SANDIA REPORT

SAND2009-2017
Unlimited Release
Printed March 2009

Detecting Combustion and Flow Features *In Situ* Using Principal Component Analysis

David Thompson, Ray W. Grout, Nathan D. Fabian, Janine C. Bennett

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia is a multiprogram laboratory operated by Sandia Corporation,
a Lockheed Martin Company, for the United States Department of Energy's
National Nuclear Security Administration under Contract DE-AC04-94-AL85000.

Approved for public release; further dissemination unlimited.

Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

NOTICE: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government, nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, make any warranty, express or implied, or assume any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represent that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof, or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof, or any of their contractors.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865) 576-8401
Facsimile: (865) 576-5728
E-Mail: reports@adonis.osti.gov
Online ordering: http://www.osti.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd
Springfield, VA 22161

Telephone: (800) 553-6847
Facsimile: (703) 605-6900
E-Mail: orders@ntis.fedworld.gov
Online ordering: http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online

SAND2009-2017
Unlimited Release
Printed March 2009

Detecting Combustion and Flow Features *In Situ* Using Principal Component Analysis

David Thompson
Sandia National Laboratories
M.S. 9159, P.O. Box 969
Livermore, CA 94551, U.S.A.
dcthomp@sandia.gov

Ray W. Grout
Sandia National Laboratories
M.S. 9051, P. O. Box 969
Livermore, CA 94551, U.S.A
rwgrout@sandia.gov

Nathan D. Fabian
Sandia National Laboratories
M.S. 1323, P. O. Box 5800
Albuquerque, NM 87185-1323, U.S.A
ndfabia@sandia.gov

Janine C. Bennett
Sandia National Laboratories
M.S. 9051, P. O. Box 969
Livermore, CA 94551, U.S.A
jcbenne@sandia.gov

Abstract

This report presents progress on identifying and classifying features involving combustion in turbulent flow using principal component analysis (PCA) and k-means clustering using an *in situ* analysis framework. We describe a process for extracting temporally- and spatially-varying information from the simulation, classifying the information, and then applying the classification algorithm to either other portions of the simulation not used for training the classifier or further simulations. Because the regions classified as being of interest take up a small portion of the overall simulation domain, it will consume fewer resources to perform further analysis or save these regions at a higher fidelity than previously possible.

The implementation of this process is partially complete and results obtained from PCA of test data is presented that indicates the process may have merit: the basis vectors that PCA provides are significantly different in regions where combustion is occurring and even when all 21 species of a lifted flame simulation are correlated the computational cost of PCA is minimal. What remains to be determined is whether k-means (or other) clustering techniques will be able to identify combined combustion and flow features with an accuracy that makes further characterization of these regions feasible and meaningful.

Acknowledgments

The authors would like to thank Jacqueline H. Chen and Chun Sang Yoo for the combustion data used in this report.

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed-Martin Company, for the United States Department of Energy under contract DE-AC04-94-AL85000.

Contents

1 Introduction 7
1.1 Lifted Ethylene Jet Simulation 7
1.2 A Review of Principal Component Analysis 8
2 Analysis Strategy 9
3 Analysis Framework 12
4 Results 13
References 17

This page intentionally left blank

1 Introduction

This report, a follow-on to [TFMI08], presents progress towards an *in situ* framework for analyzing simulation data. Where the previous report focused on design issues, this report is a case study of *in situ* analysis on turbulent combustion simulation data. Specifically, a technique for classifying regions of interest using principal component analysis (PCA) is developed.

S3D is a parallel application for simulating turbulent, reacting flows which has been run on systems with upwards of 10,000 processors. For this case study, we consider a simulation involving 21 chemical species reacting in a lifted flame.

Specifics of the simulation are discussed below and this section concludes with a brief review of PCA. The remainder of the report then focuses on the strategy for identifying and classifying regions of interest, how *in situ* analysis capabilities have been added to S3D, and finally the presentation of some intermediate results.

1.1 Lifted Ethylene Jet Simulation

In the lifted jet flame considered here a cold (550k), quickly moving, planar ethylene fuel jet is surrounded by a co-flow of hot (1550k) oxidizer. S3D solves the differential equations governing the evolution of momentum and composition through a fully compressible solution to the Navier-Stokes equations. The solution is advanced on a finite difference grid using an 8th order approximation to the spatial derivatives. In the current case, evolution equations are also advanced for each of 21 species involved in a reduced chemical mechanism for ethylene-air combustion. Due to the compressible formulation, S3D is extremely scalable as communication is only required with neighboring processes. The lifted jet flame simulation was carried out on 30,000 processes on the Cray XT4 at Oak Ridge National Labs with near-ideal weak scaling for a grid with 2025x1600x400 points in the x, y, and z direction.

In this configuration, the conditions following mixing between the fuel and oxidizer are suitable for autoignition within the domain residence time, and autoignition is a significant factor in the stabilization of the flame. The solution is statistically stationary in time; although fluctuations about the mean quantities are unsteady, the time-averaged solution is steady. There is a spatial development in the solution from separate, non-burning, fuel and oxidizer at the domain inlet (bottom) to a steady flame near the outlet at the top of the domain. As this development occurs, different underlying physics control the composition. Before any reaction has taken place, the composition will be completely determined by the amount of fuel and oxidizer present — any of the three species (O_2, N_2, C_2H_4) is sufficient to determine the complete composition. After a steady flame has developed, it is known that the composition is largely determined by amount of the mixture which *originated* in the fuel stream - an indication of this can be obtained by a linear combination of all of the species present which contain the elements found in the inlet streams. During the transition, the species which most naturally describe the thermodynamic state may have significance indicating the dominant physical processes: it is this significance that we hope

to explore using the PCA methodology described below.

1.2 A Review of Principal Component Analysis

PCA is a technique for identifying how several variables observed together are correlated across a sequence of observations. Recall that correlation is related to the covariance. Say we have variables $X = \{x_1, x_2, \ldots, x_n\}$ which are observed r times and have mean $\mu_X = \{\mu_1, \mu_2, \ldots, \mu_n\}$. Their covariance matrix is

$$\mathrm{cov}(X) = \begin{pmatrix} \sum_r (x_1-\mu_1)^2 & \sum_r (x_2-\mu_2)(x_1-\mu_1) & \cdots & \sum_r (x_n-\mu_n)(x_1-\mu_1) \\ \sum_r (x_1-\mu_1)(x_2-\mu_2) & \sum_r (x_2-\mu_2)^2 & \cdots & \sum_r (x_n-\mu_n)(x_2-\mu_2) \\ \vdots & \vdots & \ddots & \vdots \\ (\sum_r x_1-\mu_1)(x_n-\mu_n) & \sum_r (x_2-\mu_2)(x_n-\mu_n) & \cdots & \sum_r (x_n-\mu_n)^2 \end{pmatrix}.$$

Note that $\mathrm{cov}(X)$ is symmetric, positive definite, and has $\frac{n(n+1)}{2}$ unique entries. While examining the covariance matrix of a set of observations directly is possible, it can be very difficult to identify patterns of correlations between sets of more than 3 or 4 variables directly. PCA simplifies this work by presenting a set of up to n vectors which indicate correlated sets of variables.

The PCA technique computes the eigenvectors and eigenvalues of $\mathrm{cov}(X)$ and uses these eigenvectors to project observations into a space with reduced dimensionality $m < n$. Our implementation uses the singular value decomposition (SVD) of $\mathrm{cov}(X) = U\Sigma V^T$ to compute the eigenvectors and eigenvalues. Because $\mathrm{cov}(X)$ is symmetric, $U = V$ and each column of U is an eigenvector. Recall that Σ is a diagonal matrix. In our case, the square root of each diagonal entry of Σ is the eigenvalue associated with the eigenvector in the corresponding column of U.

Obtaining the eigenvalues $v = \mathrm{diag}(\Sigma)$ and eigenvectors $\mathscr{V}$ using the SVD is preferred over other methods because it behaves well in the face of degeneracy; if one variable is constant or if some combination of variables have precisely zero covariance, the SVD still produces valid results. Each eigenvector identifies a correlation among several variables and the ratio of the associated eigenvalue to other eigenvalues indicates the strength of the correlation relative to the other correlations present in $\mathscr{V}$. Most numerical implementations of SVD, including the one we use, return v as a list of eigenvalues sorted in descending order. This is very useful as high values of v indicate strong correlations – meaning that the first entries in v and $\mathscr{V}$ are the ones we should consider.

Once v and $\mathscr{V}$ are obtained, PCA uses $\mathscr{V}$ to project the original observations X into a smaller space, Y. This is accomplished by letting $\mathscr{V}_m$ be the first m columns of $\mathscr{V}$ and taking $Y = X\mathscr{V}_m$.

2 Analysis Strategy

The overall goal of the analysis is to identify, characterize, and validate models of ignition and extinction. The result of PCA on some sub-block (i, j, k) of the simulation domain at time t is a set of eigenvalues v_{ijkt} and eigenvectors $\mathscr{V}_{ijkt}$. Alone, all v_{ijkt} and $\mathscr{V}_{ijkt}$ tell us is which species are correlated in that region. However, taken as a collection over the entire simulation space-time domain, they approximate the set of possible states that the simulation has taken on. If we can further reduce the number of states in this approximation using clustering or some other classification algorithm to some tractable number, it may be possible to determine which states or time-sequence of states are locally responsible for ignition or extinction events. In the best possible scenario, a small number of clusters in state space are associated almost exclusively with ignition or extinction. In this case, we need only determine what distinguishes these clusters from others. More likely, a sub-block of the simulation containing an ignition or extinction event will move through several states during ignition or extinction and some analysis of the time-series of states in the neighborhood will be required.

In order to make classification and clustering as tractable as possible by reducing the dimension of the data, we can use PCA to project into a lower dimension; however, rather than projecting into a different space on each (i, j, k, t)-block, we can first reduce the overall dimension of the problem by eliminating combinations of species that are strongly correlated. To do this, PCA run over the entire space-time mesh can be used to identify a global projection that we can apply to each block's eigenvectors and eigenvalues before performing clustering. This global PCA is simple to calculate since the covariances for all the sub-blocks may be combined using the update formulas in [PÓ8]. The SVD of the global covariance matrix costs no more than for a single block.

PCA is a linear change of basis from one set of coordinates into an orthogonal set of vectors. In order for this change of basis to make sense, each coordinate in the original space should be interchangeable with other coordinates in some sense – it must make sense to sum multiple coordinates because weighted sums of existing coordinates are the way the new basis is created. When PCA is performed on geometric coordinates, this intuitively makes sense: moving some distance along one coordinate axis is equivalent to moving the same distance along another axis. The underlying physics are independent of the coordinate system chosen. When the coordinates have some other meaning, it is not clear that a change of some magnitude in one coordinate is identical to a change of the same magnitude in another coordinate.

For our application, the underlying coordinates are not directly interchangeable. Even small changes in some chemical concentrations are significant while large changes in others are insignificant. In order for PCA results to be meaningful, the coordinates in question should thus be normalized. External knowledge of the meaning of each coordinate is the best source of normalization. Since it is independent of observations, there is no question of circular logic creating or removing important trends in the data as part of the change in basis. Failing that, it can sometimes make sense to normalize by the variance along each initial coordinate axis. To guarantee that the PCA does not ignore variance in species with small changes in concentration, a normalization step is performed as follows:

PCA is first performed over the entire mesh at the first time step, yielding a global covariance matrix $cov(X) = C_{ab}$. Then, as PCA is performed on sub-block (i, j, k) of the mesh, its covariance $cov(X_{ijk}) = C_{ab}^{ijk}$ is normalized before SVD is applied:

$$C_{ab}^{ijk} \leftarrow \frac{C_{ab}^{ijk}}{\sqrt{C_{aa}C_{bb}}}. \tag{2.1}$$

This transformation normalizes the variances of each species directly and the covariances between pairs of species are normalized using a combination of the variances of each entry in the pair.

Once we have a set of vectors to classify, we plan to explore various clustering algorithms (including k-means) as well as other classifiers such as decision trees. As a first step, we are currently in the process of extending the VTK library to include a `vtkKMeansStatististics` class. At this point, we have developed the design of the class and are in the initial implementation stages. The `vtkKMeansStatistics` class will accept a series of raw observations as well as an optional argument consisting of initial set(s) of cluster centers (if no initial cluster centers are provided then a default number of random cluster center locations are generated). The class will be implemented as a subclass of `vtkStatisticsAlgorithm` and will thus perform its computations in a series of three phases:

Learn Updated set(s) of cluster centers are computed.

Derive Errors for the different cluster center requests are reported.

Assess Given a `vtkKMeansStatistics` model and a set of input data, the membership indicator and distance from the corresponding center for each observation is computed.

When using the k-means algorithm to classify data, there are several key issues that we must address. First and formemost, a value for k must be chosen. We plan to choose k *a posteriori* using the variance measured for each result. Recall that k-means is an optimization problem that cannot usually be solved in a global sense. The starting positions of the k cluster centers heavily affects the variance associated with the result. One way to choose k might be to examine how the range of variances associated with several different sets of starting positions changes with k. When k is too low, variances will be high since some eigenvectors will be far from any cluster center. As k increases, variances will tend to drop. However, when k is larger than necessary, some starting positions for cluster centers will yield very good results due to overfitting while other starting positions may not since there will be many local minima that can keep optimizers from converging to a reasonable value. Thus we will look for values of k that have a low overall variance and also a low distribution of variances when run with different initial cluster positions.

In addition to selecting a value for k, it important to select "good" starting positions for the initial cluster centers. When cluster centers are poorly chosen the k-means optimization algorithm will likely get stuck in a local minimum and will not converge to a globally optimal result. We plan on leveraging recent research [DH04] that has shown a close relationship between k-means clustering and PCA. Specifically, this work has proven that the continuous solutions of the discrete k-means

clustering membership indicators are the data projections on the principal eigenvectors of the covariance matrix of the input data. When $k = 2$, a near optimal clustering can be computed, and when $k > 2$ a hierarchical approach can be used to generate a good set of initial cluster centers.

3 Analysis Framework

S3D allows multiple analysis modules – each perhaps written by different authors and performing different tasks – to each be linked into its executable. Static linking is preferred since dynamic loaders may not be available on all high-performance computing (HPC) platforms and even where they are, dynamic libraries on nodes used for compilation may not be present in the run-time environment. To avoid reference loops between libraries (which again may not be supported on HPC systems), analysis modules do not make any calls to functions in S3D. Instead, S3D calls the analysis modules with all of the information they require.

In order to perform analysis *in situ*, access to simulation state is required. S3D is a finite-difference code running on a rectilinear grid with nearly evenly spaced nodes. In practice, nodes far from the flame are spread slightly further apart than nodes near the flame in order to reduce the simulation size required. Because these changes in spacing are small, we choose to make simulation data available to analyses in the form of `vtkImageData` rather than as a `vtkStructuredGrid`. This makes it possible to perform volume rendering using much faster algorithms. Since S3D can be asked to provide spatial derivatives of quantities with much higher order kernels than are present in VTK, it is unlikely this approximation will cause trouble.

On the `jaguar` machine at Oak Ridge, approximately $\frac{1}{3}$ of a node's memory is used by the simulation. Some of this is persistent state and some is in the form of a working array. Neither the persistent state nor the working array are dynamically allocated during the course of a simulation, so S3D only passes addresses to each *in situ* module once at initialization. In between time steps, S3D copies species concentrations into the working array and then calls each analysis module. Once the analyses are done, more time steps are simulated until S3D reaches a termination condition, at which point each analysis module is given an opportunity to complete any ongoing work, free any resources it has acquired, and close any open files.

Each analysis module is expected to provide a set of libraries that implement 3 procedures with the following C signatures:

```
void module_init_(  int* mpiRank, int* mpiCommunicator,
                    double* speciesVals, char* speciesNames, int* numSpecies,
                    double* x, double* y, double* z,
                    int* nx, int* ny, int* nz,
                    int* npx, int* npy, int* npz,
                    int neighbors[6] )
void module_tstep_( int* mpiRank, int* mpiCommunicator, double* time )
void module_fini_(  int* mpiRank, int* mpiCommunicator )
```

where *module* is the name of the analysis module in all lowercase. All values are passed by address instead of value since the subroutines will be called from FORTRAN. The trailing underscores on the procedure names are present because most FORTRAN compilers add an underscore to procedure names. If you are building S3D with a FORTRAN compiler that does not do this, you will have to remove the final underscores from the C implementations of the routines.

4 Results

The following results are from a small subset of simulation data provided by Jacqueline H. Chen and Chun Sang Yoo. In late April, we will be running the same analysis on a much larger dataset – with each restart file requiring 136 GiB – in order to obtain eigenvectors that can be clustered. At that point, the stretch goal for the FY09 feature recognition milestone will have been met. After examining these clusters for signs that they are involved in ignition and extinction events, the cluster centers will then be used during live simulation runs to identify regions of of interest correlated with ignition and extinction.

Figure 1 shows the principal eigenvalue produced by each block in the mesh. Figure 2 shows one component of the principal eigenvector along with its associated species concentration in the raw simulation data. Each block has 20 other eigenvalues and 441 other eigenvector components associated with it (besides those shown). However, if some species are not involved in reaction and are not being convected in large quantities to or from neighboring blocks, then some of the eigenvalues for that block will be near or exactly 0. In these cases, the associated eigenvectors carry no information other than that they are part of some orthonormal basis. Thus it is wise to consider only the subset of eigenvectors and eigenvalues that are likely to carry significant information about interactions between species.

Figures 3 and 4 show the top 152 eigenvectors and values ranked by eigenvalue. Even considering single components, H and CH3CHO concentrations can clearly be used to divide blocks into clusters. Considering pairs of components by examining the figures, CH4 and CO appear to have two distinct regions. Similarly, CO2 and CH2O; and aC3H5 and C3H6 may also be prove useful for classifying eigenvectors. Rather than relying on human inspection, we hope to apply k-means to find signatures for several different regions. Then, by examining how these signatures change over time it may be possible to determine a pattern evinced by ignition and extinction.

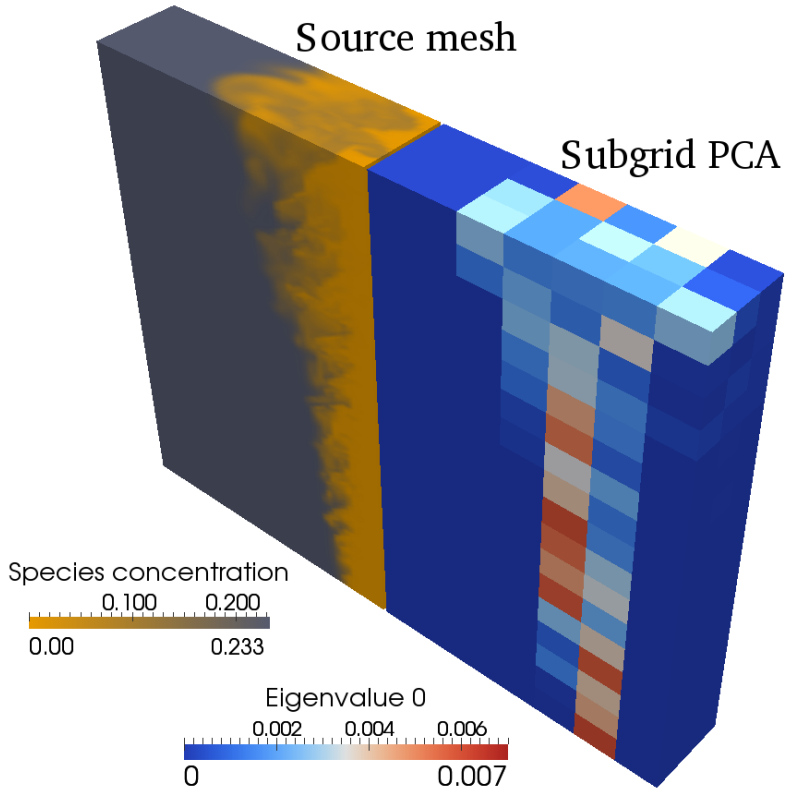

Figure 1. To the left is the source mesh colored by a species concentration. Sub-blocks on the mesh to the right are colored by the principal eigenvalue magnitude.

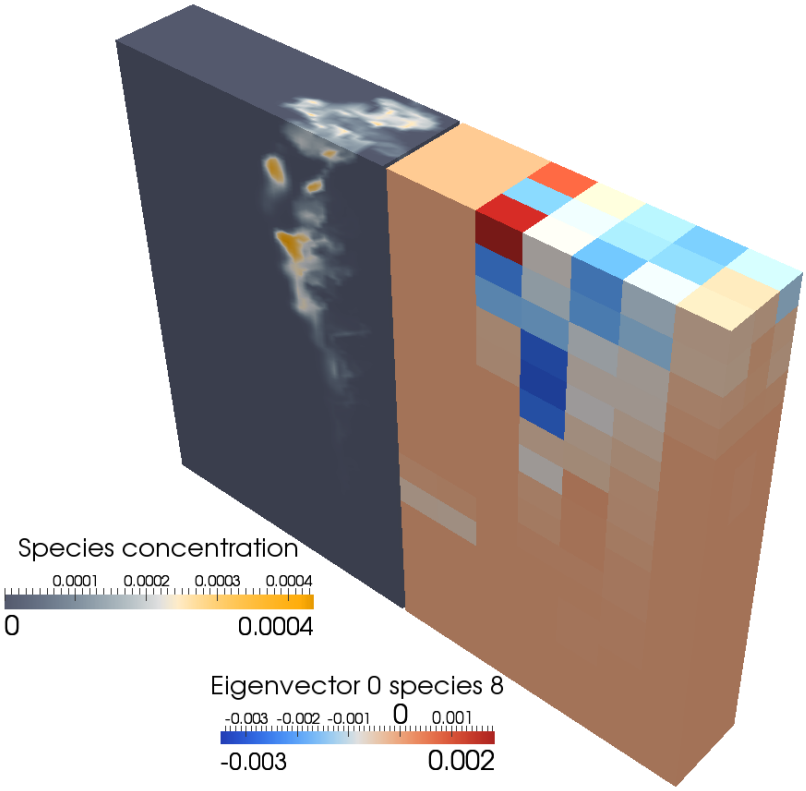

Figure 2. To the left is the source mesh colored by a species concentration. Sub-blocks on the mesh to the right are colored by the magnitude of that species' entry in the principal eigenvector.

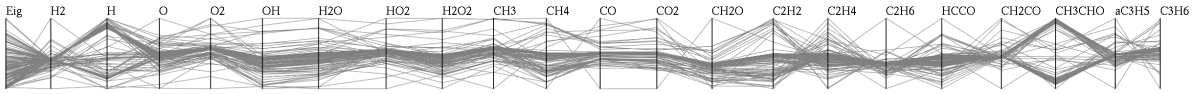

Figure 3. Parallel coordinates rendering of the preliminary results. The first column shows eigenvalues while the remaining columns are components of the associated eigenvector. Only the eigenvectors with the highest values are displayed.

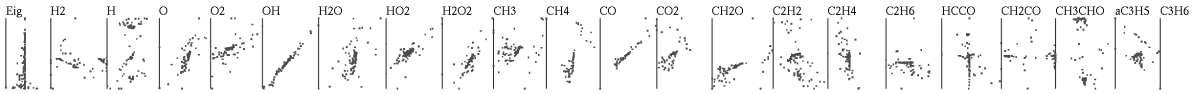

Figure 4. Scatterplots between neighboring pairs of coordinates – in the order shown in the parallel coordinates rendering of Figure 3.

References

[DH04] Chris Ding and Xiaofeng He. K-means clustering via principal component analysis. In *International Conference on Machine Learning*, August 20 2004.

[PÓ8] Philippe Pébay. Formulas for robust, one-pass parallel computation of covariances and arbitrary-order statistical moments. Technical Report SAND2008-6212, Sandia National Laboratories, 2008.

[TFMI08] David Thompson, Nathan D. Fabian, Kenneth D. Moreland, and Lisa G. Ice. Design issues for performing *in situ* analysis of simulation data. Technical Report SAND2008-XXXX, Sandia National Laboratories, 2008.

DISTRIBUTION:

1	MS 9159	David Thompson, 8963
1	MS 1323	Kenneth L. Moreland, 1424
1	MS 1323	Nathan D. Fabian, 1424
1	MS 0822	Lisa G. Ice, 9326
1	MS 9051	Janine C. Bennett, 8963
1	MS 9159	Rob Clay, 8963
1	MS 1323	David Rogers, 1424
1	MS 0823	Constantine J. Pavlakos, 9326
2	MS 9018	Central Technical Files, 8944
1	MS 0899	Technical Library, 9536

Sandia National Laboratories