Software

Integrated Modeling Environment

The Integrated Modeling Environment (IME) is a software system that establishes a centralized Web-based interface for integrating people (who may be geographically dispersed), processes, and data involved in a common engineering project. The IME includes software tools for life-cycle management, configuration management, visualization, and collaboration. It enables organized, efficient communication of engineering analyses and the statuses thereof. Key functions performed by use of the IME include creation, further development, and management of modeling analyses over their entire life cycles; publishing model and analysis information for availability and reuse throughout the user community; and managing legacy information without regard to original formats, database organizations, or computing platforms. The use of the IME creates an archive of analysis results, plus documentation that identifies the assumptions and data elements used for each analysis. This archive is configured to enable reuse of previous analysis results, and tracing of types and versions of software used for each step of each analysis. The IME utilizes a customized version of a commercial product-life-cycle-management app-lication program that provides rich capabilities for managing configurations, workflows, data, and access through a single Web-based environment.

This program was written by Gary Mosier of **Goddard Space Flight Center***, and Paul Stone and Christopher Holtery of Constellation Software Engineering Corp. Further information is contained in a TSP (see page 1).*
GSC-14827-1

Modified Recursive Hierarchical Segmentation of Data

An algorithm and a computer program that implements the algorithm that performs recursive hierarchical segmentation (RHSEG) of data have been developed. While the current implementation is for two-dimensional data having spatial characteristics (e.g., image, spectral, or spectral-image data), the generalized algorithm also applies to three-dimensional or higher dimensional data and also to data with no spatial characteristics. The algorithm and software are modified versions of a prior RHSEG algorithm and software, the outputs of which often contain processing-window artifacts including, for example, spurious segmentation-image regions along the boundaries of processing-window edges. The modification consists of the addition of an efficient subroutine through which pairs of regions are identified that may contain pixels that are actually more similar to other regions in the pair. Once these pairs of regions are identified, pixels in one region that are more similar to pixels in the other region are reassigned to the other region. The subroutine is computationally efficient because it focuses only on those regions that could potentially contribute to the processing-window artifacts. In addition, any adverse effect of the subroutine on the computational efficiency of the algorithm is minimized by executing the subroutine at a point in the algorithm such that switching of pixels between regions that are subsequently merged is avoided.

This program was written by James C. Tilton of **Goddard Space Flight Center***. For further information, contact the Goddard Innovative Partnerships Office at (301) 286-5810. GSC-14681-1*

Sizing Structures and Predicting Weight of a Spacecraft

EZDESIT is a computer program for choosing the sizes of structural components and predicting the weight of a spacecraft, aircraft, or other vehicle. In designing a vehicle, EZDESIT is used in conjunction with a finite-element structural-analysis program: Each structural component is sized within EZDESIT to withstand the loads expected to be encountered during operation, then the weights of all the structural finite elements are added to obtain the structural weight of the vehicle. The sizing of the structural components elements also alters the stiffness properties of the finite-element model. The finite-element analysis and structural component sizing are iterated until the weight of the vehicle converges to a prescribed iterative difference. The results of the sizing can be reviewed in two ways:

1. An interactive session of the EZDESIT program enables review of the results in a table that shows component types, component weights, and failure modes; and

2. The results are read into a finite-element preprocessing-and-postprocessing program and displayed on a graphical representation of the model.

This program was written by Jeffrey Cerro and C. P. Shore of **Langley Research Center***. Further information is contained in a TSP (see page 1).*
LAR-16878-1

Stress Testing of Data-Communication Networks

NetStress is a computer program that stress-tests a data-communication network and components thereof. NetStress comprises two components running, respectively, in a transmitting system and a receiving system connected to a network under test. The novelty of the program is that is has the capability to generate/receive varied network loading traffic profiles, which prior known programs were incapable of producing (i.e., various packet sizes and various packet rates all combined to make a pseudo-random traffic pattern). The transmitting-system component generates increasingly stressful data traffic for transmission via the network. The receiving-system component analyzes the resulting traffic arriving in the receiving system, generating such statistics as the number of data packets successfully received, the number of dropped packets, and the number of packets received out of order. The packet sizes must be configured before the transmitting-system component is started, but the packet frequencies, numbers of packets in bursts, and burst times can be configured during execution. Typically, a test begins with transmission of data at low sustained rates. Then the sustained rates are increased and burst rates are modified while monitoring to determine whether the receiving-system component reports any losses. When significant losses are reported, the user seeks to determine whether a malfunction or defi-