The Journal of Open Source Software

Flint: a simulator for biological and physiological models in ordinary and stochastic differential equations

Takeshi Abe[1] **and Yoshiyuki Asai**[1]

1 Graduate School of Medicine, Yamaguchi University

DOI: 10.21105/joss.02331

Software

- Review
- Repository
- Archive

Editor: Mark A. Jensen

Reviewers:

- @mstimberg
- @dawbarton

Submitted: 22 May 2020
Published: 07 September 2020

License
Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License (CC BY 4.0).

Introduction

Understanding the dynamics of living organisms often requires a mathematical model that describes the hypotheses to be tested. It is widely recognized that the class of ordinary differential equations (ODE) is suitable for describing the time course of variables in a deterministic system, stemming from a simple assumption about the rate of their change. One such example is the chemical reaction accelerated by an enzyme following Michaelis-Menten kinetics; another is the action potential of cardiac cells driven by modulation of ion channels. By virtue of differential equations, these celullar models can be integrated into models at the tissue or organ level. In fact, ways to integrate a computational model of the physiological functions of the whole individual have been explored since the end of the last century, under the name physiome (Leem, 2016).

It is, however, technically challenging for practitioners in the field of biology or physiology to express their hypotheses on biological organisms in a precise system of ODEs. In order to make it easier to edit a model in a problem that implicitly specifies the ODEs, several domain-specific languages have been proposed and standardized, including CellML (Lloyd, Halstead, & Nielsen, 2004), the Physiological Hierarchy Markup Language (PHML) devised by Asai and colleagues (Asai et al., 2015), and the Systems Biology Markup Language (SBML) devised by Hucka and colleagues (Hucka et al., 2003). Although the design principles of each modeling language vary, computational analysis of any model in these languages comprises a shared set of procedures based on the theory of differential equations and dynamical systems.

In this work we introduce `Flint`, a simulator software for models written in the above languages. The simulator allows users to transform a given model into a system of ODEs and solve it in a numerical manner. It also supports stochastic differential equations (SDE), a non-deterministic extension of ODEs, which makes it possible to involve random elements, e.g. noise, in the dynamics.

The development of `Flint` has been tied in with the physiome.jp project (Nomura, 2010), which aims to establish a computational platform for multiscale *in silico* studies on the physiome. As part of the platform, `Flint` complements the features of an authoring software PhysioDesigner for PHML (Asai et al., 2012), while they are deliberately separate programs. Driven by demands from the project's collaborators, we have enhanced `Flint` to support different modeling standards. For example, in order to leverage a published SBML model of subcellular signaling to build tissue or higher-level physiological ones, there is a technical proposal embedding it in PHML (Asai et al., 2014). Simulating such models is a reason for adopting `Flint` even when other state-of-the-art tools are publicly available, e.g. COPASI (Hoops et al., 2006) that focuses on its own format. `Flint`'s main contribution is to provide an open, language-agnostic resource for reproducible simulation studies.

Implementation

User interface

`Flint` is a standalone program that runs on consumer desktop environments such as Microsoft Windows, Apple's macOS, and Linux with GTK. For the simplest usage, its graphical user interface runs a simulation of a given model with only two steps; open the model file, and select the Run button. Running simulations at the command line is also supported, although only a limited number of functions are available in the command line interface. The simulator delegates the task of displaying the output to gnuplot (Williams, Kelley, Merrit, & al., 2017).

Numerical algorithms to solve a system of differential equations

`Flint` compiles a model written in a supported XML language into internal bytecode for simulation, and then evaluates it with particular initial values. Our current implementation provides three algorithms for solving initial-value problems for ODEs numerically: the forward Euler method, the Runge-Kutta 4th-order method, and the adaptive-step additive Runge-Kutta scheme implemented in the SUNDIALS library (Hindmarsh et al., 2005). The Euler-Maruyama method is used for solving SDEs (Higham, 2001).

Multithreading for parallel simulation

Solving an initial-value problem numerically is only the preliminary step towards a full analysis of the dynamics of the model. Further investigation often asks for different values of initial values or parameters. For instance, hypotheses on biological switches have been stated in terms of dynamical bifurcations, and demonstrated by a series of simulations over changing values of parameters, in both deterministic (Fussmann, Ellner, Shertzer, & Hairston, 2000) and stochastic (Samoilov, Plyasunov, & Arkin, 2005) paradigms. `Flint` employs multithreading to increase the number of simulations running in parallel. The parallelization is automatically performed when the user assigns multiple values to some parameter of a model for simulation, and honors the number of available CPU cores, which can be adjusted in the preferences.

Grid search algorithm for parameter fitting

The larger the number of variables and parameters in a given model are, the more resource-consuming its simulation becomes. This is also the case when estimating plausible values of parameters consistent with the prior knowledge on the behavior of the underlying system. Taking residual sum of squares (RSS) as a measure of the goodness of fit, estimation of parameter values for ODEs turns into a non-linear least-squares problem (Madsen, Nielsen, & Tingleff, 2004). `Flint` deals with the challenge the modeler faces when fitting the value of parameters via the least-squares method, taking advantage of multithreading if available. Given grid points in the parameter space as an input set, `Flint` performs the following branch-and-bound algorithm to reduce the number of simultaneously running jobs for the grid search:

```
Input  : A set S of parameter values; timeseries data D of target
         state variables
Output: An element of S with which simulated timeseries has the
         least RSS against D in S
m ← ∞;
foreach e ∈ S do // in parallel
    s ← 0;
    start simulation with parameters e from t = 0;
    foreach t found in D do // in the ascending order of time
        simulate until t;
        increment s by the squared error at t;
        if s ≥ m then break;
    end
    if s = 0 then
        return e
    else if s < m then
        x ← e;
        m ← s;
    end
end
return x
```

Figure 1: An algorithm for grid search to fit parameter values.

Unlike existing heuristics for solving non-linear least-squares, the above algorithm can find a global minimum, provided that the input grid contains it. It is also easy to benefit from parallel computing to reduce processing time. The only shared resource among parallel processes is m in Fig. 1, namely a double floating-point number with its mutex, which means the overhead is marginal. Users can define the range of each parameter as well as the way to enumerate grid points, e.g. by a pseudorandom number generator. This feature will help researchers gain insight about a subset of parameter values of biological/physiological interest at an early stage of modeling.

Acknowledgements

We acknowledge Dr Masao Okita for his invaluable comments on shared-memory parallelism implemented in `Flint`.

References

Asai, Y., Abe, T., Li, L., Oka, H., Nomura, T., & Kitano, H. (2015). Databases for multilevel biophysiology research available at Physiome.Jp. *Frontiers in Physiology*, *6*(SEP). doi:10.3389/fphys.2015.00251

Asai, Y., Abe, T., Oka, H., Okita, M., Hagihara, K.-i., Ghosh, S., Matsuoka, Y., et al. (2014). A Versatile Platform for Multilevel Modeling of Physiological Systems: SBML-

PHML Hybrid Modeling and Simulation. *Advanced Biomedical Engineering*, *3*, 50–58. doi:10.14326/abe.3.50

Asai, Y., Abe, T., Okita, M., Okuyama, T., Yoshioka, N., Yokoyama, S., Nagaku, M., et al. (2012). Multilevel Modeling of Physiological Systems and Simulation Platform: PhysioDesigner, Flint and Flint K3 Service. In *2012 IEEE/IPSJ 12th International Symposium on Applications and the Internet* (pp. 215–219). doi:10.1109/SAINT.2012.40

Fussmann, G. F., Ellner, S. P., Shertzer, K. W., & Hairston, N. G. (2000). Crossing the Hopf Bifurcation in a Live Predator-Prey System. *Science*, *290*(5495), 1358–1360. doi:10.1126/science.290.5495.1358

Higham, D. J. (2001). An Algorithmic Introduction to Numerical Simulation of Stochastic Differential Equations, *43*(3), 525–546.

Hindmarsh, A. C., Brown, P. N., Grant, K. E., Lee, S. L., Serban, R., Shumaker, D. E., & Woodward, C. S. (2005). SUNDIALS: Suite of nonlinear and differential/algebraic equation solvers. *ACM Transactions on Mathematical Software (TOMS)*, *31*(3), 363–396.

Hoops, S., Sahle, S., Gauges, R., Lee, C., Pahle, J., Simus, N., Singhal, M., et al. (2006). COPASI—a COmplex PAthway SImulator. *Bioinformatics*, *22*(24), 3067–3074. doi:10.1093/bioinformatics/btl485

Hucka, M., Finney, A., Sauro, H. M., Bolouri, H., Doyle, J. C., Kitano, H., Arkin, A. P., et al. (2003). The systems biology markup language (SBML): A medium for representation and exchange of biochemical network models. *Bioinformatics*, *19*(4), 524–531. doi:10.1093/bioinformatics/btg015

Leem, C. H. (2016). Perspectives of physiome research. *Integrative Medicine Research*, *5*(1), 37–40. doi:10.1016/j.imr.2015.12.004

Lloyd, C. M., Halstead, M. D. B., & Nielsen, P. F. (2004). CellML: Its future, present and past. *Progress in Biophysics and Molecular Biology*, Modelling Cellular and Tissue Function, *85*(2), 433–450. doi:10.1016/j.pbiomolbio.2004.01.004

Madsen, K., Nielsen, H. B., & Tingleff, O. (2004). Methods for non-linear least squares problems (2nd ed.). Richard Petersens Plads, Building 321, DK-2800 Kgs. Lyngby: Informatics; Mathematical Modelling, Technical University of Denmark, DTU.

Nomura, T. (2010). Toward Integration of Biological and Physiological Functions at Multiple Levels. *Frontiers in Physiology*, *1*. doi:10.3389/fphys.2010.00164

Samoilov, M., Plyasunov, S., & Arkin, A. P. (2005). Stochastic amplification and signaling in enzymatic futile cycles through noise-induced bistability with oscillations. *Proceedings of the National Academy of Sciences*, *102*(7), 2310–2315. doi:10.1073/pnas.0406841102

Williams, T., Kelley, C., Merrit, E. A., & al. (2017, September). Gnuplot 5.2: An interactive plotting program. http://www.gnuplot.info/.