Parallel Implementation of the Ensemble Empirical Mode Decomposition and Its Application for Earth Science Data Analysis

Bo-Wen Shen | San Diego State University
Samson Cheung | NASA Ames Research Center
Yu-Ling Wu | University of Alabama in Huntsville
Jui-Lin F. Li | Jet Propulsion Laboratory/NASA and California Technology Institute
David Kao | NASA Ames Research Center

The parallel ensemble empirical mode decomposition (PEEMD) package efficiently performs multiscale analysis of high-resolution, global, multiple-dimensional datasets by implementing three-level parallelism into the ensemble empirical mode decomposition (EMD), achieving a scaled performance of 5,000 cores.

The core technology of the parallel ensemble empirical mode decomposition (PEEMD; see the “Glossary” sidebar for a complete list of acronyms used in this article and their definitions) is the empirical mode decomposition (EMD) and ensemble EMD (EEMD) methods, which show remarkable performance in revealing multiscale processes of the data that are non-stationary and nonlinear. The EMD method, which was invented by Norden E. Huang and his colleagues in the late 1990s, has been widely applied in many different fields.\(^1\) It decomposes one set of observational data into the so-called intrinsic mode functions (IMFs), which represent the oscillatory components of the data. The EEMD was developed to overcome the scale mixing (or mode mixing) problem of the EMD.\(^2\) By introducing finite-amplitude white noise, the EEMD deals with an ensemble of datasets. Each data member, which contains the original observational data and finite-amplitude white noise, is decomposed by the EMD to obtain a set of IMFs. The final IMFs are determined as the ensemble average of the multiple sets of IMFs, each of which corresponds to one data member. The EEMD significantly reduces the chance of scale mixing, and it still preserves a filter bank property,\(^3,4\) that is, the decomposed mean IMFs staying within the natural filter period windows.

Consequently, the computational resources required for an EEMD run are tremendous, linearly proportional to the number of ensemble trials. Furthermore, applying the 1D EEMD to analyzing multidimensional data or a variety of fields at multiple vertical levels could require substantial computational time. Thus, a parallel version of the EEMD with a three-level parallelism has been developed to reduce the time required for data decomposition.

The EMD, EEMD, and PEEMD Methods

With the EMD, we can extract the IMFs from the time (or spatial) series of the raw data \(X(j)\) and represent it by the first \(m\) IMFs and the residual of data \(R_m\) as follows:

\[
X(j) = \sum_{i=1}^{m} C_i(j) + R_m(j),
\]

(1)
where $X(j)$ represents the value of $X$ at $t = j\Delta t$ or $x = j\Delta x$ (here $\Delta t$ and $\Delta x$ are the temporal and spatial increments, respectively), and $C_i(j)$ represents the $i$th IMF. Next, we introduce the method and processes for EMD/EEMD for calculating IMFs. The following notation is adopted:

- the index $i$ used to indicate the $i$th IMF;
- $t$ as the independent variable that can be time, longitude, or latitude;
- $X(t)$ as the original data;
- $S(t)$ as the data to be shifted out;
- $C_i(t)$ as the $i$th IMF;
- $R(t)$ as the residual of the data; and
- $U(t)$ ($L(t)$) as the upper (lower) envelope of the given data by connecting maxima (minima) with a cubic spline.

Therefore, $R(t) = X(t)$, and $S(t) = R(t)$ as $i = 0$, initially. $S(t)$ and $R(t)$ are changed after the beginning of the so-called shifting processes that are summarized as follows:

1. $i = 0$ and $R(t) = X(t)$; here, $X(t)$ is the original data.
2. Let $S(t) = R(t)$.
3. Identify the extrema (both maxima and minima) of the data $S(t)$.
4. Generate the upper (lower) envelope, $U(t)$ ($L(t)$).
5. Determine the local mean, $m(t) = U(t) + L(t)$, by averaging the envelopes.
6. $S(t) = S(t) - m(t)$, that is, subtract out the mean from the data.
7. Go to step 3 unless stoppage criteria for an IMF are met.
8. $i = i + 1$, $C_i(t) = S(t)$, $R(t) = R(t) - C_i(t)$, then go to step 2.

For each IMF, we performed 10 iterations to obtain the responding IMF, that is, step 2 to step 7 repeated 10 times. Based on the above sifting processes, IMFs possess the following features. The IMFs are symmetric with respect to the local zero mean and have the same number of zero crossings and extrema. The IMFs are time (spatial) domain functions that represent the local variability of the original signal at a particular range of frequencies (wavelengths). Mathematically, a strict definition of the IMF is as follows: $U(t) + L(t) = 0$, namely, its local means are zero.\(^5\)

Using the above processes, the IMFs and residual functions are defined at the same grid points as those for the raw data. The residual, $R_m$, represents the differences between the raw data and a sum of the first $m$ IMFs. Similarly, $X(j)$ can also be represented by the first $m + 1$ IMFs and $R_{m+1}$, as follows:

$$X(j) = \sum_{i=1}^{m+1} C_i(j) + R_{m+1}(j). \quad (2)$$

Equations 1 and 2 lead to

$$R_i(j) = R_{i+1}(j) + C_{i+1}(j).$$

Because $C_{i+1}(j)$ is purely oscillatory with respect to $R_{i+1}(j)$, we can consider $R_{i+1}(j)$ as the local mean of $R_i(j)$. Thus, EMD is a “Reynolds type” decomposition for sifting out (or extracting) periodic components from the data by separating the local mean from the fluctuations using spline fits.\(^1\)

For the ensemble EMD, the following steps were performed:

- Add a noise series to the targeted data.
- Decompose the data with added noise to IMFs using the EMD.
- Iterate steps 1 and 2 repeatedly but with a different noise series each time.
- Obtain the (ensemble) means of the corresponding IMFs of the decompositions as the final result.
With the EEMD, the chance of scale mixing is reduced while the dyadic property is preserved. As indicated elsewhere, for the mean IMFs, added noises cancel out one another.

Parallel Implementation of the EEMD

In this study, performance results were obtained from the NASA Ames Pleiades supercomputer. The Pleiades supercomputer is an SGI Altix ICE system with a peak performance of 4.97 Pflop/s. With 663.6 Tbytes of total memory and 185,344 cores, it achieves Linpack performance of 4.089 Pflop/s (as of November 2015). The system contains the following types of Intel Xeon processors: E5-2680v4 (Broadwell), E5-2680v3 (Haswell), E5-2680v2 (Ivy Bridge), E5-2670 (Sandy Bridge), and X5670 (Westmere) to reach different needs and capacities on different NASA projects. Each node (physical blade unit) has 2 sockets, and each socket has an 8-core Sandy Bridge or 10-core Ivy Bridge CPU. Thus, each node has 16 Sandy Bridge cores or 20 Ivy Bridge cores. While Sandy Bridge cores (E5-2670) are used in the first two benchmarks, Ivy Bridge cores (E5-2680v2) are used in the third.

In the PEEMD for this study, a three-level parallelism is achieved as follows: decompose data spatial domains into subdomains, divide ensemble members into different groups, and split a big loop regarding the shifting processes into several chunks. The first two levels of parallelism use message-passing interface (MPI) processes to perform tasks in parallel, while the third-level parallelism uses OpenMP multithreads to perform calculations in parallel. In this section, we discuss three benchmarks to compare the computational performance of the first-level parallelism with MPI processes, the fine-grained parallelism with OpenMP threads, and the whole three-level parallelism with a hybrid of MPI processes and OpenMP threads, respectively. Two datasets used in this study are Extended Reconstructed Sea Surface Temperature (ERSST), for the first two benchmarks and an idealized solution of the Mixed Rossby-gravity (MRG) for the third. Computational results with the three benchmarks are discussed below.

In first-level parallelism, a spatial (global or limited-area) domain is divided into \( M \) different subdomains with 1D or 2D domain decomposition, running with \( M \) MPI processes in parallel. The pseudo codes for first-level parallelism, which is performed in the spatial domain, are shown in Listing 1. Here, we present the performance of PEEMD code in analyzing the ERSST data, which is on a 2D spatial grid with 180 and 89 grid points in longitude and latitude, respectively. The 2D domain is decomposed into \( M \) subdomains by dividing the longitude interval into \( I \) subintervals and the latitude interval into \( J \) subintervals, where \( M = I \times J \). Table 1 shows the performance of the code in processing the ERSST data using the first parallelism with various values of \((I, J)\). The second row with \((I, J) = (I, I)\) indicates the serial run used as the baseline for the speedup calculation. The

Listing 1. Pseudo codes for the first level parallelism of the PEEMD.
remaining rows show the speedup using only first-level MPI parallelization. Since data size is small for this case, only a small number of MPI processes was used. As Table 1 shows, the overall scalability is good with a speedup of 10.1 using 16 cores, yielding a parallel efficiency of 0.63.

In each subdomain decomposed by first-level parallelism, second-level parallelism is deployed to decompose the total number of ensemble trials into $N$ groups, running with $N$ MPI processes in parallel. Then, third-level parallelism is applied to the fine-grain OpenMP threads within each of the $N$ processes spawned by each $M$ task. To examine the performance of OpenMP threading, Table 2 shows the numerical performance of one Sandy Bridges node, which has 16 cores, using various number of OpenMP threads. While the number of the MPI processes is fixed at four, which is used to limit the communication among the MPI processors, the number of threads is increased from one to two to four to examine the performance of the OpenMP threading. The second column of Table 2 that shows the time in seconds for the calculation displays the speedup of 1.82 and 3.08 using two and four threads, respectively. In this calculation, a traditional Box-Muller transformation is employed in the random number generation routine. Having realized the disadvantages of Box-Muller transformation, random number generation of the code has been greatly improved by using Intel’s Vector Statistical Library (VSL), which is part of the Intel Math Kernel Library (MKL, https://software.intel.com/en-us/articles/intel-math-kernel-library-documentation). The third column of Table 2, showing the time in seconds for the calculation, indicates the improved scalability, with a speedup of 1.97 and 3.81 using two and four threads, respectively.

As our ultimate goal is to apply the PEEMD to analyze global model simulations at a resolution of 1/4 degree or higher, the performance of the three-level parallelism in the PEEMD is first discussed using the idealized MRG wave solution with a number of grid points comparable to that at the target model resolution. The number of grid points is $1,001 \times 1,001$ for the MRG wave case, while the global weather/climate model at a resolution of 0.25 degree has $1,001 \times 721$ grid points. The PEEMD with such a grid resolution required 400 Mbytes of memory per variable. Table 3 presents computational results from the PEEMD with three-level parallelism using up to 5,000 Ivy Bridges cores. The first two columns of the table show the values of $(I, J)$ that are used to divide the 2D domain into $I \times J$ subdomains, giving a total number of subdomains $M = I \times J$. The third column (Ens) in the table displays the number of MPI processes for second-level parallelism applied in the ensemble space, while the fourth column (OMP) shows a number of OpenMP threads in each MPI process in second-level parallelism. We selected the case with 60 cores as the baseline because of required memory in this specific application. As shown in Table 3, the PEEMD is scalable up to 5,000 cores. In particular, we obtained a parallel speedup and efficiency of 52.8 and 63 percent, respectively, by increasing the number of cores from 60 to 5,000. Figure 1 displays the log-log plot of the scaling as a function of the number of cores.

In summary, major features of the newly developed PEEMD include promising scalability that yields a parallel speedup and efficiency of 52.8 and 63 percent by increasing the number of cores from 60 to 5,000; bit-by-bit consistency that assures the binary identical results with different CPU layouts; and sustainability enabled by a well-designed interface, making it easy to plug new versions of the EMD into the software package of the PEEMD. The parallel performance and implementation of an early version of the PEEMD and its application were discussed elsewhere. The scientific performance of the PEEMD is discussed in the next section.

### Table 1. PEEMD timing with first-level parallelism in processing ERSST data.

<table>
<thead>
<tr>
<th>$I$</th>
<th>$J$</th>
<th>Timing (s)</th>
<th>Speedup</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>863.88</td>
<td>1.0</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>435.78</td>
<td>1.9824</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>257.90</td>
<td>3.3497</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>197.41</td>
<td>4.3760</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>142.43</td>
<td>6.0652</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>144.64</td>
<td>5.9727</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>111.63</td>
<td>7.7384</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>85.53</td>
<td>10.0997</td>
</tr>
</tbody>
</table>

### Table 2. Parallel performance of the PEEMD associated with various numbers of OpenMP threads.

<table>
<thead>
<tr>
<th>OMP</th>
<th>Original</th>
<th>Vector Statistical Library</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>146.52</td>
<td>135.92</td>
</tr>
<tr>
<td>2</td>
<td>80.32</td>
<td>68.94</td>
</tr>
<tr>
<td>4</td>
<td>47.55</td>
<td>35.66</td>
</tr>
</tbody>
</table>
Application for the Analysis of Earth Science Data

To verify the PEEMD’s scientific performance, we use it to analyze the following three meteorological datasets: a chaotic solution of the Lorenz model, an idealized Kelvin cat’s eye flow, and Hurricane Sandy.

Lorenz Model

The first case is designed to understand PEEMD performance by computing a nonlinear solution’s corresponding growth rate and period at different stages. To achieve our goal, we apply the tool to an envelope analysis of solutions near a nontrivial critical point in the Lorenz model. The Lorenz model with three state variables, which is referred to as the 3DLM, is first linearized with respect to its nontrivial critical point solutions as follows:

\[
\frac{dX'}{d\tau} = -\sigma X' + \sigma Y',
\]

\[
\frac{dY'}{d\tau} = (\tau - Z_c)X' - Y' - X'Z' - FN(X'Y'),
\]

\[
\frac{dZ'}{d\tau} = Y'X' + X'Y' - bZ' + FN(X'Y'),
\]

where \(\tau\) is the dimensionless time. We use \(\tau = 110\) and \(b = 8/3\) in this study; \(\sigma\) and \(\tau\) are Prandtl and normalized Rayleigh numbers, which are set to 10 and 25, respectively. The flag \(FN\) indicates whether the system is fully nonlinear \((FN = 1)\) or “linear” \((FN = 0)\). Each variable with a prime \((X')\) indicates perturbation with respect to its critical point \((X_c)\), and the total field is written as \(X = X_c + X'\). In the above equations, the choice of critical point solutions as the basic state leads to the disappearance of the nonlinear terms that involve the interaction of two basic state variables \((X_c\) and \(Y_c\)).

Table 3. PEEMD performance with full three-level parallelism in processing the MRG wave case with up to 5,000 Ivy Bridges cores used.

<table>
<thead>
<tr>
<th>(I)</th>
<th>(J)</th>
<th>Ens</th>
<th>OMP</th>
<th>Total cores</th>
<th>Time (s)</th>
<th>Speedup</th>
<th>Parallel Efficiency</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>6</td>
<td>2</td>
<td></td>
<td>60</td>
<td>6543.56</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td>2</td>
<td></td>
<td>200</td>
<td>1983.25</td>
<td>3.3</td>
<td>0.99</td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td>4</td>
<td></td>
<td>400</td>
<td>1021.1</td>
<td>6.4</td>
<td>0.96</td>
</tr>
<tr>
<td>20</td>
<td>20</td>
<td>2</td>
<td></td>
<td>800</td>
<td>531.36</td>
<td>12.3</td>
<td>0.92</td>
</tr>
<tr>
<td>20</td>
<td>20</td>
<td>4</td>
<td></td>
<td>1,600</td>
<td>289.42</td>
<td>22.6</td>
<td>0.85</td>
</tr>
<tr>
<td>25</td>
<td>40</td>
<td>2</td>
<td></td>
<td>2,000</td>
<td>231.69</td>
<td>28.2</td>
<td>0.85</td>
</tr>
<tr>
<td>25</td>
<td>25</td>
<td>4</td>
<td></td>
<td>2,500</td>
<td>200.6</td>
<td>32.6</td>
<td>0.78</td>
</tr>
<tr>
<td>25</td>
<td>25</td>
<td>4</td>
<td>2</td>
<td>5,000</td>
<td>129.68</td>
<td>50.4</td>
<td>0.60</td>
</tr>
<tr>
<td>50</td>
<td>50</td>
<td>2</td>
<td></td>
<td>5,000</td>
<td>123.85</td>
<td>52.8</td>
<td>0.63</td>
</tr>
</tbody>
</table>

Figure 1. PEEMD parallel performance with three-level parallelism in a log-log plot. The vertical axis represents the time spent for the calculation, and the horizontal axis indicates number of cores. The first case run with 60 cores takes 6,543.56 seconds while the last with 5,000 cores takes 123.85 seconds.
initial evolution of the solution in the full 3DLM when initial perturbations are small, which is described below.

The (local) growth rate and period near the critical point of the 3DLM are estimated using the solution’s envelope (consecutive maximum). Numerical solutions are obtained with the total integration time of $\tau = 110$ and an integration time step $\Delta \tau = 0.001$. Initial conditions are $(X', Y', Z') = (0, 1, 0)$. Shown in Figure 2 are the local growth rates and periods using the numerical solutions with $FN = 1$ and $FN = 0$, which are indicated by blue and red open circles, respectively. The theoretical eigenvalue of $\lambda = (7.92109 \times 10^{-3}, 19.67213)$ obtained from the eigenvalue problem of the 3DLM gives the linear growth rate of 0.0079 and period of 0.6496 ($= 2\pi/9.67213$), which are shown in black lines. A comparison in Figure 2 suggests that the envelope analysis with the PEEMD produces the linear growth rate and period in agreement with the theoretical values and that a (small) initial perturbation grows at a linear growth rate initially and then at a nonlinear growth rate.

A Kelvin Cat’s Eye Flow

Based on analysis of observations, researchers proposed the marsupial paradigm, suggesting that a Kelvin cat’s eye circulation could appear in association with the critical level (or latitude, CL) of an easterly wave and lead to the formation of a “pouch,” which in turn provides protective environment for small-scale convections to grow to enhance mesoscale (vortex) circulation and lead to TC genesis. The CL is defined as the level where the phase speed of a wave mode is equal to the speed of the environment flow. The second case is designed to detect the Kelvin cat’s eye flow and its association with tropical cyclogenesis.

With the confidence in the performance of the PEEMD in analyzing the idealized cases such as the mixed Rossby gravity (MRG) wave, the second dataset with grid points of $(1,001, 1,001)$ in the $(X, Y)$ directions are created as follows. Horizontal and vertical wind perturbations ($U'$ and $W'$) are generated using analytical solutions of a terrain-induced large-scale flow. The basic wind ($U$) represents a linear back-sheared flow, $U = U_0 (1 - Z/H)$, where $U_0$ represents the surface wind, $Z$ is the height, and $H$ is the height of wind reversal. These solutions with the basic wind display the so-called Kelvin cat’s eye flow, showing the streamline from the total horizontal wind ($U' + U$) and vertical wind ($W'$). Individual components of horizontal winds ($U'$ and $U$) and the total horizontal wind ($U' + U$) are shown in Figures 3a through 3c, respectively. The total wind (Figure 3c) is then decomposed into nine IMFs using the PEEMD. Only the sixth IMF (IMF6), the ninth IMF (IMF9), and their summation (IMF6 + IMF9) are displayed in the middle panels of Figure 3. These IMFs are selected because of high correlation coefficients between each of the IMFs and wind perturbation (or basic wind). Note that the IMF9 is indeed the nonoscillatory residual of the data. It’s clear that the sixth IMF (IMF6 in Figure 3d) and ninth IMF (IMF9 in Figure 3e) represent the wind perturbation (Figure 3a) and basic wind (Figure 3b), respectively. In addition, we obtain a very high correlation coefficient of nearly 1 (0.999) between the $U'$ ($U$) and the IMF9 (IMF9), as indicated by the scatter plots in Figures 3g and 3h. Encouraged by these promising results from the idealized case and real-world cases, we applied the PEEMD to analyze the environmental flows associated with Hurricane Sandy, which is discussed next.

Hurricane Sandy

For our third case, we use the PEEMD to analyze Hurricane Sandy data. Sandy first appeared as a
tropical storm in the southern Caribbean Sea on 22 October 2012, moved northeastward, turned northward, and made landfall near Brigantine, New Jersey, in late October. The storm devastated surrounding areas, caused an estimated damage of US$65 billion, and became the second costliest tropical cyclone (TC) in US history, surpassed only by Hurricane Katrina in 2005. Previous work demonstrated the model’s capability to realistically predict Sandy’s genesis with a lead time of up to six days and subsequent evolution for the next two-day period of 22-24 October. The researchers suggested that upper-level tropical waves (MRG wave) and a low-level westerly wind belt could contribute to Sandy’s formation. Below, we demonstrate the performance of the PEEMD in revealing the upper-level tropical waves.

Figure 4 shows the time longitude diagram of the upper-level meridional winds from the ERA-Interim reanalysis data and the corresponding IMFs. Among the IMFs, the fourth (IMF4 in Figure 4c) displays a more visible wave mode than the original data (Figure 4a). IMF4 is then used to determine what kind of waves may appear. In Figure 4c, black dashed lines, indicating the constant phase, determine the phase speeds, while a green line is used to estimate a wavelength of 45 degrees. Theoretical dispersion relations with this wavelength predict a period of 4.91 days for a MRG wave and 7.41 days for an equatorial Rossby wave. The results suggest that the initial formation and movement of Hurricane Sandy (as indicated by the cross signs in Figure 4c) could be influenced by the MRG wave and equatorial Rossby wave.
In this study, we discussed the implementation of parallelism in the PEEMD and presented computational and scientific results with the PEEMD on the Pleiades supercomputer. Three-level parallelism in the PEEMD leads to a parallel speedup and efficiency of 52.8 and 63 percent by increasing the number of cores from 60 to 5,000. Further improvement in computational and scientific performance is being made. We also plan to use the PEEMD in a multiyear data analysis to reveal the statistical relationship among different scale weather and climate parameters.

Acknowledgments
We’re grateful to the following organizations for their support: the College of Science at San Diego State University and the NASA Advanced Information Systems Technology Program. Resources supporting this work were provided by the NASA High-End Computing Program and the NASA Advanced Supercomputing Division at Ames Research Center.

References

Figure 4. Decompositions of the upper-level (200-hPa) meridional winds from the ERA-Interim reanalysis data. (a) Time-longitude diagram of meridian winds at 200-hPa averaged over 20 and 30 degrees north latitude.13 (b)-(d) The third, fourth, and fifth IMFs extracted from the 200-hPa meridional winds, respectively. The locations of Sandy at 00Z 22-24 October are shown in black multiplication signs.

Bo-Wen Shen is an associate professor in the Department of Mathematics and Statistics at San Diego State University. His research interests include high-resolution global and regional modeling, numerical hurricane and weather predictions, high-end computing, chaotic dynamics and predictability, and nonlinear multiscale data analysis. Shen has a PhD in atmospheric sciences from North Carolina State University. Contact him at bowen.shen@gmail or bshen@mail.sdsu.edu.

Samson Cheung is a senior software engineer for CSRA and a member of the Application Group at the NASA Ames Research Center’s Advanced Supercomputing facility. His research interests are in application performance and parallel I/O on new architectures. Samson has a PhD in applied mathematics from the University of California, Davis. Contact him at samson.h.cheung@nasa.gov.

Yu-Ling Wu is a research associate at the Earth Science System Center (ESSC) of the University of Alabama in Huntsville. Her research interests include land-atmosphere interaction, planetary boundary layer meteorology, and air quality. Wu has a PhD from the University of Alabama in Huntsville. Contact her at wuy@nssosc.uah.edu.

Jui-Lin F. Li has extensive experience in global/regional coupled climate modeling and model physical parameterizations with emphasis on boundary layer, clouds, convection, and radiation as well as utilizing new and emerging satellite datasets to study weather and climate to advance our model simulation and future climate projection. Li has a PhD in atmospheric and oceanic sciences from the University of Wisconsin-Madison. Contact him at jli@jpl.nasa.gov.

David Kao is a researcher in the NASA Advanced Supercomputing Division at Ames Research Center. His fields of interest include computer graphics, scientific visualization, information visualization, and numerical flow visualization. Kao has a PhD in computer science from Arizona State University. Contact him at David.L.Kao@nasa.gov.