Software

Active Mirror Predictive and Requirements Verification Software (AMP-ReVS)

This software is designed to predict large active mirror performance at various stages in the fabrication lifecycle of the mirror. It was developed for 1-meter class powered mirrors for astronomical purposes, but is extensible to other geometries. The package accepts finite element model (FEM) inputs and laboratory measured data for large optical-quality mirrors with active figure control. It computes phenomenological contributions to the surface figure error using several built-in optimization techniques. These phenomena include stresses induced in the mirror by the manufacturing process and the support structure, the test procedure, high spatial frequency errors introduced by the polishing process, and other process-dependent deleterious effects due to light-weighting of the mirror. Then, depending on the maturity of the mirror, it either predicts the best surface figure error that the mirror will attain, or it verifies that the requirements for the error sources have been met once the best surface figure error has been measured.

The unique feature of this software is that it ties together physical phenomenology with wavefront sensing and control techniques and various optimization methods including convex optimization, Kalman filtering, and quadratic programming to both generate predictive models and to do requirements verification. This software combines three distinct disciplines: wavefront control, predictive models based on FEM, and requirements verification using measured data in a robust, reusable code that is applicable to any large optics for ground and space telescopes.

The software also includes state-of-the-art wavefront control algorithms that allow closed-loop performance to be computed. It allows for quantitative trade studies to be performed for optical systems engineering, including computing the best surface figure error under various testing and operating conditions. After the mirror manufacturing process and testing have been completed, the software package can be used to verify that the underlying requirements have been met.

This work was done by Scott A. Basinger 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-47667.

Navigation/Prop Software Suite

Navigation (Nav)/Prop software is used to support shuttle mission analysis, production, and some operations tasks. The Nav/Prop suite containing configuration items (CIs) resides on IPS/Linux workstations. It features lifecycle documents, and data files used for shuttle navigation and propellant analysis for all flight segments. This suite also includes trajectory server, archive server, and RAT software residing on MCC/Linux workstations.

Navigation/Prop represents tool versions established during or after IPS Equipment Rehost-3 or after the MCC Rehost.

This work was done by Tomas Bruchmiller, Sanh Tran, Mathew Lee, Scott Bucker, Catherine Bupane, Charles Bennett, Sergio Cantu, Ping Kwong, and Carolyn Propst of the United Space Alliance for Johnson Space Center. For further information, contact the JSC Innovation Partnerships Office at (281) 483-3809. MSC-24957-1

Personal Computer Transport Analysis Program

The Personal Computer Transport Analysis Program (PCTAP) is C++ software used for analysis of thermal fluid systems. The program predicts thermal fluid system and component transients. The output consists of temperatures, flow rates, pressures, delta pressures, tank quantities, and gas quantities in the air, along with air scrubbing component performance.

PCTAP's solution process assumes that the tubes in the system are well insulated so that only the heat transfer between fluid and tube wall and between adjacent tubes is modeled. The system described in the model file is broken down into its individual components; i.e., tubes, cold plates, heat exchangers, etc. A solution vector is built from the components and a flow is then simulated with fluid being transferred from one component to the next. The solution vector of components in the model file is built at the initiation of the run. This solution vector is simply a list of components in the order of their inlet dependency on other components. The component parameters are updated in the order in which they appear in the list at every time step. Once the solution vectors have been determined, PCTAP cycles through the components in the solution vector, executing their outlet function for each time-step increment.

This work was done by Frank DiStefano III, Craig Wobick, Kirt Chapman, and Peter McCloud of The Boeing Company for Johnson Space Center. For further information, contact the JSC Innovation Partnerships Office at (281) 483-3809.

Title to this invention has been waived under the provisions of the National Aeronautics and Space Act {42 U.S.C. 2457(f)}, to The Boeing Company. Inquiries concerning licenses for its commercial development should be addressed to:

Terrance.Mason@Boeing.com or
Phone No.: (562) 797-9034

Refer to Boeing ID No. 10-0614 & MSC-24971-1, volume and number of this NASA Tech Briefs *issue, and the page number.*

Pressure Ratio to Thermal Environments

A pressure ratio to thermal environments (PRatTlE.pl) program is a Perl language code that estimates heating at requested body point locations by scaling the heating at a reference location times a pressure ratio factor. The pressure ratio factor is the ratio of the local pressure at the reference point and the requested point from CFD (computational fluid dynamics) solutions.

This innovation provides pressure ratio-based thermal environments in an automated and traceable method. Previously, the pressure ratio methodology was implemented via a Microsoft Excel spreadsheet and macro scripts. PRatTlE is able to calculate heating environments for 150 body points in less than two minutes.

PRatTlE is coded in Perl programming language, is command-line-driven, and has been successfully executed on both the HP and Linux platforms. It supports multiple concurrent runs. PRatTlE