

Ponio : vous reprendrez bien un peu d'intégrateur en temps ?

Josselin MASSOT, CMAP - Palaiseau

Il existe de nombreuses stratégies d'intégration en temps pour résoudre une EDP par la méthode des lignes : les méthodes de type Runge-Kutta (explicites, diagonal-implicites, implicites), les méthodes de Lawson induites par des méthodes Runge-Kutta, les méthodes exponentielles Runge-Kutta, les méthodes IMEX (implicites-explicites, aussi appelées *additive Runge-Kutta*), les méthodes de Runge-Kutta Chebyshev (aussi appelées méthodes à stabilité étendue), et ce, sans oublier les méthodes de *splitting* (de Lie ou de Strang). Un catalogue assez restreint peut être retrouvé dans un écosystème C++, à l'aide, par exemple, des logiciels odeint [4] ou GSL [5], mais rien de l'envergure de DifferentialEquation.jl [6] en Julia. En C++, la solution est souvent un appel à des routines Fortran¹. C'est en partant de ce constat et de la volonté d'avoir des méthodes d'intégration de nouvelle génération (type ROCK4 [3], PIROCK [1], ESERK [2], etc.) écrites en C++ qu'est né le développement du logiciel ponio.

Dans cet exposé, nous présenterons différentes méthodes pour résoudre différents modèles, dont un problème raide classique : l'équation de Curtiss-Hirschfelder, et illustrerons ceci par des résultats numériques issus du logiciel ponio. L'ensemble de ces travaux est disponible via le logiciel *open source* ponio qui se trouve à cette adresse : <https://github.com/hpc-maths/ponio>.

- [1] Pirock : *A swiss-knife partitioned implicit-explicit orthogonal runge-kutta chebyshev integrator for stiff diffusion-advection-reaction problems with or without noise*. Journal of Computational Physics, **242**, 869–888, 2013. doi :<https://doi.org/10.1016/j.jcp.2013.02.009>.
- [2] Eserk5 : *A fifth-order extrapolated stabilized explicit runge-kutta method*. Journal of Computational and Applied Mathematics, **356**, 22–36, 2019. doi :<https://doi.org/10.1016/j.cam.2019.01.040>.
- [3] A. Abdulle. *Fourth order chebyshev methods with recurrence relation*. SIAM Journal on Scientific Computing, **23(6)**, 2041–2054, 2002. doi :10.1137/S1064827500379549.
- [4] K. Ahnert, M. Mulansky. *Odeint – Solving Ordinary Differential Equations in C++*. AIP Conference Proceedings, **1389(1)**, 1586–1589, 2011. doi :10.1063/1.3637934.
- [5] B. Gough. *GNU scientific library reference manual*. Network Theory Ltd., 2009.
- [6] C. Rackauckas, Q. Nie. *DifferentialEquations.jl—a performant and feature-rich ecosystem for solving differential equations in Julia*. Journal of Open Research Software, **5(1)**, 2017.

Contact : josselin.massot@polytechnique.edu

¹. Comme celles proposées sur la page de Ernst Hairer de l'Université de Genève <http://www.unige.ch/~hairer/software.html>