The SBML ODE Solver Library: a native API for symbolic
and fast numerical analysis of reaction networks

Rainer Machné\textsuperscript{a}, Andrew Finney\textsuperscript{b}, Stefan Müller\textsuperscript{c}, James Lu\textsuperscript{c}, Stefanie Widder\textsuperscript{a}, Christoph Flamm\textsuperscript{a,d}

\textsuperscript{a}Theoretical Biochemistry Group, Institute for Theoretical Chemistry, Univ. of Vienna, Währingerstr. 17, 1090 Vienna, Austria, \textsuperscript{b}Physiomics PLC, Magdalen Centre, Oxford Science Park, Oxford, OX4 4GA, UK, \textsuperscript{c}Johann Radon Institute for Computational and Applied Mathematics (RICAM), Austrian Academy of Sciences, Altenbergerstrasse 69, 4040 Linz, Austria, \textsuperscript{d}Bioinformatics, Institute for Computer Science, Univ. of Leipzig, Härtelstr. 16-18, 04107 Leipzig, Germany

\textbf{ABSTRACT}

The SBML ODE Solver Library (SOSlib) is a programming library for symbolic and numerical analysis of chemical reaction network models encoded in the Systems Biology Markup Language (SBML). It is written in ISO C and distributed under the open source LGPL license. The package employs \textit{libSBML} structures for formula representation and associated functions to construct a system of ordinary differential equations (ODEs), their Jacobian matrix and other derivatives. SUNDIALS CVODES is incorporated for numerical integration and sensitivity analysis. Preliminary benchmarking results give a rough overview on the behavior of different tools and are discussed in the supplementary material. The native API provides fine-grained interfaces to all internal data structures, symbolic operations and numerical routines, enabling the construction of very efficient analytic applications and hybrid or multi-scale solvers with interfaces to SBML and non SBML data sources. Optional modules based on \textit{XMGrace} and \textit{Graphviz} allow quick inspection of structure and dynamics.

\textbf{Availability:} www.tbi.univie.ac.at/$\sim$raim/odeSolver/

\textbf{Contact:} \{raim, xtof\}@tbi.univie.ac.at

1 INTRODUCTION

Mathematical modeling of (bio)chemical reaction networks involves a variety of techniques and theories and has long been applied for many purposes in research and technology. The need for exchange of models between different computational tools motivated collaborative efforts to develop standard formats for describing the common chemical reaction networks underlying the differing derived mathematical descriptions. Of the two XML based standards SBML (Hucka \textit{et al.}, 2003) and CellML (Lloyd \textit{et al.}, 2004), the former is supported by a growing number of applications and an official programming library, \textit{libSBML} (http://sbml.org/software/libsbml/). While available tools (see http://www.sbml.org) cover a variety of methods to edit and analyze reaction networks and their dynamics and/or structure, they are mostly designed as platform specific standalone tools, accessible mostly via complex graphical user interfaces. In contrast, \textit{SOSlib} combines \textit{libSBML} with the SUNDIALS package (http://www.llnl.gov/CASC/sundials/) to provide a detailed API (application program interface) to both a derived ODE system and various derivatives thereof (e.g. the Jacobian matrix) and to efficient integration and sensitivity analysis routines. The fine-grained interfaces allow to access integration routines at all levels, e.g. to operate on parameter and variable values during integration or to incorporate external data. All functionalities of the library are implemented in several well-documented example programs and a simple command-line application with additional visualization modules.

2 METHODS

The \textit{SOSlib} is a straightforward integration of the features of \textit{libSBML}, the official library for parsing and editing SBML, with CVODES\textsuperscript{	extregistered} methods for solving stiff and non-stiff ODE systems and their parameter sensitivities. Figure 1 depicts the basic architecture, which can be outlined as follows:

1. Construction of an ODE system \((dx/dt = f(x, p, t))\), where \(p\) are parameters of the system) from a reaction network follows the usual procedure, as described in many text books. SBML’s “kinetic law” construct represents reaction kinetics in \((\textit{item or mole})/\textit{second} \text{instead of the usual mole}/\textit{liter}/\textit{second})\, allowing a very simple handling of multi-compartment systems. The SBML “rate rule” construct enables representation of ODEs directly, and \textit{SOSlib} actually constructs a derived SBML model, consisting only of rate rules (ODEs) for the system’s variables. Simple recursive functions for formula evaluation, symbolic differentiation and simplification based on \textit{libSBML}'s abstract syntax tree (AST) representation of formulae further allow to construct the Jacobian matrix \((J : \delta f / \delta x)\) and differentiation with respect to parameters of the ODE system \((P : \delta f / \delta p)\). Differentiation is implemented for all formulae representable in AST. Structures in \( f \) that can’t be differentiated with respect to \( y \) will produce messages in the returned formula for \( df / dy \) as well as in \textit{SOSlib}'s error management system.

2. The constructed ODE system and its derivative \( J \) are then used to initiate numerical integration by CVODE’s implementation of the BDF (backward differentiation formula) or the Adams-Moulton (AM) method.
with Newton or Functional iteration to calculate $z(t)$ for a requested series of time points. BDF and AM methods are used for stiff and non-stiff systems, respectively. Changing variables $z(t)$ at any time point by either event assignments (see below) or by calling applications simply requires reinitialization of the integrator structures with new initial values. \textit{SOSlib} incorporates \textit{CVODES}' methods for forward sensitivity analysis using the parametric derivatives in $P$ to calculate $\delta z(t)/\delta p$. The performance of the methods for iteration as well as for forward sensitivity analysis (staggered direct, simultaneous or staggered corrector) depend on the numerical properties of the specific system. When construction of $J$ or $P$ fails due to undifferentiable structures in ODE, internal approximation routines of CVODES are employed. Both, numerical integration and sensitivity analysis are compatible with online variable manipulation. \textit{SBML} event triggers are evaluated at every time step and executed if fired. The flaws of this approach are that the accuracy of event detection depends on the chosen time step and that the order of events fired at the same time step is not further resolved.

(3) Two optional modules support visualization of a model’s structure and dynamics. Time courses of concentrations, rates, reaction fluxes, and Jacobian matrix values can be directly visualized in XMGraue (http://plasma-gate.weizmann.ac.il/Grace/). The Graphivc library is employed for graph drawing (http://www.graphviz.org). Besides the usual bipartite reaction network, a species interaction and a parameter dependency graph based on $J$ and $P$ respectively, proved useful for visual exploration of dominating feedback cycles and parameter dependencies.

### 3 ACCURACY, SCOPE AND APPLICATIONS

#### 3.1 Numerical Analysis

\textit{SOSlib}'s integration routines have been tested with the official \textit{SBML Semantic Test Suite}. All of the tests without algebraic rules, events or delays were successfully solved. We have benchmarked the performance of \textit{SOSlib} and four other \textit{SBML ODE} solvers, namely \textit{Jarnac}, \textit{Copasi}, \textit{Dizzy} and the \textit{SBMLToolbox} for Matlab, using 4 models from the BioModels DB (Le Novère et al., 2006) and a variant of the \textit{repressilator} model in a stiff parameter regime (Müller et al., 2005). Models 22 and the \textit{repressilator} model in a stiff parameter regime fail due to undifferentiable structures in ODE, internal approximation routines of CVODES are employed. Both, numerical integration and sensitivity analysis are compatible with online variable manipulation. \textit{SBML} event triggers are evaluated at every time step and executed if fired. The flaws of this approach are that the accuracy of event detection depends on the chosen time step and that the order of events fired at the same time step is not further resolved.

(3) Two optional modules support visualization of a model’s structure and dynamics. Time courses of concentrations, rates, reaction fluxes, and Jacobian matrix values can be directly visualized in XMGraue (http://plasma-gate.weizmann.ac.il/Grace/). The Graphivc library is employed for graph drawing (http://www.graphviz.org). Besides the usual bipartite reaction network, a species interaction and a parameter dependency graph based on $J$ and $P$ respectively, proved useful for visual exploration of dominating feedback cycles and parameter dependencies.

### Table 1. Benchmarking of the \textit{SOSlib} and other \textit{SBML} Solvers

<table>
<thead>
<tr>
<th>Biomodel DB Id</th>
<th>Type</th>
<th>NEQ/Time</th>
<th>Stable</th>
<th>Oscil.</th>
<th>Oscil./stiff</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>stable</td>
<td>22/150</td>
<td>86/300</td>
<td>28/60</td>
<td>10/2000</td>
</tr>
<tr>
<td>14</td>
<td>stable</td>
<td></td>
<td></td>
<td></td>
<td>6/10,000</td>
</tr>
<tr>
<td>33</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>22</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>repress.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

CPU times of single runs in milliseconds for ODE construction and numerical integration on a Pentium 4 CPU with 3.4 GHz and 1GB RAM. See supplementary material for details. The column numbers are the models' IDs at the BioModels DB (Le Novère et al., 2006), except for 'repress.' ('repressilator') which was taken from Müller et al., 2005. NEQ: number of ODEs. Time: end time of integration in the model's built in default unit. Type: refers to the model’s dynamic behavior.

#### 4 DISCUSSION

While systems biology has gained much attention only recently, it’s current methods are in large parts old and well-founded on mathematical and biochemical theories. We follow the spirit of \textit{SBML} and \textit{libSBML} to provide application developers with detailed interfaces to already existing standard methods under a very liberal licensing policy. In other words: we are not trying to ‘reinvent the wheel’, but offer its best possible implementation to enable rapid scientific progress and unrestricted further development in this field. \textit{SOSlib} uniquely provides a detailed native API, independent of any GUI or scripting environment, that allows access to all the components of a deterministic reaction network simulator enabling scientists to construct efficient applications that are tailored to their research needs. We are not aware of any other open source \textit{SBML} analysis package that offers the time course sensitivity analysis provided by \textit{SOSlib} through \textit{CVODES}. The next releases (during 2006) will include exact event handling and extend the \textit{CVODES} interface to provide adjoint sensitivities and sophisticated parameter identification routines that employ inverse methods. \textit{SUNDIALS}’s \textit{IDA} solver for differential algebraic equation systems will be interfaced to provide integration for models with arbitrary algebraic rules. Bifurcation and feedback analysis would be obvious next steps.

### ACKNOWLEDGMENT

This work was supported by the WWTF, project number MA05. RM gratefully acknowledges financial support by his parents.

### REFERENCES


