

Avant-propos

Le livre que vous avez entre les mains est celui d'un mathématicien, programmeur du dimanche, certes, mais assidu à ce hobby – avec parfois le prétexte d'une motivation professionnelle.

Si je fais aujourd'hui la promotion de Julia, c'est pour deux raisons : d'abord, aucun langage ne m'a autant enthousiasmé depuis ma découverte du Turbo Pascal en 1988. Ensuite, je crois utile de porter un contrepoids à la position dominante du langage Python, qui freine la diffusion de solutions performantes plus récentes¹. Un effort particulier est donc nécessaire si on veut susciter la curiosité du public naturel (étudiants en mathématiques, enseignants, ingénieurs, chercheurs) pour d'autres langages, en particulier pour Julia. Pour ma part, je pense que la formation initiale (au lycée, à l'université) ne doit pas se contenter de répliquer des standards éphémères. Il est pertinent d'illustrer des concepts universels avec des outils divers, et parmi eux, de faire une place à ceux qui innovent dans des directions prometteuses. Ainsi, les méthodes multiples et la compilation *Just in Time* permettent à Julia de concilier écriture souple et vitesse d'exécution ; une situation qui pourra devenir fréquente puisqu'on manie davantage qu'hier des jeux de données de grande taille.

Quelques mots pour expliquer comment Julia est devenu mon couteau suisse de mathématicien : j'ai découvert Julia il y a cinq ans sur le blog de mon collègue Djilil Chafai. À l'époque, mes activités de programmation se partageaient surtout entre Scilab et C/C++. J'appréciais Scilab pour sa gestion des vecteurs, qui permet d'écrire très vite un code assez compact et de tester rapidement une petite conjecture de routine. En revanche, le manque de rapidité était rédhibitoire pour l'étude de modèles sur les grands réseaux. Pour les programmes d'envergure, passage au C/C++, donc, mais non sans peine. Comme la plupart des mathématiciens, je n'ai pas assimilé sérieusement les bases de la programmation objet ; or la pensée objet est au coeur des interfaces des bibliothèques C++, ce qui donne un coût d'entrée assez élevé

1. Python a été rendu public en 1991, soit 21 ans avant Julia.

à son utilisation. Quand on doit réécrire des bouts de bibliothèques écrits par d'autres suite à des évolutions du langage (en particulier de la couche objet), c'est encore pire. Et on n'a toujours pas la simplicité de syntaxe de Scilab.

J'ai ainsi vu dans Julia la promesse de la simplicité d'une syntaxe à la Scilab avec une rapidité d'exécution voisine du C. Je n'ai pas été déçu, et j'ai de plus obtenu des satisfactions inattendues. La première surprise, c'est la vitesse avec laquelle s'est développé une large base de bibliothèques facilement installables. J'ai ainsi pu porter en Julia des petites choses de calcul formel que je faisais avec Sage ou Maxima, des outils que je ne pratique pas assez pour bien progresser. Julia permet d'avancer à son rythme, de progresser petit à petit dans la connaissance du langage, tout en produisant déjà du code utile, et sans avoir la sensation de recopier des formules magiques.

Avec ce petit opuscule, j'essaie de partager mon expérience avec Julia : l'installation, les bases du langage, et une description de quelques bibliothèques qui me semblent utiles à un usage généraliste. On rencontrera donc dans ce livre des graphiques et des statistiques, des équations différentielles, mais aussi des groupes, de la théorie des nombres, des graphes, du dénombrement, des systèmes aléatoires de particules, ainsi que du calcul symbolique au service d'un problème théorique de développement asymptotique.

Bien sûr, mon livre n'épuise pas le sujet. Le web regorge de bien plus de ressources que cet ouvrage ne peut contenir. Cependant, ces ressources s'adressent bien souvent à un lecteur averti, déjà grand polyglotte informatique ou familier du calcul scientifique. J'espère ici toucher un public plus large, car je suis convaincu que l'outil informatique a sa place sur le bureau de nombreux mathématiciens, alors que mon expérience d'enseignant et de chercheur me montre que, loin des professions de foi modernistes, il ne touche aujourd'hui que des communautés trop restreintes.

À vous de jouer ! Tous les commentaires seront les bienvenus, vous pouvez me les envoyer par mail à

`Olivier.Garet@univ-lorraine.fr`.

Les éventuels errata seront disponibles à l'adresse

http://www.iecl.univ-lorraine.fr/~Olivier.Garet/livre_julia/.

Vous y trouverez aussi le code de la plupart des programmes.

Olivier Garet, le 31 décembre 2019.

Remerciements. Je tiens à remercier les collègues et amis qui ont bien voulu me donner leur avis sur une version préliminaire de ce texte : Bruno et Bruno, Djalil, Vladimir, Remarque, avec une mention particulière pour le travail incroyable du contributeur Saturne du site les-mathematiques.net.