



# Software

## AutoChem

AutoChem is a suite of Fortran 90 computer programs for the modeling of kinetic reaction systems. AutoChem performs automatic code generation, symbolic differentiation, analysis, and documentation. It produces a documented stand-alone system for the modeling and assimilation of atmospheric chemistry. Given databases of chemical reactions and a list of constituents defined by the user, AutoChem automatically does the following:

1. Selects the subset of reactions that involve a user-defined list of constituents and automatically prepares a document listing the reactions;
2. Constructs the ordinary differential equations (ODEs) that describe the reactions as functions of time and prepares a document containing the ODEs;
3. Symbolically differentiates the time derivatives to obtain the Jacobian and prepares a document containing the Jacobian;
4. Symbolically differentiates the Jacobian to obtain the Hessian and prepares a document containing the Hessian; and
5. Writes all the required Fortran 90 code and datafiles for a stand-alone chemical modeling and assimilation system (implementation of steps 1 through 5).

Typically, the time taken for steps 1 through 5 is about 3 seconds. The modeling system includes diagnostic components that automatically analyze each ODE at run time, the relative importance of each term, time scales, and other attributes of the model.

*This program was written by David John Lary of Goddard Space Flight Center. Further information is contained in a TSP (see page 1). GSC-14862-1*

## Virtual Machine Language

Virtual Machine Language (VML) is a mission-independent, reusable software system for programming for spacecraft operations. Features of VML include a rich set of data types, named functions, parameters, IF and WHILE control structures, polymorphism, and on-the-fly creation of spacecraft commands from calculated values. Spacecraft functions can be abstracted into named blocks that re-

side in files aboard the spacecraft. These named blocks accept parameters and execute in a repeatable fashion. The sizes of uplink products are minimized by the ability to call blocks that implement most of the command steps. This block approach also enables some autonomous operations aboard the spacecraft, such as aerobraking, telemetry conditional monitoring, and anomaly response, without developing autonomous flight software. Operators on the ground write blocks and command sequences in a concise, high-level, human-readable programming language (also called "VML"). A compiler translates the human-readable blocks and command sequences into binary files (the operations products). The flight portion of VML interprets the uplinked binary files. The ground subsystem of VML also includes an interactive sequence-execution tool hosted on workstations, which runs sequences at several thousand times real-time speed, affords debugging, and generates reports. This tool enables iterative development of blocks and sequences within times of the order of seconds.

*This program was written by Christopher Grasso, Dennis Page, and Taifun O'Reilly with support from Ralph Fleichert, Patricia Lock, Imin Lin, Keith Naviaux, and John Sisino of Caltech for NASA's Jet Propulsion Laboratory. Further information is contained in a TSP (see page 1).*

*This software is available for commercial licensing. Please contact Karina Edmonds of the California Institute of Technology at (818) 393-2827. Refer to NPO-40365.*

## Two-Dimensional Ffowcs Williams/Hawkings Equation Solver

FWH2D is a Fortran 90 computer program that solves a two-dimensional (2D) version of the equation, derived by J. E. Ffowcs Williams and D. L. Hawkings, for sound generated by turbulent flow. FWH2D was developed especially for estimating noise generated by airflows around such approximately 2D airframe components as slats. The user provides input data on fluctuations of pressure, density, and velocity on some surface. These data are combined with information about the geometry of the surface to calculate histories of thickness and loading terms. These histories are fast-

Fourier-transformed into the frequency domain. For each frequency of interest and each observer position specified by the user, kernel functions are integrated over the surface by use of the trapezoidal rule to calculate a pressure signal. The resulting frequency-domain signals are inverse-fast-Fourier-transformed back into the time domain. The output of the code consists of the time- and frequency-domain representations of the pressure signals at the observer positions. Because of its approximate nature, FWH2D overpredicts the noise from a finite-length (3D) component. The advantage of FWH2D is that it requires a fraction of the computation time of a 3D Ffowcs Williams/Hawkings solver.

*This program was written by David P. Lockard of Langley Research Center. Further information is contained in a TSP (see page 1).*

*LAR-16338-1*

## Full Multigrid Flow Solver

FMG3D (full multigrid 3 dimensions) is a pilot computer program that solves equations of fluid flow using a finite difference representation on a structured grid. Infrastructure exists for three dimensions but the current implementation treats only two dimensions. Written in Fortran 90, FMG3D takes advantage of the recursive-subroutine feature, dynamic memory allocation, and structured-programming constructs of that language. FMG3D supports multi-block grids with three types of block-to-block interfaces: periodic, C-zero, and C-infinity. For all three types, grid points must match at interfaces. For periodic and C-infinity types, derivatives of grid metrics must be continuous at interfaces. The available equation sets are as follows: scalar elliptic equations, scalar convection equations, and the pressure-Poisson formulation of the Navier-Stokes equations for an incompressible fluid. All the equation sets are implemented with nonzero forcing functions to enable the use of user-specified solutions to assist in verification and validation. The equations are solved with a full multigrid scheme using a full approximation scheme to converge the solution on each succeeding grid level. Restriction to the next coarser mesh uses direct injection for variables and full weighting for residual quantities; prolongation of the coarse grid correction from the coarse

mesh to the fine mesh uses bilinear interpolation; and prolongation of the coarse grid solution uses bicubic interpolation.

*This program was written by Raymond E. Mineck, James L. Thomas, and Robert T. Biedron of Langley Research Center and Boris Diskin of the National Institute of Aerospace. Further information is contained in a TSP (see page 1).  
LAR-16608-1*

## Doclet To Synthesize UML

The RoseDoclet computer program extends the capability of Java doclet software to automatically synthesize Unified Modeling Language (UML) content from Java language source code. [Doclets are Java-language programs that use the doclet application programming interface (API) to specify the content and format of the output of Javadoc. Javadoc is a program, originally designed to generate API documentation from Java source code, now also useful as an extensible engine for processing Java source code.] RoseDoclet takes advantage of Javadoc comments and tags already in the source code to produce a UML model of that code. RoseDoclet applies the doclet API to create a doclet passed to Javadoc. The Javadoc engine applies the doclet to the source code, emitting the output format specified by the doclet. RoseDoclet emits a Rose model file and populates it with fully documented packages, classes, methods, variables, and class diagrams identified in the source code. The way in which UML models are generated can be controlled by use of new Javadoc comment tags that RoseDoclet provides. The advantage of using RoseDoclet is that Javadoc documentation becomes leveraged for two purposes: documenting the as-built API and keeping the design documentation up to date.

*This program was written by Matthew R. Barry and Richard N. Osborne of United Space Alliance for Johnson Space Center. For further information, contact the Johnson Technology Transfer Office at (281) 483-3809.  
MSC-23580*

## Computing Thermal Effects of Cavitation in Cryogenic Liquids

A computer program implements a numerical model of thermal effects of cavitation in cryogenic fluids. The

model and program were developed for use in designing and predicting the performances of turbopumps for cryogenic fluids. Prior numerical models used for this purpose do not account for either the variability of properties of cryogenic fluids or the thermal effects (especially, evaporative cooling) involved in cavitation. It is important to account for both because in a cryogenic fluid, the thermal effects of cavitation are substantial, and the cavitation characteristics are altered by coupling between the variable fluid properties and the phase changes involved in cavitation. The present model accounts for both thermal effects and variability of properties by incorporating a generalized representation of the properties of cryogenic fluids into a generalized compressible-fluid formulation for a cavitating pump. The model has been extensively validated for liquid nitrogen and liquid hydrogen. Using the available data on the properties of these fluids, the model has been shown to predict accurate temperature-depression values.

*This program was written by Ashvin Hosangadi, Vineet Ahuja, and Sanford M. Dash of Combustion Research and Flow Technology, Inc., for Marshall Space Flight Center. For further information, contact Ashvin Hosangadi at hosangad@craft-tech.com.  
MFS-32140*

## GUI for Computational Simulation of a Propellant Mixer

Control Panel is a computer program that generates a graphical user interface (GUI) for computational simulation of a rocket-test-stand propellant mixer in which gaseous hydrogen ( $\text{GH}_2$ ) is injected into flowing liquid hydrogen ( $\text{LH}_2$ ) to obtain a combined flow having desired thermodynamic properties. The GUI is used in conjunction with software that models the mixer as a system having three inputs (the positions of the  $\text{GH}_2$  and  $\text{LH}_2$  inlet valves and an outlet valve) and three outputs (the pressure inside the mixer and the outlet flow temperature and flow rate). The user can specify valve characteristics and thermodynamic properties of the input fluids via user-friendly dialog boxes. The user can enter temporally varying input values or temporally varying desired output values. The GUI provides (1) a set-point calculator function for determining fixed valve positions that yield desired output values and

(2) simulation functions that predict the response of the mixer to variations in the properties of the  $\text{LH}_2$  and  $\text{GH}_2$  and manual- or feedback-control variations in valve positions. The GUI enables scheduling of a sequence of operations that includes switching from manual to feedback control when a certain event occurs.

*This program was written by Fernando Figueroa of Stennis Space Center, Hanz Richter of the National Research Council, and Enrique Barbieri and Jamie Granger Austin of Tulane University.*

*Inquiries concerning rights for the commercial use of this invention should be addressed to the Intellectual Property Manager, Stennis Space Center, (228) 688-1929. Refer to SSC-00213.*

## Control Program for an Optical-Calibration Robot

A computer program provides semiautomatic control of a moveable robot used to perform optical calibration of video-camera-based optoelectronic sensor systems that will be used to guide automated rendezvous maneuvers of spacecraft. The function of the robot is to move a target and hold it at specified positions. With the help of limit switches, the software first centers or finds the target. Then the target is moved to a starting position. Thereafter, with the help of an intuitive graphical user interface, an operator types in coordinates of specified positions, and the software responds by commanding the robot to move the target to the positions. The software has capabilities for correcting errors and for recording data from the guidance-sensor system being calibrated. The software can also command that the target be moved in a predetermined sequence of motions between specified positions and can be run in an advanced control mode in which, among other things, the target can be moved beyond the limits set by the limit switches.

*This program was written by Albert (Nick) Johnston of Marshall Space Flight Center. For further information, contact Sammy Nabors, MSFC Commercialization Assistance Lead, at sammy.a.nabors@nasa.gov. MFS-31925-1*

## SQL-RAMS

SQL-RAMS (where "SQL" signifies Structured Query Language and "RAMS" signifies Rocketdyne Automated Management System) is a succes-