It adds new features to code automatically, such as recovering from a lost node or the ability to modify the code while running.

In this project, the innovators at the time of this reporting intend to develop two distinct technologies that build upon each other and both of which serve as building blocks for more efficient HPC usage. First is the scheduling and dynamic execution framework, and the second is scalable linear algebra libraries that are built directly on the former.

This work was done by John Humphrey and Kyle Spagnoli of EM Photonics, Inc. for Goddard Space Flight Center. Further information is contained in a TSP (see page 1). GSC-16472-1

Deepak Condenser Model (DeCoM)

Goddard Space Flight Center, Greenbelt, Maryland

Development of the DeCoM comes from the requirement of analyzing the performance of a condenser. A component of a loop heat pipe (LHP), the condenser, is interfaced with the radiator in order to reject heat. DeCoM simulates the condenser, with certain input parameters. Systems Improved Numerical Differencing Analyzer (SINDA), a thermal analysis software, calculates the adjoining component temperatures, based on the DeCoM parameters and interface temperatures to the radiator. Application of DeCoM is (at the time of this reporting) restricted to small-scale analysis, without the need for in-depth LHP component integrations. To efficiently develop a model to simulate the LHP condenser, DeCoM was developed to meet this purpose with least complexity. DeCoM is a single-condenser, single-pass simulator for analyzing its behavior. The analysis is done based on the interactions between condenser fluid, the wall, and the interface between the wall and the radiator.

DeCoM is based on conservation of energy, two-phase equations, and flow equations. For two-phase, the Lockhart-Martinelli correlation has been used in order to calculate the convection value between fluid and wall. Software such as SINDA (for thermal analysis analysis) and Thermal Desktop (for modeling) are required. DeCoM also includes the ability to implement a condenser into a thermal model with the capability of understanding the code process and being edited to user-specific needs. DeCoM requires no license, and is an open-source code. Advantages to DeCoM include time dependency, reliability, and the ability for the user to view the code process and edit to their needs.

This work was done by Deepak Patel of Goddard Space Flight Center. Further information is contained in a TSP (see page 1). GSC-16296-1

Flight Software Math Library

Goddard Space Flight Center, Greenbelt, Maryland

The flight software (FSW) math library is a collection of reusable math components that provides typical math utilities required by spacecraft flight software. These utilities are intended to increase flight software quality reusability and maintainability by providing a set of consistent, well-documented, and tested math utilities. This library only has dependencies on ANSI C, so it is easily ported.

Prior to this library, each mission typically created its own math utilities using ideas/code from previous missions. Part of the reason for this is that math libraries can be written with different strategies in areas like error handling, parameters orders, naming conventions, etc. Changing the utilities for each mission introduces risks and costs. The obvious risks and costs are that the utilities must be coded and revalidated. The hidden risks and costs arise in miscommunication between engineers. These utilities must be understood by both the flight software engineers and other subsystem engineers (primarily guidance navigation and control).

The FSW math library is part of a larger goal to produce a library of reusable Guidance Navigation and Control (GN&C) FSW components. A GN&C FSW library cannot be created unless a standardized math basis is created. This library solves the standardization problem by defining a common feature set and establishing policies for the library's design. This allows the libraries to be maintained with the same strategy used in its initial development, which supports a library of reusable GN&C FSW components.

The FSW math library is written for an embedded software environment in C. This places restrictions on the language features that can be used by the library. Another advantage of the FSW math library is that it can be used in the FSW as well as other environments like the GN&C analyst's simulators. This helps communication between the teams because they can use the same utilities with the same feature set and syntax.

This work was done by David McComas of Goddard Space Flight Center. Further information is contained in a TSP (see page 1). GSC-16102-1