

contains error trapping and input file format verification, which allows clear visibility into the input data structure and intermediate calculations

This work was done by Pedro Lopez and Winston Wang of The Boeing Co. for Johnson Space Center. For further information, contact the JSC Innovation Partnerships Office at (281) 483-3809. MSC-24963-1

Probabilistic Fatigue Damage Program (FATIG)

FATIG computes fatigue damage/fatigue life using the stress rms (root mean square) value, the total number of cycles, and S-N curve parameters. The damage is computed by the following methods: (a) traditional method using Miner's rule with stress cycles determined from a Rayleigh distribution up to $3 \times \sigma$; and (b) classical fatigue damage formula involving the Gamma function, which is derived from the integral version of Miner's rule. The integration is carried out over all stress amplitudes.

This software solves the problem of probabilistic fatigue damage using the integral form of the Palmgren-Miner rule. The software computes fatigue life using an approach involving all stress amplitudes, up to $N \times \sigma$, as specified by the user.

It can be used in the design of structural components subjected to random dynamic loading, or by any stress analyst with minimal training for fatigue life estimates of structural components.

This work was done by Constantine Michalopoulos of The Boeing Co. for Johnson Space Center. For further information, contact the JSC Innovation Partnerships Office at (281) 483-3809. MSC-24975-1

ASCENT Program

The ASCENT program solves the three-dimensional motion and attendant structural loading on a flexible vehicle incorporating, optionally, an active analog thrust control system, aerodynamic effects, and staging of multiple bodies.

ASCENT solves the technical problems of loads, accelerations, and displacements of a flexible vehicle; staging of the upper stage from the lower stage; effects of thrust oscillations on the vehicle; a payload's relative motion; the effect of fluid sloshing on vehicle; and the effect of winds and gusts on the vehicle (on the ground or aloft) in a continuous analysis.

The ATTACH ASCENT Loads program reads output from the ASCENT flexible body loads program, and calcu-

lates the approximate load indicators for the time interval under consideration. It calculates the load indicator values from pre-launch to the end of the first stage.

This work was done by Richard Brown, Gary Collier, Richard Heckenlaible, Edward Dougherty, James Dolenz, and Iain Ross of The Boeing Company for Johnson Space Center. For further information, contact the JSC Innovation Partnerships Office at (281) 483-3809. MSC-24978-1/9-1

JPL Genesis and Rapid Intensification Processes (GRIP) Portal

Satellite observations can play a very important role in airborne field campaigns, since they provide a comprehensive description of the environment that is essential for the experiment design, flight planning, and post-experiment scientific data analysis. In the past, it has been difficult to fully utilize data from multiple NASA satellites due to the large data volume, the complexity of accessing NASA's data in near-real-time (NRT), as well as the lack of software tools to interact with multi-sensor information.

The JPL GRIP Portal is a Web portal that serves a comprehensive set of NRT observation data sets from NASA and NOAA satellites describing the atmospheric and oceanic environments related to the genesis and intensification of the tropical storms in the North Atlantic Ocean. Together with the model forecast data from four major global atmospheric models, this portal provides a useful tool for the scientists and forecasters in planning and monitoring the NASA GRIP field campaign during the 2010 Atlantic Ocean hurricane season.

This portal uses the Google Earth plug-in to visualize various types of data sets, such as 2D maps, wind vectors, streamlines, 3D data sets presented at series of vertical cross-sections or pointwise vertical profiles, and hurricane best tracks and forecast tracks. Additionally, it allows users to overlap multiple data sets, change the opacity of each image layer, generate animations on the fly with selected data sets, and compare the observation data with the model forecast using two independent calendars. The portal also provides the capability to identify the geographic location of any point of interest.

In addition to supporting the airborne mission planning, the NRT data and portal will serve as a very rich source of information during the post-field campaign analysis stage of the airborne

experiment. By including a diverse set of satellite observations and model forecasts, it provides a good spatial and temporal context for the high-resolution, but limited in space and time, airborne observations.

This work was done by Brian W. Knosp, P. Peggy Li, Quoc A. Vu, Francis J. Turk, Tsae-Pyng J. Shen, Svetla M. Hristova-Velova, Stephen J. Licata, and William L. Poulsen of Caltech for NASA's Jet Propulsion Laboratory. For more information, contact iaoffice@jpl.nasa.gov.

This software is available for commercial licensing. Please contact Daniel Broderick of the California Institute of Technology at danielb@caltech.edu. Refer to NPO-47787.

Data::Downloader

Downloading and organizing large amounts of files is challenging, and often done using ad hoc methods. This software is capable of downloading and organizing files as an OpenSearch client. It can subscribe to RSS (Really Simple Syndication) feeds and Atom feeds containing arbitrary metadata, and maintains a local content addressable data store. It uses existing standards for obtaining the files, and uses efficient techniques for storing the files. Novel features include symbolic links to maintain a sane directory structure, checksums for validating file integrity during transfer and storage, and flexible use of server-provided metadata.

This work was done by Brian Duggan of Adnet Systems and Curt Tilmes of NASA for Goddard Space Flight Center. For further information, contact the Goddard Innovative Partnerships Office at (301) 286-5810. GSC-16203-1.

Fault Tolerance Middleware for a Multi-Core System

Fault Tolerance Middleware (FTM) provides a framework to run on a dedicated core of a multi-core system and handles detection of single-event upsets (SEUs), and the responses to those SEUs, occurring in an application running on multiple cores of the processor. This software was written expressly for a multi-core system and can support different kinds of fault strategies, such as introspection, algorithm-based fault tolerance (ABFT), and triple modular redundancy (TMR). It focuses on providing fault tolerance for the application code, and represents the first step in a plan to eventually include fault tolerance in message passing and the FTM itself.

In the multi-core system, the FTM resides on a single, dedicated core, separate from the cores used by the application. This is done in order to isolate the FTM from application faults and to allow it to swap out any application core for a substitute. The structure of the FTM consists of an interface to a fault tolerant strategy module, a responder module, a fault manager module, an error factory, and an error mapper that determines the severity of the error.

In the present reference implementation, the only fault tolerant strategy implemented is introspection. The introspection code waits for an application node to send an error notification to it. It then uses the error factory to create an error object, and at this time, a severity level is assigned to the error. The introspection code uses its built-in knowledge base to generate a recommended response to the error. Responses might include ignoring the error, logging it, rolling back the application to a previously saved checkpoint, swapping in a new node to replace a bad one, or restarting the application. The original error and recommended response are passed to the top-level fault manager module, which invokes the response. The responder module also notifies the introspection module of the generated response. This provides additional information to the introspection module that it can use in generating its next response. For example, if the responder triggers an application rollback and errors are still occurring, the introspection module may decide to recommend an application restart.

This work was done by Raphael R. Some, Paul L. Springer, Hans P. Zima, Mark James, and David A. Wagner of Caltech for NASA's Jet Propulsion Laboratory. For more information, contact tiaoffice@jpl.nasa.gov.

This software is available for commercial licensing. Please contact Daniel Broderick of the California Institute of Technology at danielb@caltech.edu. Refer to NPO-47806.

DspaceOgreTerrain 3D Terrain Visualization Tool

DspaceOgreTerrain is an extension to the DspaceOgre 3D visualization tool that supports real-time visualization of various terrain types, including digital elevation maps, planets, and meshes. DspaceOgreTerrain supports creating 3D representations of terrains and placing them in a scene graph. The 3D representations allow for a continuous level of detail, GPU-based ren-

dering, and overlaying graphics like wheel tracks and shadows. It supports reading data from the SimScape terrain-modeling library.

DspaceOgreTerrain solves the problem of displaying the results of simulations that involve very large terrains. In the past, it has been used to visualize simulations of vehicle traverses on Lunar and Martian terrains. These terrains were made up of billions of vertices and would not have been renderable in real-time without using a continuous level of detail rendering technique.

This work was done by Steven Myint, Abhinandan Jain, and Marc I. Pomerantz 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 Daniel Broderick of the California Institute of Technology at danielb@caltech.edu. Refer to NPO-47976.

Trick Simulation Environment 07

The Trick Simulation Environment is a generic simulation toolkit used for constructing and running simulations. This release includes a Monte Carlo analysis simulation framework and a data analysis package. It produces all auto documentation in XML. Also, the software is capable of inserting a malfunction at any point during the simulation. Trick 07 adds variable server output options and error messaging and is capable of using and manipulating wide characters for international support. Wide character strings are available as a fundamental type for variables processed by Trick.

A Trick Monte Carlo simulation uses a statistically generated, or predetermined, set of inputs to iteratively drive the simulation. Also, there is a framework in place for optimization and solution finding where developers may iteratively modify the inputs per run based on some analysis of the outputs. The data analysis package is capable of reading data from external simulation packages such as MATLAB and Octave, as well as the common comma-separated values (CSV) format used by Excel, without the use of external converters. The file formats for MATLAB and Octave were obtained from their documentation sets, and Trick maintains generic file readers for each format.

XML tags store the fields in the Trick header comments. For header files,

XML tags for structures and enumerations, and the members within are stored in the auto documentation. For source code files, XML tags for each function and the calling arguments are stored in the auto documentation. When a simulation is built, a top level XML file, which includes all of the header and source code XML auto documentation files, is created in the simulation directory. Trick 07 provides an XML to TeX converter. The converter reads in header and source code XML documentation files and converts the data to TeX labels and tables suitable for inclusion in TeX documents.

A malfunction insertion capability allows users to override the value of any simulation variable, or call a malfunction job, at any time during the simulation. Users may specify conditions, use the return value of a malfunction trigger job, or manually activate a malfunction. The malfunction action may consist of executing a block of input file statements in an action block, setting simulation variable values, call a malfunction job, or turn on/off simulation jobs.

The variable server output options and error messaging capabilities allow the variable server to return data in a tab delimited ASCII format, or in a record-based binary format. The binary record includes information about the type and size of a variable not present in the ASCII format. The binary option is capable of transmitting full C/C++ structures with one request. With this option, error messaging is returnable to client applications. In this software, the variable server may also return time synchronous data, which is gathered and sent at the end of the simulation frame to guarantee data consistency. Also, the Trick 07 variable server is capable of delivering the simulation data to multiple clients using multicast sockets. This allows multiple machines to receive the same data without increasing the computational load on the simulation.

In addition to Linux and MacOSX, Trick 07 now supports three real-time operating systems: QNX, LynxOS, and RedHawk Linux. Each RTOS has unique system calls accessing real-time features such as setting process priorities, processor assignment, and accessing the real-time clock. Trick uses the unique real-time features of each OS.

This work was done by Alexander S. Lin of Johnson Space Center and John M. Penn, Dan A. Strauss, and Keith Vetter of L-3 Communications Corporation. Further informa-