SymPy par la pratique

K.I.A.Derouiche

PUBLISHED BY PUBLISHER

BOOK-WEBSITE.COM

First printing, March 2018

Contents

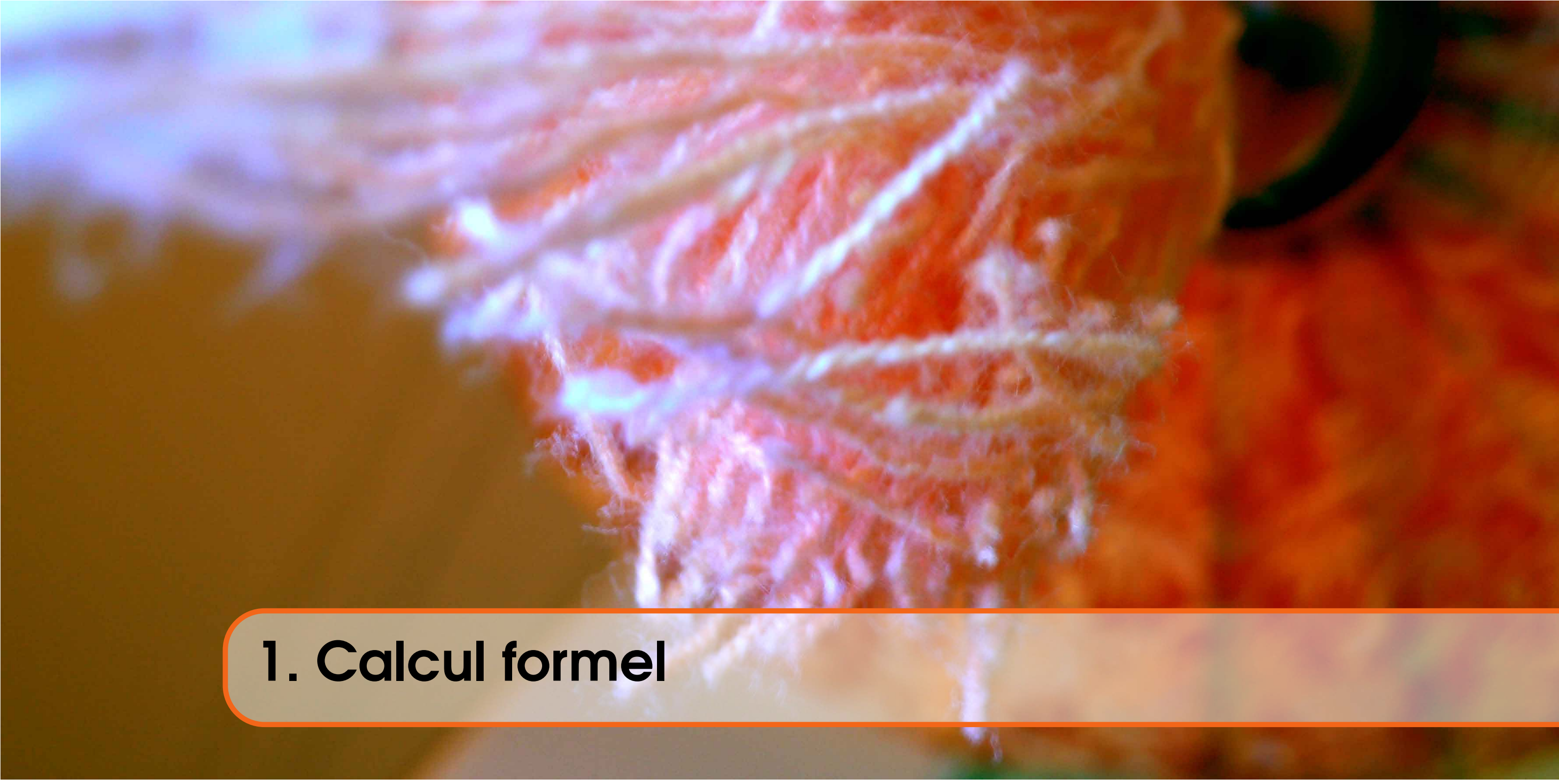
0.1 Introduction

Ce recueil d'exercices et de problèmes de programmation s'adresse aussi bien aux débutants qu'aux programmeurs confirmés. Il présente en effet plusieurs états d'esprit dont les deux principaux sont la programmation classique en Pascal pour les étudiants du premier cycle universitaire, et la programmation fonctionnelle en Lisp pour le second cycle.

Ce livre constitue un panorama (non exhaustif, mais suffisant) sur les langages de programmation, et offre une grande variété dans les sujets traités : graphiques, calcul matriciel, traitements de chaînes de caractères, graphes, intelligence artificielle...
La première partie du livre sera consacré à la résolution par une approche symbolique au divers questions posées au étudiants et toute personnes qui aiment savoir et voir s'initier pour des niveaux et des questions rencontrés, la deuxième partie du livre sera questions aux problèmes plus rencontrés pour des étudiants passionnée des questions entre mathématiques et technologies, chercheurs et développeurs d'applications scientifiques, la troisième partie plus consacré aux questions poussées

0.1.1 Pourquoi programmer en symbolique

Part One



1. Calcul formel

L'approche La simulation numérique est devenue essentielle dans de nombreux domaines tels que la mécanique des fluides et des solides, la météo, l'évolution du climat, la biologie ou les semi-conducteurs. Elle permet de comprendre, de prévoir, d'accéder là où les instruments de mesures s'arrêtent.

Ce livre présente des méthodes performantes du calcul scientifique : matrices creuses, résolution efficace des grands systèmes linéaires, ainsi que de nombreuses applications à la résolution par éléments finis et différences finies. Alternant algorithmes et applications, les programmes sont directement présentés en langage C++. Ils sont sous forme concise et claire, et utilisent largement les notions de classe et de généricité du langage C++.

Le contenu de ce livre a fait l'objet de cours de troisième année à l'école nationale supérieure d'informatique et de mathématiques appliquées de Grenoble (ENSIMAG) ainsi qu'au mastère de mathématiques appliquées de l'université Joseph Fourier. Des connaissances de base d'algèbre matricielle et de programmation sont recommandées. La maîtrise du contenu de cet ouvrage permet d'appréhender les principaux paradigmes de programmation du calcul scientifique. Il est alors possible d'appliquer ces paradigmes pour aborder des problèmes d'intérêt pratique, tels que la résolution des équations aux dérivées partielles, qui est abordée au cours de ce livre. La diversité des sujets abordés, l'efficacité des algorithmes présentés et leur écriture directe en langage C++ font de cet ouvrage un recueil fort utile dans la vie professionnelle d'un ingénieur.

Le premier chapitre présente les bases fondamentales pour la suite : présentation du langage C++ à travers la conception d'une classe de quaternions et outils d'analyse asymptotique du temps de calcul des algorithmes. Le second chapitre aborde l'algorithme de transformée de Fourier rapide et développe deux applications à la discrétisation d'équations aux dérivées partielles par la méthode des différences finies. Le troisième chapitre est dédié aux matrices creuses et à l'algorithme du gradient conjugué. Ces notions sont appliquées à la méthode des éléments finis. En annexe sont groupés des exemples de génération de maillage et de visualisation graphique.

S'il est cependant recommandé de maîtriser les notions du premier chapitre pour aborder le reste du livre, les chapitres deux et trois sont complètement indépendants et peuvent être abordés séparément. Ces chapitres sont complétés par des exercices qui en constituent des développements, ainsi que des notes bibliographiques retraçant l'historique des travaux et fournissant des références sur des logiciels et librairies récents implémentant ou étendant les algorithmes présentés.

1.1 Système de calcul formel

Dans le cas de donnée des explications et représentations très proche, car la question qui relie entre le logiciel et l'ordinateur d'un coté et la démonstration mathématique, est un pas décisive est très important l'exemple de l'intervalle beaucoup plus technique, donc il y a beaucoup de CAS

■ **Example 1.1** Let $G = \{x \in \mathbb{R}^2 : |x| < 3\}$ and denoted by: $x^0 = (1,1)$; consider the function:

$$f(x) = \begin{cases} \mathrm{e}^{|x|} & \text{si } |x - x^0| \leq 1/2 \\ 0 & \text{si } |x - x^0| > 1/2 \end{cases} \tag{1.1}$$

The function f has bounded support, we can take $A = \{x \in \mathbb{R}^2 : |x - x^0| \leq 1/2 + \varepsilon\}$ for all $\varepsilon \in]0; 5/2 - \sqrt{2}[$. ■

qui exprime ce qui nous permet notre choix pour un CAS qui possède des caractéristiques techniques et sur le plan du coût très important quand peut résumer dans les points suivants:

1. Leger et
2. S'appuie sur le langage de programmation Python
3. Portabilité dans toute transparence

L'un des systèmes qui peut nous permettre d'écrire cette exemple avec un ordinateurs avec SymPy qui semble mieu intégré

1.2 Bibliothèque SymPy

Dans un cas plus simple l'exemple 1.1 se formule beaucoup plus dans un outil comme SymPy est une bibliothèque de calcul formel elle est aussi un environnement pour l'apprentissage de l'algèbre, l'analyse, géométrie, combinatoire, cryptographie, mécanique classique et quantique pour le lycée et l'université mais aussi un environnement de développement et de recherche. SymPy écrit entièrement en Python un langage de programmation facile à apprendre et adapté à l'apprentissage, elle fourni aux étudiant *SymPyGamma* une application web notamment des primitives générales de traitement des expressions algébriques (développement, factorisation, ...), des aides à l'organisation des objets mathématiques intervenant dans la résolution d'un problème ainsi qu'une assistance à la preuve. Il permet au professeur de préparer et de suivre le travail de l'élève. Différentes maquettes ont été développées et testées auprès d'élèves. Dans la plus récente, nous nous sommes attachés à explorer une nouvelle forme d'activité algébrique. Alors que le calcul en papier crayon et les logiciels standards considèrent les expressions de façon isolée, l'environnement que nous développons organise en réseau les différentes expressions intervenant dans la résolution d'un problème. L'ordinateur peut facilement mettre à jour ce réseau quand l'utilisateur modifie certains de ses éléments. Il devient ainsi possible, pour aborder un problème générique, d'explorer facilement des cas particuliers et de conduire une généralisation. Les relations entre expressions algébriques sont mieux mises en évidence du fait de leur invariance dans les modifications du réseau. De façon très concise, Casyopée peut être défini

1.2.1 SymPyGamma

Est une interface onWeb marche avec un navigateur contient plusieurs catégorie liée de calcul, dynamique. L'interet de cette outil qu'il est facilement partageable adapté pour l'enseignement et surtout l'auto-apprentissage

1.2.2 Besoin de rester dans le symbolique

Le symbolique est une grande importance d'un point de vue technique, car il permet de limité les risques de bug dans l'exécution des programmes, dans le contexte de la vérification formelle si en prend le programme suivant:

1.2.3 Passage du symbolique au numérique

Généralement, le symbolique parmis c'est

1.2.4 Faire des dessins

1.3 Citation

This statement requires citation [**article_key**]; this one is more specific [**book_key**].

1.4 Lists

Lists are useful to present information in a concise and/or ordered way[1].

1.4.1 Numbered List

1. The first item
2. The second item
3. The third item

1.4.2 Bullet Points

- The first item
- The second item
- The third item

1.4.3 Descriptions and Definitions

Name Description
Word Definition
Comment Elaboration

[1] Footnote example...

2. Problèmes et exercices

2.1 Premiers pas en Python symbolique

This is an example of theorems.

2.1.1 Variable et affection

Exercise 2.1 Affectez les variables temps t et distance d par les valeurs 6.892 et 19.7. Calculez et affichez la valeur de la vitesse. Améliorez l'affichage en imposant un chiffre après le point décimal. ■

Solution 2.1 Pour, affectez des variables est les rendre symbolique comme c'est décrit dans le mémo ou il sera expliquer temps t et distance d par les valeurs 6.892 et 19.7. Calculez et affichez la valeur de la vitesse. Améliorez l'affichage en imposant un chiffre après le point décimal.

2.1.2 Contrôle du flux d'instructions

This is a theorem consisting of just one line.

Exercise 2.2 A set $\mathscr{D}(G)$ in dense in $L^2(G), |\cdot|_0$. ■

Solution 2.2

2.2 Les Fonctions

This is an example of a definition. A definition could be mathematical or it could define a concept.

Exercise 2.3 Écrire une fonction cube qui retourne le cube de son argument ■

Exercise 2.4 Écrire une fonction *volumeSphere* qui calcule le volume d'une sphère de rayon r fourni en argument et qui utilise la fonction cube . Tester la fonction *volumeSphere* par un appel dans le programme principal. ■

Exercise 2.5 Écrire une fonction maFonction qui retourne $f(x) = 2x^3 + x - 5$ ■

Exercise 2.6 Écrire une fonction tabuler avec quatre paramètres : $fonction$, $borneInf$, $borneSup$ et $nbPas$. Cette procédure affiche les valeurs de $fonction$, de $borneInf$ à $borneSup$, tous les nbPas . Elle doit respecter $borneInf < borneSup$. Tester cette fonction par un appel dans le programme principal après avoir saisi les deux bornes dans une floatbox et le nombre de pas dans une integerbox (utilisez le module easyguiB). ■

Exercise 2.7 Écrire une fonction $volMasse$ Ellipsoide qui retourne le volume et la masse d'un ellipsoïde grâce à un tuple. Les paramètres sont les trois demi-axes et la masse volumique. On donnera à ces quatre paramètres des valeurs par défaut.
On donne: $v = \frac{3}{4}\pi abc$
Tester cette fonction par des appels avec différents nombres d'arguments. ■

Exercise 2.8 Une fonction $f(x)$ est linéaire et a une valeur de 29 à $x = -2$ et 39 à $x = 3$. Trouver sa valeur à $x = 5$. ■

Exercise 2.9 Pour l'ensemble N de nombres naturels et une opération binaire $f : NxN \longrightarrow N$, on appelle un élément $z \ \varepsilon \ N$ une identité pour f, si $f(a,z) = a = f(z,a)$, pour tout a ε N. Lesquelles des opérations binaires suivantes ont une identité?:

1. $f(x,y) = x + y - 3$
2. $f(x,y) = max(x,y)$
3. $f(x,y) = x^y$

■

Solution 2.3 le deuxième et le troisième

2.3 Structures de données

2.3.1 Logique

Exercise 2.10 Dans la carte de Karnaugh ci-dessous, X indique un terme sans intérêt. Quelle est la forme minimale de la fonction représentée par la carte de Karnaugh? ■

2.3.2 Ensembles

La notion d'objet immuable en Python est fondamentale, une structure qui rappel les ensembles en mathématiques que soit fini ou infini est *set*, importante, bien que dans le cadre de SymPy elle s'appui entièrement sur Python avec certain modification, avec la collection d'objet.

La fonction set accepte donc en argument un objet de type quelconque et s'efforce de le traduire dans un ensemble. Lorsqu'on ne passe aucun argument à set (option 2), ou qu'on lui passe une liste vide, set renvoie naturellement un ensemble vide; on aurait pu utiliser aussi bien, de la même manière, set(()), set(), ou même set(") pour arriver au même résultat.

Exercise 2.11 Définir deux ensembles $X = \{a,b,c,d\}$ et $Y = \{s,b,d\}$, puis affichez les résultats suivants :

1. les ensembles initiaux.

2. le test d'appartenance de l'élément c à X.
3. le test d'appartenance de l'élément a à Y.
4. les ensembles $X - Y$ et $Y - X$.
5. l'ensemble $X \cup Y$ (union).
6. l'ensemble $X \cap Y$ (intersection).

Solution 2.4 Il faut noter qu'il existe une solution qui se base sur le Python builtuints en utilisant la structure de donnée *sets*. Mais comme en n'est dans la logique en utilise

```
from sympy import FiniteSet

X = FiniteSet('a', 'b', 'c', 'd')
Y = FiniteSet('s', 'b', 'd')

class MyClass(Yourclass):
    def __init__(self, my, yours):
        bla = '5 1 2 3 4'
        print bla
```

```
class MyClass(Yourclass):
    def __init__(self, my, yours):
        bla = '5 1 2 3 4'
        print bla
```

2.3.3 Polynômes

Exercise 2.12 Considérons le polynôme $p(x) = a_0 + a_1x + a_2x^2 + a_3x^3$, où $a_i \neq 0\ \forall i$. Le nombre minimum de multiplications nécessaires pour évaluer p sur une entrée x est:

2.3.4 Nombres parfaits et nombres chanceux

Exercise 2.13 $\sqrt{12}$

2.4 Remarks

This is an example of a remark.

R The concepts presented here are now in conventional employment in mathematics. Vector spaces are taken over the field $\mathbb{K} = \mathbb{R}$, however, established properties are easily extended to $\mathbb{K} = \mathbb{C}$.

2.5 Corollaries

This is an example of a corollary.

Corollary 2.5.1 — Corollary name. The concepts presented here are now in conventional employment in mathematics. Vector spaces are taken over the field $\mathbb{K} = \mathbb{R}$, however, established properties are easily extended to $\mathbb{K} = \mathbb{C}$.

2.6 Propositions

This is an example of propositions.

2.6.1 Several equations

Proposition 2.6.1 — Proposition name. It has the properties:

$$\big| ||\mathbf{x}|| - ||\mathbf{y}|| \big| \leq ||\mathbf{x} - \mathbf{y}|| \tag{2.1}$$

$$||\sum_{i=1}^{n} \mathbf{x}_i|| \leq \sum_{i=1}^{n} ||\mathbf{x}_i|| \quad \text{where } n \text{ is a finite integer} \tag{2.2}$$

2.6.2 Single Line

Proposition 2.6.2 Let $f,g \in L^2(G)$; if $\forall \varphi \in \mathscr{D}(G)$, $(f,\varphi)_0 = (g,\varphi)_0$ then $f = g$.

2.7 Examples

This is an example of examples.

2.7.1 Equation and Text

■ **Example 2.1** Let $G = \{x \in \mathbb{R}^2 : |x| < 3\}$ and denoted by: $x^0 = (1,1)$; consider the function:

$$f(x) = \begin{cases} \mathrm{e}^{|x|} & \text{si } |x - x^0| \leq 1/2 \\ 0 & \text{si } |x - x^0| > 1/2 \end{cases} \tag{2.3}$$

The function f has bounded support, we can take $A = \{x \in \mathbb{R}^2 : |x - x^0| \leq 1/2 + \varepsilon\}$ for all $\varepsilon \in]0;5/2 - \sqrt{2}[$. ■

2.7.2 Paragraph of Text

■ **Example 2.2 — Example name.** Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris. ■

2.8 Exercises

This is an example of an exercise.

> **Exercise 2.14** This is a good place to ask a question to test learning progress or further cement ideas into students' minds. ■

2.9 Problems

Problem 2.1 What is the average airspeed velocity of an unladen swallow?

2.10 Vocabulary

Define a word to improve a students' vocabulary.
Vocabulary 2.1 — Word. Definition of word.

Part Two

3. POO

3.1 Programmation Orientée Objet

Notation 3.1. *Given an open subset G of $\mathbb{R}^n$, the set of functions φ are:*

1. *Bounded support G;*
2. *Infinitely differentiable;*

a vector space is denoted by $\mathscr{D}(G)$.

3.1.1 POO

Exercise 3.1 Définir une classe Vecteur2D avec un constructeur fournissant les coordonnées par défaut d'un vecteur du plan (par exemple : $x = 0$ et $y = 0$). Dans le programme principal, instanciez un Vecteur2D sans paramètre, un Vecteur2D avec ses deux paramètres, et affichez-les. ■

Solution 3.1 en utilise le module sympy.geometry ce module fait appel à tout les outils et theories qui peuvents entre utiliser dans le cade de la géométrie dans le Plan.

```
from sympy.geometry
```

Exercise 3.2 Enrichissez la classe Vecteur2D précédente en lui ajoutant une méthode d'affichage et une méthode de surcharge d'addition de deux vecteurs du plan. Dans le programme principal, instanciez deux Vecteur2D , affichez-les et affichez leur somme. ■

Solution 3.2

3.1.2 Notions de COO et d'encapsulation

Part Three

4. Nonlinear Problem

Dans cette section ou se mele symbolique et numérique pour modélisé et résoudre des problèmes complexes, l'accents et prix sur des questions un

4.1 Chaos

Prenons une pause dans l'apprentissage de nouvelles techniques et algorithmes informatiques pour un peu, et passer du temps en utilisant ce que nous avons appris jusqu'à présent pour enquêter sur quelque chose d'intéressant. Nous allons commencer avec quelque chose de familier: le simple pendule.

4.1.1 Pendule simple

Le pendule simple figure

4.1.2 Pendule à deux bras

4.1.3 Les mouvements d'un robot

4.2 Solution non linéaire d'equation algébrique

Qu'est ce que non-linéaire et qu'est ce que une équation algébrique

Une équation algébrique est un polynôme de la forme $P(x)$

$$\exp(-x)\sin(x) = \cos(x) \tag{4.1}$$

4.3 Transport optimal

C'est quoi le *transport optimal*?, exemple simple..., le domaine du transport optimal est très

4.4 Le calcul des variations

4.5 Figure

Treatments	**Response 1**	**Response 2**
Treatment 1	0.0003262	0.562
Treatment 2	0.0015681	0.910
Treatment 3	0.0009271	0.296

Table 4.1: Table caption

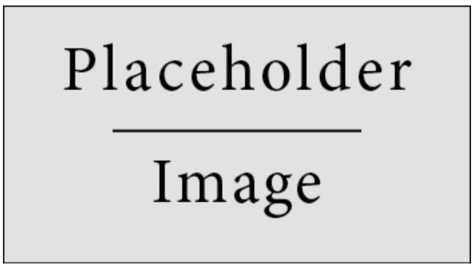


Figure 4.1: Figure caption

Bibliography

Index