Estimating Basic Preliminary Design Performances of Aerospace Vehicles

Aerodynamics and Performance Estimation Toolset is a collection of four software programs for rapidly estimating the preliminary design performance of aerospace vehicles represented by doing simplified calculations based on ballistic trajectories, the ideal rocket equation, and supersonic wedges through standard atmosphere. The program consists of a set of Microsoft Excel worksheet subprograms. The input and output data are presented in a user-friendly format, and calculations are performed rapidly enough that the user can iterate among different trajectories and/or shapes to perform "what-if" studies. Estimates that can be computed by these programs include:

1. Ballistic trajectories as a function of departure angles, initial velocities, initial positions, and target altitudes; assuming point masses and no atmosphere. The program plots the trajectory in two-dimensions and outputs the position, pitch, and velocity along the trajectory.
2. The "Rocket Equation" program calculates and plots the trade space for a vehicle's propellant mass fraction over a range of specific impulse and mission velocity values, propellant mass fractions as functions of specific impulses and velocities.
3. "Standard Atmosphere" will estimate the temperature, speed of sound, pressure, and air density as a function of altitude in a standard atmosphere, properties of a standard atmosphere as functions of altitude.
4. "Supersonic Wedges" will calculate the free-stream, normal-shock, oblique-shock, and isentropic flow properties for a wedge-shaped body flying supersonically through a standard atmosphere. It will also calculate the maximum angle for which a shock remains attached, and the minimum Mach number for which a shock becomes attached, all as functions of the wedge angle, altitude, and Mach number.

This work was done by Paul L. Luz and Reginald Alexander of **Marshall Space Flight Center**. *For further information, contact Caroline Wang, MSFC Software Release Authority, at (256) 544-3887 or Caroline.K.Wang@nasa.gov. Refer to MFS-31795.*

Framework for Development of Object-Oriented Software

The Real-Time Control (RTC) Application Framework is a high-level software framework written in C++ that supports the rapid design and implementation of object-oriented application programs. This framework provides built-in functionality that solves common software development problems within distributed client-server, multi-threaded, and embedded programming environments. When using the RTC Framework to develop software for a specific domain, designers and implementers can focus entirely on the details of the domain-specific software rather than on creating custom solutions, utilities, and frameworks for the complexities of the programming environment. The RTC Framework was originally developed as part of a Space Shuttle Launch Processing System (LPS) replacement project called Checkout and Launch Control System (CLCS). As a result of the framework's development, CLCS software development time was reduced by 66 percent. The framework is generic enough for developing applications outside of the launch-processing system domain. Other applicable high-level domains include command and control systems and simulation/training systems.

This software was written by Gus Perez-Poveda of 360 Software Corporation, Tony Ciavarella of United Space Alliance, and Dan Nieten of **Kennedy Space Center**. *For further information, access http://www.360SoftwareCorp.com.*

In accordance with Public Law 96-517, the contractor has elected to retain title to this invention. Inquiries concerning rights for its commercial use should be addressed to:

360 Software Corporation
12472 Lake Underhill Rd. #174
Orlando, FL 32828
Phone: (407) 694-2227

Refer to KSC-12499, volume and number of this NASA Tech Briefs *issue, and the page number.*

Analyzing Spacecraft Telecommunication Systems

Multi-Mission Telecom Analysis Tool (MMTAT) is a C-language computer program for analyzing proposed spacecraft telecommunication systems. MMTAT utilizes parameterized input and computational models that can be run on standard desktop computers to perform fast and accurate analyses of telecommunication links. MMTAT is easy to use and can easily be integrated with other software applications and run as part of almost any computational simulation. It is distributed as either a stand-alone application program with a graphical user interface or a linkable library with a well-defined set of application programming interface (API) calls. As a stand-alone program, MMTAT provides both textual and graphical output. The graphs make it possible to understand, quickly and easily, how telecommunication performance varies with variations in input parameters. A delimited text file that can be read by any spreadsheet program is generated at the end of each run. The API in the linkable-library form of MMTAT enables the user to control simulation software and to change parameters during a simulation run. Results can be retrieved either at the end of a run or by use of a function call at any time step.

This program was written by Mark Kordon, David Hanks, Roy Gladden, and Eric Wood 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 Don Hart of the California Institute of Technology at (818) 393-3425. Refer to NPO-40298.

Collaborative Planning of Robotic Exploration

The Science Activity Planner (SAP) software system includes an uplink-planning component, which enables collaborative planning of activities to be undertaken by an exploratory robot on a remote planet or on Earth. Included in the uplink-planning component is the SAP-Uplink Browser, which enables users to load multiple spacecraft activity plans into a single window, compare them, and merge them. The uplink-planning component includes a subcomponent that implements the Rover Markup Language Activity Planning format (RML-AP), based on the Extensible Markup Language (XML) format that enables the representation, within a single document, of planned spacecraft and robotic activities together with the scientific reasons for the activities. Each such document is highly parseable and can be validated easily. Another subcomponent of the uplink-planning component is the Activity Dictionary Markup Language (ADML), which eliminates the need for two mission activity dictionaries — one in a human-readable format and one in a machine-readable format. Style sheets that have been developed along with the ADML format enable users to edit one dictionary in a user-friendly environment without compromising the machine-readability of the format.

This program was written by Jeffrey Norris, Paul Backes, Mark Powell, Marsette Vona, and Robert Steinke 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 Don Hart of the California Institute of Technology at (818) 393-3425. Refer to NPO-30676.

Tools for Administration of a UNIX-Based Network

Several computer programs have been developed to enable efficient administration of a large, heterogeneous, UNIX-based computing and communication network that includes a variety of computers connected to a variety of subnetworks. One program provides secure software tools for administrators to create, modify, lock, and delete accounts of specific users. This program also provides tools for users to change their UNIX passwords and log-in shells. These tools check for errors. Another program comprises a client and a server component that, together, provide a secure mechanism to create, modify, and query quota levels on a network file system (NFS) mounted by use of the VERITAS File SystemJ software. The client software resides on an internal secure computer with a secure Web interface; one can gain access to the client software from any authorized computer capable of running web-browser software. The server software resides on a UNIX computer configured with the VERITAS software system. Directories where VERITAS quotas are applied are NFS-mounted. Another program is a Web-based, client/server Internet Protocol (IP) address tool that facilitates maintenance lookup of information about IP addresses for a network of computers.

These programs were written by Stephen LeClaire and Edward Farrar of Netlander, Inc., for **Kennedy Space Center**. *For further information, contact the Kennedy Commercial Technology Office at (321) 867-1463. KSC-12269/68/71*

Preparing and Analyzing Iced Airfoils

SmaggIce version 1.2 is a computer program for preparing and analyzing iced airfoils. It includes interactive tools for (1) measuring ice-shape characteristics, (2) controlled smoothing of ice shapes, (3) curve discretization, (4) generation of artificial ice shapes, and (5) detection and correction of input errors. Measurements of ice shapes are essential for establishing relationships between characteristics of ice and effects of ice on airfoil performance. The shape-smoothing tool helps prepare ice shapes for use with already available grid-generation and computational-fluid-dynamics software for studying the aerodynamic effects of smoothed ice on airfoils. The artificial ice-shape generation tool supports parametric studies since ice-shape parameters can easily be controlled with the artificial ice. In such studies, artificial shapes generated by this program can supplement simulated ice obtained from icing research tunnels and real ice obtained from flight test under icing weather condition. SmaggIce also automatically detects geometry errors such as tangles or duplicate points in the boundary which may be introduced by digitization and provides tools to correct these. By use of interactive tools included in SmaggIce version 1.2, one can easily characterize ice shapes and prepare iced airfoils for grid generation and flow simulations.

This program was written by Mary B. Vickerman, Marivell Baez, Donald C. Braun, Barbara J. Cotton, Yung K. Choo, Rula M. Coroneos, James A. Pennline, Anthony W. Hackenberg, and Herbert W. Schilling of **Glenn Research Center**. *John W. Slater with Yung K. Choo contributed to the conception of the code. In addition, Kevin M. Burke, Gerald J. Nolan, and Dennis Brown of InDyne, Inc., contributed to developing training material. For further information, access http://icebox-esn.grc.nasa.gov/ext/design/smaggice.html.*

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-17399.

Evaluating Performance of Components

Parallel Component Performance Benchmarks is a computer program developed to aid the evaluation of the Common Component Architecture (CCA) — a software architecture, based on a component model, that was conceived to foster high-performance computing, including parallel computing. More specifically, this program compares the performances (principally by measuring computing times) of componentized versus conventional versions of the Parallel Pyramid 2D Adaptive Mesh Refinement library — a software library that is used to generate computational meshes for solving physical problems and that is typical of software libraries in use at NASA's Jet Propulsion Laboratory.

This program was written by Daniel Katz, Edwin Tisdale, and Charles Norton 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 Don Hart of the California Institute of Technology at (818) 393-3425. Refer to NPO-30693.