The SOSI Methodology (New in V.5.3 DMSolver)

To streamline setting up simple problems (especially for the beginning user) we have developed a source-code-generating methodology called SOSI (Single-module Only Simple Input).

The requirements of SOSI are that all the equations are expressed as single assignment statements i.e. single formulas, and there is only one block of equations (the SYSTEM module).

To make initial learning easier the SOSI formula notation does not have anything peculiar to Pascal --- if you work with formulas in C, Basic, or Excel you already know the notation.

After you click on the button to convert the SO file to the PMS file you can add things to it, but there is a framework already set up for you. (In many cases you can program function subprograms in Pascal that get called in the SOSI equation statements.)

An example of a very tiny and simple problem which can be expressed entirely with SOSI is the classic chaos problem of Lorenz ("Deterministic Nonperiodic Flow", J. Atmos. Sci. 20, pp. 130-141, 1963). The SO file is

```
Parameters
   Sigma=10; R=28; BB=2.666667;
Equations
   Tderiv(X)=SIGMA*(Y-X);
   Tderiv(Y)=-X*Z+RR*X-Y;
   Tderiv(Z)=X*Y-BB*Z;
end
```

There are only two parts to a SOSI file, designated as Parameters and Equations. Each parameter has a default value as shown in the example, and the parameter definitions are terminated by semicolons, as are the equation definitions. Spaces and multiple-line statements can be used as in C or Pascal, but there is currently no allowance for comments.

There are three types of equations:

An algebraic equation has a variable name on the left-hand side and an expression involving parameters and variables (but NOT the left-hand-side variable) on the right-hand side.

An ODE with a time derivative left-hand side has Tderiv( ) indicating the first time derivative of the variable in the parentheses (all the example equations are like this).

An equation setting the right-hand-side expression equal to a constant corresponds to the PMSFIX statement usage in the conventional DMSolver source code:

Fix(cv, fv) = <rhs expression>

In the case of the Lorenz example problem shown above, absolutely no user Pascal programming is needed to create the DMSolver source code file and proceed with the analysis. After

preprocessing, sequencing, compiling, and running the problem you have to use the Display/edit command to set the initial conditions for the variables and input a set of arbitrary "range" values (see the beginning of DMSexamp.pdf).

For another test of SOSI, we programmed verbatim the equations of the EUROKIN kinetics problem 1A as follows:

```
Parameters
   kk1=0.0105;
   kk2=0.00186;
   kk3=0.0000150;
   KA=6.119;
   KB=2.496;
   Keq=176.0;
   Astart=0.65;
Equations
   Tderiv(CCA)=(-kk1*KA*(CCA-CCB/Keq)-kk3*KA*CCA)/(1.0+KA*CCA+KB*CCB);
   CCB = Astart-CCA-CCC-CCD;
   Tderiv(CCC)=kk2*KB*CCB/(1.0+KA*CCA+KB*CCB);
   Tderiv(CCD)=kk3*KA*CCA/(1.0+KA*CCA+KB*CCB);
end
```

The notation is the same as in our EUROKIN test report (www.DAofTX.com/eurorpt.pdf), where CCA, CCB, CCC, and CCD are concentrations of the 4 chemical components A, B, C, and D. There is one algebraic equation and three with time derivatives. To run one case it is only necessary to set the "ranges" and initial conditions in the Display/edit window, so no Pascal programming is involved.

The SOSI input is of course much shorter compared with our EUROKIN test report appendix example. It gives the same results for one case, but (a) in the test report we programmed the denominator that is repeated in three equations as a separate variable to avoid redundant evaluations of it, and (b) we included all the temperature-dependent formulas for the rate constants and equilibrium constants and also the linkage to least-squares so that the complete EUROKIN parameter estimation analysis could be performed. (b) requires programming in Pascal.

In some problems you may find it convenient to start with the parameters and equations expressed through SOSI and then add program elements written in Pascal to the DMSolver source code file. For example, in a dynamic case where a parameter is a function of the time variable _T you can put Pascal code into the TIMESET procedure to change the parameter(s) as time progresses.