

## **Developing Numerical Integrators for The Kinetic Pre-Processor**

Eller, Paul; Hobbs, Nicholas; Jacob, Don; Jiang, Tingting  
Department of Computer Science, Virginia Tech

The Kinetic Pre-Processor is a software assistant to aid the simulation of chemical kinetic systems, as the concentrations change over time according to the differential law of mass action kinetics. Using Fortran77, Fortran90, C, or Matlab code KPP runs numerical integrators along with the concentration time derivative function, its Jacobian, and its Hessian. KPP consists of a variety of numerical integrators that are very efficient with low to medium accuracy. These numerical integrators implement the tangent linear model and adjoint model, and distinguish between sensitivities with respect to initial values and parameters for efficiency. KPP produces the building blocks for simulations and for sensitivity calculations, and provides application programming templates, which will allow the user to develop their own applications based on KPP.

The integrators have been optimized for computational efficiency through exploiting the sparsity in the Jacobian and Hessian. Each of the integrators takes in a set of integer input parameters and real input parameters to specify conditions such as method selection, upper and lower bounds for step size, and tolerances. These integrators then output a set of arguments that include the value at which the solution was computed, the numerical solution, and data on the number of function calls and steps. The environment has been modularly designed to easily allow additional integrators to be added to the suite. The software functions to give choice of both integrators and code languages, along with a variety of other options, to suit the user's requirements for the simulation they are running.

The integrators for KPP have primarily been developed in the Fortran programming language. We have been expanding the functionality of KPP to allow more of the numerical integrators to be used with the C programming language. We have focused primarily on developing versions of the Rosenbrock, Runge-Kutta, and Sdirk(Single Diagonally Implicit Runge-Kutta method) integrators that can be used by KPP with the C programming language.