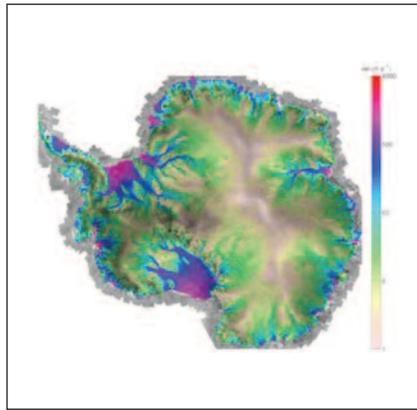


uted. It can run on the largest clusters, and is fully scalable. This allows ISSM to tackle models the size of continents.

ISSM is embedded into MATLAB and Python, both open scientific platforms. This improves its outreach within the science community. It is entirely written in C/C++, which gives it flexibility in its design, and the power/speed that C/C++ allows. ISSM is svn (subversion) hosted, on a JPL repository, to facilitate its development and maintenance.

ISSM can also model propagation of rifts using contact mechanics and mesh splitting, and can interface to the Dakota software. To carry out sensitivity analysis,



Modeled Antarctic Surface Velocity using ISSM.

mesh partitioning algorithms are available, based on the Scotch, Chaco, and Metis partitioners that ensure equal area mesh partitions can be done, which are then usable for sampling and local reliability methods.

*This work was done by Eric Larour and John E. Schiermeier of Caltech, and Helene Seroussi and Mathieu Morlinghem of Ecole Centrale Paris for NASA's Jet Propulsion Laboratory. For more information, see <http://issm.jpl.nasa.gov/>.*

*This software is available for commercial licensing. Please contact Dan Broderick at [Daniel.F.Broderick@jpl.nasa.gov](mailto:Daniel.F.Broderick@jpl.nasa.gov). Refer to NPO-48164.*

## Automated Loads Analysis System (ATLAS)

Lyndon B. Johnson Space Center, Houston, Texas

ATLAS is a generalized solution that can be used for launch vehicles. ATLAS is used to produce modal transient analysis and quasi-static analysis results (i.e., accelerations, displacements, and forces) for the payload math models on a specific Shuttle Transport System (STS) flight

using the shuttle math model and associated forcing functions. This innovation solves the problem of coupling of payload math models into a shuttle math model. It performs a transient loads analysis simulating liftoff, landing, and all flight events between liftoff and landing.

ATLAS utilizes efficient and numerically stable algorithms available in MSC/NASTRAN.

*This work was done by Stephen Gardner, Scot Frere, and Patrick O'Reilly of The Boeing Company for Johnson Space Center. For further information, contact the JSC Innovation Partnerships Office at (281) 483-3809. MSC-24987-1*

## Integrated Main Propulsion System Performance Reconstruction Process/Models

Lyndon B. Johnson Space Center, Houston, Texas

The Integrated Main Propulsion System (MPS) Performance Reconstruction process provides the MPS post-flight data files needed for post-flight reporting to the project integration management and key customers to verify flight performance. This process/model was used as the baseline for the currently ongoing Space Launch System (SLS) work.

The process utilizes several methodologies, including multiple software programs, to model integrated propulsion system performance through space shuttle ascent. It is used to evaluate integrated propulsion systems, including propellant tanks, feed systems, rocket engine, and pressurization systems performance throughout ascent based on flight pressure and temperature data. The latest re-

vision incorporates new methods based on main engine power balance model updates to model higher mixture ratio operation at lower engine power levels.

*This work was done by Eduardo Lopez, Katie Elliott, Steven Snell, and Michael Evans of The Boeing Company for Johnson Space Center. For further information, contact the JSC Innovation Partnerships Office at (281) 483-3809. MSC-25066-1*

## Phoenix Telemetry Processor

NASA's Jet Propulsion Laboratory, Pasadena, California

Phxtelemproc is a C/C++ based telemetry processing program that processes SFDU telemetry packets from the Telemetry Data System (TDS). It generates Experiment Data Records (EDRs) for several instruments includ-

ing surface stereo imager (SSI); robotic arm camera (RAC); robotic arm (RA); microscopy, electrochemistry, and conductivity analyzer (MECA); and the optical microscope (OM). It processes both uncompressed and compressed teleme-

try, and incorporates unique subroutines for the following compression algorithms: JPEG Arithmetic, JPEG Huffman, Rice, LUT3, RA, and SX4.

This program was in the critical path for the daily command cycle of the

Phoenix mission. The products generated by this program were part of the RA commanding process, as well as the SSI, RAC, OM, and MECA image and science analysis process. Its output products were used to advance science of the near polar regions of Mars, and were used to prove that water is found in abundance there.

Phxtelemproc is part of the MIPL (Multi-mission Image Processing Laboratory) system. This software produced Level 1 products used to analyze images returned by *in situ* spacecraft. It ultimately assisted in operations, planning, commanding, science, and outreach.

*This work was done by Alice Stanboli of Caltech for NASA's Jet Propulsion Laboratory. For more information, contact [iaoffice@jpl.nasa.gov](mailto:iaoffice@jpl.nasa.gov).*

*This software is available for commercial licensing. Please contact Dan Broderick at [Daniel.F.Broderick@jpl.nasa.gov](mailto:Daniel.F.Broderick@jpl.nasa.gov). Refer to NPO-47801.*

---

## Contact Graph Routing Enhancements Developed in ION for DTN

*NASA's Jet Propulsion Laboratory, Pasadena, California*

The Interplanetary Overlay Network (ION) software suite is an open-source, flight-ready implementation of networking protocols including the Delay/Disruption Tolerant Networking (DTN) Bundle Protocol (BP), the CCSDS (Consultative Committee for Space Data Systems) File Delivery Protocol (CFDP), and many others including the Contact Graph Routing (CGR) DTN routing system. While DTN offers the capability to tolerate disruption and long signal propagation delays in transmission, without an appropriate

routing protocol, no data can be delivered.

CGR was built for space exploration networks with scheduled communication opportunities (typically based on trajectories and orbits), represented as a contact-graph. Since CGR uses knowledge of future connectivity, the contact graph can grow rather large, and so efficient processing is desired. These enhancements allow CGR to scale to predicted NASA space network complexities and beyond.

This software improves upon CGR by adopting an earliest-arrival-time cost met-

ric and using the Dijkstra path selection algorithm. Moving to Dijkstra path selection also enables construction of an earliest-arrival-time tree for multicast routing. The enhancements have been rolled into ION 3.0 available on sourceforge.net.

*This work was done by John S. Segui and Scott Burleigh of Caltech for NASA's Jet Propulsion Laboratory. For more information, contact [iaoffice@jpl.nasa.gov](mailto:iaoffice@jpl.nasa.gov).*

*This software is available for commercial licensing. Please contact Dan Broderick at [Daniel.F.Broderick@jpl.nasa.gov](mailto:Daniel.F.Broderick@jpl.nasa.gov). Refer to NPO-48186.*

---

## GFECchutes Lo-Fi

*Lyndon B. Johnson Space Center, Houston, Texas*

NASA needed to provide a software model of a parachute system for a manned re-entry vehicle. NASA has parachute codes, e.g., the Descent Simulation System (DSS), that date back to the Apollo Program. Since the space shuttle did not rely on parachutes as its primary descent control mechanism, DSS has not been maintained or incorporated into modern simulation architectures such as Osiris and Antares, which are used for new mission simulations. GFECchutes Lo-Fi is an object-oriented implementation of conventional parachute codes designed for use in modern simulation environments.

The GFE (Government Furnished Equipment), low-fidelity (Lo-Fi) parachute model (GFECchutes Lo-Fi) is a software package capable of modeling the effects of multiple parachutes, deployed concurrently and/or sequentially, on a vehicle during the subsonic phase of re-entry into planetary atmosphere. The term "low-fidelity" distinguishes models

that represent the parachutes as simple forces acting on the vehicle, as opposed to independent aerodynamic bodies. GFECchutes Lo-Fi was created from these existing models to be clean, modular, certified as NASA Class C software, and portable, or "plug and play."

The GFE Lo-Fi Chutes Model provides basic modeling capability of a sequential series of parachute activities. Actions include deploying the parachute, changing the reefing on the parachute, and cutting away the parachute. Multiple chutes can be deployed at any given time, but all chutes in that case are assumed to behave as individually isolated chutes; there is no modeling of any interactions between deployed chutes. Drag characteristics of a deployed chute are based on a coefficient of drag, the face area of the chute, and the local dynamic pressure only. The orientation of the chute is approximately modeled for purposes of obtaining torques on the vehicle, but the dy-

amic state of the chute as a separate entity is not integrated — the treatment is simply an approximation.

The innovation in GFECchutes Lo-Fi is to use an object design that closely followed the mechanical characteristics and structure of a physical system of parachutes and their deployment mechanisms. Software objects represent the components of the system, and use of an object hierarchy allows a progression from general component outlines to specific implementations. These extra chutes were not part of the baseline deceleration sequence of drogues and mains, but still had to be simulated. The major innovation in GFECchutes Lo-Fi is the software design and architecture.

*This work was done by Emily Gist, Gary Turner, Robert Shelton, Mana Vautier, and Ashraf Shaikh of Odyssey Space Research, LLC for Johnson Space Center. Further information is contained in a TSP (see page 1). MSC-25004-1*