage.

b. Description des groupes de recherche

Groupe de recherche automatique

Les activités menées par ce groupe s'articulent autour de 2 thématiques :

- l'Approche Robuste et Intégrée de l'Automatique (équipe **ARIA**)
- les Systèmes à dérivées Non Entières (équipe **CRONE**)

Les recherches faites par l'équipe **ARIA** concernent les domaines suivants :

- Surveillance à base de modèle et contrôle d'intégrité : les travaux réalisés ont déjà permis de poser les fondements d'une méthodologie générale en analyse (μ g-analyse) et de clarifier les liens entre les outils de surveillance et de commande robuste
- Planification de trajectoires, méthodes de guidage/pilotage : les travaux portent sur le développement des méthodologies de guidage et pilotage pour les systèmes non linéaires. Les applications associées portent d'une manière globale sur le système GNC (Guidance Navigation & Control) des engins aéronautiques et spatiaux.
- Commande tolérante aux fautes : une démarche méthodologique a été développée pour la conception d'un système de commande tolérante aux fautes, intégrant un filtre de diagnostic. Cette approche permet de concevoir un système FTC indépendamment du correcteur nominal déjà en place. Il s'agit d'un aspect important qui facilite grandement sa portabilité en simplifiant les questions relatives aux aspects de validation/certification.

De même, on cite les thèmes des recherches effectuées par l'équipe **CRONE** :

- Théorie des systèmes : les travaux relèvent de la recherche fondamentale et concerne l'analyse, la synthèse et la représentation des systèmes linéaires d'ordres non entiers (ou fractionnaires), réels ou complexes
- Automatique : cette activité relève de la recherche fondamentale et méthodologique. Utilisant les résultats établis en théorie des systèmes, elle concerne la commande CRONE, et plus récemment l'identification par modèle non entier.
- Robotique : Cette activité relève de la recherche méthodologique et appliquée et utilise les résultats obtenus en théorie des systèmes et en automatique. Noyau du PRC-GDR du CNRS "La dérivation non entière en isolation vibratoire", elle concerne la suspension CRONE et la modélisation d'environnement à travers la planification non entière de trajectoire et la génération d'un terrain fractal.

Groupe de recherche GRAI-Productique

Les activités menées par ce groupe s'articulent autour de 4 thématiques. Pour chacune d'entre elles, on cite les domaines étudiés.

- Analyse des dysfonctionnements des systèmes
 - Sûreté de fonctionnement des systèmes
 - Fiabilité humaine et organisationnelle
 - Conduite des systèmes de production et aide à la décision
 - Modélisation agrégée des systèmes de production
 - Modèles de données et aide à la décision
 - Ingénierie de la conception
 - Systèmes de conception
 - Modèle GRAI
 - Modélisation technico-économique des systèmes de production
 - Performance industrielle
 - Modélisation des systèmes complexes socio-techniques
 - Conception des systèmes de production et des produits
- Groupe de recherche Traitement du Signal et de l'image

Les domaines étudiés se répartissent dans ce groupe selon 5 thématiques :

- Extension des approches fondées sur un modèle 1D au cas 2D
 - approche paramétrique haute résolution
 - modélisation paramétrique de champs stochastiques
 - filtrage non linéaire de type Volterra
- Traitement du signal pour les applications multimédias
 - analyse, rehaussement et compression du signal de parole
 - synthèse de la parole pour les langues sémitiques, traitement automatique de la prosodie
- tatouage de signaux audio
- Analyse de signaux biomédicaux (EEG)
- Traitement des textures orientées
 - Modélisation des textures
 - Estimation robuste en tout point de l'orientation
 - Caractérisation des textures
 - Nouveaux outils de modélisation
- Modèles particuliers
 - modèles déformables particuliers
 - morphologie adaptative

c. Politique partenariale

Le département LAPS développe une recherche fondamentale, méthodologique et appliquée ainsi que des actions de transfert de recherche dans le domaine de l'analyse et de la conduite des systèmes, qu'il s'agisse de systèmes artificiels ou naturels. La valorisation des résultats de recherche s'effectue tant au niveau national (EDF, CNES, CEA, THOMSON-CSF SEXTANT, PSA, ALSTOM, KSB-AMRI, EADS, SNECMA, Schlumberger, Total, Réseau Technologique RNTL, ...) qu'au niveau international, notamment à travers les programmes européens (5^e PCRD,...), ou avec des industriels (Robert Bosch). La capacité du LAPS à valoriser ses points forts confère une pérennité aux collaborations contractuelles avec les grandes entreprises (PSA, EADS, Laboratoire commun IMS/LAPS-CNRS-Total). Le LAPS s'inscrit donc dans le schéma général de la recherche technologique en s'attachant à répondre avec des solutions novatrices, originales et efficaces à une demande du milieu socio-économique. La recherche technologique s'avère particulièrement exigeante, en ce sens que les contraintes dictées par le réel sont souvent plus sévères que les hypothèses d'étude

adoptées dans la recherche académique. Le capital de résultats de recherche fondamentale et méthodologique du LAPS permet d'aborder le quadriennal 2007-2010 avec l'ambition de lever des verrous technologiques au travers, particulièrement :

- du démarrage d'une ERT LAPS-PSA
- du laboratoire virtuel européen INTEROP- VLab issu du ReX INTEROPd
- du développement du laboratoire commun CNRS IMS/LAPS-Total

II. Fonction de transfert de système à dérivées non entières

Cette partie a été réalisée en reprenant l'introduction de la thèse de Mr XX (?), qu'il en soit remercié.

1. Introduction

La première idée de différenciation à des ordres non entiers remonte à *Leibniz*, et cependant plus d'un siècle s'est écoulé sans qu'elle n'ait produit de résultats. Bien qu'il soit plusieurs fois question de ces différenciations dans les ouvrages d'*Euler*, de *Laplace* et de *Fourier*, ce n'est que dans le XIX ème siècle que *Liouville* et *Riemann* en ont nettement fixé le sens et dissipé le vague enveloppant cette théorie. Le lecteur pourra se référer à [Let68, Ros74, Old74, Mil93, Dug94] pour un développement historique complet de la dérivation non entière.

Ce chapitre est une introduction aux opérateurs non entiers de dérivation et d'intégration et aux systèmes non entiers. Il est ponctué par trois parties. La première rappelle les définitions mathématiques de la dérivation et de l'intégration non entières et leur transformée de *Laplace*. La seconde traite des différentes représentations des systèmes non entiers. Leur condition de stabilité est examinée dans la troisième partie.

2. Intégration et dérivation réelles

a. Intégration réelle

Définition

Soit f une fonction réelle de la variable t continue par morceaux et intégrable sur $[0, +\infty[$.

L'intégration de f à un ordre entier n s'exprime par la formule de *Cauchy* :

$$\underbrace{\int \dots \int_0^t}_{n \text{ fois}} f(\tau) d\tau = \mathbf{I}^n f(t) = \frac{1}{(n-1)!} \int_0^t \frac{f(\tau)}{(t-\tau)^{1-n}} d\tau. \quad (1)$$

Dans un article de *Riemann*, écrit en 1847 lorsqu'il était étudiant et publié pour la première fois après sa mort en 1876, le mathématicien allemand a défini l'expression générale de l'intégrale d'ordre réel $\nu \in \mathbb{R}_+^*$ d'une fonction f , en généralisant la formule de *Cauchy* :

$$\mathbf{I}^\nu f(t) \stackrel{\Delta}{=} \frac{1}{\Gamma(\nu)} \int_0^t \frac{f(\tau)}{(t-\tau)^{1-\nu}} d\tau, \quad (2)$$

où Γ est la fonction d'*Euler* telle que:

$$\Gamma(\nu) = \int_0^\infty e^{-x} x^{\nu-1} dx, \forall \nu \in \mathbb{R}^* \setminus \mathbb{Z}_-. \quad (3)$$

b. Transformée de *Laplace* de l'intégrale d'une fonction

La transformée de *Laplace* de l'intégrale d'ordre ν d'une fonction f réelle telle que $f(t) = 0$ quand $t \leq 0$ est donnée par [Mil93]:

$$\mathbf{L}(\mathbf{I}^\nu(f(t))) = \frac{1}{s^\nu} \mathbf{L}(f(t)), \quad (5)$$

où s désigne la variable de *Laplace*.

Cette formule est la généralisation de la transformée de *Laplace* d'une intégrale entière.

c. Dérivation réelle

Bien qu'il existe plusieurs approches de généralisation de la dérivée d'ordre entier à des ordres réels, ce paragraphe limite leur présentation à trois définitions, celles de *Riemann*, *Caputo* et *Grünwald*.

Définitions

Riemann a défini la dérivée d'ordre réel $\nu \in \mathbb{R}_+$ d'une fonction f comme la dérivée d'ordre entier $\lfloor \nu \rfloor + 1$ de l'intégrale d'ordre non entier $(\lfloor \nu \rfloor + 1 - \nu)$, soit [Sam93] :

$$\mathbf{D}^\nu f(t) \stackrel{\Delta}{=} \mathbf{D}^{\lfloor \nu \rfloor + 1} \left(\mathbf{I}^{\lfloor \nu \rfloor + 1 - \nu} f(t) \right), \quad (7)$$

où $\lfloor \nu \rfloor$ est le plus petit entier inférieur ou égal à ν ($\lfloor \nu \rfloor = \text{floor}(\nu)$).

Compte tenu de la définition de l'intégrale (2), la dérivée d'ordre ν de f s'écrit:

$$\mathbf{D}^\nu f(t) = \frac{1}{\Gamma([\nu] - \nu)} \frac{d^{[\nu]+1}}{dt^{[\nu]+1}} \left(\int_0^t \frac{f(\tau)}{(t-\tau)^{\nu+1-[\nu]}} d\tau \right). \quad (8)$$

La figure (3) schématise graphiquement l'idée de l'approche de *Riemann*.

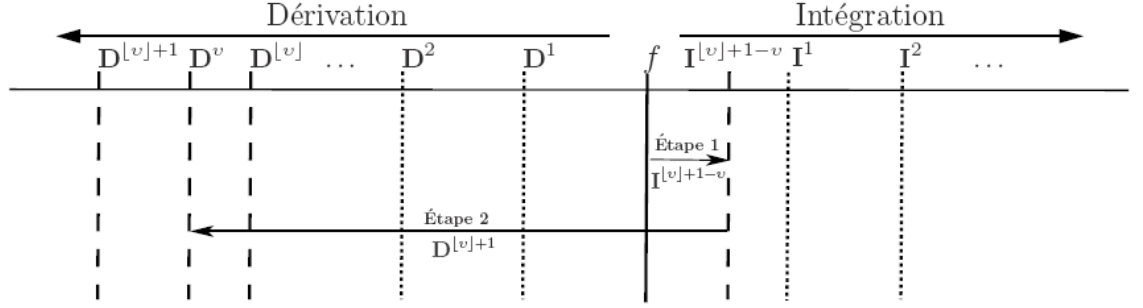


Figure 3. Approche de *Riemann*

d. Transformée de *Laplace* de la dérivée d'une fonction

Comme il existe plusieurs définitions de la dérivée non entière, l'expression de la transformée de *Laplace* n'est pas unique. Les différences portent uniquement sur les *conditions initiales*. La transformée de *Laplace* de la dérivée d'ordre réel ν d'une fonction définie par *Riemann* est donnée par [Mil93]:

$$\mathbf{L}(\mathbf{D}^\nu f(t)) = s^\nu \mathbf{L}(f(t)) - \sum_{k=0}^{[\nu]-1} s^k \mathbf{D}^{\nu-k-1} f(t) \Big|_{t=0}. \quad (17)$$

La transformée de *Laplace* de la dérivée définie par *Caputo*, s'écrit quant à elle :

$$\mathbf{L}(\mathbf{D}^\nu f(t)) = s^\nu \mathbf{L}(f(t)) - \sum_{k=0}^{[\nu]-1} s^{\nu-1-k} \mathbf{D}^k f(t) \Big|_{t=0}. \quad (18)$$

Pour des ordres de dérivation entiers, les deux formules sont identiques à la transformée de *Laplace* de la dérivée d'ordre entier d'une fonction.

3. Représentation des systèmes non entiers

Nous traitons, dans ce mémoire, des systèmes monovariables, linéaires, à temps continu, invariants dans le temps, causaux, strictement propres et à dérivées non entières. Ces systèmes, non entiers, peuvent être représentés sous plusieurs formes:

- équation différentielle
- représentation d'état
- fonction de transfert.

a. Équation différentielle

Un système non entier peut être décrit par une équation différentielle de la forme:

$$y(t) + a_1 \mathbf{D}^{\nu_{a_1}} y(t) + a_2 \mathbf{D}^{\nu_{a_2}} y(t) + \dots + a_n \mathbf{D}^{\nu_{a_n}} y(t) = b_0 \mathbf{D}^{\nu_{b_0}} u(t) + b_1 \mathbf{D}^{\nu_{b_1}} u(t) + b_2 \mathbf{D}^{\nu_{b_2}} u(t) + \dots + b_m \mathbf{D}^{\nu_{b_m}} u(t), \quad (21)$$

où $u(t)$ et $y(t)$ désignent respectivement l'entrée et la sortie.

Les ordres de dérivation $\nu_{a_1}, \dots, \nu_{a_n}, \nu_{b_0}, \dots, \nu_{b_m}$ sont des nombres réels positifs tels que:

$$0 < \nu_{a_1} < \dots < \nu_{a_n} \quad \text{et} \quad 0 < \nu_{b_0} < \dots < \nu_{b_m}. \quad (22)$$

Comme dans le cas d'une équation différentielle classique à dérivées d'ordres entiers, les ordres de dérivation doivent vérifier la contrainte $\nu_{a_n} < \nu_{b_m}$ pour que le système soit strictement propre.

Généralement, l'équation (21) peut être réécrite sous la forme d'une équation différentielle de type séquentiel [Mil93, Coi02]:

$$y(t) + a_1 \overbrace{\mathbf{D}^{\nu_{a_1}} \left(\dots \left(\mathbf{D}^{\nu_{a_1}} y(t) \right) \right)}^{\frac{\nu_{a_1}}{\nu} \text{ fois}} + \dots + a_n \overbrace{\mathbf{D}^{\nu_{a_n}} \left(\dots \left(\mathbf{D}^{\nu_{a_n}} y(t) \right) \right)}^{\frac{\nu_{a_n}}{\nu} \text{ fois}} = \underbrace{b_0 \mathbf{D}^{\nu_{b_0}} \left(\dots \left(\mathbf{D}^{\nu_{b_0}} u(t) \right) \right)}_{\frac{\nu_{b_0}}{\nu} \text{ fois}} + \dots + \underbrace{b_m \mathbf{D}^{\nu_{b_m}} \left(\dots \left(\mathbf{D}^{\nu_{b_m}} u(t) \right) \right)}_{\frac{\nu_{b_m}}{\nu} \text{ fois}}, \quad (23)$$

équation dans laquelle $\frac{\nu_{a_i}}{\nu}, i = 1, \dots, n$ et $\frac{\nu_{b_j}}{\nu}, j = 0, \dots, m$ sont des nombres entiers.

En modifiant la notation, l'équation (23) équivaut à :

$$y(t) + a'_1 \mathbf{D}^\nu y(t) + a'_2 \mathbf{D}^\nu (\mathbf{D}^\nu y(t)) + \dots + a'_n \overbrace{\mathbf{D}^\nu \left(\dots (\mathbf{D}^\nu y(t)) \right)}^{n' \text{ fois}} =$$

$$b'_0 \mathbf{D}^\nu u(t) + b'_1 \mathbf{D}^\nu (\mathbf{D}^\nu y(t)) + \dots + b'_m \underbrace{\mathbf{D}^\nu \left(\dots (\mathbf{D}^\nu u(t)) \right)}_{m' \text{ fois}}, \quad (24)$$

avec $n' = \frac{\nu_{a_n}}{\nu} \in \mathbb{N}$ et $m' = \frac{\nu_{b_m}}{\nu} \in \mathbb{N}$.

L'ordre de dérivation ν est un réel positif généralement fixé à l'ordre commensurable dont la définition suit.

b. Définition : ordre commensurable

L'ordre commensurable est ici défini comme le plus grand diviseur réel positif des ordres de dérivation de l'équation différentielle (21) $\{\nu_{a_1}, \dots, \nu_{a_n}, \nu_{b_0}, \dots, \nu_{b_m}\}$. Il vérifie :

$$\frac{\nu_{a_i}}{\nu} \in \mathbb{N}, \forall i = 1, \dots, n \quad \text{et} \quad \frac{\nu_{b_j}}{\nu} \in \mathbb{N}, \forall j = 0, \dots, m. \quad (25)$$

Dans le cas d'un système rationnel, l'ordre commensurable est un nombre entier généralement égal à 1.

c. Fonction de transfert

La transformée de *Laplace* de l'équation différentielle (21), soit

$$Y(s) + a_1 s^{\nu_{a_1}} Y(s) + a_2 s^{\nu_{a_2}} Y(s) + \dots + a_n s^{\nu_{a_n}} Y(s) =$$

$$b_0 s^{\nu_{b_0}} U(s) + b_1 s^{\nu_{b_1}} U(s) + b_2 s^{\nu_{b_2}} U(s) + \dots + b_m s^{\nu_{b_m}} U(s), \quad (36)$$

détermine directement la forme classique d'une fonction de transfert non entière :

$$F(s) = \frac{Y(s)}{U(s)} = \frac{b_0 s^{\nu_{b_0}} + b_1 s^{\nu_{b_1}} + b_2 s^{\nu_{b_2}} + \dots + b_m s^{\nu_{b_m}}}{1 + a_1 s^{\nu_{a_1}} + a_2 s^{\nu_{a_2}} + \dots + a_n s^{\nu_{a_n}}}. \quad (37)$$

Si le système est commensurable à l'ordre ν , cette fonction de transfert peut être réécrite selon :

$$F(s) = \frac{Y(s)}{U(s)} = \frac{b_0 \left(s^\nu \right)^{\frac{\nu_{b_0}}{\nu}} + b_1 \left(s^\nu \right)^{\frac{\nu_{b_1}}{\nu}} + b_2 \left(s^\nu \right)^{\frac{\nu_{b_2}}{\nu}} + \dots + b_m \left(s^\nu \right)^{\frac{\nu_{b_m}}{\nu}}}{1 + a_1 \left(s^\nu \right)^{\frac{\nu_{a_1}}{\nu}} + a_2 \left(s^\nu \right)^{\frac{\nu_{a_2}}{\nu}} + \dots + a_n \left(s^\nu \right)^{\frac{\nu_{a_n}}{\nu}}}, \quad (38)$$

avec $\frac{\nu_{a_i}}{\nu} \in \mathbb{N}, \forall i = 1, \dots, n$ et $\frac{\nu_{b_j}}{\nu} \in \mathbb{N}, \forall j = 0, \dots, m$.

Une notation particulière d'un transfert commensurable peut être définie par la fonction rationnelle R_ν :

$$\frac{Y(s)}{U(s)} = R_\nu(s^\nu), \quad (39)$$

avec :

$$R_\nu(x) = \frac{Q_\nu(x)}{P_\nu(x)} = \frac{b_0 x^{\frac{\nu_{b_0}}{\nu}} + b_1 x^{\frac{\nu_{b_1}}{\nu}} + b_2 x^{\frac{\nu_{b_2}}{\nu}} + \dots + b_m x^{\frac{\nu_{b_m}}{\nu}}}{1 + a_1 x^{\frac{\nu_{a_1}}{\nu}} + a_2 x^{\frac{\nu_{a_2}}{\nu}} + \dots + a_n x^{\frac{\nu_{a_n}}{\nu}}}, \quad (40)$$

où Q_ν et P_ν sont deux polynômes à puissances entières et ν l'ordre commensurable. Les zéros de Q_ν et P_ν sont respectivement les zéros en s^ν du numérateur et du dénominateur de F .

III. La Toolbox CRONE

1. Qu'est-ce que la toolbox CRONE ?

Une toolbox Matlab est l'agrégation d'objets mathématiques et des fonctions qui leurs sont associées dans le but de faciliter la réalisation de traitement dans un domaine précis.

L'objet de base de Matlab (Matrix Laboratory) étant une matrice, la toolbox permet de définir des structures associant plusieurs matrices, et représentant pour l'utilisateur les objets que l'on veut modéliser.

Dans le cas de la toolbox CRONE, les objets en question sont les fonctions de transfert de système à dérivées non entières.

2. Analyse de l'existant

Intéressons-nous à la manière dont les fonctions de transfert sont représentées. Nous retrouvons une structure hiérarchique à base d'héritage et d'agrégation courantes dans les applications orientées objets.

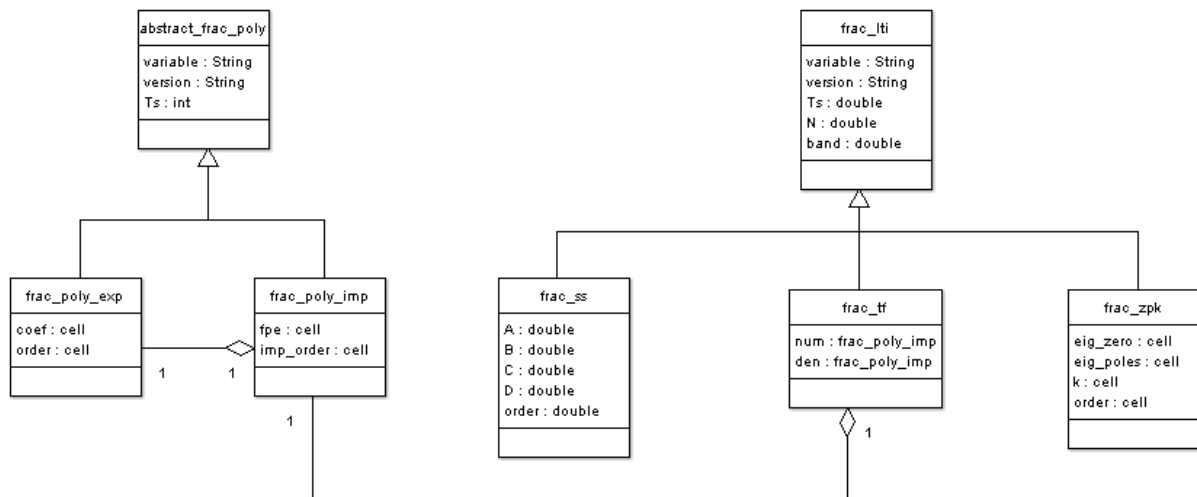


Figure 1 - Diagramme de classe de la toolbox existante

Les fonctions de transfert sont représentées par un objet (**frac_lti**), abstrait, c'est-à-dire qui n'est pas manipulable en tant que tel. Ce sont les classes « filles » qui seront manipulées dans le cadre de l'étude. Elles correspondent à trois représentations possibles des fonctions de transfert (*voir I*).

La pratique étant de définir une fonction de transfert par sa représentation `frac_tf`, puis éventuellement de la convertir en `frac_zpk` ou `frac_ss`, celle-ci est décomposée en deux attributs. Le numérateur et le dénominateur sont tous les deux des polynômes fractionnaires implicites. Un polynôme fractionnaire implicite étant lui-même en partie composé d'un polynôme fractionnaire explicite. Tous deux sont des classes filles d'une classe abstraite *`abstract_frac_poly`*.

Critique de l'architecture :

- Relation entre `fpi` et `fpe`.

Le rapport d'unicité liant un `fpi` à un `fpe` ne convient pas à une agrégation. En effet, un polynôme fractionnaire implicite est un polynôme fractionnaire explicite où l'on rajoute un attribut, l'ordre implicite. De ce fait un `fpi` n'est autre qu'un `fpe` spécialisé.

- L'existence de la classe `abstract_frac_poly`

Une classe chapeautant par héritage d'autres classes ayant des caractéristiques communes a pour but de factoriser et donc d'alléger le code tant à propos des attributs de classe que des méthodes. Examinons l'apport de la classe `abstract_frac_poly` sur ces deux aspects.

Au niveau des fonctions, la factorisation est impossible étant donné que celles-ci ont besoin pour traitement des attributs propre aux polynômes. Il s'en trouve que de nombreuses fonctions communes à `fpe` et `fpi` sont implémentées deux fois.

En ce qui concerne les attributs, la factorisation est faite mais induit une erreur. Effectivement, l'architecture retenue dissocie ces attributs selon la classe à laquelle ils appartiennent. C'est-à-dire qu'un `fpi` dont la variable a été précisée peut être composé d'un `fpe` ayant une variable différente.

De plus la manipulation des polynômes n'ayant pas de sens scientifique direct la présence de ces attributs n'est pas indispensable. Elle est même nuisible à l'étage supérieur, une fonction de transfert peut alors avoir des variables différentes au numérateur et au dénominateur, elles mêmes

différentes de la variable intrinsèque de la fonction de transfert (attributs de `frac_lti`)

Critique de l'implémentation :

- Redondance horizontale de code

L'existence d'objets multidimensionnels induit dans chaque fonction la présence d'un code de déconstruction jusqu'à un élément unitaire (polynôme de taille $[1 \ 1]$) pour lui appliquer la fonction. Ce code étant sensiblement identique dans toutes les fonctions il peut être factorisé.

- Redondance verticale de code

Comme nous l'avons vu précédemment, à cause de l'architecture actuelle beaucoup de fonctions se répètent dans `fpi` et `fpe`.

- Cas particuliers

Certains algorithmes sont améliorables, d'autres sont à reprendre complètement.

3. Synthèse des modifications à apporter

- Refonte du diagramme UML

La classe `abstract_frac_poly` disparaît avec ses attributs.

La relation entre `fpe` et `fpi` est modifiée en un héritage.

Pour ce qui est du reste du diagramme UML, aucune modification n'est envisagée.

Nouveau diagramme de classe :

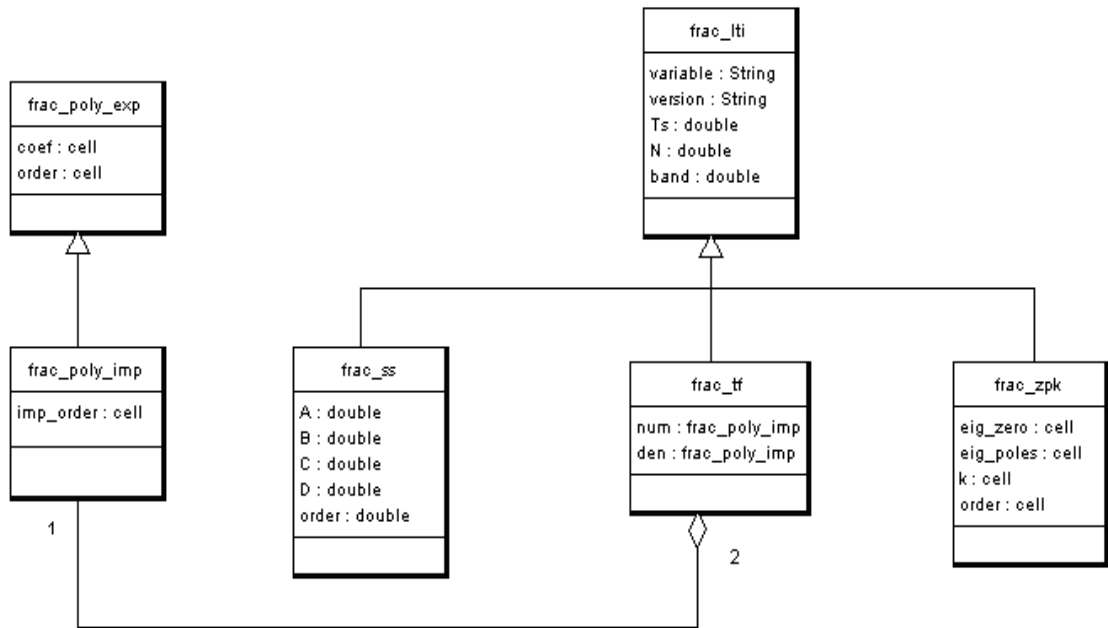


Figure 2 - Nouveau diagramme de classe :

- Généralisation de l'héritage

L'héritage entre `fpe` et `fpi` permet de n'implémenter la plupart des fonctions que dans la classe `fpe`. Lesquelles seront réutilisées dans la classe `fpi` par héritage.

- Création de la fonction « multi »

Une fonction « multi » réalisant l'externalisation du traitement des polynômes multidimensionnels est créée. Cette fonction est générique, elle reçoit en argument le nom de la fonction à exécuter en plus des arguments de cette dernière. Elle met en œuvre la déconstruction puis lance un appel de la fonction de base sur les polynômes unitaires.

- Amélioration d'algorithme

Anecdotiquement certains algorithmes subissent un toilettage.

Cette synthèse tient lieu de fil conducteur et permet de cadrer la réalisation des travaux.

IV. Réalisation

Présenter ici un descriptif exhaustif de la mise en œuvre du cahier des charges serait beaucoup trop long et ne correspondrait pas au format de ce rapport. C'est pourquoi le développement est abordé au moyen de quelques cas significatifs.

Par ailleurs la documentation complète est disponible sous la forme du « User's Guide » de la toolbox CRONE (*voir VJ*).

1. Traitement des polynômes remarquables

Le développement des fonctions a amené à faire certains choix quant au comportement des polynômes remarquables en particulier vis-à-vis des opérations arithmétiques. Ces polynômes sont les polynômes nul, unité, vide et nan.

Enoncé des règles retenues :

- Pour les opérations les polynômes nul et unité se comportent de la même manière que leurs homologues entiers.
- Le polynôme vide se comporte comme un élément neutre de l'addition.
- Le polynôme vide ne peut être opérande d'une multiplication autre qu'avec lui même
En effet une multiplication n'est possible qu'avec des polynômes de même taille et le polynôme vide est le seul de taille [0 0].
- Le polynôme nan se comporte comme un élément absorbant avec toutes les opérations (excepté la multiplication avec le polynôme vide).

2. Hiérarchisation et encapsulation des fonctions

L'objet de la hiérarchisation des fonctions est de découper des fonctions assez lourdes et volumineuses en une structure hiérarchique de fonctions. La fonction principale réalise le moins de travail possible en déléguant au maximum le traitement aux sous fonctions auxquelles elle fait appel.

De la hiérarchisation découle l'encapsulation. Cette dernière consiste à rigoureusement définir les entrées et sorties d'une fonction tout en occultant son fonctionnement interne. Ce

fonctionnement modulaire permet, outre de protéger les données, de faciliter l'évolutivité d'un projet.

Illustration par le cas de la fonction clean

La fonction clean a pour but d'ordonner et de simplifier (*nettoyer*) les polynômes, qu'ils soient mal saisis ou consécutivement à des opérations. Les trois tâches de la fonction sont :

- Classer les termes du polynôme selon les sens décroissant des ordres.
- Additionner les termes de même ordre pour supprimer les doublons
- Supprimer les termes ayant un coefficient nul

Précédemment la fonction clean réalisait toutes ces tâches grâce à un code développé d'un seul bloc. De ce fait découlait un code peu lisible et assez lourd.

Un nouveau découpage a été mis en place, créant une fonction correspondant à chacune des tâches. Le rôle de la fonction principale (*clean*) ne consiste plus qu'à appeler ces sous-fonctions.

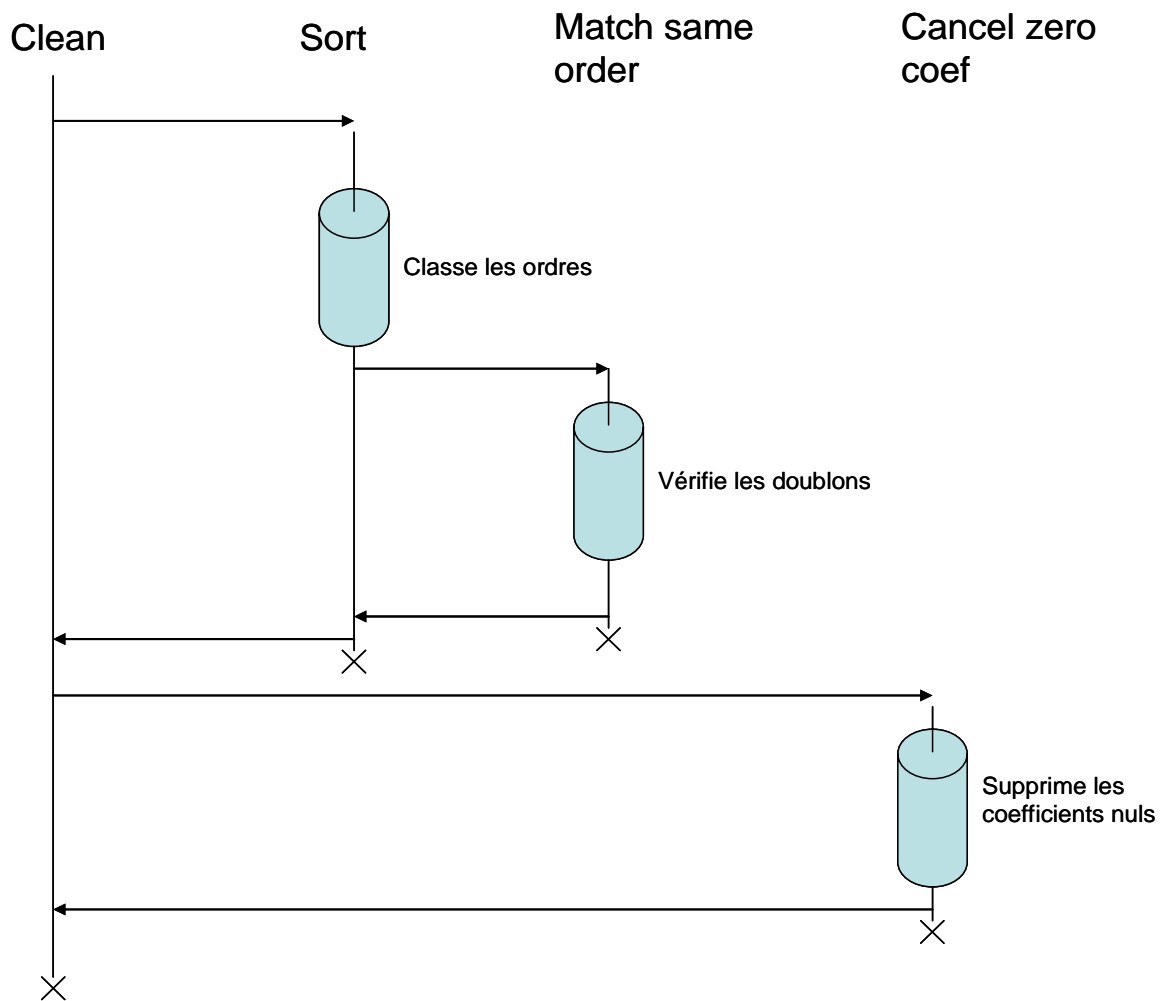


Figure 3 - Description du nouveau fonctionnement de la fonction clean

Les trois sous-fonctions correspondant à chacune des tâches sont :

- Sort
- Match same order
- Cancel zero coef

On remarque qu'aucun traitement n'est réalisé à proprement parler dans la fonction *clean*, celle-ci se bornant à appeler les sous-fonctions.

3. Mise en place de la fonction « multi »

Il s'agit de développer une fonction permettant d'externaliser la déconstruction des polynômes multidimensionnels, le traitement des polynômes unitaires et enfin le réassemblage du résultat pour reformer un polynôme multidimensionnel.

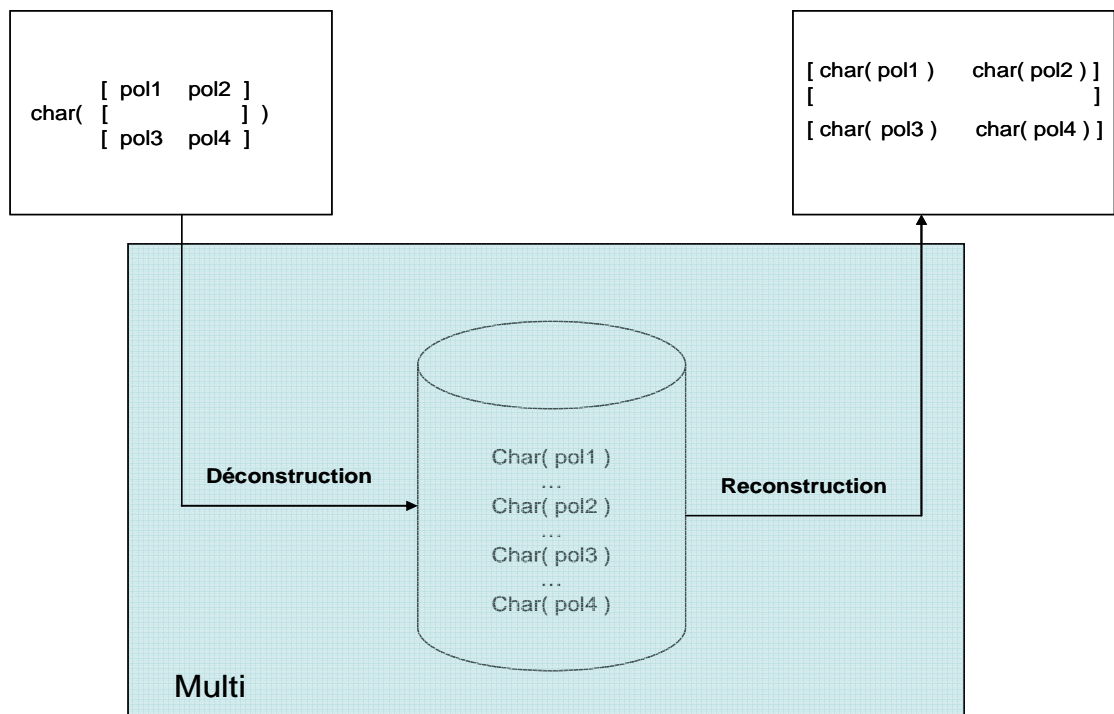


Figure 4 - Description du fonctionnement de la fonction multi

A première vue la fonction multi a besoin en argument du nom de la fonction traitée ainsi que des arguments originellement passés à cette fonction.

L'algorithme se déroule comme suit :

- Déconstruction du polynôme à l'aide d'une double boucle.

For i = 1 to largeur

For j = 1 to longueur

Isoler le polynôme

End

End

- Exécution de la fonction de base
- Reconstruction

Il s'agit de mettre en place une structure de la même taille que le polynôme multidimensionnel.

Il apparaît que pour réaliser la reconstruction il importe de connaître le nombre d'arguments que renvoie la fonction de base. Ceci permet alors de définir combien de case de la même taille que le polynôme multidimensionnel il faut prévoir. De même, de quel type doivent être ces cases ? Il faut là aussi connaître le type d'argument renvoyé. Ceux-ci peuvent être de deux natures. Soit un polynôme, auquel cas il s'agit de renvoyer un polynôme multidimensionnel, la méthode étant la concaténation. Soit d'autres variables (int, char...), cas où une cellule multidimensionnelle contiendra les résultats.

Il faut donc passer deux arguments supplémentaires à la fonction multi. Le premier lui indique combien d'arguments elle doit attendre en retour de la fonction demandée. Le deuxième, le nombre de polynômes parmi ces arguments.

De plus certaines fonctions acceptent plusieurs polynômes en entrée (*addition, soustraction...*). Il faut aussi que la fonction multi sache combien parmi les arguments d'entrée sont des polynômes. Ce qui lui permet ensuite de savoir combien d'argument sont à déconstruire. Cette information peut être obtenue par simple examen des arguments d'entrée et ne nécessite pas d'argument supplémentaire.

Nouveau plan de l'algorithme :

- Examen du nombre de polynômes en entrée
- Création de la structure d'accueil des résultats selon les informations passés en arguments (nombre total de cases, et type de ces cases).
- Déconstruction du (ou des) polynôme(s) à l'aide d'une double boucle.
- Exécution de la fonction de base
- Reconstruction en formant une structure

Code de la fonction :

```
% Recupération du nombre de fpe dans varargin
nbr_fpe=0;

while ((nbr_fpe < (nargin - 3)) &&
(isa(varargin{nbr_fpe+1}, 'frac_poly_exp'))
    nbr_fpe = nbr_fpe +1;
end

[x y]=size(varargin{1});
debut_var = nbr_fpe + 1;
string_res='';
obj(1)=varargin{1};
string_arg_fpe='subsref(obj(1),S)';

%Depending on the number of fpe argin, defining the objects
for m=2:nbr_fpe
    obj(m)=varargin{m};
    string_arg_fpe=[string_arg_fpe ',subsref(obj(' num2str(m) '),S)'];
end

if nbrout > 0

    %creation of the string passed in the function "eval". This string
depends

    %on the number of arg out

    string_res='[res1{i,j}';

    for k=2:nbrout

        string_res=[string_res ',res' num2str(k) '{i,j}'];

    end

    string_res=[string_res ']='];

%     end
end

%For each unidimensionnal system composing the multi dimensionnal one the
%function "fun_name" is called
```

```

for i=1:x
    for j=1:y
        %Select each unidim polynome
        S.type='(';
        S.subs={i,j};
        %tmp=subsref(obj,S);
        if nargin == nbr_fpe + 3
            eval([string_res fun_name '(' string_arg_fpe ');' ]);
        else
            eval([string_res fun_name '(' string_arg_fpe ','
varargin{debut_var:end} ');' ]);
        end
    end
end

if nbrout_fpe > 0
    for k=1:nbrout_fpe
        for i=1:x
            eval(['fp{i}=horzcat(res' num2str(k) '{i,:}');']);
        end
        eval(['varargout{k}=vertcat(fp{:});' ]);
    end
    for k=nbrout_fpe+1:nbrout
        eval(['varargout{k}=res' num2str(k) ';' ]);
    end
else
    for k=1:nbrout
        eval(['varargout{k}=res' num2str(k) ';' ]);
    end
end
end

```

4. Modification de l'algorithme de la fonction « commensurate »

L'algorithme de la fonction commensurate a été totalement remanié.

Cette fonction calcule le plus grand facteur commun des ordres d'un polynôme de telle sorte qu'en le factorisant le vecteur des ordres ne contienne que des entiers. Ce facteur est nommé *step order*.

Exemple :

$$\text{Commensurate}(s^{0,2} + s^{0,1} + s^{0,7}) = [0,1 \quad [2 \ 1 \ 7]]$$

Plan de l'algorithme :

S'il s'agit d'un polynôme unidimensionnel

- Récupération du tableau des ordres (ce sont des réels)
- Multiplication des ces ordres par la tolérance, pour être sur de travailler sur des entiers
- Détermination du pgcd des ordres (entiers). Ce pgcd divisé par la tolérance fournit le *step order*.
- Si le *step order* vaut 0 on renvoie le tableau des ordres identiques ainsi que le *step order*
Sinon on renvoie un tableau contenant les anciens ordres divisés par le *step order*.

S'il s'agit d'un polynôme multidimensionnel

- Un appel de la fonction multi fournit, pour chaque polynôme de taille unité le composant, un *step order* et un tableau contenant les nouveaux coefficients.
- Détermination du pgcd de tous les *step order* (ne pas oublier de multiplier les ordres puis de diviser le résultat par la tolérance pour travailler avec des entiers). Ce résultat est le *step order* général du polynôme multidimensionnel.

- Tous les tableaux de coefficients sont chacun pondérés par le *step order* leur correspondant divisé par le *step order* général.
- Est renvoyé un scalaire (*step order*) et une cellule contenant les tableaux de nouveaux coefficients.

Code de la fonction :

```
% Premier test : cas NaN et Empty

if isnan(p)

    step_order=[];

    new_order=[];

    disp('The object is NaN polynomial');

    return

elseif isempty(p)

    step_order=[];

    new_order=[];

    disp('The object is Empty polynomial');

    return

end

% Cas unidimensionnel

if size(p)== [1 1]

    % Récupération des ordres

    frac_order=get(p,'order');

    frac_order_int=round(frac_order{1}/tolord);

    % Recherche de l'ordre commensurable

    step_int=frac_order_int(1);

    taille=size(frac_order_int,2);

    for i=2:taille

        step_int=gcd(step_int,frac_order_int(i));

    end

    % Affectation des nouvelles valeurs
```

```

step_order=step_int*tolord;

if step_order == 0

    new_order{1}=frac_order{1};

else

    new_order{1}=frac_order_int/step_int;

end

else

[k l]=size(p);

% Appel la fonction multi

[step_order,new_order]=multi('commensurate',2,0,p);

% Calcul du pgcd de la cellule step order

step_int=(step_order{1,1}/tolord);

for i=1:k

    for j=1:l

        step_int=gcd(step_int,(round(step_order{i,j}/tolord)));

    end

end

step=step_int*tolord;

% recalcul des ordres

for i=1:k

    for j=1:l

        new_order{i,j}= (step_order{i,j}/step)*new_order{i,j}{1};

    end

end

% le step_order renvoyé est step (scalaire)

step_order=step;

end

```


V. Procédures de test

Le développement d'un projet tel que la toolbox CRONE nécessite la mise au point de procédures de test rigoureuses en vue de la validation du travail effectué. De plus s'agissant d'un projet de longue haleine impliquant successivement plusieurs intervenants, il est important de mettre en place un manuel expliquant et détaillant le travail réalisé.

1. Test et démo des différentes fonctions

La réponse aux exigences de test et de démo a été apportée par une seule et même réalisation. Pour chaque fonction, ont été développées des fonctions test se voulant une liste exhaustive des cas d'utilisation de ces fonctions.

L'exécution d'un fichier test à travers la commande *echodemo* lance une démonstration didactique de la fonction. Tout d'abord sont présentés le prototype complet de la fonction ainsi que son objectif. Puis, pas à pas, tous les cas d'utilisation sont développés. Ainsi chaque cas est, non seulement explicité mais, aussi exécuté s'assurant de son bon fonctionnement.

Les cas d'utilisation usuels sont les suivants :

- Fonctionnement avec les polynômes remarquables (nul, unité, vide et nan)
- Fonctionnement suivant les divers prototypes (arguments optionnels)
- Fonctionnement du mode multidimensionnel

```
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
Click Next to continue or Stop to end test\_commensurate.m (1/4)
-----
DESCRIPTION

Test function on :
    - function : commensurate
    - class    : frac_poly_exp

Prototype :
    res=commensurate(fpe)

Argument in :
    fpe : Frac_poly_exp object

Argument out :
    step_order : commun non integer step (scalar)
    new_order  : new order vector of p (vector or cell of vector)

Commensurate computes the commun factor of the order vector and
returns the integer vector with the commun factor step_order.
-----
>> |
```

Figure 5 - Première page : prototype et décripsion de la fonction

```
Command Window
i New to MATLAB? Watch this Video, see Demos, or read Getting Started.
Click Next to continue or Stop to end test\_commensurate.m (2/4)
-----
THE CLASSIC CASE : UNIDIMENSIONNAL
-----

pol1=frac_poly_exp([1,1,1,1],[0.2,0.4,0.8,1.2]);

% Consider the following : pol1 = s^1.2 +s^0.8 +s^0.4 +s^0.2
%
% The function commensurate gives back the vector of integer order.
% Which orders form (with the same coefficients) the equivalent polynomial
% through the transformation p = s^(step_order)
%
% pol1 <==> p^6+ p^4 + p^2 + p^1 and p = s^0.2

[step_order, new_order]=commensurate(pol1);
step_order
new_order

step_order =

    0.2000

new_order =

    [1x4 double]

>> |
```

Figure 6 - Pages suivantes : divers cas d'utilisation

2. Rapport de test par classe

Un deuxième processus complète le système de test par fonction. Pour chaque classe, un fichier teste le bon déroulement de l'ensemble des fichiers test précédemment exposés.

Ce rapport de test comporte plusieurs intérêts :

- Permettre une visualisation succincte de l'ensemble des fonctions d'une classe.
- Repérer rapidement quelle fonction sont erronées

- Faciliter la maintenance du projet. En effet après la modification d'une fonction, on peut ainsi rapidement vérifier qu'elle n'a pas impactée le fonctionnement des autres fonctions de la classe.

Rapport de test de la classe frac_poly_exp :

 New to MATLAB? Watch this [Video](#), see [Demos](#), or read [Getting Started](#).

```
*****
*               Test Report of               *
*           the class Frac_poly_exp           *
*               v 0.1                         *
*****

*** press a key ***

****  Frac_poly_exp
|  Checked  |  Result  |  Command
|    1     |    1     |  test\_frac\_poly\_exp.m
|    1     |    1     |  test\_append.m
|    1     |    1     |  test\_cancel\_zero\_coef.m
|    1     |    1     |  char.m
|    1     |    1     |  test\_clean.m
|    1     |    1     |  test\_coef.m
|    1     |    1     |  test\_commensurate.m
|    1     |    1     |  test\_display.m
|    1     |    1     |  test\_eig.m
|    1     |    1     |  test\_eq.m
|    1     |    1     |  test\_get.m
|    1     |    1     |  test\_horzcat.m
|    1     |    1     |  test\_iscomplex.m
|    1     |    1     |  test\_isempty.m
|    1     |    1     |  test\_isnan.m
|    1     |    1     |  test\_ldivide.m
|    1     |    1     |  test\_match\_same\_order.m
|    1     |    1     |  test\_minus.m
|    1     |    1     |  test\_mpower.m
|    1     |    1     |  test\_mtimes.m
|    1     |    1     |  test\_multi.m
|    1     |    1     |  test\_ne.m
|    1     |    1     |  test\_order.m
|    1     |    1     |  test\_plus.m
|    1     |    1     |  test\_rdivide.m
|    1     |    1     |  test\_set.m
|    1     |    1     |  test\_size.m
|    1     |    1     |  test\_sort.m
|    1     |    1     |  test\_times.m
|    1     |    1     |  test\_transpose.m
|    1     |    1     |  test\_uminus.m
|    1     |    1     |  test\_vertcat.m

32 of Functions tested

Found:  0 error(s)
Fully functional.
>> |
```

Figure 7 - Rapport de test de la classe frac_poly_exp

VI. Documentation : le User's Guide de la toolbox Matlab

Outre les démos et l'aide consultable en ligne, la toolbox Matlab est accompagnée d'une documentation papier. Cette documentation reprend les aspects théoriques des systèmes à dérivées non entières, l'organisation au sens UML de la toolbox, l'interface graphique appréhendée par l'utilisateur et enfin les références des fonctions.

L'apport à la documentation s'est ici borné aux parties relatives à l'organisation UML et au référencement des fonctions. Pour l'organisation, il consiste en un nouveau diagramme de classe (*voir III] 3*). Quant aux références, étant déjà basées sur le trio : syntaxe, description, exemple, ce schéma a été conservé.

Exemple d'une référence :

ne

Test for unequality

Syntax

```
fp1 ~= fp1  
bool=ne(fp1, fp2)
```

Description

ne(fp1, fp2) is called when the syntax `fp1 ~= fp2` is used.

`fp1 ~= fp2` compares each element of fp1 with the corresponding element of fp2, and returns a logical 1 (true) if fp1 and fp2 are unequal, or logical 0 (false) if they are equal.

By convention, two NaN polynomials are not equal and two Empty polynomials are equal.

Arguments

Argument in :

fp1 : frac_tf or frac_poly_exp

fp2 : frac_tf or frac_poly_exp

Argument out :

bool : boolean

Example

```
>>pol1=frac_poly_exp([1,2,4],[3,2,0]);  
>>pol1bis=frac_poly_exp([2,4,8],[3,2,0]);
```

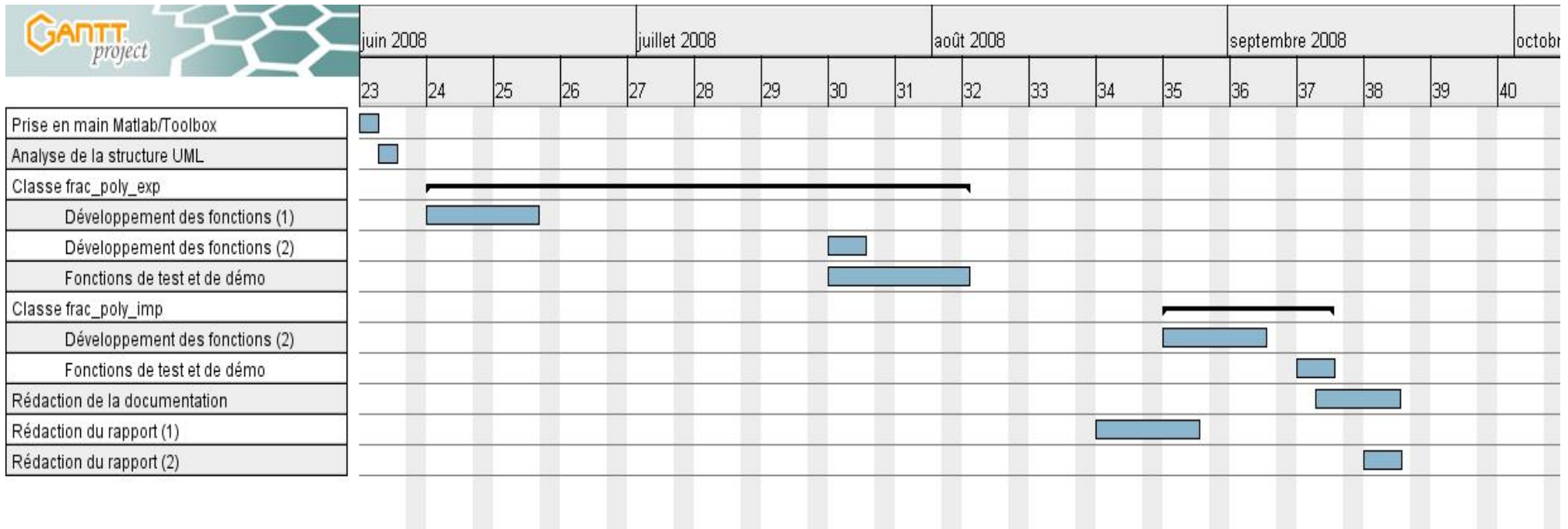
```
>>ne(poll,poll)
ans = 0
>>ne(poll,pollbis)
ans = 1
>>ne(pol_multi,pol_multi)
ans =  [0]    [1]
       [0]    [1]
```

VII. Conclusion

1. Organisation du travail

Mon travail lors de ce stage s'est déroulé en trois phases. Tout d'abord l'analyse structurelle de la toolbox CRONE ainsi que la mise au point du cahier des charges. La deuxième étape a été le développement et le test des fonctions relatives aux deux classes abordées durant mon stage. Enfin, la finalisation du stage a consisté en la rédaction de la documentation relative à ces fonctions.

Ci-après figure un diagramme de Gantt sur lequel a été représentée l'organisation temporelle de mon travail. On dénotera deux pauses dans la réalisation des diverses tâches, la première lors du début du mois de juillet, la seconde au début du mois d'Aout. La seconde pause correspond en effet à la fermeture annuelle du laboratoire. La première, quant à elle, est due à un arrêt de travail consécutif à un problème de santé.



2. Bilan du stage

Ce stage, au sein du laboratoire IMS-LAPS, m'as permis de mettre en pratique une expertise dans l'organisation fonctionnel de système technologique, ici informatique. En effet la partie du travail consistant à analyser la structure de la toolbox et à en proposer une nouvelle correspond tout à fait à mes aspirations en tant que futur ingénieur.

Le développement et le test des fonctions à fait appel à mes connaissances d'algorithmique tout en me faisant découvrir la possibilité du développement objet sous matlab. Le développement des fonctions matlab en elles-mêmes ne m'a pas été d'un apport considérable, mais à mis en évidence un point très important pour un ingénieur, à savoir de pouvoir s'adapter très rapidement à un environnement ou une technologie entièrement nouvelle.

Bien que par certain côté assez fastidieuse, la rédaction de la documentation a été très enrichissante. Celle-ci à permis, à travers un échange, de recadrer le travail effectué. En l'occurrence, de bien déterminer les informations pertinentes à transmettre à la fois dans une perspective d'évolution et d'utilisation.

Pour conclure j'ajouterais qu'il est regrettable que mon arrêt de travail n'est amputé une partie du temps qui m'était alloué. Je n'ai pu aborder la classe *frac_tf*, ce qui aurait permis d'amener mon travail jusqu'au stade opérationnel.