The Journal of Open Source Software

HOODESolver.jl: A Julia package for highly oscillatory problems

Yves Mocquard[*1], **Pierre Navaro**[1, 2], **and Nicolas Crouseilles**[1]

1 Inria (MINGuS team) Rennes - Bretagne Atlantique, and IRMAR CNRS, UMR 6625, France. **2** University of Rennes 1, IRMAR CNRS, UMR 6625, France

DOI: DOIunavailable

Software

- Review
- Repository
- Archive

Editor: Pending Editor

Reviewers:

- @Pending Reviewers

Submitted: N/A
Published: N/A

Summary

Highly oscillatory ordinary differential equations (ODEs) has a long history since they are ubiquitous to describe dynamical multi-scale physical phenomena in physics or chemistry. They can be obtained by appropriate spatial discretization of a partial differential equations or can directly describe the behavior of dynamical quantities. In addition to the standard difficulties coming their numerical resolution, highly oscillatory ODEs involve a stiffness (characterized by a parameter $\varepsilon \in]0,1]$) creating high oscillations in the solution. Hence, to capture these small scales (or high oscillations), conventional methods have to consider a time step smaller than ε leading to unacceptable computational cost.

We present here `HOODESolver.jl`[1], a general-purpose library written in Julia dedicated to the efficient resolution of highly oscillatory ODEs. In the documentation[2] you will see how to simulate highly oscillatory ODEs using a Uniformly Accurate (UA) method *ie* the method able to capture the solution while keeping the time step (and then the computational cost) independent of the degree of stiffness ε.

Statement of need

There are many software packages to efficiently solve ODEs (one can quote DifferentialEquations in Julia (Rackauckas & Nie, 2017) but also many others in Python (completer/citer ?)). In spite of the fact that these packages include recent and advanced numerical techniques, the highly oscillatory character of the solution makes the use of these packages very limited. Indeed, when ε is small, the solution presents oscillations whose period is proportional to ε. As a consequence, conventional methods struggle to solve such multi-scale phenomena since they require to use tiny time steps to capture high oscillations and become computationally very costly. On the one side, specific methods inspired by the averaging theory have been designed to deal with the regime $\varepsilon \ll 1$. On the other side, when $\varepsilon \sim 1$ the problem ceases to be stiff and a classical integrator gives accurate result in a reasonable time. The true difficulty emerges for intermediate values of ε, for which averaging techniques are not accurate enough and, due to computational cost, standard methods are inefficient. Thus, a new paradigm has been recently introduced, the so-called uniform accuracy: uniformly accurate (UA) methods are indeed able to solve the original highly oscillatory problem with a precision and a computational cost that are independent of the value ε. In particular, these methods allows to skip several oscillations in a single time step, reducing the number of iterations (and then the cost of the simulation) drastically. (met-on des figures pour illustrer ?)

`HOODESolver.jl` intends to gather and unify recent research around highly oscillatory problems (Bao & Zhao, 2019; Chartier et al., 2015; Chartier, Crouseilles, et al., 2020; Crouseilles et al., 2017) its development has been motivated by these research needs and it has already

*Corresponding author.
[1] https://github.com/ymocquar/HOODESolver.jl
[2] https://ymocquar.github.io/HOODESolver.jl/stable/

been used in some papers (Chartier, Lemou, et al., 2020) `HOODESolver.jl` is written entirely in Julia and provides software implementations of several theoretical ideas contained in the recent literature around the so-called *two-scale* method. In particular, a very recent extension proposed in (Chartier, Lemou, et al., 2020) enables to reach high order accuracy. The implementation focuses on a multistep method (namely Adams-Bashforth method) coupled a spectral method for the discretization of the additional variable representing the fast scale. Hence, `HOODESolver.jl` provides an efficient way for researchers to solve a highly oscillatory ODE system, and as such it can be used by the scientific community

- researchers interested in solving highly oscillatory problems arising in their research field (electromagnetic waves, quantum mechanics, plasma physics, molecular dynamics, ...),
- it can guide some future possible numerical or theoretical developments,
- it will serve as a reference to benchmark a new method designed by researchers.

Features

`HOODESolver.jl` is designed to solve the following highly oscillatory ordinary differential system

$$\dot{u}(t) = \frac{1}{\varepsilon} A u(t) + f(t, u), \qquad t \in [t_{start}, t_{end}], \qquad u(t = t_{start}) = u_{in}, \tag{1}$$

where

- $u \colon [t_{start}, t_{end}] \mapsto \mathbb{R}^n, \qquad t_{start}, t_{end} \in \mathbb{R},\ t \mapsto u(t).$
- $u_{in} \in \mathbb{R}^n$,
- $A \in \mathcal{M}_{n,n}(\mathbb{R})$ such that $\tau \mapsto \exp(\tau A)$ is periodic,
- $f \colon \mathbb{R} \times \mathbb{R}^n \mapsto \mathbb{R}^n,\ (t, u) \mapsto f(t, u).$

The numerical solution of Equation 1 is computed by simply entering the different components of the equation (A, f, ε, t_{start}, t_{end}, u_{in}) following the required format. The user simply chooses an order of the Adams-Bashforth time integrator and the time step. The result is given as a function object which can be evaluated in an arbitrary time t, not just at the discrete times. In particular, the Julia command $plot(t \mapsto u(t)[1])$ enables to plot the solution. In addition to the methodology introduced in `HOODESolver.jl` includes

- the use of BigFloat types (optional),
- a new technique to compute the first iterations required for the initialization of the Adams-Bashforth method,
- the extension of the two-scale method to non-homogeneous problems.

Related research and software

The development of the HOODESolver package was initially motivated by the need of efficient multiscale solvers for the charged particles dynamics in an external strong magnetic field. Indeed, due to the Lorentz force, charged particles undergo rapid circular motion around the magnetic field lines. This constitutes the basis of the magnetic confinement of a plasma in a chamber. Obviously, computing a highly oscillatory dynamics is a long-standing problem occurring in many relevant applications (Engquist et al., 2009; Hairer et al., 2006) However, we are not aware of other software packages with similar purpose, excepting the very recent (py)oscode package (Agocs, 2020) which combines WKB techniques and standard integration methods to ensure a user-specified tolerance.

HOODESolver.jl has been thought to be in close connection to the DifferentialEquation.jl Julia package (Rackauckas & Nie, 2017). Whereas DifferentialEquation.jl is a very popular and powerful Julia package, it does not provide an algorithm to overcome the stiffness due to the presence of ε and thus needs to use very small time steps leading to lengthy computations or unstable results. In particular, the HOODESolver.jl is intended to offer a common interface with the package DifferentiaEquations.jl by extending the Split ODE problem type. Users could compare more easily our method with methods implemented in (Rackauckas & Nie, 2017).

The following research projects are connected to HOODESolver in the sense that most of which have led to its development (Chartier, Lemou, et al., 2020; Chartier et al., 2015; Chartier, Crouseilles, et al., 2020; Crouseilles et al., 2013, 2017)

- introduction of two-scale method to design Uniformly Accurate methods
- numerical analysis of the two-scale method
- coupling of the two-scale method with Particle-In-Cell approach
- extension to high order

Aknowledgements

Much of the HOODESolver developement was performed by Y. Mocquard while he was supported by Inria through the AdT (Aide au developpement technologique) J-Plaff of the center Rennes-Bretagne Atlantique.

References

Agocs, F. J. (2020). Fast solutions of oscillatory ODEs. *Journal of Open Source Software*, *5(56)*, 2830–2833. https://doi.org/https://doi.org/10.21105/joss.02830

Bao, W., & Zhao, X. (2019). Comparison of numerical methods for the nonlinear Klein-Gordon equation in the non relativistic limit regime. *Journal of Computational Physics*, *398*. https://doi.org/https://doi.org/10.1016/j.jcp.2019.108886

Chartier, P., Crouseilles, N., Lemou, M., & Mehats, F. (2015). Uniformly accurate numerical schemes for highly oscillatory klein-gordon and nonlinear schrödinger equations. *Numerische Mathematik*, *129*.

Chartier, P., Crouseilles, N., Lemou, M., Mehats, F., & Zhao, X. (2020). Uniformly accurate methods for three dimensional vlasov equations under strong magnetic field with varying direction. *SIAM Journal of Scientific Computing*, *42*. https://doi.org/https://doi.org/10.1137/19M127402X

Chartier, P., Lemou, M., Mehats, F., & Zhao, X. (2020). Derivative-free high order uniformly accurate schemes for highly-oscillatory systems. *ArXiv e-Prints*, 12–18.

Crouseilles, N., Lemou, M., & Mehats, F. (2013). Asymptotic preserving schemes for highly oscillatory kinetic equations. *Journal of Computational Physics*, *248*. https://doi.org/https://doi.org/10.1016/j.jcp.2013.04.022

Crouseilles, N., Lemou, M., Mehats, F., & Zhao, X. (2017). Uniformly accurate Particle-In-Cell method for the long time solution of the two-dimensional Vlasov-Poisson equation with uniform strong magnetic field. *Journal of Computational Physics*, *346*. https://doi.org/https://doi.org/10.1016/j.jcp.2017.06.011

Engquist, B., Fokas, A., Hairer, E., & Iserles, A. (2009). *Highly oscillatory problems*. Cambridge University Press.

Hairer, E., Lubich, C., & Wanner, G. (2006). *Geometric numerical integration*. Springer.

Rackauckas, C., & Nie, Q. (2017). Differentialequations.jl–a performant and feature-rich ecosystem for solving differential equations in julia. *Journal of Open Research Software*, *5*(1).