A computer program known as “PyPele,” originally written as a Python-language extension module of a C++ language program, has been rewritten in pure Python language. The original version of PyPele dispatches and coordinates parallel-processing tasks on cluster computers and provides a conceptual framework for spacecraft-mission-design and analysis software tools to run in an embarrassingly parallel mode. The original version of PyPele uses SSH (Secure Shell — a set of standards and an associated network protocol for establishing a secure channel between a local and a remote computer) to coordinate parallel processing. Instead of SSH, the present Python version of PyPele uses Message Passing Interface (MPI) [an unofficial de-facto standard language-independent application programming interface for message-passing on a parallel computer] while keeping the same user interface.

The use of MPI instead of SSH and the preservation of the original PyPele user interface make it possible for parallel application programs written previously for the original version of PyPele to run on MPI-based cluster computers. As a result, engineers using the previously written application programs can take advantage of embarrassing parallelism without need to rewrite those programs.

This program was written by George Hockney and Seungwon Lee of Caltech for NASA’s Jet Propulsion Laboratory.

This software is available for commercial licensing. Please contact Karina Edmonds of the Telecommunication Systems Directorate, 4500 NASA Parkway, Building 1103, Room 233, Stennis Space Center, MS 39529, (626) 395-2322. Refer to NASA Tech Briefs issue and the page number.

**Data Assimilation Cycling for Weather Analysis**

This software package runs the atmospheric model MM5 in a data assimilation cycling mode to produce an optimized weather analysis, including the ability to insert or adjust a hurricane vortex. The program runs MM5 through a cycle of short forecasts every three hours where the vortex is adjusted to match the observed hurricane location and storm intensity. This technique adjusts the surrounding environment so that the proper steering current and environmental shear are achieved. MM5cycle uses a Cressman analysis to blend observation into model fields to get a more accurate weather analysis. Quality control of observations is also done in every cycle to remove bad data that may contaminate the analysis. This technique can assimilate and propagate data in time from intermittent and infrequent observations while maintaining the atmospheric field in a dynamically balanced state.

The software consists of a C-shell script (MM5cycle.driver) and three FORTRAN programs (splitMM5files.F, comRegrid.F, and insert_vortex.F), and are contained in the pre-processor component of MM5 called “Regriddr.” The model is first initialized with data from a global model such as the Global Forecast System (GFS), which also provides lateral boundary conditions. These data are separated into single-time files using splitMM5.F. The hurricane vortex is then bogussed in the correct location and with the correct wind field using insert_vortex.F. The modified initial and boundary conditions are then recombined into the model fields using comRegrid.F. The model then makes a three-hour forecast. The three-hour forecast data from MM5 now become the analysis for the next short forecast run, where the vortex will again be adjusted. The process repeats itself until the desired time of analysis is achieved. This code can also assimilate observations if desired.

This program was written by Nam Tran, Yongzuo Li, and Patrick Fitzpatrick of the GeoResources Institute at Mississippi State University for Stennis Space Center.

Inquiries concerning rights for its commercial use should be addressed to:

GeoResources Institute
Mississippi State University
Building 1103, Room 233
Stennis Space Center, MS 39529
Phone No.: (228) 688-4218
Web site: www.gri.msstate.edu

Refer to NASA Tech Briefs issue and the page number.

---

**PyPele Rewritten To Use MPI**

A computer program known as "PyPele," originally written as a Python-