Fast and flexible Bayesian platform for pharmacometric analysis

MY NAME
MY AFFILIATION

March 25, 2016

An NIH STTR (R41) proposal, submitted in response to
PAR-13-157: Early Stage Development of Technologies in Biomedical Computing, Informatics, and Big Data Science.

INSERT DATE HERE

Dear NIH:

I am submitting a grant proposal entitled “Fast and flexible Bayesian platform for pharmacometric analysis.”

- Please assign this application to the National Institute for General Medical Sciences
- Please assign this application to the following study section Modeling and Analysis of Biological Systems Study Section [MABS] or the SBIR/STTR study section Small Business: Computational, Modeling, and Biodata Management [IMST (14)].

I have discussed this application with INSERT NAME OF PROGRAM OFFICER at INSERT NAME OF INSTITUTE. They recommended the above two assignments.

Thank you very much.

Sincerely,

INSERT NAME
INSERT AFFILIATION

Project Summary/Abstract

Project narrative

Specific aims

The overall objective of this proposal is to develop additional functionality within Stan to support pharmacometric applications. This includes the addition of functions for implementing compartmental PK/PD models and schedules of dosing and other discrete events. The resulting enhancements to Stan will also support other applications that would benefit from Bayesian data analysis using complex models requiring numerical solution of systems of ordinary differential or algebraic equations.

Technical Objectives

The combined scope of the proposed Phase I and Phase II work will result in a pharmacometric modeling and simulation platform that is more flexible and that provides order(s) of magnitude more scalable and more efficient Bayesian computation than available tools. This will be accomplished by extending the capabilities of Stan to include support for pharmacometric models. The goal is a software tool that combines

- numerical solution of systems of ordinary differential equations (ODEs),
- numerical solution of systems of algebraic equations,
- Bayesian methodology that permits integration of new data with prior information,
- efficient computation of posterior densities and expectations with respect to them (e.g., for event probability estimation and prediction),
- a flexible model specification language that permits
  - probabilistic models that allow multiple levels and sources of variability and uncertainty,
  - implementation of complex models that combine submodels with different stochastic hierarchies. E.g., translational models that combine linked submodels for simultaneous analysis of clinical, animal and in vitro experiments,
  - model calculations that depend on a potentially complicated sequence of events that affect the model predictions (e.g., multiple dosing, changes in non-drug treatments, time-varying model parameters).

Specific Aims for Phase I

- Detailed planning and feasibility assessment for development of a suite of Stan enhancements to support pharmacometric modeling and simulation
- Development of Stan components for non-steady-state calculations for compartmental PK/PD models
- Handling of discrete events, e.g., bolus or constant rate input, piecewise constant parameters,
- Implementation of functions for analytic solutions for standard linear one and two compartment PK models with or without first order absorption.
- Functions that integrate numerical solution of ODEs with the event handler
- Non-stiff ODEs solved with Adams-Moulton or Runge-Kutta methods
- Stiff ODEs solved with Backward Differentiation Formulas (BDF)
- R package for PK/PD data handling and model implementation.
- Linear ODEs solved semi-analytically using matrix exponentials
- Function that integrates numerical solution of linear ODEs with the event handler

Research strategy

Significance

Nonlinear ordinary differential equations (ODEs) are widely used for modeling the dynamics of complex systems in the physical, biological, and social sciences, as well as across the engineering disciplines. Given (noisy) measurements of system inputs and outputs, researchers are faced with the "inverse problem" of estimating parameters from data relative to a statistical model of the phenomenon at hand and the measurement process. Examples of application areas include drug-disease response in pharmacology/toxicology [1, 2], thermal conductivity and heat transfer for rocketry, airflow analysis over control surfaces in in aeronautical engineering [3], facial recognition in computer vision [4, 5], dynamics of gaseous bodies with respect to background magnetic fields in astrophysics [6], population diffusion and spatial dynamics in ecology [7], continuous-time asset pricing in econometrics [8], soil-carbon respiration in biogeochemistry [9], cellular regulation in systems biology [10, 11, 12], fatigue in land and sea combat [13], and pressure changes and oceanic flow in fluid dynamics for meteorology [14], to name but a few.

Inference for ODEs can be challenging, both from a statistical perspective (the solution space of a nonlinear inverse problem can have a complex geometric structure, far from the Gaussian distributions used in standard asymptotic inference) and also computationally: solving the inverse problem in a probabilistic sense requires tracing out the space of parameter values that are consistent with data and prior information. Even the simplest ODEs generally do not admit closed-form solutions, hence computational approaches are necessarily iterative, requiring a search through parameter space. Such a search is most effectively performed using gradients, which in turn raises the challenging problem of computing gradients of differential equation solutions. In this setting, problems of parameter estimation, prediction, uncertainty quantification, and hence decision-making under uncertainty, are intractable analytically and notoriously difficult to solve numerically.

Stan [15, 16, 17] is a widely used, open-source, probabilistic programming language and Bayesian inference engine. Stan currently provides state-of-the-art Bayesian inference algorithms for inverse problems based on gradients. Full Bayesian inference is performed exactly (up to some specified precision) using Hamiltonian Monte Carlo (HMC, [18]), a Markov chain Monte Carlo (MCMC) method for sampling from the posterior and computing expectations; approximate algorithms, which can provide exploratory analysis and even useful estimates at orders of magitude greater throughput, include mean-field and full-rank variational inference (VI, [19]) and simple optimization-based Laplace approximation [20]. Stan includes a simple Runge-Kutta initial-value solver for ordinary differential equations.

We propose to extend Stan to incorporate state-of-the-art solvers for ordinary differential equations and differential algebraic equations into its inference procedures. Based on the performance of our non-stiff solvers, we believe we can achieve an order of magnitude or more speedup over the existing state-of-the-art, due to Stan's built-in efficient automatic differentiation library and efficient samplers for full Bayesian inference and optimizers for approximation Bayes through variational inference and maximum marginal likelihood. Stan provides the means to create coupled systems of differential equations through nested automatic differentiation, which provide full sensitivities (derivatives of solutions with respect to parameters) [21, 22, 23] and thus allow Stan's derivative-based inference engines to be used with statistical models involving non-linear differential equations. The key for our proposed work will be calculating the sparse, structured Jacobians of these coupled systems to permit solutions of stiff systems of equations (typically characterized by varying time scales among the components, such as the varying absorption, distribution, metabolism, and excretion of toxins in bone, fat, blood, and organs such as kidneys).

We will also extend Stan to deal with events arising from external inputs such as discrete (bolus) dosing in pharmacology or leaf-litter decomposition by enzymes in soil-carbon modeling. These represent general improvements in the Stan language which should serve specific customers in the present commercial development. In that way, we propose at Columbia University to serve the general science and engineering community with improvements in open-source software while at the same time at Metrum producing a product and services for which there is a strong market.

For concreteness, we will evaluate the tools produced using pharmacometric data with a range of sophisticated statistical and mathematical models in common use [e.g., 24, 25]. Our goal is to extend the size of data sets, sophistication of models (e.g., varying effects by patient, missing data, meta-analysis of multiple drug studies and placebo controls, increased granularity of spatio-temporal modeling, hierarchical modeling of (sub)populations), and speed of solvers by an order of magnitude or more over the state of the art, a target we believe is realistic given Stan's performance with simple non-stiff ODE solvers [26]. The result will be an even more flexible Bayesian statistics platform that supports analysis of heterogeneous collections of data conditioned on models of great stochastic and deterministic complexity and quantitative prior knowledge, leading to better calibrated and sharper (more precise) predictions [see, e.g., 27].

Background

Modern research in science and engineering is characterized by big, messy data. Scientific datasets are not just bigger in that there are more instances of the same thing—a mere increase in sample size would make our job as data analysts easier. Rather, there is more *breadth* and *complexity* to the data: more subgroups, locations, or time granularity than is currently being modeled, more partial and noisy measurements that cannot easily be incorporated into standard models, more related studies available for meta-analysis and prior formulation, more information on the physical mechanisms of measurement, more information on the population units being measured, and more fine-grained information on the predictions desired.

Consequently, in order to utilize big data we need complementary big models that can, for example, adjust the data to match sample to population, to match treatment to control groups in causal inference, or learn about interactions and variation and make individualized predictions rather than be limited to averages. Once we have big models we inevitably have multiple levels of uncertainty and variation, which can be studied using Bayesian inference, as is demonstrated in a large literature of applied statistics, including our own books and research articles. The big computational challenge is scalability: developing algorithms and implementations that work well and do not require too much storage and computation time as datasets become larger and the amount of predictive information increases.

Modeling and simulation in drug research and development

Pharmacometric data typically consists of longitudinal measurements gathered from different sources such as drug concentrations in blood plasma and pharmacological effects often resulting from a sequence of treatment events, e.g., drug doses. Mathematical modeling and simulation facilitates decision-making and risk-benefit assessment by drug developers regulators, health care providers, and patients.

Mathematical models describing measurements over time are often compartmental models formulated as systems of first order ordinary differential equations (ODEs). These models often also involve a hierarchy of random effects required to describe sources of variability, e.g., inter-individual, inter-occasion and residual variation. Highly complex systems pharmacology models describe physiologic, biochemical, and pharmacological processes at multiple scales (molecular, cellular, tissue, organ, organism and population) [e.g., 11, 1]. Even relatively simple compartmental models used for pharmacokinetic data analysis require solution of systems of ODEs often by computationally demanding numerical methods.

Although there are well-developed methods for solving continuous ODEs given initial values, tools for inference in models that incorporate ODEs as part of the statistical model are less-developed. Discrete events such as bolus doses into a model compartment induce discontinuities in the ODEs or their solution, which adds additional computational complexity. We propose to evaluate and extend software tools to incorporate better methods for solving ODEs and handle discrete events.

Systems pharmacology models are very useful for applications in translational drug research–predictions of drug disposition and effects in humans based on preclinical research. The development of such models is highly challenging and resource intensive. It requires integration of large amounts of prior biochemical and physiological knowledge with a heterogeneous collection of data from in vitro, animal and human studies. Bayesian methods provide a rational approach to the model-based analysis of data conditioned on such complex models and quantitative problem-specific knowledge about the model parameters [e.g., 2, 12].

Innovation

Comparison to Existing Tools

Various existing software tools address subsets of those features, but none address them all. Shortcomings of currently available software are outlined below. Commercial tools that specialize in pharmacometric applications such as NONMEM, Monolix and Phoenix offer

- Good support for compartmental models based on ODEs and for typical schedules of dosing and other discrete events.
- Including ODE solvers specialized for stiff, non-stiff and linear ODEs
- Limited flexibility w.r.t. probabilistic model components.
- NONMEM & Monolix employ relatively inefficient MCMC algorithms for Bayesian analyses.
- Phoenix provides no support for Bayesian analysis.
- Stochastic structure of supported models is limited:
- Only Normal-Wishart priors
- Only normally distributed variability (except for residual variability)
- Cannot combine submodels with different stochastic hierarchies

Stan is in part modeled on the previous state-of-the-art tool for Bayesian inference via MCMC, WinBUGS [28], which contains a module PKBugs for ODE solving, and for which we have the BUGSModelLibrary of sample models for testing [29]. WinBUGS and other Gibbs-sampling based approaches provide

- inefficient MCMC algorithms compared to Stan, which is
  - two orders of magnitude more scalable, and
  - between two times faster and orders of magnitude faster, with greater speedups in more challenging problems.

Advantages of Stan

In contrast, Stan provides

- more efficient posterior simulation algorithms (HMC/NUTS) than other pharmacometrics and general purpose Bayesian analysis tools,
- approximate Bayesian (ADVI) and optimization methods,
- numerical solvers for non-stiff ODEs,
- only limited, primitives-based support for pharmacometrics models,
- no tools for handling typical schedules of dosing and other discrete events,
- a single simple ODE solver (no stiff solvers, no efficient solvers for linear ODEs), and
- no tools for solving nonlinear algebraic equations, which are required for steady-state solutions for ODES, a commonly encountered challenge in pharmacometric analyses.

These properties of Stan make it a highly attractive general purpose program for Bayesian data analysis and arguably the best available platform upon which to build additional features. As outlined above, Stan lacks some important capabilities required for many pharmacometric applications.

Approach

0.1 Phase I Base

Phase I Base will be limited to (1) detailed planning and feasibility assessment for development of a suite of Stan enhancements to support pharmacometric modeling and simulation, and (2) developing an initial subset of the planned components.

0.1.1 Planning and feasibility assessment

The first stage of the work will be to develop more detailed functional specifications for Stan components that fulfill the following general functional requirements: Functions that calculate the amount in each compartment of a compartmental PK/PD model at a specified time given the amount in those compartments at a previous time, Calculations based on analytic solutions for the standard linear one and two compartment PK models with or without first order absorption. Calculations based on numerical solution of a user-specified system of ordinary differential equations (ODEs). Specific functions will be required for different categories of ODEs [30]:

- Stiff ODEs solved numerically via methods based on backward differentiation formulas (BDF).
- Non-stiff differential equations solved numerically via Adams-Moulton or Runge-Kutta methods.
- Linear ODEs solved semi-analytically via matrix exponential methods.
- Functions that calculate the amount in each compartment of a compartmental PK/PD model at a vector of specified times given a schedule of discrete events, e.g., bolus or constant rate doses and piecewise constant parameters. These functions will be developed by integrating the functions for calculating the compartment amounts at a single time with software for handling the event schedule.
- One or more functions for numerical calculation of the solution to a system of nonlinear algebraic equations.
- Functions that calculate the amount in each compartment of a compartmental PK/PD model at a vector of specified times under quasi-steady-state conditions resulting from a periodic input, e.g., repeated administration of equal doses at equally spaced time intervals. This requires numerical solution of both ODEs and algebraic equations in order to solve a boundary value problem.
- Integral with the development of the above functional specifications will be assessing the feasibility of implementing them within Stan. Feasibility assessment may also extend to consideration of additional functionality that may be desirable for pharmacometric and other scientific modeling applications, e.g., solutions of delay differential equations, stochastic differential equations, differential algebraic equations and partial differential equations.

0.1.2 Development of Stan components for non-steady-state calculations for compartmental PK/PD models

The software developed in this project will be written in C++. The C++ code will result in new functions in the Stan model specification language. Subject to revision based on the planning and feasibility effort described above, C++ code consistent with Stan architecture will be developed for the following tasks. A function will be constructed that calculates the solution of a system of ODEs at a specified sequence of times given a schedule of discrete events, e.g., bolus or constant rate input or piecewise constant parameters. This will be a general event handler that calls another function that returns the solution to a specific ODE system at a single time given initial values, i.e., implementing a recursive approach in which the ODEs are solved over each continuous interval given initial conditions for that interval calculated from the solution for the prior interval. Functions for calculating analytic solutions for standard linear one and two compartment PK models with or without first order absorption will be created. A function for calculating numerical solutions of non-stiff ODEs using an Adams-Moulton method will also be implemented. This will probably employ code from the Sundials collection [31]. Stan functions have already been developed for calculating numerical solutions of ODEs using Backward Differentiation Formulas (BDF) for stiff ODEs and Runge-Kutta methods for non-stiff ODEs.

Additional C++ code will be written that integrates the ODE solution functions with the event handler and exposes the results as functions in the Stan language. In addition a general purpose ODE solver function using the Adams-Moulton method will be added to Stan. The function will be consistent with the current Stan implementations of the Runge-Kutta and BDF solvers.

0.1.3 Software development practices

Stan is developed using current best-practices for production software development. These rely on a combination of test-driven interface design, established coding standards, thorough code reviews, exhaustive unit tests, and ongoing continous integration testing on all supported platforms.

Stan's software distributions, including the source version control repositories and the official releases, along with the Wiki used for design documents, and the issue tracker for feature requests and bug reports, are hosted by GitHub.[1] The entire distribution and the in-development code are open access, and available through the public-facing GitHub web site. All work is carried out in branches and submitted to be merged with the development branch through pull requests which are openly code reviewed. All behavior must be thoroughly tested, both with unit tests for the local functionality and integration tests to guard against breaking things end-to-end with local changes.

0.1.4 User-facing changes

Stan has over 1500 registered users on its users list and has seen over 200 academic publications across science, engineering, business, and sports. The results of this project will involve extensive additions to the user-facing documentation. This includes interface-by-interface walk-through tutorials and example models. Stan is increasingly being included in third-party textbooks, and we strive for backward compatibility so that these references are not made obsolete. Stan also has a dedicated web site, which is also managed through GitHub.

0.1.5 Collaboration plan

The project will be a close collaboration between personnel at Metrum Research Group and the Stan development team based at Columbia University. Metrum scientists provide expertise in pharmacometrics. In most cases they will be responsible for initial functional specifications and will often develop prototype code for the desired functionality in Stan. The Stan team will collaborate with Metrum scientists to refine the functional specifications. They will adapt or replace the prototype code to develop Stan code suitable for inclusion in publicly released versions of Stan. Part of that process will include rigorous testing to assure the code functions as intended.

The main coordination mechanisms we use are GitHub for version control and code review, Jenkins for continuous integration testing, weekly Google+ hangouts for online meetings as well as additional meetings as needed. In this particular case, we are within public transportation distance of each other, which will facilitate regular face-to-face meetings with whiteboards.

0.2 Phase I Option

0.2.1 R package development

R (`https://www.r-project.org/`) is a statistics software package widely used by pharmacometricians and other scientific fields. Users can develop additional R components and make them available as easily installed packages. An R package will be developed to provide a more familiar and user-friendly tool for the following tasks:

- Translation of data sets formatted for NONMEM (`http://www.iconplc.com/innovation/solutions/nonmem/`) into Stan-compatible formats

[1] `http://github.com/`

- Model specification
- Model execution using either the rstan or cmdstan interfaces to Stan
- Tabular and graphical analyses of Stan output specialized for pharmacometric applications
- The component functions will be programmed in the R language. They will be made available to the community via open distribution sites, e.g., CRAN (`https://cran.r-project.org/`) or R-Forge (`https://r-forge.r-project.org/`).

0.2.2 Development of additional Stan components for linear compartmental PK/PD models

A large fraction of pharmacometric models are based on systems of linear ODEs with constant coefficients. Such models can be solved semi-analytically via a matrix exponential approach. A matrix exponential method can solve applicable ODEs much more rapidly than the multi-step numerical methods implemented as part of Phase I Base. A Stan function for calculating non-steady-state and steady-state solutions of linear ODEs using matrix exponentials will be developed. The specific computational approach will be identified during the Phase I planning. A variety of algorithms have been proposed [32] and implemented in various languages, e.g., FORTRAN [33] and C++ (`https://people.sc.fsu.edu/~jburkardt/cpp_src/matrix_exponential/matrix_exponential.html`).

As with the ODE solvers described in section 3.1.2, a Stan function integrating the matrix exponential solver with the event handler will be developed.

Literature Cited

[1] Mark C. Peterson and Matthew M. Riggs. A physiologically based mathematical model of integrated calcium homeostasis and bone remodeling. *Bone*, 46(1):49–63, 2010.

[2] Andrew Gelman, Frederic Bois, and Jiming Jiang. Physiological pharmacokinetic analysis using population modeling and informative prior distributions. *Journal of the American Statistical Association*, 91(436):1400–1412, 1996.

[3] Oleg M Alifanov. *Inverse heat transfer problems*. Springer, 2012.

[4] Gilles Aubert and Pierre Kornprobst. *Mathematical Problems in Image Processing: Partial Differential Equations and the Calculus of Variations*, volume 147. Springer Science & Business Media, second edition, 2006.

[5] Alan C Bovik. *Handbook of Image and Video Processing*. Academic press, second edition, 2010.

[6] SM Tobias, K Dagon, and JB Marston. Astrophysical fluid dynamics via direct statistical simulation. *The Astrophysical Journal*, 727(2):1–12, 2011.

[7] Kondalsamy Gopalsamy. *Stability and Oscillations in Delay Differential Equations of Population Dynamics*. Springer, 2013.

[8] Michael Johannes and Nicholas Polson. Mcmc methods for continuous-time financial econometrics. In Yacine Aït-Sahalia and Lars Peter Hansen, editors, *Handbook of Financial Econometrics: Applications*. Elsevier, 2010.

[9] Stefano Manzoni and Amilcare Porporato. Soil carbon and nitrogen mineralization: theory and models across scales. *Soil Biology and Biochemistry*, 41(7):1355–1379, 2009.

[10] Kyle T. Baron and Marc R. Gastonguay. Simulation from ODE-based population PK/PD and systems pharmacology models in R with `mrgsolve`. *J Pharmacokinet Pharmacodyn*, 42(W-23):S84–S85, 2015.

[11] Kyle Baron, Matthew M. Riggs, Ryoko Sawamura, Takako Shimizu, Fumihiko Okada, Jin Zhou, Takahiro Shibayama, and Mendel Jansen. An evaluation of calcilytic effects on parathyroid hormone and bone mineral density response using a physiologically-based, multiscale systems pharmacology model. *Journal of Bone and Mineral Research*, 28(S1), 2013.

[12] Eric Leclerc, Jeremy Hamon, and Frederic Yves Bois. Investigation of ifosfamide and chloroacetaldehyde renal toxicity through integration of in vitro liver-kidney microfluidic data and pharmacokinetic-system biology models. *Journal of Applied Toxicology*, 36(2):330–339, 2016.

[13] Xavier Rubio-Campillo. Model selection in historical research using approximate Bayesian computation. *PloS one*, 11(1):e0146491, 2016.

[14] Jule G. Charney and N. A. Phillips. Numerical integration of the quasi-geostrophic equations for barotropic and simple baroclinic flows. *Journal of Meteorology*, 10(2):71–99, 1953.

[15] Bob Carpenter, Andrew Gelman, Matt Hoffman, Daniel Lee, Ben Goodrich, Michael Betancourt, Marcus Brubaker, Jiqiang Guo, Peter Li, , and Allen Riddell. Stan: A probabilistic programming language. *Journal of Statistical Software*, in press, 2016.

[16] Stan Development Team. Stan: a C++ library for probability and sampling, version 2.9.0. 2016.

[17] Richard McElreath. *Statistical Rethinking: A Bayesian Course with Examples in R and Stan*. Chapman & Hall/CRC Press, 2016.

[18] Radford Neal. MCMC using Hamiltonian dynamics. In Steve Brooks, Andrew Gelman, Galin L. Jones, and Xiao-Li Meng, editors, *Handbook of Markov Chain Monte Carlo*, pages 116–162. Chapman and Hall/CRC, 2011.

[19] M. J. Wainwright and M. I. Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1:1–305, 2008.

[20] Andrew Gelman, J. B. Carlin, Hal S. Stern, David B. Dunson, Aki Vehtari, and Donald B. Rubin. *Bayesian Data Analysis*. Chapman &Hall/CRC Press, London, third edition, 2013.

[21] Steven L. Lee and Paul D. Hovland. Sensitivity analysis using parallel ODE solvers and automatic differentiation in C: SensPVODE and ADIC. In George Corliss, Christèle Faure, Andrewas Griewank, Laurent Hascoët, and Uwe Naumann, editors, *Automatic differentiation of algorithms*, pages 223–229. Springer, 2002.

[22] Radu Serban and Alan C Hindmarsh. Cvodes: An ode solver with sensitivity analysis capabilities. Technical Report Technical Report UCRL-JP-200039, Lawrence Livermore National Laboratory, 2003.

[23] Bob Carpenter, Matthew D. Hoffman, Marcus Brubaker, Daniel Lee, Peter Li, and Michael Betancourt. The Stan math library: Reverse-mode automatic differentiation in C++. *arXiv*, 1509.07164, 2015.

[24] Ene I. Ette and Paul J. Williams. *Pharmacometrics: The Science of Quantitative Pharmacology*. John Wiley & Sons, 2007.

[25] Stephan Schmidt and Hartmut Derendorf. *Applied Pharmacometrics*, volume 14. Springer, 2014.

[26] S. Weber, B. Carpenter, D. Lee, F. Bois, A. Gelman, and A. Racine. Bayesian drug disease model with Stan—using published longitudinal data summaries in population models. *Abstracts of the Annual Meeting of the Population Group in Europe*, 2014.

[27] Tilmann Gneiting, Fadoua Balabdaoui, and Adrian E Raftery. Probabilistic forecasts, calibration and sharpness. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69(2):243–268, 2007.

[28] D. J. Lunn, A. Thomas, N. Best, and D. Spiegelhalter. WinBUGS – a Bayesian modelling framework: concepts, structure, and extensibility. *Statistics and Computing*, 10:325–337, 2000.

[29] Marc R. Gastonguay1, William R. Gillespie, and Robert J. Bauer. Comparison of MCMC simulation results using NONMEM 7 or WinBUGS with BUGSModelLibrary. *Abstracts of the Annual Meeting of the Population Group in Europe*, 2010.

[30] George D. Byrne and Alan C. Hindmarsh. A polyalgorithm for the numerical solution of ordinary differential equations. *ACM Transactions on Mathematical Software (TOMS)*, 1(1):71–96, 1975.

[31] Alan C Hindmarsh, Peter N Brown, Keith E Grant, Steven L Lee, Radu Serban, Dan E Shumaker, and Carol S Woodward. Sundials: Suite of nonlinear and differential/algebraic equation solvers. *ACM Transactions on Mathematical Software (TOMS)*, 31(3):363–396, 2005.

[32] Cleve Moler and Charles Van Loan. Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later. *SIAM review*, 45(1):3–49, 2003.

[33] R.B̃. Sidje. Expokit: A software package for computing matrix exponentials. *ACM Trans. Math. Softw.*, 24(1):130–156, 1998.

Consortium/contractual arrangements

Letters of support

Resource sharing plan(s)

Data sharing plan

Budget justification

Personnel

-

Facilities & Other Resources

Office

Computer

Other

Equipment

Metrum Research Group

Since the primary focus of this project is computer software and development, the equipment requirements are primarily suitable computer hardware and software resources. All Metrum scientists are equipped with a recent model Apple MacBook Pro. Most production computation is done on Amazon Web Services (AWS) cloud compute servers using the Ubuntu operating system. The software required for the proposed project is installed on the standard machine image configuration used by Metrum, e.g., Gnu C++ compiler, R, Gnu Fortran, NONMEM, LaTeX, Stan, git and subversion version control software.

Columbia University

At Columbia University, all participants are equipped with recent Apple Macbook Pro personal computers. The Stan project has three dedicated servers with 64GB and 8 cores each, one for each of Mac OS X, Linux, and Windows, along with two dedicated workstations for web hosting and remote debugging on Windows and Linux. The Stan team has access to two Linux clusters, one shared by statistics and astronomy with 384 cores, and one shared by 24 departments with 2672 CPU cores, 35 of which are high end 256-gigabyte memory nodes, and 45,000+ NVIDIA K40 and K20 GPU cores. All of the software required for individual computers for this project (see the list above) is free and open source. Source version control, issue trackers, and project web servers are provided by GitHub, mailing lists for user and developer groups is supplied by Google, and continuous integration for the interfaces by Travis; all of these services are free to open-source projects.