Atmos. Chem. Phys. Discuss., 5, S2585–S2588, 2005
www.atmos-chem-phys.org/acpd/5/S2585/
European Geosciences Union

Atmospheric Chemistry and Physics Discussions

Interactive comment on "Technical Note: Simulating chemical systems in Fortran90 and Matlab with the Kinetic PreProcessor KPP-2.1" *by* A. Sandu and R. Sander

Anonymous Referee #1

Received and published: 13 September 2005

Review of

A. Sandu and R. Sander:

Technical note: Simulating chemical systems in Fortran90 and Matlab with the Kinetic PreProcessor KPP-2.1

The paper addresses an important practical issue in atmospheric chemistry: What are suitable tools when simulating the reaction kinetics of complex chemical systems? When performing such simulation an appropriate numerical method must be chosen.

Such systems are often numerically stiff, which requires using backwards differencing formulae for integration of the system of coupled differential equations.

In addition, atmospheric chemists may prefer tools, which do not require deep knowledge of the involved numerical problems and are easy to use. In past, commercial software packages have been frequently and very successfully used. However, such packages may be quite expensive and commercial interests of the software companies sometimes lead to software development, which is impractical for the users. Specifically, source codes of commercial software are not available, which may prevent a skilled used to exploit all features of a certain software package. Here, software which is licensed using the GNU public license has two advantages: It's free and the source code is public. Thus everybody can uses the code and, if needed adjust it to the own needs.

The authors present the KPP pre-processor as a quite practical solution to this problem: A free code, which is easy to use, and able to handle stiff differential equations, as often found in atmospheric chemistry.

The paper is, as the title says, a technical note, as the numerical machine of the KPP is already published in a previous paper by Sandu et al. (Sandu et al. J. Comp. Phys., 129, 101-110 (1996)), which is cited in the presented paper. However, as ACP addresses a wider community than the J. Comp. Phys., I think that (despite little hard 'scientifically' new results) the paper is of interest to ACP, because it may stimulate using open source software tools in the atmospheric science community. The availability of such tool in this specific community would be of great practical use and in turn great scientific impact on the community.

I have some specific comments which the authors should address in a revised version: Generally, I think that the paper has a little bit too much the character of a software manual. The authors might consider reducing the detailed description of the software details and describe the features of the new implementation in a more general way. It

ACPD

5, S2585–S2588, 2005

would be of benefit to the paper to compare the numerical accuracy of their tool with other frequently used software packages.

1. Implementation of the software. The possibility for different platforms for implementation should be discussed. As far as I can see, the code download from the internet page given, is suited for UNIX/LINUX based environment. Is there a windows version? Can cygwin be used as platform?

2. The spelling of the links should be rechecked, there are problems with capital letters.

3. Section 2.1, end. The SUN command should either be omitted, or explained: What is day and night for the integration routine? 4. Section 2.3: The authors claim: "All methods in the KPP library have excellent stability properties for stiff problems". If all methods are excellent, I wonder why there are different methods implemented. Thus, this statement needs quantification. What degree of stiffness can be solved? Where are the limits? What can be said about the accuracy of the different methods? How do they compare? Here I would suggest to add an additional section which discusses these issues. It might also be of interest to compare with the frequently used FACSIMILE software.

5. section 2.4: The authors refer to the 3D-code. This reference needs an explanation. Can 3-D problem be solved using predefined tools? What is implemented in KPP?

6. Section 3.7 - 3.10: These sections have very much the character of a Software handbook. In a paper it might be sufficient to refer to the software manual, which comes with KPP. The authors may consider to shorten and to clarify this section without describing technical details, such as the directory structure of KPP, and the explicit definition of commands, such as #STOCIOMAT. It might be enough to briefly describe the new features of the new KPP version in general terms.

I have these specific questions:

Section 3.1: Do the explicit values (e.g. NHESS = 10) refer to the example in section

ACPD

5, S2585–S2588, 2005

2.1?

Section 3.3: The subroutine FUN, can be understood by someone who has programming experience. I suggest that all parameters should be defined either in the text of in a table in the appendix.

Section 3.5: This section can be understood by someone who has some knowledge about differential equations. As the paper addresses the atmospheric science community, I would at least introduce the differential equation explicitly, and based on this explain how equation (1) is related to the differential equation.

Section 3.6: It should be explained where the Hess Matrix is used in the numerical solver.

Section 3.7: I do not understand this section. It should be omitted, or the command #STOICMAT should be explained explicitly using and example of equations.

Section 3.8. Here I also have problems to understand. What do the authors mean by the formulation in the first sentence (Ĕ allows a direct computation with respect to rate coefficientsĔ)? This section might be omitted as well.

Section 3.9 and 3.10: These sections should be omitted as well, as they contain technical information, which should be provided in the KPP manual.

Chapter 5: Applications.

The authors performed benchmark tests of the different types of integration routine. For all systems, integration times are short (0.5 - 2 minutes). It would be of interest how the numerical accuracy of the different methods compares. Also, a comparison with other commercial packages would be of interest for the paper.

Interactive comment on Atmos. Chem. Phys. Discuss., 5, 8689, 2005.