Report on SUNDIALS: Suite of Nonlinear and Differential/Algebraic Equation Solvers

March 12, 2004

General

This is a clearly written paper setting out the current state of the Lawrence Livermore National Laboratory project described in the title. It outlines how existing solvers were re-written with the aims of: better support for solving large systems in parallel, including a wide variety of linear system solvers both direct and iterative; better extraction of software components usable in several solvers; a user interface suitable for both 'simple' and 'expert' use. Also described is the partly completed work to add sensitivity analysis to the solvers.

The basic solvers (ODE, DAE, algebraic) are described. Much of course is standard, but there is with a wealth of detail about algorithm heuristics that have evolved over years of experience. This is invaluable information for any practitioner in the field. The more so because appearing in the open literature helps protect it from interest groups trying to patent common knowledge, as is currently happening in some areas of scientific computing.

A detailed description of preconditioning methods and options follows.

Next is a description of the way in which forward and adjoint methods are applied to compute sensitivities. It is done at the model level, that is by discretizing the derivatives of the mathematical equations rather than the Automatic Differentiation route of differentiating the discretization.

There follow sections on the code organization and on how a user sets up the system to solve a problem, and extracts the results.

The abstract and conclusions are appropriate. The references cover theory, usage and experience with the solvers and run from the seventies to the present.

I recommend publication subject to some minor improvements suggested below. I would not wish to see the revised version before publication.

Software architecture

In the abstract, the authors say "a highly modular structure to allow incorporation of different preconditioning and/or linear solver methods; and clear interfaces allowing for users to provide their own data structures underneath the solvers"; and in the conclusion section they state: "The design philosophy of providing clear interfaces to the user and allowing the user to supply their own data structures make the solvers reasonably easy to add into existing simulation codes".

Designing good interfaces in complex numerical software, such as SUNDIALS, is not a trivial task. The authors have accumulated valuable experience doing that, and others should benefit from this experience. An illustration of how "clear" interfaces are achieved and how new methods can be added would be helpful. A discussion on design decisions is valuable too. If it is impractical to add to the present paper and is explained elsewhere, references to where the reader can find such information should be given.

Heuristics

Many heuristics are discussed in this paper. The authors give values for numerous constants, but do not explain why they chose these values. On p.6, why is the safety factor 1/6? On p.5, the stepsize h_n is replaced by $h_n/4$. Why not $h_n/2$ for example? On p.6, the safety factor is limited above by 0.2. Why is 0.2 chosen?

I accept this paper is not the place to justify 'magic numbers' in heuristics. But a researcher or a practitioner may be interested in knowing the reasons for choosing a certain value of a constant or a parameter. If there are technical reports where they *are* justified, it would be useful to have references to them.

Also, wherever a snippet of rationale is provided, as p.12 'The above strategy balances ...', it immediately makes the text more accessible to the non-specialist reader.

Specific comments

General Quite a few paragraphs would read better if split. I have made some suggestions, and leave to the authors to decide on better paragraph splitting.

Page 4. The notation $y_{n(m)}$ jars. It means the approximation, at the mth (Newton or other) iteration, to the computed solution y_n at time point n, but of course if read literally it indicates that n is somehow a function of m. I know you use y_n^i to mean the ith component of y_n so superscript m is out, but I think $y_n(m)$ or $y_{n,m}$ reads a lot better.
line -17. A formula for $\|\cdot\|_{\mathrm{WRMS}}$ would be helpful.

Page 5. line 2. What is "non-fatal convergence failure"?

Page 6. line 6. A reference to Curtis-Powell-Reid might be included.
line 14. 'A critical part of CVODE ... ' The comma should become a dash, or vice versa. But is this sentence needed? Anyone reading this far knows well enough that professional ODE software controls local error.
line -9. "the step is restarted from scratch" It is not clear what this means.
line -8. The ratio h'/h is limited above to 0.2. It is not clear if 0.2 stays until the end of the integration. This should be clarified; the same for 0.1.

Page 7. line 1. 'on the step just completed ... done'. Would 'on the previous step ... done on this step' be clearer?
line $19, 22$. It is little confusing with LTE denoting two different things. I think a formula for LTE(q') would help the reader.

Page 8. Inexact Newton Iteration. Using m rather than n fits with earlier notation.

The assignment $u_{n+1} = u_n + \delta_n$ threw me, when it is clear from the next paragraph that a 'cautious Newton' $u_{n+1} = u_n + \lambda\delta_n$ is actually used, and only part way down p9 is it explained that either of these may be used. Possibly change 2(b) to 'Set $u_{n+1} = u_n +$ multiple of δ_n'. Alter p9 text accordingly.
line -13. 'These solutions are only approximate. ... new approximate solution.' mixes up two meanings of approximate solution. It might better read 'The update δ_n is only

an approximate solution of system 2(a). At each stage in the iteration process, a scalar multiple of the update is added to the previous approximate solution, u_n, to produce a new approximate solution.'

Page 10. line $3, 5, 7, 9, 11$. The η symbol is also used on p. 7, line 14,16, but for a different purpose.
line 13. You should explain what KINSOL does if the solution it is seeking violates a constraint such as $u^i > 0$.

Page 13. line -18. The strategy for starting IDA is described, but there is no corresponding description for CVODE. For consistency, such explanation could be given.

Page 14. The order selection approach may be easier to read if presented in the form of an algorithm.

Page 21. Sect 4.3 and elsewhere. I found it slightly confusing that a (purely) algebraic system is referred to as a 'nonlinear system' as if 'nonlinear' is the key aspect. I suggest 'algebraic system'.

Page 22. Sect 5. It would be useful to give reasons for choosing C rather than Fortran 90/95, which (as far as I can see) can accommodate each of the listed design features. 'Our programmers know C best' is an acceptable reason.

Wording and paragraphs

Page 2. New paragraph e.g. at "SUNDIALS is being expanded…".

Page 3. line $2, 3$. "contained" is redundant.
line -22. "which" refers to "N-space", not "ODE...". Reword.
line -21. You may omit "abstract".
line -9. New paragraph at "For nonstiff problems, CVODE...".
line -4. "recent history of the step sizes" is not very precise. You may say which step sizes.

Page 4. line 3. New paragraph at "Functional iteration...".
line 18. New paragraph at "For large..."
line -5. Comma after "as possible".

Page 5. line $5, 7$. See "reevaluate" and "re-evaluate".
line 20. New paragraph at "We initialize..."
line -20. Comma before "but instead".

Page 8. line 1. New paragraph at "The implementation...".

Page 10. line 14. You may say that i denotes the ith component of u.
line -5. New paragraph at "A second...".

Page 13. line 10. For consistency, the subscript WRMS may be omitted.

Page 18. line −9. Period after "...each step".
line −5. Period after "matrix".
line −1. Period after "[Feehery ...]".

Page 22. line 14. Remove " " in "NVECTOR".

Page 29. The formats of the first and the 17th reference are not consistent.

Software

The serial version installs nicely on Sun Solaris with the GNU C compiler, and all the examples execute without any problems. Since I do not have currently access to a parallel machine, I have not tried the parallel version of SUNDIALS.

However, I had slight difficulties installing SUNDIALS on Macintosh running OS X. I believe this can be easily fixed by the authors, and I encourage them to do so.

The SUNDIALS suite is very well organized across directories. The programs are clearly written and well commented. As a minor remark, I find the main solver files quite long: kinsol.c 1410 lines, cvode.c 3048, ida.c 3062 and cvodes.c 5791.