

# *Mini-symposium Python*

## *Du calcul symbolique à l'apprentissage, à bord du Python-Express*

*Mini-symposium porté par le GdR Calcul*

### Résumé

L'objectif de ce mini-symposium est de donner un aperçu des fonctionnalités offertes par l'écosystème des logiciels libres autour de Python pour les mathématiques pures et appliquées : modélisation, calcul interactif et haute performance, visualisation, traitement de données, apprentissage, le tout accessible depuis un environnement interactif unifié.

Ce mini-symposium sera suivi dans la soirée par un tutoriel informel, avec aide à l'installation, etc. Les jours suivants les organisateurs seront à la disposition des participants pour une aide plus personnalisée.

### Organisateur(s)

1. **Paul Zimmermann**, Inria Nancy - Grand Est and LORIA.
2. **Nicolas M. Thiéry**, Laboratoire de Recherche en Informatique, Université Paris Sud.

### Liste des orateurs

1. **Maximilian Albert**, Computational Modelling Group, University of Southampton  
*Titre* : De la modélisation à la visualisation avec Jupyter/IPython.
2. **Nicolas M. Thiéry**, Laboratoire de Recherche en Informatique, Université Paris Sud  
*Titre* : Algèbre et mathématique discrètes avec Sage.
3. **Loïc Gouarin**, Laboratoire de Mathématique d'Orsay, Université Paris Sud  
*Titre* : Quelques idées d'optimisation de code numérique écrit en Python.
4. **Fabian Pedregosa**, CEREMADE/Université Dauphine  
*Titre* : Apprentissage statistique avec scikit-learn.

**Paul Zimmermann**, [www.loria.fr/~zimmerma/](http://www.loria.fr/~zimmerma/), [Paul.Zimmermann@loria.fr](mailto:Paul.Zimmermann@loria.fr)

**Maximilian Albert**, <http://cmg.soton.ac.uk/people/mha2e09/>, [maximilian.albert@gmail.com](mailto:maximilian.albert@gmail.com)

**Nicolas M. Thiéry**, [Nicolas.Thiery.name](mailto:Nicolas.Thiery.name), [Nicolas.Thiery@u-psud.fr](mailto:Nicolas.Thiery@u-psud.fr)

**Loïc Gouarin**, <http://www.math.u-psud.fr/~gouarin/>, [Loic.Gouarin@math.u-psud.fr](mailto:Loic.Gouarin@math.u-psud.fr)

**Fabian Pedregosa**, <http://fa.bianp.net>, [f\(at\)bianp.net](mailto:f(at)bianp.net)

## De la modélisation à la visualisation avec Jupyter/IPython

We introduce the Jupyter environment (formerly known as IPython notebook, but made independent of the Python programming language) and outline its capabilities to integrate live code, descriptive text, rendered equations and media such as images/plots into the same document. There is a wide range of applications of Jupyter in mathematics, science and engineering – including explorative computation with support for many different programming languages, reproducible computation, and convenient documentation and communication of results.

In this presentation we give a hands-on demonstration of how the Jupyter environment can help with all aspects of the scientific workflow – from interactive, exploratory analysis to a finished document explaining the results that can be shared with (and verified by!) other scientists.

## Algèbre et mathématiques discrètes avec Sage

Dans cet exposé, nous présenterons un tour rapide du système de calcul mathématique généraliste libre Sage ([sagemath.org](http://sagemath.org)). Basé sur le langage Python et sur une multitude de bibliothèques spécialisées, Sage est développé depuis une dizaine d'années par une communauté internationale de plusieurs centaines d'enseignants et de chercheurs. Ses fonctionnalités couvrent entre autres le calcul symbolique, la théorie des nombres, la théorie des groupes, l'algèbre linéaire exacte, les mathématiques discrètes, tout en s'intégrant dans l'écosystème Python pour les besoins en calcul scientifique.

## Quelques idées d'optimisation de code numérique écrit en Python

Le langage Python est maintenant largement utilisé au sein de la communauté du calcul numérique en s'appuyant fortement sur le module NumPy qui permet de gérer les tableaux. Bien que ce module soit optimisé pour les opérations élémentaires (addition de deux vecteurs, multiplication de matrices, ...), il est souvent nécessaire d'améliorer certaines parties du code avec d'autres outils afin d'obtenir des performances proches d'un langage bas niveau (C ou Fortran). Une large palette est offerte aux développeurs pour optimiser leur code : Cython, Numba, PyPy, Pythran, ... La plupart de ces outils utilise de la compilation "just in time" permettant de générer du code optimisé à l'exécution du programme. Leur syntaxe est très simple et ils peuvent être utilisés sans connaissances particulières. Lors de cette présentation, nous aborderons quelques uns de ces outils ainsi que leurs utilisations sur un cas concret : l'implantation d'une méthode lattice Boltzmann appelée  $D_2Q_9$  pour résoudre les équations de Saint Venant.

## Apprentissage statistique avec scikit-learn

Dans cet exposé, nous présenterons scikit-learn ([scikit-learn.org](http://scikit-learn.org)), une bibliothèque d'apprentissage statistique (machine learning) en Python. Cette bibliothèque fournit une API cohérente pour une vaste collection d'algorithmes : de l'apprentissage non supervisé au supervisé, en passant par la sélection de variables ou la sélection de modèles. Grâce à la facilité d'utilisation du langage Python et à sa grande collection d'exemples, la bibliothèque scikit-learn rend accessible de puissants algorithmes d'apprentissage à des utilisateurs novices et confirmés. La maturité de son code et sa performance rend cette bibliothèque attractive aussi dans des environnements industriels ; ceci est démontré par exemple par plusieurs témoignages et le grand nombre de compétitions kaggle ([www.kaggle.com](http://www.kaggle.com)) qui ont été remportées en utilisant scikit-learn. Dans cette présentation, nous présenterons la philosophie et la stratégie du projet, ainsi que l'API et de brefs exemples de code.