Software

Avionics System Architecture Tool

Avionics System Architecture Tool (ASAT) is a computer program intended for use during the avionics-system-architecture-design phase of the process of designing a spacecraft for a specific mission. ASAT enables simulation of the dynamics of the command-and-data-handling functions of the spacecraft avionics in the scenarios in which the spacecraft is expected to operate. ASAT is built upon I-Logix Statemate MAGNUM, providing a complement of dynamic system modeling tools, including a graphical user interface (GUI), modeling checking capabilities, and a simulation engine. ASAT augments this with a library of predefined avionics components and additional software to support building and analyzing avionics hardware architectures using these components.

This program was written by Savio Chau, Ronald Hall, and Marcus Traylor of Caltech, and Adrian Whitfield of I-Logix 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-30629.

Updated Chemical Kinetics and Sensitivity Analysis Code

An updated version of the General Chemical Kinetics and Sensitivity Analysis (LSENS) computer code has become available. A prior version of LSENS was described in "Program Helps to Determine Chemical-Reaction Mechanisms" (LEW-15758), *NASA Tech Briefs*, Vol. 19, No. 5 (May 1995), page 66. To recapitulate: LSENS solves complex, homogeneous, gas-phase, chemical-kinetics problems (e.g., combustion of fuels) that are represented by sets of many coupled, nonlinear, first-order ordinary differential equations. LSENS has been designed for flexibility, convenience, and computational efficiency. The present version of LSENS incorporates mathematical models for (1) a static system; (2) steady, one-dimensional inviscid flow; (3) reaction behind an incident shock wave, including boundary-layer correction; (4) a perfectly stirred reactor; and (5) a perfectly stirred reactor followed by a plug-flow reactor. In addi-

tion, LSENS can compute equilibrium properties for the following assigned states: enthalpy and pressure, temperature and pressure, internal energy and volume, and temperature and volume. For static and one-dimensional-flow problems, including those behind an incident shock wave and following a perfectly stirred reactor calculation, LSENS can compute sensitivity coefficients of dependent variables and their derivatives, with respect to the initial values of dependent variables and/or the rate-coefficient parameters of the chemical reactions.

This program was written by Krishnan Radhakrishnan of the Institute for Computational Mechanics in Propulsion for **Glenn Research Center***. Further information is contained in a TSP (see page 1).*

Inquiries concerning rights for the commercial use of this invention should be addressed to NASA Glenn Research Center, Commercial Technology Office, Attn: Steve Fedor, Mail Stop 4–8, 21000 Brookpark Road, Cleveland, Ohio 44135. Refer to LEW-17519-1.

Predicting Flutter and Forced Response in Turbomachinery

TURBO-AE is a computer code that enables detailed, high-fidelity modeling of aeroelastic and unsteady aerodynamic characteristics for prediction of flutter, forced response, and blade-row interaction effects in turbomachinery. Flow regimes that can be modeled include subsonic, transonic, and supersonic, with attached and/or separated flow fields. The three-dimensional Reynolds-averaged Navier-Stokes equations are solved numerically to obtain extremely accurate descriptions of unsteady flow fields in multistage turbomachinery configurations. Blade vibration is simulated by use of a dynamic-grid-deformation technique to calculate the energy exchange for determining the aerodynamic damping of vibrations of blades. The aerodynamic damping can be used to assess the stability of a blade row. TURBO-AE also calculates the unsteady blade loading attributable to such external sources of excitation as incoming gusts and blade-row interactions. These blade loadings, along with aerodynamic damping, are used to calculate the forced responses of blades to predict their fatigue lives. Phase-lagged boundary conditions based on the direct-store

method are used to calculate nonzero interblade phase-angle oscillations; this practice eliminates the need to model multiple blade passages, and, hence, enables large savings in computational resources.

This program was written by Dale E. Van-Zante and John J. Adamczyk of **Glenn Research Center***; Rakesh Srivastava, Milind A. Bakhle, and Aamir Shabbir of the University of Toledo; Jen-Ping Chen and J. Mark Janus of Mississippi State University; Wai-Ming To of AP Solutions, Inc.; and John Barter of GE Aircraft Engines. Further information is contained in a TSP (see page 1).*

Inquiries concerning rights for the commercial use of this invention should be addressed to NASA Glenn Research Center, Commercial Technology Office, Attn: Steve Fedor, Mail Stop 4–8, 21000 Brookpark Road, Cleveland, Ohio 44135. Refer to LEW-17514-1.

Upgrades of Two Computer Codes for Analysis of Turbomachinery

Major upgrades have been made in two of the programs reported in "Five Computer Codes for Analysis of Turbomachinery" (LEW-16851), *NASA Tech Briefs*, Vol. 23, No. 11 (November 1999), page 28. The affected programs are:

- Swift — a code for three-dimensional (3D) multiblock analysis; and
- TCGRID, which generates a 3D grid used with Swift.

Originally utilizing only a central-differencing scheme for numerical solution, Swift was augmented by addition of two upwind schemes that give greater accuracy but take more computing time. Other improvements in Swift include addition of a shear-stress-transport turbulence model for better prediction of adverse pressure gradients, addition of an H-grid capability for flexibility in modeling flows in pumps and ducts, and modification to enable simultaneous modeling of hub and tip clearances. Improvements in TCGRID include modifications to enable generation of grids for more complicated flow paths and addition of an option to generate grids compatible with the ADPAC code used at NASA and in industry. For both codes, new test cases were developed and documentation was updated. Both codes were converted to Fortran 90, with dynamic memory allocation. Both codes were also modified for ease of use in