Compartmental Modeling Software: a fast, discrete stochastic framework for biochemical and epidemiological simulation

Christopher W. Lorton[1⋆], Joshua L. Proctor[1*], Min K. Roh[1*], and Philip A. Welkhoff[2]

[1] Institute for Disease Modeling, Bellevue WA 98005, USA CMS@idmod.org
[2] Bill and Melinda Gates Foundation, Seattle WA 98109, USA

Abstract. The compartmental modeling software (CMS) is an open source computational framework that can simulate discrete, stochastic reaction models which are often utilized to describe complex systems from epidemiology and systems biology. In this article, we report the computational requirements, the novel input model language, the available numerical solvers, and the output file format for CMS. In addition, the CMS code repository also includes a library of example model files, unit and regression tests, and documentation. Two examples, one from systems biology and the other from computational epidemiology, are included that highlight the functionality of CMS. We believe the creation of computational frameworks such as CMS will advance our scientific understanding of complex systems as well as encourage collaborative efforts for code development and knowledge sharing.

Keywords: stochastic simulation · compartmental · open source.

1 Introduction

Developing fast, efficient, and scalable computational frameworks is integral to investigating a broad set of epidemiological and biological systems. Here, we present a new open-source computational framework, called the compartmental modeling software (CMS), which enables the simulation of discrete, stochastic reaction models. The CMS framework is highly flexible: the new model description language enables rapid model development; a broad set of available numerical algorithms allows users to optimize simulations based on model structure or computational speed requirements; and the standardized model output empowers a wide-variety a visualization tools. In this article, we report the functionality of the CMS framework while highlighting the key components of the software.

Open-source frameworks have been previously developed for discrete, stochastic compartmental modeling [20], including advancements that account for event handling [25], access via open-source software such as Python [6], and adaptations to cloud-based platforms [15]. Our new CMS framework broadens the scope and scale of previous open-source software. In addition to the standard exact

[⋆] Co-first author

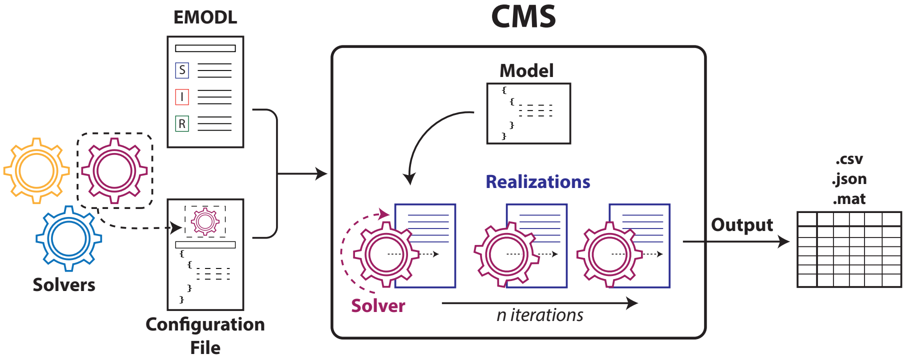

Fig. 1. An overview of the CMS software.

and approximate numerical algorithms, CMS includes more recently developed rare-event probability estimation algorithms [13, 24]. Spatial simulation algorithms [22, 14, 9] are also integrated to accommodate spatial diffusion processes. Reaction propensities, time delays [8, 10], and state- and time- dependent events can be fully customized in the CMS framework. Coupling these features with software documentation, unit and regression tests, and object-oriented development, CMS is a novel and flexible framework to allow for modeling of complex, physical systems.

2 Compartmental Modeling Software

In this section, we briefly explain the major components of CMS. An overview of the CMS schematics is given in Figure 1. For more detail, see the CMS documentation [1] as well as the GitHub repository for the associated code [2].

2.1 Platform and Computational Requirements

CMS was developed in C# and C++ and tested with NUnit framework [4]. It runs on 64-bit Windows 7 or later with any 64-bit Intel CPU. The CMS Visual Studio solution file has been updated to work with Visual Studio 2017 and targets the .NET Framework version 4.6.

2.2 Execution Pathways

Command line invocation is the most common usage for CMS. However, CMS can be executed in any language or scripting tool that can load Microsoft .NET technology. For example, CMS can be integrated into MATLAB or into Python through a package such as pythonnet [5].

2.3 Input Language and Configuration

A custom model description language, named the epidemiological modeling language (EMODL), was created to support the unique features of CMS and accommodate future expansion. EMODL syntax allows for efficient formulation of time- and state-based events as well as delays. It also supports custom propensity formulation in addition to the traditional propensities for mass-action kinetics. Runtime and solver-specific parameters are listed in a JavaScript Object Notation (JSON) [3] formatted configuration file. Parameters such as solver name, ensemble size, length of simulation, random number generator and output format are included in this configuration file.

2.4 Discrete Stochastic Solvers in CMS

CMS offers a comprehensive suite of stochastic solvers. It contains a total of 16 solvers, whose capabilities range from exact sampling of the true underlying probability distribution of the master equation (ME), fast approximation of the ME, spatial (network) simulations, and rare event probability estimation. We briefly describe each category below. In each subsection, we refer the reader to the original publication for mathematical derivations and computational complexity.

Exact Methods The time evolution of systems are fully described by the solution of the ME, which provides the probability of every possible system state configuration at any given time. The ME is analytically intractable for most systems, but exact numerical solvers can estimate these probabilities numerically by sampling from the ME. Gillespie's stochastic simulation algorithm (SSA) [18] is the most well-known exact method for simulating systems in stochastic chemical kinetics. CMS offers two known implementations of SSA—direct method and first reaction method—that are theoretically equivalent to each other. CMS also features Gibson and Bruck's next reaction method [17] as well as the SSA with time delays [8, 10].

Approximate Methods While the exact methods produce accurate time trajectories, explicitly simulating every reaction may be prohibitively slow for some systems. Approximate algorithms have been developed to accelerate the simulation at the expense of accuracy. CMS includes three approximate solvers—two implementations of τ-leaping [12] and one of R-leaping [7].

Tau-leaping is the most popularly used approximate method, and CMS offers two implementations for choosing a time step (τ)—adaptive and fixed. The adaptive time step selection mechanism also includes a check to avoid negative population [11] by reverting to SSA when appropriate. The fixed time step method assumes that the chosen τ is small enough to produce accurate trajectories. When this assumption is not met, negative population counts can occur.

Spatial Simulation Methods Spatial simulation in CMS is possible via three different solvers—inhomogeneous SSA (ISSA) [22], diffusive finite state projection (DFSP) [14], and fractional diffusion (FD) [9]. ISSA divides a system into homogeneous subvolumes, and diffusive transfers are treated as a unimolecular reaction. Therefore, it can be prohibitively slow when fast diffusion is present. The other two methods are more efficient than the ISSA when diffusion occurs frequently with respect to the number of reaction events. DFSP solves the diffusion master equation by adapting the Finite State Projection (FSP) method [23], while FD is based on Lie-Trotter operator splitting of the diffusion and reaction terms. Unlike ISSA and DFSP, fractional diffusion allows for jumping to a distant locale with non-zero probability.

Rare Event Probability Methods In addition to generation of time trajectories, CMS allows for efficient estimation of a rare event probability via the doubly weighted SSA (dwSSA) [13] and the state-dependent dwSSA (sdwSSA) [24]. Both algorithms utilize importance sampling, whose optimal parameters are determined by the information-theoretic concept of cross entropy. While the dwSSA assigns a single importance sampling parameter per reaction, the sdwSSA creates a list of state-dependent importance sampling parameters in order to further reduce the variance in the rare event probability estimate.

Exploratory Methods The CMS framework is designed to enable efficient prototyping of new methods. A new solver can be easily implemented by extending the base solver. Four prototype methods are included in the CMS, and we refer to the documentation page [1] for further details. We note that solvers in this category are included as an example of ongoing method development and are not currently supported by the developers.

2.5 Output Files

There are three output formats available in the CMS—comma-separated values (CSV), JSON [3], and MATLAB (MAT). By default, CMS creates trajectories.csv in the output directory, with the realization index appended to each observable name specified in the EMODL file. Output-related options, such as compression and heading, can be specified in the configuration file.

3 Examples

3.1 Schlögl Process

The Schlögl process is a canonical example of a chemical system exhibiting bistability. This model consists of the following four reactions:

$$B_1 + 2X \underset{k_2}{\overset{k_1}{\rightleftharpoons}} 3X \qquad \text{and} \qquad B_2 \underset{k_4}{\overset{k_3}{\rightleftharpoons}} X.$$

We take the system description including parameters and initial condition from [19] that produces bistable behavior in X: $k_1 = 3 \times 10^{-7}$, $k_2 = 10^{-4}$, $k_3 = 10^{-3}$, $k_4 = 3.5$, and $\mathbf{x}_0 = [10^5,\ 2 \times 10^5,\ 250]$, where $\mathbf{x}_0$ denotes the initial population of B_1, B_2, and X. Using the SSA solver, an ensemble of $N = 1\text{e}5$ simulations were generated to produce the distribution of X shown in Figure 2(a).

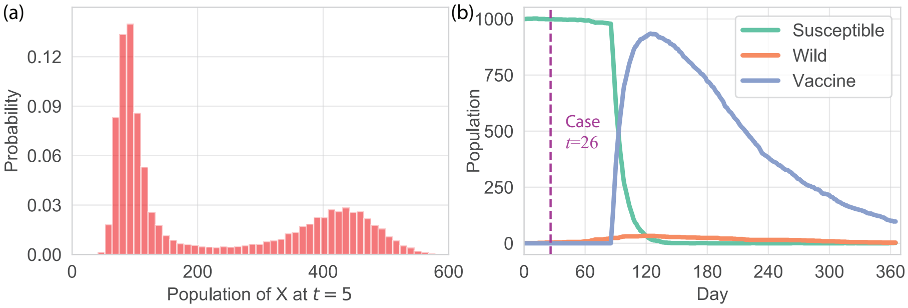

Fig. 2. (a) illustrates the bistable distribution of population of species X at the final time $t = 5$. (b) illustrates one realization of a stochastic Polio outbreak, a paralyzed child (case) detection, and the delayed start of a vaccination campaign.

3.2 Vaccination campaigns for eradicating poliomyelitis

Globally, the number of poliomyelitis cases has dramatically decreased over the past two decades and may be the second human infectious disease to be eradicated [21]. The broad functionality of CMS can be utilized to model current vaccination questions such as programmatic mobilization time to arrest a small polio outbreak. We implement an established discrete, stochastic compartmental model of the spread of Polio [16]. We add two essential components to this model: a probabilistic case detection matching the 1 to 200 case to infection rate for polio and a vaccination campaign that is initiated after a fixed time duration following case detection.

We simulate the model starting with a population of 1000 individuals and one imported infection. When a case is detected, the model enforces a sixty day operational delay before a vaccination campaign begins. Figure 2(b) illustrates one realization of this model; note that a single case is detected on day 26 and vaccination begins on day 86. Due to the vaccination in this scenario, we do not detect another paralyzed child. To generate this realization, we utilize the SSA numerical algorithm, allow for a simulation duration of 365 days, and sample the state of the system each day. The model file, config file, and output file for this example can be found in the examples folder at [2].

4 Conclusion

Simulating biological and epidemiological processes with low population counts, such as nearing elimination of an infectious disease, requires the ability to simulate discrete, stochastic reaction models. The compartmental modeling software (CMS) is a novel, extensible framework used to generate an ensemble of trajectories that approximate the true underlying probability distribution described by the model and initial condition. CMS was designed to enable rapid model development with a custom model language, allow for user flexibility in choosing model-specific numerical solvers, and output the model trajectories into easy to visualize formats. Moreover, CMS is an open-source project allowing for community development; the object-oriented programming and class structure of the code allows for intuitive modifications of the code-base such as the inclusion of new numerical solvers or random number generators.

A number of challenges face the widespread adoption of our framework as a modeling tool. The source code has been written in C# which is not widely utilized in university settings. Also, CMS does not currently make use of multi-threading, multi-core CPUs, or GPU resources. Despite these limitations, CMS can be deployed across multiple virtual machines since the memory and disk requirements are minimal. We also plan on providing a docker image for development and a reproducible environment for execution as well as developing an API from Python to enable users to call into the CMS executable. More broadly, we believe computational tools such as CMS will help provide insights into realistic biological and epidemiological systems.

Acknowledgements

JLP, MKR, CWL, and PW would like to thank Bill and Melinda Gates for their active support of the Institute for Disease Modeling and their sponsorship through the Global Good Fund. The authors would also like to thank Mandy Izzo for her assistance illustrating Figure 1.

References

1. CMS Documentation. `http://idmod.org/docs/cms/`
2. CMS Repository. `https://github.com/InstituteforDiseaseModeling/IDM-CMS`
3. JSON organization. `http://www.json.org`
4. NUnit 3.6.1. `https://github.com/nunit/nunit/releases/3.6.1`
5. Python for .NET. `http://pythonnet.github.io/`
6. Abel, J.H., Drawert, B., Hellander, A., Petzold, L.R.: GillesPy: a Python package for stochastic model building and simulation. IEEE life sciences letters **2**(3), 35–38 (2016)
7. Auger, A., Chatelain, P., Koumoutsakos, P.: R-leaping: Accelerating the stochastic simulation algorithm by reaction leaps. The Journal of Chemical Physics **125**(8), 084103 (aug 2006). https://doi.org/10.1063/1.2218339
8. Barrio, M., Burrage, K., Leier, A., Tian, T.: Oscillatory regulation of hes1: Discrete stochastic delay modelling and simulation. PLoS Computational Biology **2**(9), e117 (2006). https://doi.org/10.1371/journal.pcbi.0020117
9. Bayati, B.S.: Fractional diffusion-reaction stochastic simulations. The Journal of Chemical Physics **138**(10), 104117 (mar 2013). https://doi.org/10.1063/1.4794696
10. Cai, X.: Exact stochastic simulation of coupled chemical reactions with delays. The Journal of Chemical Physics **126**(12), 124108 (mar 2007). https://doi.org/10.1063/1.2710253
11. Cao, Y., Gillespie, D.T., Petzold, L.R.: Avoiding negative populations in explicit poisson tau-leaping. The Journal of Chemical Physics **123**(5), 054104 (aug 2005). https://doi.org/10.1063/1.1992473

12. Cao, Y., Gillespie, D.T., Petzold, L.R.: Efficient step size selection for the tau-leaping simulation method. The Journal of Chemical Physics **124**(4), 044109 (jan 2006). https://doi.org/10.1063/1.2159468
13. Daigle, B.J., Roh, M.K., Gillespie, D.T., Petzold, L.R.: Automated estimation of rare event probabilities in biochemical systems. The Journal of Chemical Physics **134**(4), 044110 (jan 2011). https://doi.org/10.1063/1.3522769
14. Drawert, B., Lawson, M.J., Petzold, L., Khammash, M.: The diffusive finite state projection algorithm for efficient simulation of the stochastic reaction-diffusion master equation. The Journal of Chemical Physics **132**(7), 074101 (feb 2010). https://doi.org/10.1063/1.3310809
15. Drawert, B., Trogdon, M., Toor, S., Petzold, L., Hellander, A.: Molns: A cloud platform for interactive, reproducible, and scalable spatial stochastic computational experiments in systems biology using pyurdme. SIAM Journal on Scientific Computing **38**(3), C179–C202 (2016)
16. Eichner, M., Dietz, K.: Eradication of poliomyelitis: when can one be sure that polio virus transmission has been terminated? American journal of epidemiology **143**(8), 816–822 (1996)
17. Gibson, M.A., Bruck, J.: Efficient exact stochastic simulation of chemical systems with many species and many channels. The Journal of Physical Chemistry A **104**(9), 1876–1889 (mar 2000). https://doi.org/10.1021/jp993732q
18. Gillespie, D.T.: Exact stochastic simulation of coupled chemical reactions. The Journal of Physical Chemistry **81**(25), 2340–2361 (dec 1977). https://doi.org/10.1021/j100540a008
19. Gillespie, D.T.: Markov Processes: An Introduction for Physical Scientists. ACADEMIC PR INC (1991), `https://www.ebook.de/de/product/3655742/daniel_t_gillespie_markov_processes_an_introduction_for_physical_scientists.html`
20. Hucka, M., Finney, A., Sauro, H.M., Bolouri, H., Doyle, J.C., Kitano, H., Arkin, A.P., Bornstein, B.J., Bray, D., Cornish-Bowden, A., et al.: The systems biology markup language (sbml): a medium for representation and exchange of biochemical network models. Bioinformatics **19**(4), 524–531 (2003)
21. Kew, O., Pallansch, M.: Breaking the last chains of poliovirus transmission: progress and challenges in global polio eradication. Annual review of virology **5**, 427–451 (2018)
22. Lampoudi, S., Gillespie, D.T., Petzold, L.R.: The multinomial simulation algorithm for discrete stochastic simulation of reaction-diffusion systems. The Journal of Chemical Physics **130**(9), 094104 (mar 2009). https://doi.org/10.1063/1.3074302
23. Munsky, B., Khammash, M.: The finite state projection algorithm for the solution of the chemical master equation. The Journal of Chemical Physics **124**(4), 044104 (jan 2006). https://doi.org/10.1063/1.2145882
24. Roh, M.K., Daigle, B.J., Gillespie, D.T., Petzold, L.R.: State-dependent doubly weighted stochastic simulation algorithm for automatic characterization of stochastic biochemical rare events. The Journal of Chemical Physics **135**(23), 234108 (dec 2011). https://doi.org/10.1063/1.3668100
25. Sanft, K.R., Wu, S., Roh, M., Fu, J., Lim, R.K., Petzold, L.R.: Stochkit2: software for discrete stochastic simulation of biochemical systems with events. Bioinformatics **27**(17), 2457–2458 (2011)