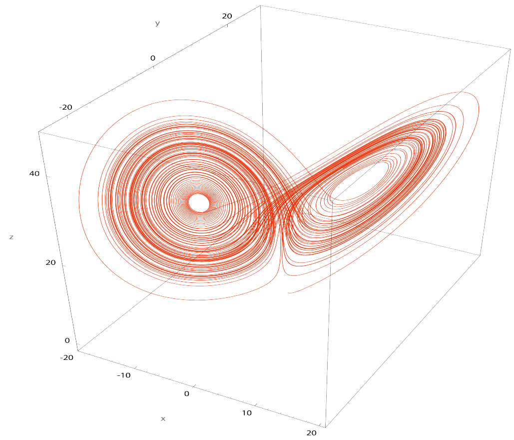
Validated Numerics and Formal Proof for Differential Equations

- **where ?** CRIStAL, Université de Lille — in the research group CFHP (Computer Algebra and HPC).
- **who ?** *student :* Master 2 in CS or mathematics.
  advisors : • Florent Bréhard (CRIStAL, Univ. Lille),
  • Damien Pous (LIP, ENS de Lyon),
  • Nicolas Brisebarre (LIP, ENS de Lyon).
- **when ?** spring 2024 (4 – 6 months)



Differential equations in scientific computing: a new challenge for the Coq proof assistant

Differential equations are ubiquitous in physics, chemistry or biology since they model systems where the evolution is determined by the current state. Solving them in various flavours is of prior importance for scientific computing: numerical evaluation of the solution, characterization of the trajectories, parameter estimation or design of control laws. Except for very specific cases, differential equations admit no closed-form solutions, and numerical methods executed on computers with finite memory and running time necessarily introduce errors due to the discretization of continuous variables and the use of floating-point arithmetic to approximate real numbers, among other things. This accumulation of numerical errors, together with possible implementation mistakes when coding such methods, is a major obstacle for applications with high reliability requirements, such as safety-critical engineering applications (e.g., control of a spacecraft on orbit) or computer-assisted proofs in mathematics (e.g., in dynamical systems theory [8]).

In order to overcome such issues and propose highly reliable software to mathematicians and engineers, we advocate the use of validated numerics [6, 7] and formal proof via the Coq proof assistant. The main objective of this internship is **the development and formalization in Coq of efficient and validated numerical algorithms to rigorously approximate solutions of differential equations**.

Methodology of the internship

We here consider *ordinary differential equations* (ODEs), which are differential equations in one independent real variable t of the form $y^{(r)}(t) = F(t, y(t), y'(t), \dots, y^{(r-1)}(t))$. Here are some famous examples:

$$y'' = ty \qquad (1) \qquad\qquad y'' = 6y^2 + t \qquad (2) \qquad\qquad \begin{cases} x' = 10(y-x) \\ y' = x(28-z) - y \\ z' = xy - 8/3\, z \end{cases} \qquad (3)$$

The first one (1) is the Airy equation widely used in physics: it is *linear*, *second-order* and *time-dependant*. The second one is called Painlevé I and is additionally *nonlinear*. Finally, the last one is a *autonomous (i.e., time-independent) nonlinear system* of dimension 3 and order 1, discovered by the meteorologist Lorenz.

Given sufficiently many initial conditions, computer algebra methods [4, 1, 9] allow for computing efficiently the coefficients of the Taylor expansion $y(t) = \sum_{n \geqslant 0} a_n t^n$ of the solution. The key idea is to cleverly define a Newton operator such that each iteration doubles the number of correctly computed coefficients a_n. Such a strategy is however purely symbolic: it requires the coefficients to be exactly representable numbers (e.g., rational or algebraic numbers) and it does not provide error bounds when truncated the series to finite degree.

In this internship, we aim at extending this approach to a numerical setting while preserving rigorous mathematical statements by also computing error bounds. A possible roadmap is the following:

(1) Translate the method of [1] in a numerical setting: instead of exact power series, we consider approximations of the form $\widetilde{y}(t) = \sum_{n=0}^{N} a_n f_n(t)$ in a well-chosen Banach space of functions, like Fourier ($f_n(t) =$

e^{int}) or Chebyshev ($f_n(t) = T_n(t)$) approximations. Then the Newton method is expected to "square" the error at each iteration, even if the initial conditions are given with some errors or not directly known (e.g., boundary conditions rather than initial ones). Furthermore, more "exotic" bases can be investigated to approximate solutions with *singularities*.

(2) Implement an a posteriori validation algorithm in Coq, which takes an approximation $\widetilde{y}$ computed in the first step, and returns a rigorous error bound with respect to the exact solution y. Such error bound reconstruction can be obtained by a suitable application of a *fixed-point theorem* on a similar Newton operator (see [2]). All the necessary tools (interval arithmetic, Chebyshev approximations, fixed-point theorem, etc) will be provided by the Coq librairies Interval[1] [5] and ApproxModels[2] [3].

(3) Test the resulting prototype implementation on some examples (e.g., Equations (1), (2) and (3)), and compare its accuracy, efficiency and reliability to other software, either purely numerical, or relying on validated numerics, or already resorting to formal proof.

Student's profile

This internship targets Master 2 students in computer science and/or mathematics. It requires typical undergraduate knowledge in mathematics – especially in analysis for differential equations – together with a minimum experience with the Coq proof assistant. Even though not mandatory, additional competences in computer algebra or approximation theory will be considered.

Références

[1] Alin Bostan, Frédéric Chyzak, François Ollivier, Bruno Salvy, Éric Schost, and Alexandre Sedoglavic. Fast computation of power series solutions of systems of differential equations. In *SODA'07*, pages 1012–1021. Society for Industrial and Applied Mathematics, January 2007.

[2] Florent Bréhard. A symbolic-numeric validation algorithm for linear odes with newton–picard method. *Mathematics in Computer Science*, 15(3):373–405, 2021.

[3] Florent Bréhard, Assia Mahboubi, and Damien Pous. A certificate-based approach to formally verified approximations. In *ITP 2019-Tenth International Conference on Interactive Theorem Proving*, pages 1–19, 2019.

[4] Richard P Brent and Hsiang T Kung. Fast algorithms for manipulating formal power series. *Journal of the ACM (JACM)*, 25(4):581–595, 1978.

[5] Érik Martin-Dorel and Guillaume Melquiond. Proving tight bounds on univariate expressions with elementary functions in coq. *Journal of Automated Reasoning*, 57(3):187–217, Oct 2016.

[6] Ramon E. Moore, R. Baker Kearfott, and Michael J. Cloud. *Introduction to interval analysis*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2009.

[7] Warwick Tucker. *Validated numerics: A short introduction to rigorous computations*. Princeton University Press, Princeton, NJ, 2011.

[8] Jan Bouwe van den Berg and Jean-Philippe Lessard. Rigorous numerics in dynamics. *Notices Amer. Math. Soc.*, 62(9):1057–1061, 2015.

[9] Joris van der Hoeven. Newton's method and fft trading. *Journal of Symbolic Computation*, 45(8):857–878, 2010.

[1] `https://coqinterval.gitlabpages.inria.fr/`
[2] `https://gitlab.inria.fr/amahboub/approx-models`