

SANDIA REPORT

SAND2009-6032

Unlimited Release

Printed September 2009

Scalable Analysis Tools for Sensitivity Analysis and UQ (3160) Results

Lisa Ice Nathan Fabian Kenneth D. Moreland Janine C. Bennett
David C. Thompson David B. Karelitz W. Alan Scott

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-AC04-94AL85000.

Approved for public release; further dissemination unlimited.

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 order: <http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online>



SAND2009-6032
Unlimited Release
Printed September 2009

Scalable Analysis Tools for Sensitivity Analysis and UQ (3160) Results

Lisa Ice David B. Karelitz and W. Alan Scott
Scientific Applications and User Support Department

Nathan Fabian and Kenneth D. Moreland
Data Analysis and Visualization Department

David C. Thompson and Janine C. Bennett
Viz and Scientific Computing

Sandia National Laboratories
P.O. Box 5800
Albuquerque, New Mexico 87185-0822

Abstract

The 9/30/2009 ASC Level 2 Scalable Analysis Tools for Sensitivity Analysis and UQ (Milestone 3160) contains feature recognition capability required by the user community for certain verification and validation tasks focused around sensitivity analysis and uncertainty quantification (UQ). These feature recognition capabilities include crater detection, characterization, and analysis from CTH simulation data; the ability to call fragment and crater identification code from within a CTH simulation; and the ability to output fragments in a geometric format that includes data values over the fragments. The feature recognition capabilities were tested extensively on sample and actual simulations. In addition, a number of stretch criteria were met including the ability to visualize CTH tracer particles and the ability to visualize output from within an S3D simulation.

CONTENTS

1. EXECUTIVE SUMMARY	7
2. MOTIVATION	7
3. MILESTONE OBJECTIVE SUCCESS CRITERIA	9
4. MILESTONE OBJECTIVE STRETCH GOALS	9
5. INTRODUCTION	10
5.1 Overview	10
5.2 Crater Calculation	10
5.3 In-Situ Feature Recognition and Visualization	10
6. RESULTS	11
6.1 Milestone Objective Success Criteria	11
6.2 Milestone Objective Stretch Goals	23
7. CONCLUSION	34
References	34

FIGURES

Figure 1: Four images from an in-situ feature recognition and visualization task	12
Figure 2: Example: exodus output of an extracted feature. The Exodus output is on the left. The original simulation is on the right	13
Figure 3: Low-resolution cone shaped crater example. The original dataset is on the left. The crater is on the right.	14
Figure 4: High-resolution cone shaped crater example. The original dataset showing the crater is on the right. The crater volume is on the left.	15
Figure 5: Low-resolution cylinder shaped crater example. The original dataset is on the right. The crater is on the left.	16
Figure 6: High-resolution cylinder shaped crater example. The original dataset is on the right. The crater is on the left.	17
Figure 7: Low-resolution hemisphere shaped crater example. The original dataset is on the right. The crater is on the left.	17
Figure 8: High-resolution hemisphere shaped crater example. The original dataset is on the right. The crater is on the left.	18
Figure 9: Low resolution cone at a Material Volume Fraction of 0.25	19
Figure 10: Low resolution cone at a Material Volume Fraction of 0.75	19
Figure 11: Example script for in-situ visualization	22
Figure 12: Scaling performance of the ParaView fragment connectivity filter as a function of the number of processes	23
Figure 13: Example impact crater. The left side shows the crater. The right side shows the original sphere impacting the block	24
Figure 14: Chart plotting Crater Volume and Depth versus Time step	25
Figure 15: Time series visualization of the magnitude of the magnetic field in a test magnetohydrodynamic solver problem in Alegra, the "Perry-White Rotor Problem"	26

Figure 16: These two images show the magnitude of the first two eigenvalues of each 8x8x8 block in the S3D simulation.....	28
Figure 17: This image shows one component of the second eigenvector, corresponding to carbon monoxide concentration.....	29
Figure 18: This plot shows for each of the 21 eigenvector/eigenvalue pairs, the log of the largest eigenvalue over the entire image.	30
Figure 19: Parallel coordinate plots of observations may provide insight into initial cluster center values in addition to the ideal number of clusters for a set of observations. These images are plots of parallel coordinates for eigenvalue/eigenvectors pair observations generated from a subset of the S3D data. Potential clusters are drawn in red.	31
Figure 20: This image shows the fragment filter run on an Exodus simulation of a pipe bursting.	32
Figure 21: Visualization of CTH tracer particles. The spheres show where the tracers are at the current time step. The lines show the path they traveled.	33

1. EXECUTIVE SUMMARY

Sandia has met the requirements of the 9/30/2009 ASC Level 2 Scalable Analysis Tools for Sensitivity Analysis and UQ (Milestone 3160). All feature recognition functionality required by the user community is present and has been tested extensively. Feature recognition capabilities provided and/or tested by the user community for this milestone include crater detection, characterization, and analysis from CTH simulation data; the ability to call fragment and crater identification code from within a CTH simulation; and the ability to output fragments in a geometric format that includes data values over the fragments. The capabilities were tested extensively on sample and actual simulations. In addition, a number of stretch criteria were met including the ability to visualize CTH tracer particles and the ability to visualize output from within an S3D simulation. Significant progress was made towards characterizing the efficiency of the fragment detection code at high numbers of processors and a plan is in place to enable feature recognition and fragment characterization operations on simulation data other than CTH. In addition, first steps were taken towards integrating the ParaView in-situ module with the Alegria and Sierra simulation codes. All SNL tasks are documented in the *ASC 2009 Level II Scalable Analysis Tools for Sensitivity Analysis and UQ Success and Stretch Criteria* (WFS1030938) document. All results are documented in this document (WFS1030939).

2. MOTIVATION

Modeling and simulation problems of importance to Sandia's NW program produce results that are large, complex, and difficult to investigate. These data contain interrelated quantities, features, and anomalies that are of interest to analysts - things that at best are difficult to find. Post processing tools that promote interactive analysis of these results rely on data residing on disk, which is typically only a fraction of the data computed during a simulation. As we advance to compute platforms such as Zia and Trinity, there will be an increasing disparity between compute power and I/O speeds, so it will be increasingly challenging to rely on a post-processing workflow to capture and understand the science behind these simulations.

Much better than simply post-processing data is providing methods and libraries for analyzing and visualization the data in conjunction with the simulation – spending part of a compute cycle performing operations such as fragment identification, crater identification, and creating images on the fly. Other artifacts, such as reduced geometry for features of significance, computed quantities that are not present in the simulation, and statistics can also be computed during the simulation, for exploration later. The benefits of this approach are clear – a higher fraction of relevant information can be written to disk, while optimizing the I/O bandwidth available to a simulation. In addition, data written in conjunction with the simulation can achieve higher temporal fidelity than is practical when simply post-processing. Images, raw data, and reduced geometry can be written every time step, if necessary, and because analysis algorithms have access to the simulation's entire runtime dataset, it becomes possible to achieve things such as feature tracking, which would be impossible with low temporal fidelity data.

Codes such as CTH currently implement useful tools to perform these functions, utilizing an effective scripting API that analysts can use to great effect. However, this capability is currently tied to CTH. A common analysis and visualization capability – one that could be accessed from different codes, and provide a common way of performing important operations – would be of significant value to the modeling and simulation community at Sandia, and elsewhere.

To address this need, Sandia's Data Analysis and Visualization department is working closely with a small set of important customers to implement a general capability that provides a flexible analysis capability to a range of problems. This capability should be useable by a broad range of users, be adaptable to current and future workflows, and build upon the tools, experience, and expertise of developers and analysts who use the current generation of these tools.

Of critical importance to any general capability is the requirement to scale to Zia-sized processor numbers in a way that does not unnecessarily interfere with the computation. As part of the effort to deliver a general capability, we have pursued ASC Level II Milestones in FY08 and FY09 to deliver useful capability for application to real NW problems, to establish technical collaborations required to achieve integrated analysis capability across a set of codes.

3. MILESTONE OBJECTIVE SUCCESS CRITERIA

The Completion Criteria specified in the Milestone were to provide scalable analysis capabilities to support validation of large data with particular emphasis on comparison of simulation and test data.

The project delivered feature recognition capabilities supporting specific aspects of large data analysis that will be of use to a range of customers.

From the completion criteria specified above, the following objective success criteria were developed early in the calendar year by the milestone team, consisting of Lisa Ice, Dino Pavlakos, David Rogers, Ken Moreland, Nathan Fabian, David Karelitz, Jason Wilke, Stephen Attaway, Richard Hills, and Martin Heinstein. To successfully complete the milestone, the following baseline feature recognition capabilities were developed and tested against example and actual datasets to the satisfaction of the customers, Stephen Attaway, Jason Wilke, and Richard Hills.

1. Deploy fragment identification and geometric output capabilities demonstrated in ParaView for the FY08 Post-Processing V&V Level II ASC Milestone (2843) as a set of library calls to be used within a CTH simulation.
2. Produce geometric surface output of identified fragments in a format that includes data values.
3. Calculate crater statistics on a cone-shaped crater:
 - a. Crater volume, defined as the volume of a cavity measured from a user-defined plane to the remaining surface
 - b. Crater surface area:
 - i. Surface area of the depression geometry
 - ii. Surface area of the intersection of the user-defined plane and the crater volume
 - c. Depth of penetration, measured as the longest distance from the user-defined plane to the furthest point, normal to the plane.
4. Provide scripting capability for inclusion in an input deck associated with CTH that is sufficient to execute the feature recognition tasks developed for this milestone.

4. MILESTONE OBJECTIVE STRETCH GOALS

1. Characterize the efficiency of the fragment identification algorithms to prove scaling as the number of processors increases, up to at least 1000
2. Calculate crater statistics on actual data.
 - a. Crater volume, defined as the volume of a cavity measured from a user-defined plane to the remaining surface
 - b. Crater surface area:
 - i. Surface area of the depression geometry
 - ii. Surface area of the intersection of the user-defined plane and the crater volume
 - c. Depth of penetration, measured as the longest distance from the user-defined plane to the furthest point, normal to the plane.
3. Demonstrate prototype fragment identification capability within a Presto simulation
4. Demonstrate prototype capability for executing VTK components within an S3D simulation.
5. Demonstrate fragment identification within ParaView on other formats:
 - a. Exodus file set
6. Demonstrate visualization of CTH tracer particles within ParaView.

5. INTRODUCTION

5.1 Overview

Sensitivity Analysis and UQ require analysis and investigation of ensembles of runs, in which large numbers of runs (anywhere from tens to hundreds) must be analyzed as a group. In practice, analysts cannot examine a significant fraction of the data, and are reduced to examining simple quantities (such as a maximum or minimum of a significant variable) to understand the behavior of the ensemble. Feature recognition capabilities developed in this milestone leveraged the scalable scientific and information visualization technologies within ParaView to enable analysts a broader understanding of the ensemble, by providing additional results, both quantitative and qualitative, from within a simulation run. These results will allow the future development of abstract views of the ensemble, connected to readily-available drill-down to specific data from particular simulation runs.

5.2 Crater Calculation

Two common results of CTH simulations are datasets producing either fragments or craters. Feature recognition in the form of quantitative analysis of fragments and craters provides a means both for validation, comparing the simulation to an actual experiment, as well as examining the sensitivity of a simulation to changes in the input conditions. The feature recognition work performed for last year's Level II milestone resulted in the ability to quantify the fragments produced by a CTH simulation. This year's milestone extends that work to quantify craters produced by a CTH simulation.

Calculating craters statistics is very similar to calculating fragment statistics because a crater is an inverted fragment. To extract a crater instead of a fragment, cells are kept if they have volume fractions less than the material volume fraction threshold versus fragment extraction which keeps cells if they have volume fractions greater than the material volume fraction threshold. These cells are then clipped with an implicit plane or sphere to isolate the volume of the crater below the original surface of the material. The volume of the crater is integrated in the same manner as the standard fragment connectivity filter; the volume of each cell in the fragment is computed and then summed. For convenience, the maximum depth of the crater along the clipping plane's normal is also recorded as the cells are integrated.

5.3 In-Situ Feature Recognition and Visualization

Visualization and data analysis is most commonly performed as a post-processing step, after a simulation has completed running. In-situ feature recognition, data analysis, and visualization move these steps from post-processing to become part of the simulation run. Simulations typically only save data for post-processing at a very coarse temporal resolution due to size and speed limitations with the result being that data for analysis is only available at that coarse temporal resolution with the potential to miss important simulation events. Performing the feature recognition, data analysis, and visualization during the simulation allows for a much higher temporal fidelity, resulting in smoother animation and a much lower chance of missing important events. In addition, if specific extracted features, quantities, or visualizations are the only desired results, then these extractions, quantities, or visualizations can be computed during the simulation without saving any data, resulting in a dramatic time and space savings.

6. RESULTS

6.1 *Milestone Objective Success Criteria*

1. Deploy fragment identification and geometric output capabilities demonstrated in ParaView for the FY08 Post-Processing V&V Level II ASC Milestone (2843) as a set of library calls to be used within a CTH simulation

As part of the in-situ feature extraction and visualization work, the first step was to modify CTH to allow for the execution of a static ParaView pipeline in place of spyplot, the native in-situ visualization tool used for CTH. This was accomplished by redirecting the CTH-to-spyplot API into equivalent calls in ParaView. A majority of the task on the ParaView side involved taking pointers to in-memory CTH data structures and reinterpreting them as ParaView data structures without copying that data. Finally the static pipeline was replaced with the execution of a python script using the batch python scripting capability in ParaView.

Figure 1 below show four time steps out of eighty-four from an in-situ feature recognition and visualization task using a ParaView pipeline. The simulation shows an example dataset of a ball impacting a brick, with the ball and brick materials each a different color. The temporal resolution of these images is over ten times greater than what is typically saved for post-processing. The full image set is available through Web FileShare (WFS1030872).

This capability is deployed in a beta form in the CTH source-code repository. A plan is in development to integrate the capability with the CTH software deployment process so it is easily available to all CTH users at Sandia.

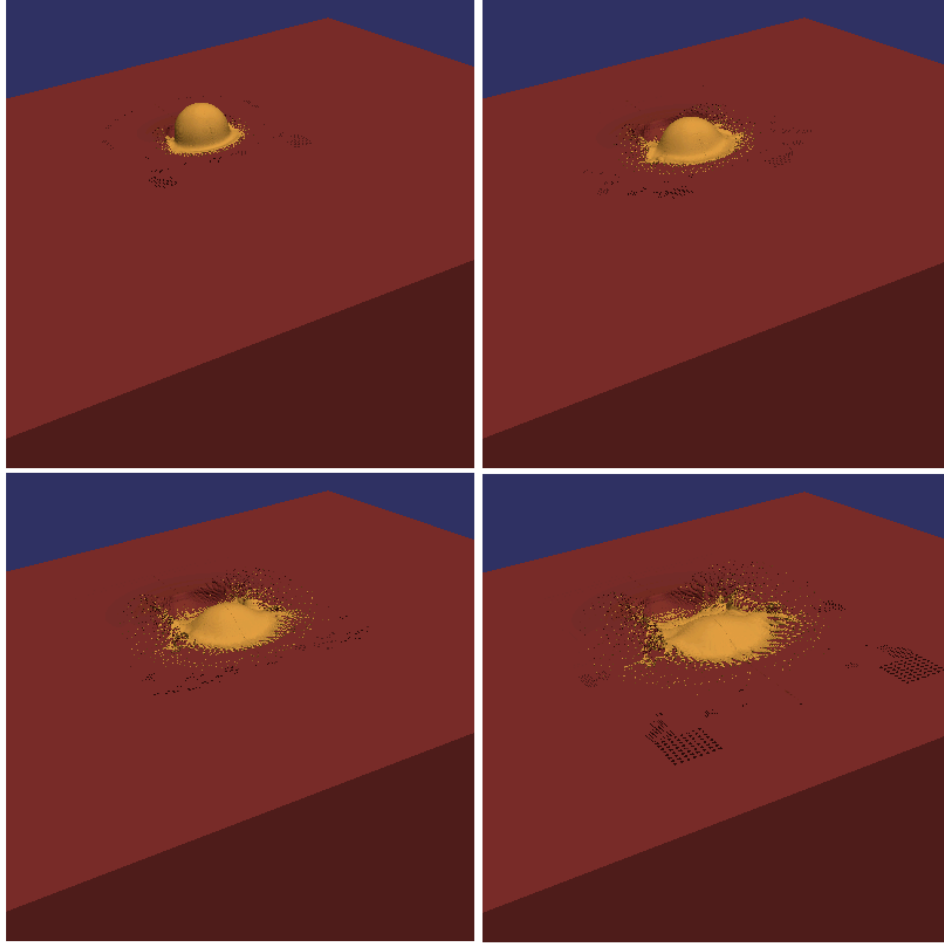


Figure 1: Four images from an in-situ feature recognition and visualization task

2. Produce geometric surface output of identified fragments in a format that includes data values

In order to save extracted features such as fragment data, the capability to export the outer surface of identified fragments as an exodus file, complete with data values, was added to ParaView. The capability is available both during in-situ processing as well as from within typical ParaView post-processing. Figure 2 shows the original CTH simulation on the right and the Exodus output on the left.

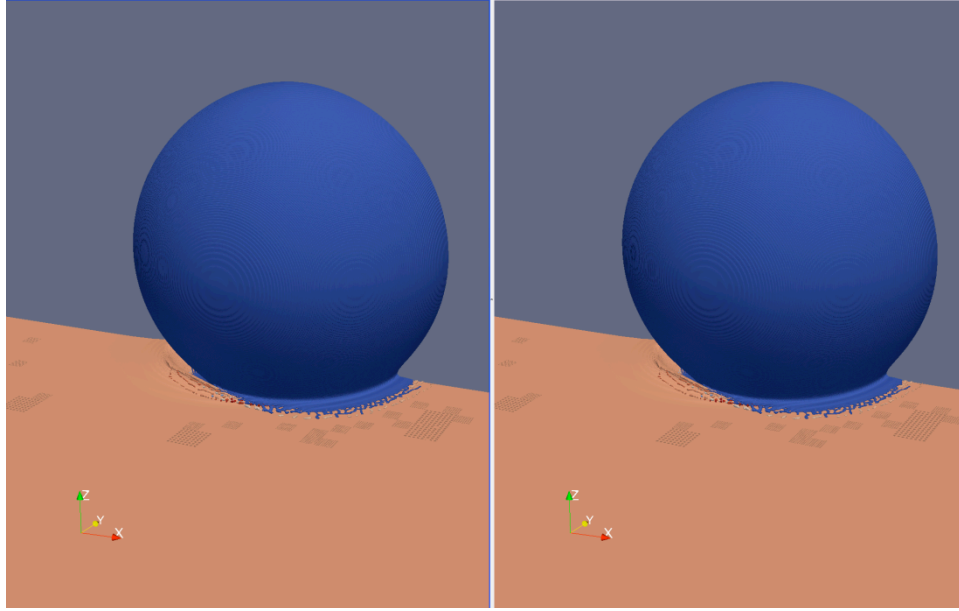


Figure 2: Example: exodus output of an extracted feature. The Exodus output is on the left. The original simulation is on the right

3. Calculate crater statistics on a cone-shaped crater:
 - a. Crater volume, defined as the volume of a cavity measured from a user-defined plane to the remaining surface
 - b. Crater surface area:
 - i. Surface area of the depression geometry
 - ii. Surface area of the intersection of the user-defined plane and the crater volume
 - c. Depth of penetration, measured as the longest distance from the user-defined plane to the furthest point, normal to the plane.

A key requirement for sensitivity analysis and uncertainty quantification for CTH simulations is the ability to accurately quantify craters produced during a simulation. Features identified and quantified include crater volume, surface area, and depth. Two meshes were generated for this example; a low-resolution 22 million element CTH AMR mesh, and a high-resolution 118 million element CTH AMR mesh. Both meshes contained three canonical shapes, a cone, a hemisphere, and a cylinder. Crater measurements were generated from both the low resolution and high resolution meshes and compared to the theoretical results for each canonical shape.

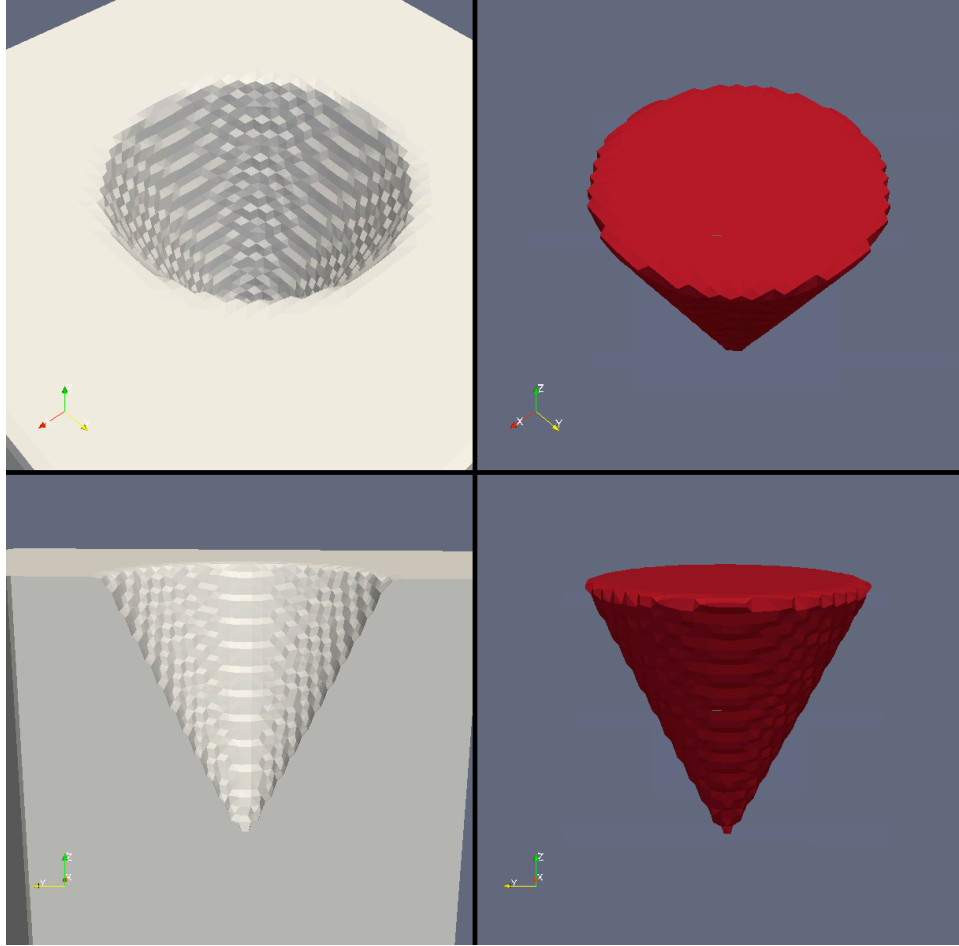


Figure 3: Low-resolution cone shaped crater example. The original dataset is on the left. The crater is on the right.

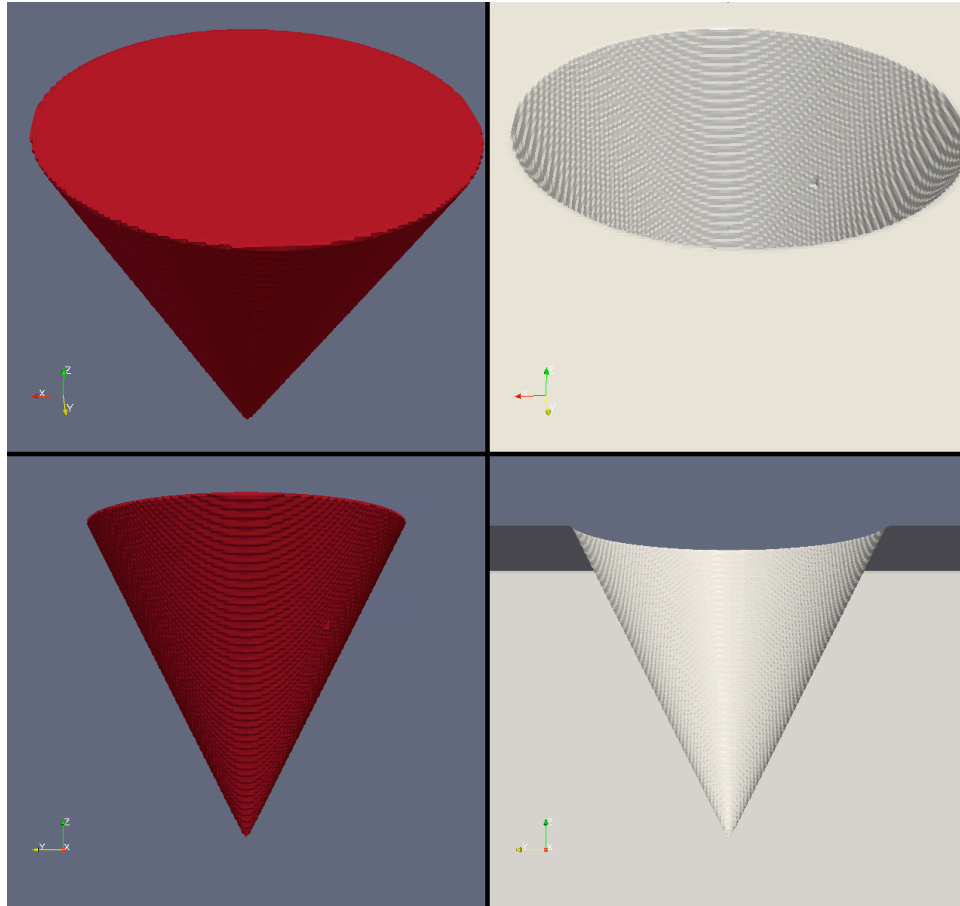


Figure 4: High-resolution cone shaped crater example. The original dataset showing the crater is on the right. The crater volume is on the left.

The following statistics were calculated for the low-resolution and high-resolution cone-shaped craters and compared to the actual values.

Cone	Low-Resolution	High-Resolution	Theoretical Result
Volume	0.184	0.189	0.191
Surface Area of Top	0.478	0.611	0.636
Surface Area of Crater	1.522	1.46	1.423
Maximum Depth	0.852	0.89	0.900
Radius	0.440	0.445	0.45

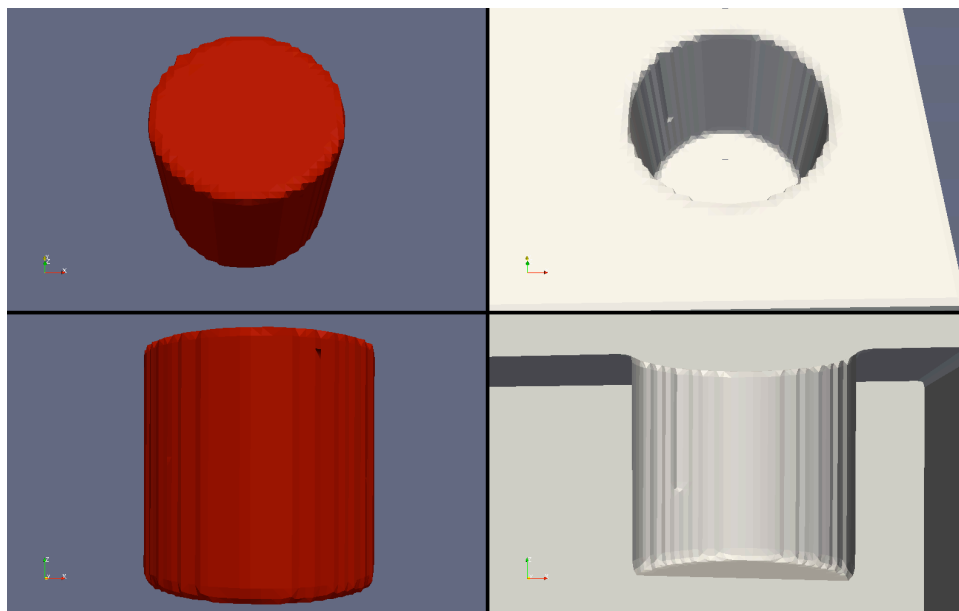


Figure 5: Low-resolution cylinder shaped crater example. The original dataset is on the right. The crater is on the left.

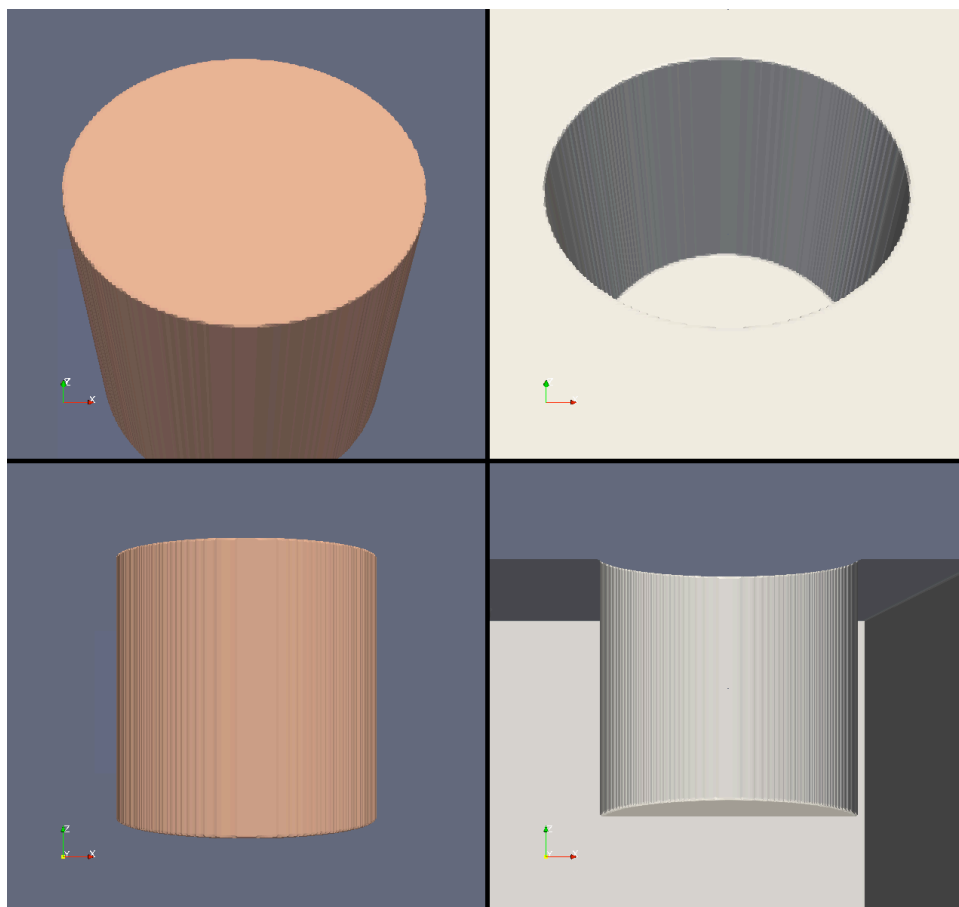


Figure 6: High-resolution cylinder shaped crater example. The original dataset is on the right. The crater is on the left.

The following statistics were calculated for the low-resolution and high-resolution cylinder-shaped craters and compared to the actual values.

Cylinder	Low-Resolution	High-Resolution	Theoretical
Volume	0.766	0.779	0.785
Surface Area of Top	0.768	0.782	0.785
Surface Area of Crater	3.870	3.94	3.927
Maximum Depth	1.000	1.000	1.0
Radius	0.497	0.500	0.5

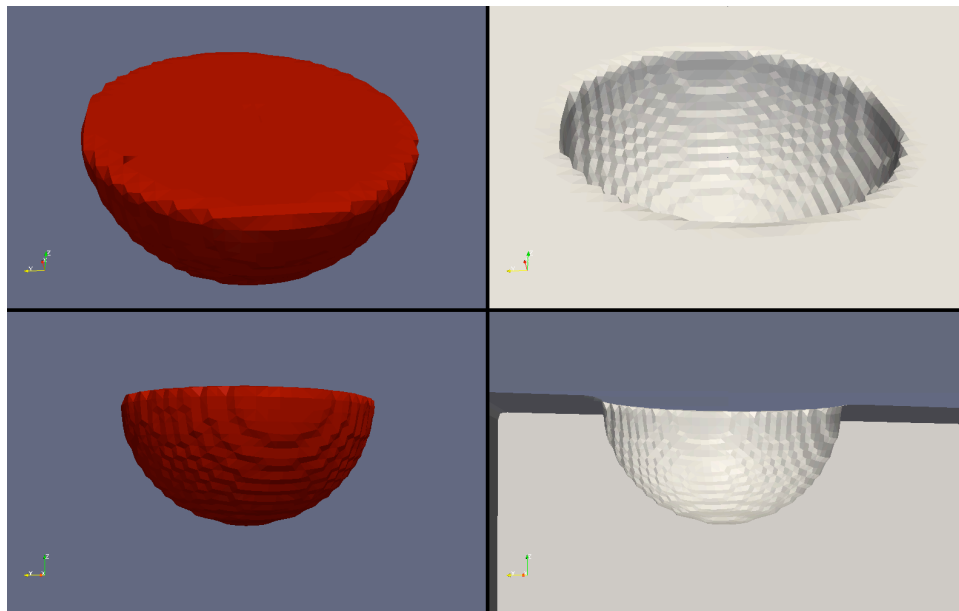


Figure 7: Low-resolution hemisphere shaped crater example. The original dataset is on the right. The crater is on the left.

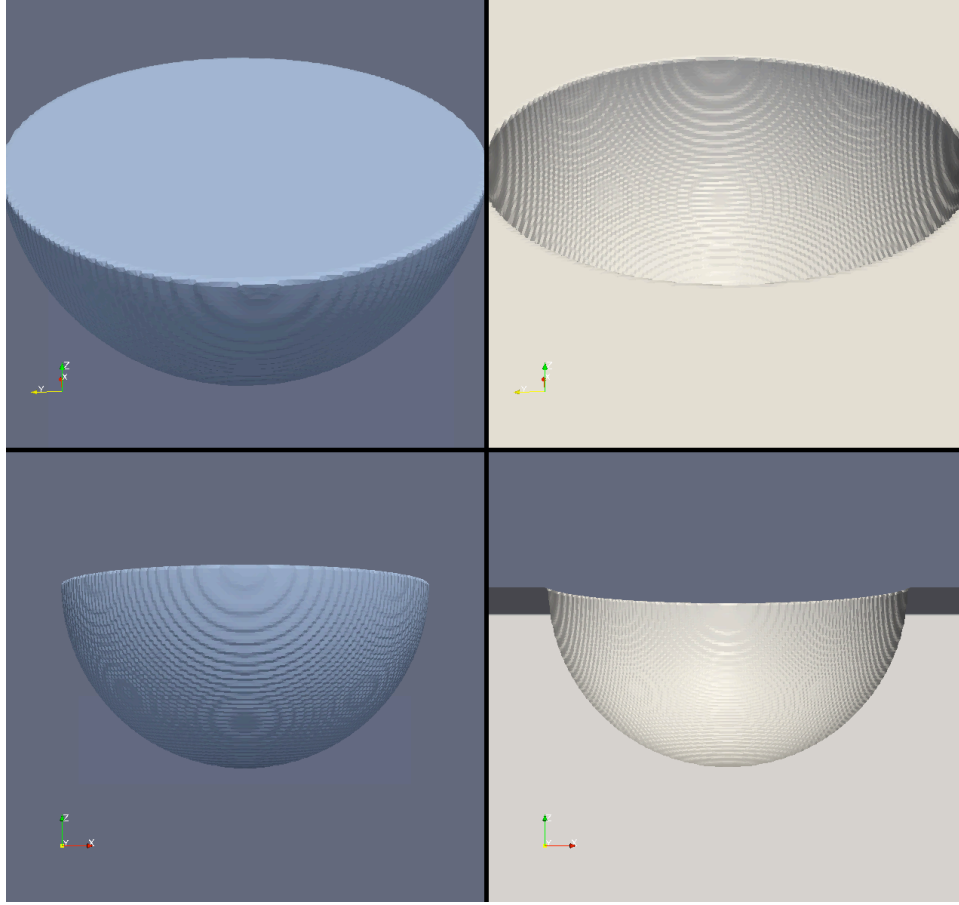


Figure 8: High-resolution hemisphere shaped crater example. The original dataset is on the right. The crater is on the left.

The following statistics were calculated for the low-resolution and high-resolution hemisphere-shaped craters and compared to the actual values.

Hemisphere	Low-Resolution	High-Resolution	Theoretical
Volume	0.254	0.189	0.262
Surface Area of Top	0.772	0.779	0.785
Surface Area of Crater	1.558	1.593	1.571
Maximum Depth	0.501	0.499	0.5
Radius	0.495	0.501	0.5

It's important to note that for all 3 crater shapes, as the mesh resolution increases, the simulation results converge to the theoretical results.

In addition, the statistics of the cone were studied as a function of the material volume fraction threshold. As the threshold is decreased, the craters grow larger, and as it's increased, the craters grow smaller. This is due to the fact that the craters are computed by inverting the original geometry, so an increase in the material volume fraction makes the original geometry larger and the crater smaller.

Adjusting the material volume fraction threshold can reduce the errors due to mesh under-resolution for this ideal example. We have not looked into whether adjusting the threshold helps in simulations of non-canonical shapes.

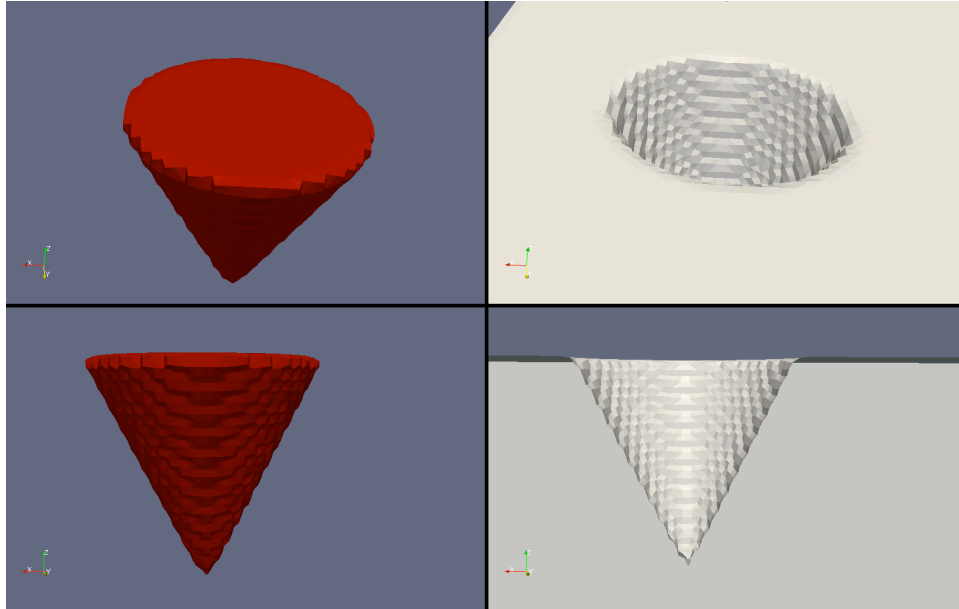


Figure 9: Low resolution cone at a Material Volume Fraction of 0.25

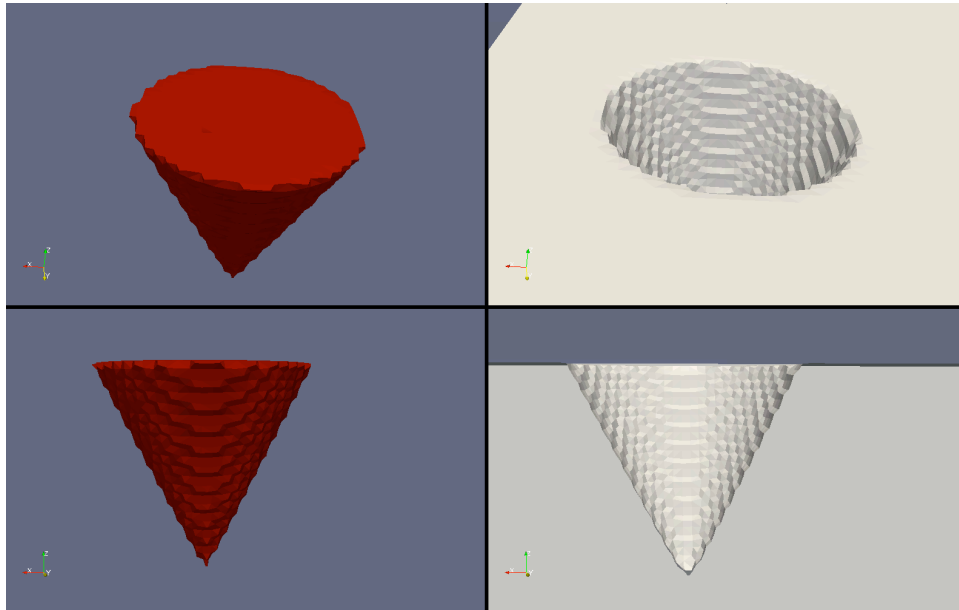


Figure 10: Low resolution cone at a Material Volume Fraction of 0.75

Low-Res Cone	MVF 0.25	MVF 0.5	MVF 0.75	Theoretical Result
Volume	0.189	0.184	0.176	0.191
Surface Area of Top	0.651	0.585	0.448	0.636
Surface Area of Crater	1.524	1.415	1.409	1.423
Maximum Depth	0.893	0.852	0.835	0.900
Radius	0.45	0.440	0.42	0.45

4. Provide scripting capability for inclusion in an input deck associated with CTH that is sufficient to execute the feature recognition tasks developed for this milestone

The first success criterion in the milestone was simply to call a static ParaView feature recognition pipeline from within a CTH simulation. This success criterion builds on that by allowing the user to specify a ParaView python script to execute within the CTH input deck. The script shown in Figure 11 shows the ParaView script used to generate the images shown in Figure 1.

```
# make the ParaView source that interfaces with the CTH simulation
cthsource = CTHSource ()

# Create a view of the ball impacting the brick
ballblock = ExtractCTHFrags (cthsource)
ballblock.MaterialFractionThreshold = 0.5
ballblock.SelectMaterialFractionArrays = ["VOLM1", "VOLM2"]
ballblock.SelectMassArrays = ["M1", "M2"]

# Reflect the simulation result about the simulation axis of symmetry
normBall = GenerateSurfaceNormals (ballblock)
normBall.FeatureAngle = 90
normBall.Splitting = 1
normBall.PieceInvariant = 1
reflBall = Reflect (normBall)
reflBall.Center = 0.0
reflBall.Plane = "Y"

# Function to make the appropriate colorbar
def MakeGoldRedLT():
    lt = servermanager.rendering.PVLookupTable ()
    servermanager.Register(lt)
    lt.RGBPoints = [1, 0.56, 0.23, 0.23, 0, 0.9568, 0.6928, 0.3294]
    lt.ColorSpace = "RGB"
    return lt

# Function to initialize the display
def SetupDisplay(proxy, lt):
    SetDisplayProperties (proxy, SuppressLOD = 1)
    SetDisplayProperties (proxy, StaticMode = 1)
    SetDisplayProperties (proxy, LookupTable = lt)
    SetDisplayProperties (proxy, Specular = 0.1)
    SetDisplayProperties (proxy, SpecularPower = 40)
    SetDisplayProperties (proxy, SpecularColor = [1.0, 1.0, 1.0])

# Setup the initial display
print "Updating display parameters"
SetupDisplay (reflBall, MakeGoldRedLT())
Show (reflBall)
print "done Updating display parameters"

# Eventually used for camera view rotation
angle = 3.0

# update is called after every CTH simulation iteration
def update(process,cycle,ptime,pdt):
    global angle

    # don't care about the first 1200 cycles of the simulation
    if (cycle < 1200):
        return

    # only perform the visualization every 25th cycle
    if ((cycle % 25) == 0):

        # Setup the coloring for the geometry
        print "Setting refl properties"
        SetDisplayProperties (reflBall, ColorAttributeType = "POINT_DATA")
```

```

SetDisplayProperties (reflBall, ColorArrayName = "Material")
print "done Setting refl properties"

# Set the simulation time for the visualization
print "Updating visual pipeline"
SetViewProperties (ViewTime = ptime)
# set view properties
SetViewProperties (Background = [0.2, 0.2, 0.4])
SetViewProperties (LightIntensity = 0)
SetViewProperties (MaintainLuminance = 0)
SetViewProperties (UseLight = 1)

#Set the camera position and where it's looking
c = GetActiveCamera ()
c.SetPosition ([10, -4, 4])
c.SetFocalPoint ([1.5, 0, -1.0])
c.SetViewUp([-0.423999152002544,0.317999364001908,0.847998304005088])
c.Azimuth (angle)

# save the image
WriteImage ("image_%(p)03d_%(c)06d.png" % {'p':process,'c':cycle})

# since every process saves an image, remove the ones we don't need
# This is just a workaround and should
# eventually be fixed in ParaView
if (process != 0):
    os.remove ("image_%(p)03d_%(c)06d.png" % {'p':process,'c':cycle})

```

Figure 11: Example script for in-situ visualization

6.2 Milestone Objective Stretch Goals

1. Characterize the efficiency of the fragment identification algorithms to prove scaling as the number of processors increases, up to at least 1000

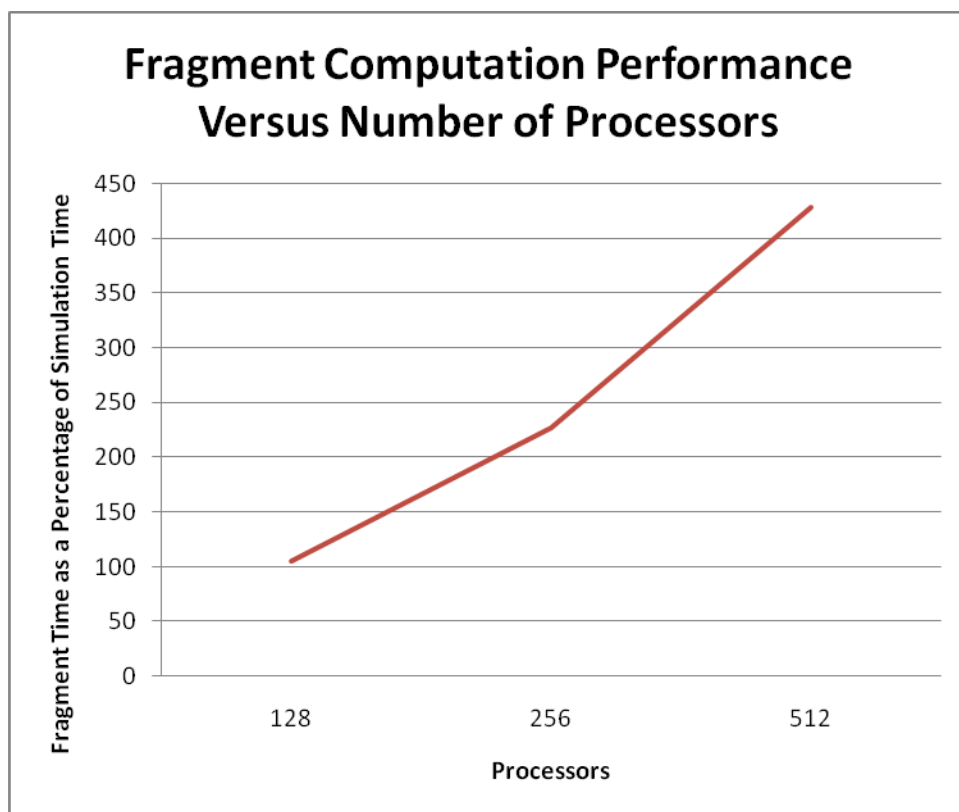


Figure 12: Scaling performance of the ParaView fragment connectivity filter as a function of the number of processes

Work was done to characterize the efficiency of the CTH/ParaView interface as well as the efficiency of the feature recognition algorithm used for fragment identification and characterization. A simulation was run at various scales from 128 to 512 processors using the Thunderbird simulation cluster. The results show that the fragment identification and characterization code does not scale correctly as the number of processors increases. As the number of processors increases, the fragment time should remain fixed as a percentage of simulation time, not increase linearly with the number of processors.

This poor result was not what was expected. The team has prioritized the ongoing work of diagnosing and correcting the scaling issues. The team is consulting experts in scaling algorithms in order to evaluate the scaling of the fragment connectivity filter algorithm, itself.

We did not successfully run a simulation at 1024 processors in order to test ParaView component scaling on that many processors; however the issues identified with lower numbers of processors must be corrected prior to continuing the scaling study.

2. Calculate crater statistics on actual data.
 - a. Crater volume, defined as the volume of a cavity measured from a user-defined plane to the remaining surface
 - b. Crater surface area:
 - i. Surface area of the depression geometry
 - ii. Surface area of the intersection of the user-defined plane and the crater volume
 - c. Depth of penetration, measured from the longest distance from the user-defined plane to the furthest point, normal to the plane.

A simple CTH AMR simulation was run in which a sphere impacted a cube. Statistics were calculated on the roughly hemispherical crater and are shown below.

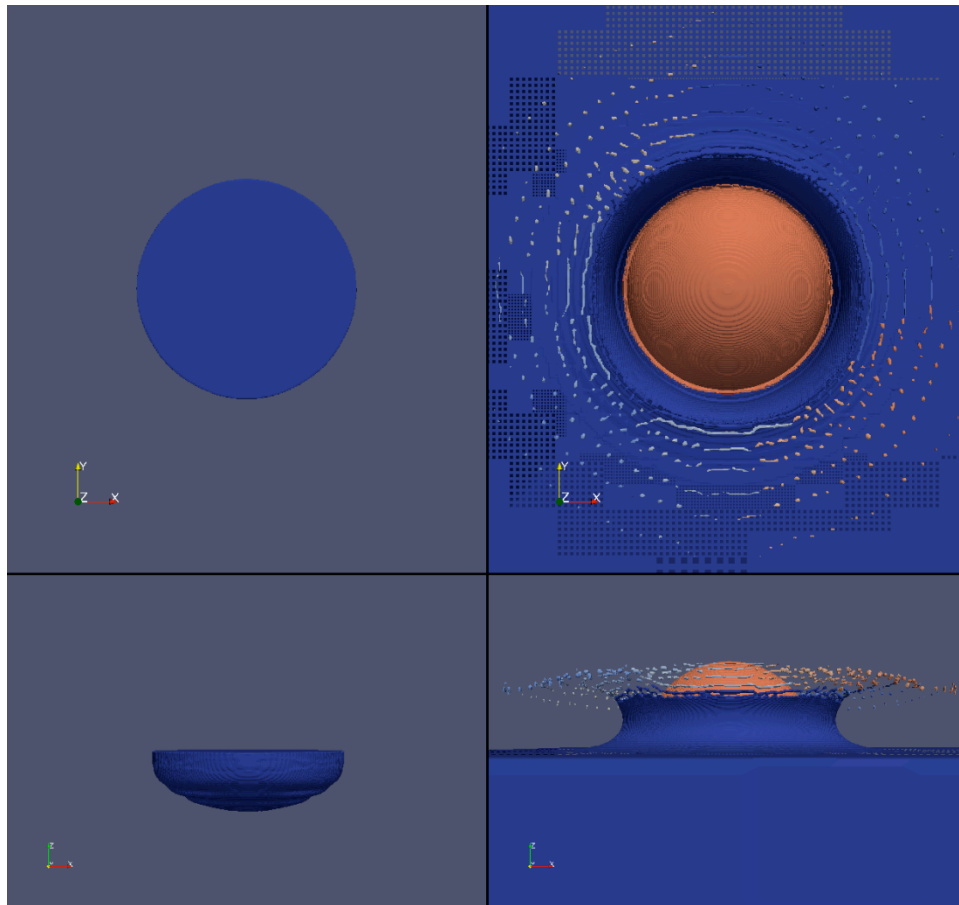


Figure 13: Example impact crater. The left side shows the crater. The right side shows the original sphere impacting the block

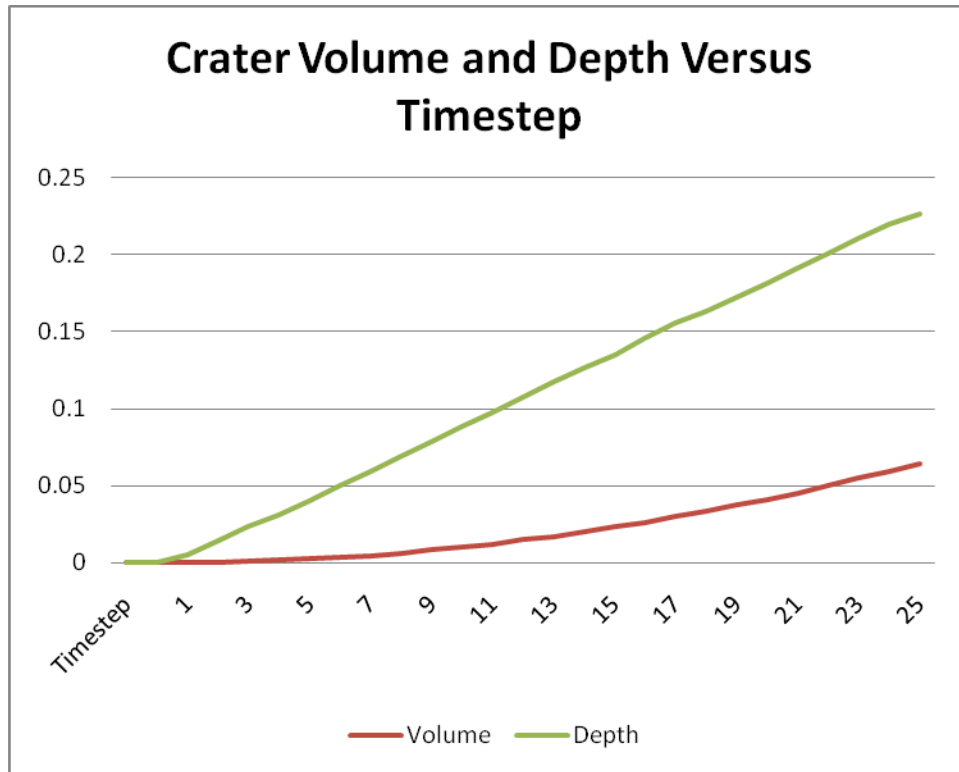


Figure 14: Chart plotting Crater Volume and Depth versus Time step

The table below shows the crater statistics at the time step shown in Figure 13. Plots of volume and depth vs. time are shown in Figure 13.

Volume	0.064
Surface Area of Top	0.389
Surface Area of Crater	0.677
Maximum Depth	0.226

3. Demonstrate prototype fragment identification capability within a Presto simulation

The team focused early this year on integrating the ParaView in-situ feature recognition and visualization library with Alegra, a high energy density physics simulation code. The team has recently begun to extend this work to integration with Presto, one of the simulation codes within the Sierra framework.

An alpha version of the Alegra integration is complete with some preliminary images available. While full scripting is not yet included, the Alegra/ParaView binding has been successfully demonstrated. A plan is in place for the Sierra integration path forward, but no results are available yet.

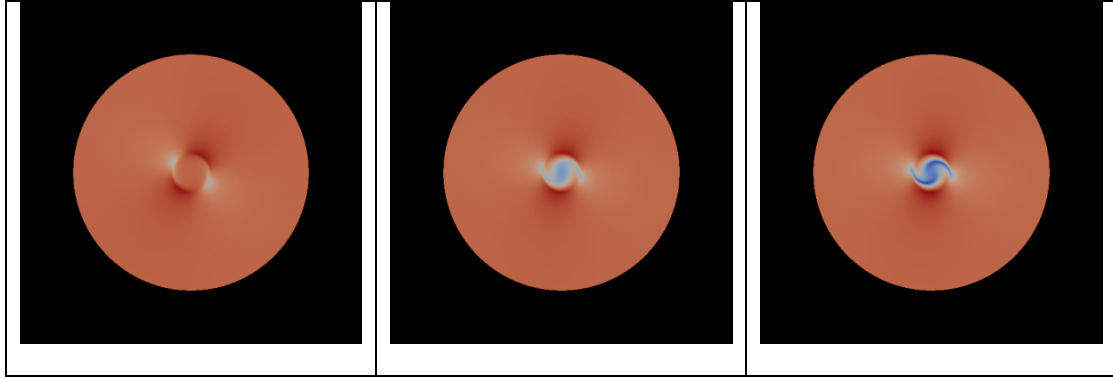


Figure 15: Time series visualization of the magnitude of the magnetic field in a test magnetohydrodynamic solver problem in Alegra, the "Perry-White Rotor Problem"

4. Demonstrate prototype capability for executing VTK components within an S3D simulation

S3D is a parallel application for simulating turbulent, reacting flows developed by scientists at the Combustion Research Facility to study fundamental turbulence-chemistry interactions. The code has been run on systems with upwards of 10,000 processors and the data it generates is large and high-dimensional. A primary goal of the scientists is to identify, classify and track features in their data in an effort to understand the complex physical phenomena contributing to events such as ignition and extinction in reacting flows. Towards this end, we are developing software that performs principal component analysis and K-Means clustering to reduce the dimensionality of the data and identify features of interest.

This case study was a simulation involving 21 chemical species reacting in a lifted jet flame. In this configuration a cold (550 K), quickly moving, planar ethylene fuel jet is surrounded by a co-flow of hot (1550 K) 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 eighth 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.

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 that 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 principal component analysis (PCA) and K-Means clustering methodologies described below.

PCA

PCA is a technique for identifying how several variables observed together are correlated across a set of observations. Recall that correlation is defined by the covariance matrix. Say we have variables

$$X = \{x_1, x_2, \dots, x_n\}$$

which are observed r times and have mean

$$\mu_X = \{\mu_1, \mu_2, \dots, \mu_n\}$$

Then their covariance matrix is

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

Note that $\text{cov}(X)$ is symmetric, positive definite, and has $(n+2)n/2$ 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 $\text{cov}(X)$ and uses these eigenvectors to project observations into a space with reduced dimensionality $m < n$. Our implementation uses the singular value decomposition (SVD) of $\text{cov}(X) = U \Sigma V^T$ to compute the eigenvectors and eigenvalues. Because $\text{cov}(X)$ 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 $v = \text{diag}(\Sigma)$ and eigenvectors \mathcal{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 \mathcal{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 \mathcal{V} are the ones we should consider. Once v and \mathcal{V} are obtained, PCA uses \mathcal{V} to project the original observations X into a smaller space, Y . This is accomplished by letting \mathcal{V}_m be the first m columns of \mathcal{V} and taking $Y = X \mathcal{V}_m$.

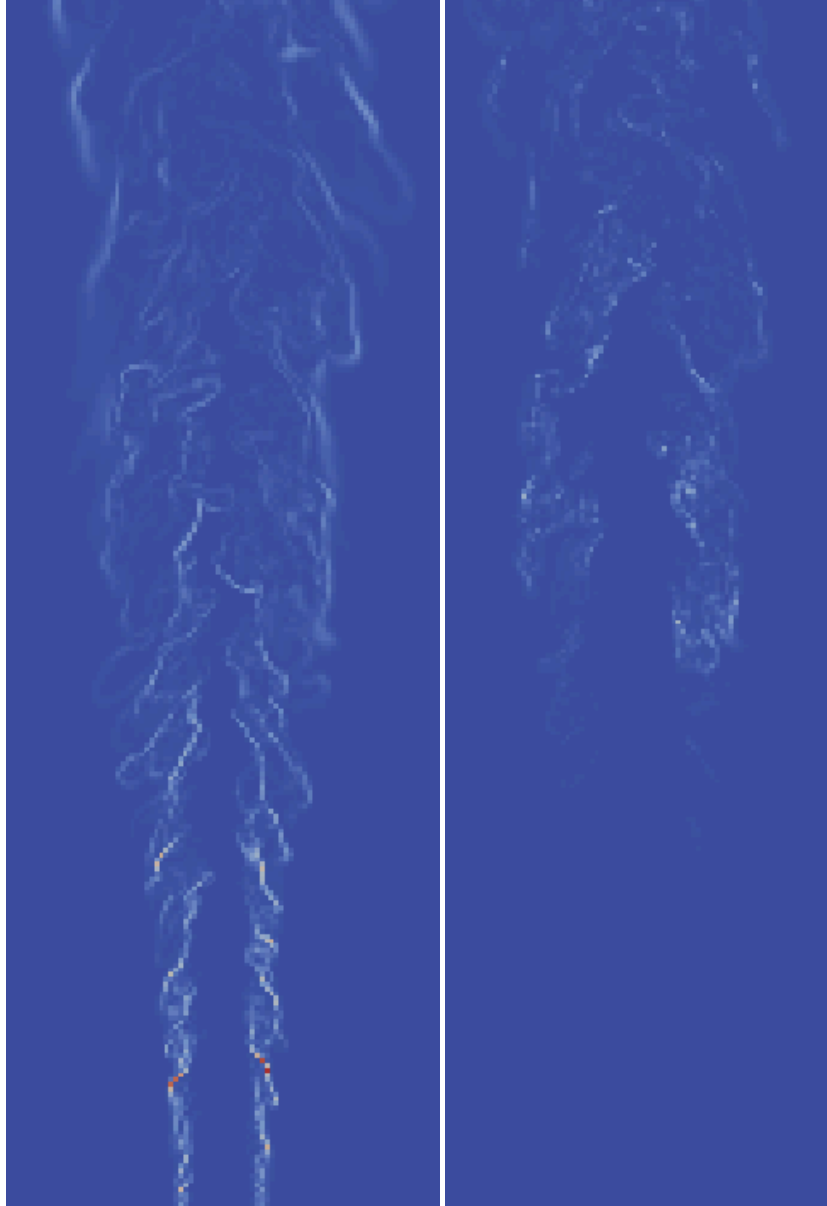


Figure 16: These two images show the magnitude of the first two eigenvalues of each $8 \times 8 \times 8$ block in the S3D simulation.

The result of this analysis is a second grid that is $8 \times 8 \times 8 = 512$ times more coarse than the original. Each cell in the second grid has a set of eigenvalues and eigenvectors associated with it and represents one $8 \times 8 \times 8$ block of the original mesh. Figure 16 shows two renderings of a z-slice in the second grid; the first shows each cell colored by the first eigenvalue and the next by the second eigenvalue. Because the simulation is periodic in the z direction, these renderings are representative of the entire domain. Figure 17 shows the same grid colored by the component of the second eigenvector related to carbon monoxide concentration. While values are provided over the entire mesh, they are only significant where the associated eigenvalue is significantly above zero. Finally, Figure 18 is a logarithmic plot of the maximum of the i^{th} eigenvalue over the entire second grid versus i . The graph consists of 2 sections, each of which is nearly linear, that meet at $i = 4$.

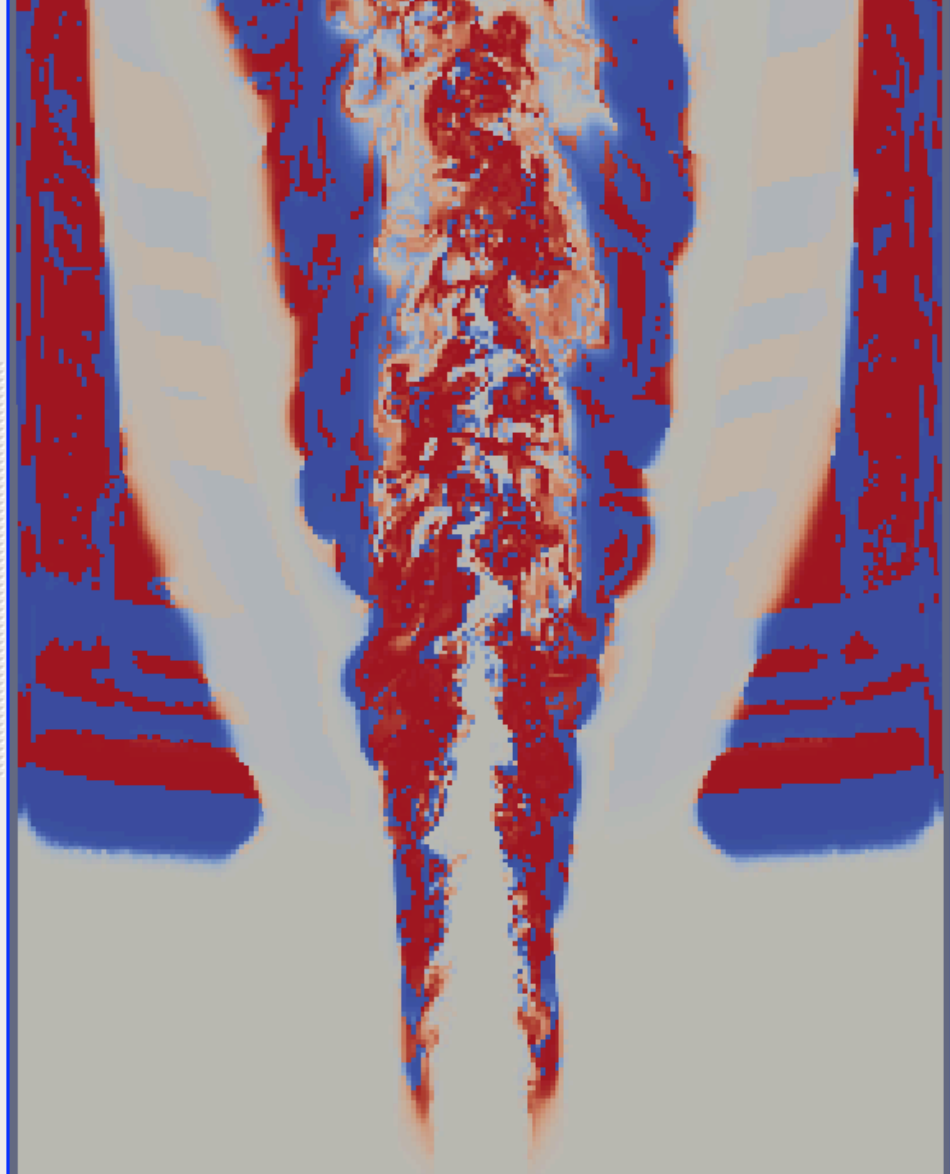


Figure 17: This image shows one component of the second eigenvector, corresponding to carbon monoxide concentration

These results are interesting because PCA appears to have captured features related to autoignition in the second eigenvector (Figure 16, right). We suspect, due to the shape of the graph in Figure 18, that the first 4 eigenvectors contain information relevant to ignition and extinction events.

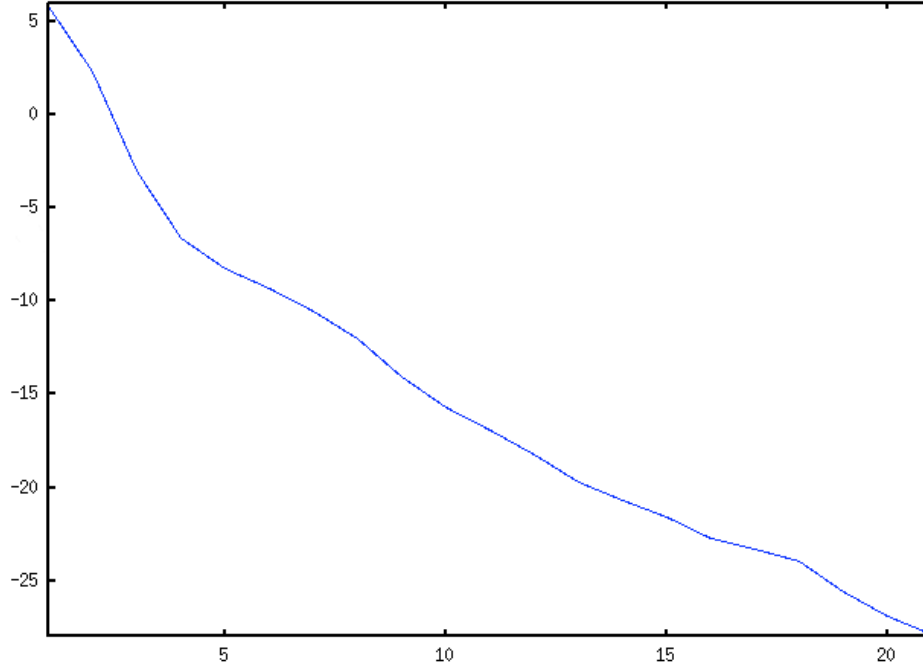


Figure 18: This plot shows for each of the 21 eigenvector/eigenvalue pairs, the log of the largest eigenvalue over the entire image.

The simulation from which these results were obtained was run on over 10,000 nodes of the Jaguar platform at Oak Ridge. Because no large simulations have been run since the development of the *in situ* framework, data for these figures was obtained by replaying checkpoint data in S3D on 1,500 nodes of Jaguar as if it were being simulated and then using the *in situ* framework described in this report to compute covariances, compute eigenvectors, and then save the results for later inspection. The eigenvalues and eigenvectors are only a small fraction of the full simulation state.

K-Means Clustering

The term cluster analysis is used to describe a series of unsupervised learning algorithms that partition objects into groups according to a measure of association. K-Means clustering [Mac67] is one such algorithm that classifies objects into k clusters, with $k > 0$, by minimizing some distance metric between the data and the corresponding cluster centers.

The general algorithm consists of the following steps:

1. Choose the number of clusters, k .
2. Determine k initial cluster centers.
3. Assign each data observation to the nearest cluster center.
4. Re-compute the new cluster centers.
5. Repeat the steps 3 and 4 until a prescribed convergence criterion is met (either the maximum number of iterations is reached or the number of cluster assignment changes reaches some minimum value).

A parallel version of the K-Means clustering algorithm was implemented within VTK and is currently deployed within the *in situ* analysis framework for S3D. After performing PCA, each eigenvalue/eigenvector pair is a potential statistical observation for the K-Means clustering algorithm (an option is provided to filter out pairs whose eigenvalue falls below a particular threshold). Initial studies show that cluster analysis will facilitate further dimensionality reduction of S3D data in a meaningful way, however there are two main

disadvantages of the naïve implementation: the value of k must be pre-determined and the results are heavily reliant on the initial cluster centers used.

Relatively recent work [ZHD+01, DH] has proven that the continuous solutions of the discrete K-Means clustering membership indicators are the data projections on the principal eigenvectors of the associated covariance matrix. This is equivalent to showing that the cluster centroids are given by spectral expansion of the data covariance matrix truncated at $k-1$ terms. An effort is under way to implement a spectral version of the K-Means algorithm, and methods are being investigated in which topological analysis in conjunction with parallel coordinates may provide additional insight into the appropriate number of clusters as well as their initial starting points. Figure 19 shows parallel coordinate plots of the observations for a subset of the eigenvalue/eigenvector pairs of S3D data, with sample clusters highlighted in red.

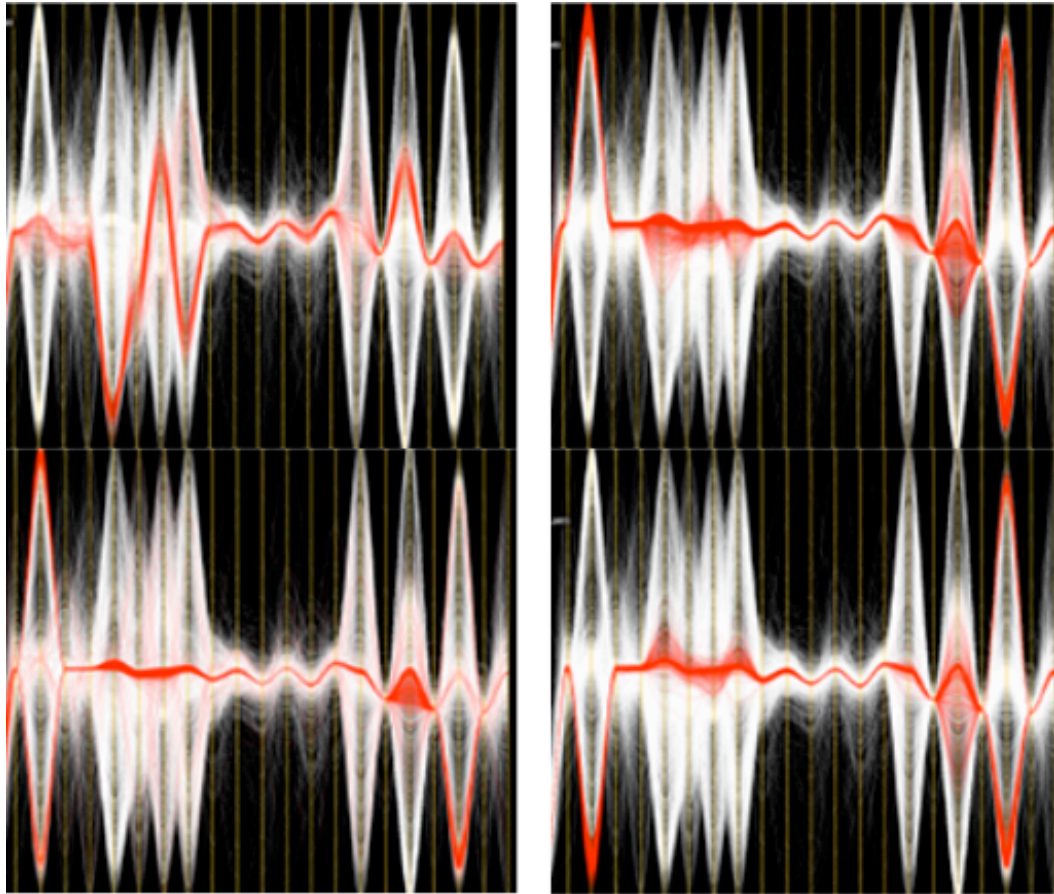


Figure 19: Parallel coordinate plots of observations may provide insight into initial cluster center values in addition to the ideal number of clusters for a set of observations. These images are plots of parallel coordinates for eigenvalue/eigenvectors pair observations generated from a subset of the S3D data. Potential clusters are drawn in red.

5. Demonstrate fragment identification within ParaView on other formats:
 - a. Exodus file set

The current version of the fragment identification filter only works on Adaptive Mesh Resolution (AMR) CTH data. This stretch goal seeks to extend this feature recognition code to work on Exodus files in addition to AMR CTH files. Figure 20 shows the result of running a version of the fragment detection filter on an Exodus dataset. The visualization shows a simulation of a pipe bursting and is colored by fragment ID. While this capability is currently only available in serial executions of ParaView, work is ongoing to support parallel execution.

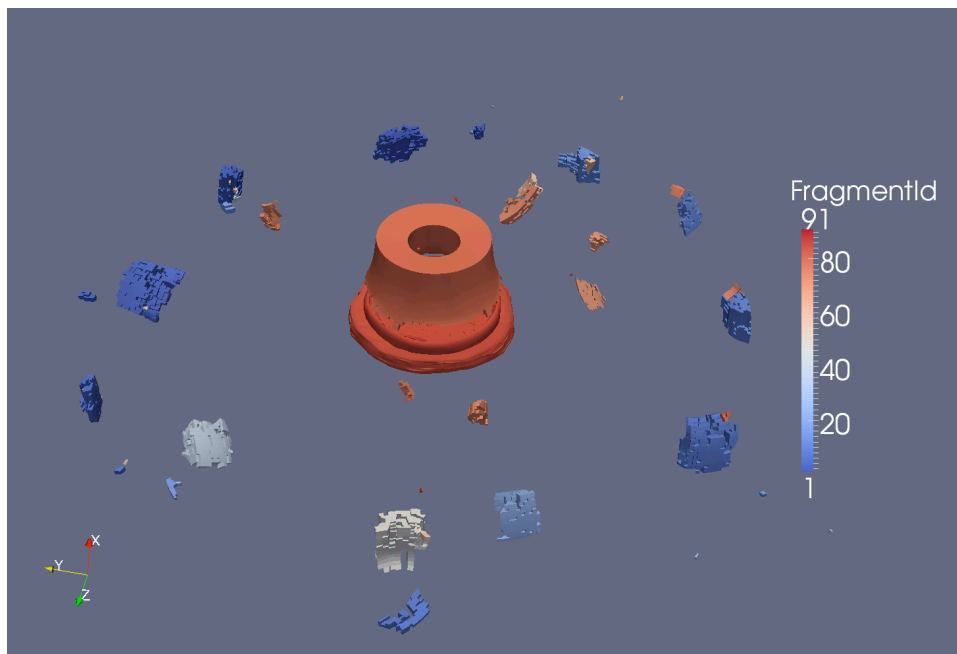


Figure 20: This image shows the fragment filter run on an Exodus simulation of a pipe bursting.

6. Demonstrate visualization of CTH tracer particles within ParaView

CTH provides the ability to attach tracer particles to points on the model. As the simulation advances, the tracers follow the point they were initially attached to. This stretch goal allows ParaView to visualize the current position of tracers as well as see the path they took. Figure 21 shows a visualization of tracer particles from the ball brick simulation. A movie showing the visualization over time is available through Web Fileshare (WFS1030872).

Currently, tracer particles can be extracted from spth geometry files. The team plans further extend this capability, including developing the ability to extract tracer particles from the higher temporal fidelity history ("hisplot") files.

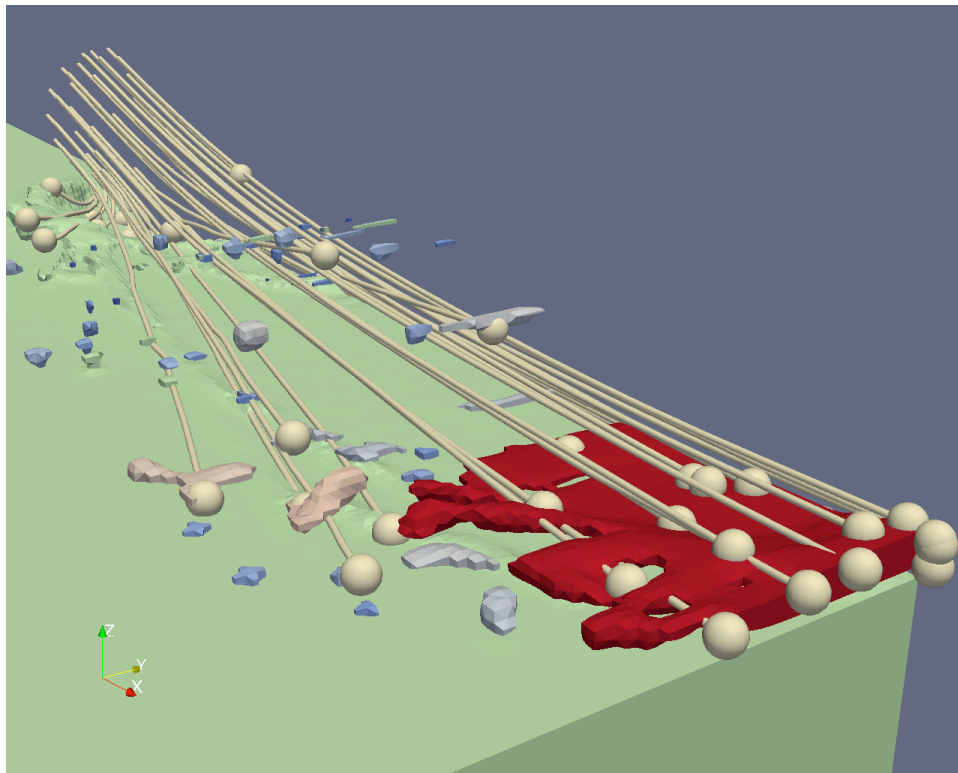


Figure 21: Visualization of CTH tracer particles. The spheres show where the tracers are at the current time step. The lines show the path they traveled.

7. CONCLUSION

A number of new feature recognition capabilities were developed by this project to meet all of the minimum success criteria and several of the stretch goals of Milestone 3160. These new capabilities will aid verification and validation performed for sensitivity analysis and uncertainty quantification (UQ). These feature recognition capabilities include:

- crater detection, characterization, and analysis from CTH simulation data;
- the ability to call feature recognition and visualization code contained with ParaView and its underlying toolkit, VTK, from within a CTH simulation;
- fragment identification in unstructured mesh data in the Exodus format, albeit performed serially at this point;
- output of extracted features in a geometric format that includes data values; and
- utilization of VTK components to identify and visualize features within S3D.

In addition, for this milestone, the project developed visualization capability for CTH tracers and performed scalability analysis on the fragment identification algorithm.

References

- [DH] Chris H. Q. Ding and Xiaofeng He. K-means clustering via principal component analysis.
- [Mac67] J. MacQueen. Some methods for classification and analysis of multivariate observations. In L. M. LeCam and J. Neyman, editors, *Proc. of the 5th Berkeley Symp. On Mathematics Statistics and Probability*, 1967.
- [ZHD+01] Hongyuan Zha, Xiaofeng He, Chris H. Q. Ding, Ming Gu, and Horst D. Simon. Spectral relaxation for K-means clustering. In Thomas G. Dietterich, Suzanna Becker, and Zoubin Ghahramani, editors, *Advances in Neural Information Processing Systems 14* [Neural Information Processing Systems: Natural and Synthetic, NIPS 2001, December 3-8, 2001, Vancouver, British Columbia, Canada], pages 1057–1064. MIT Press, 2001.

DISTRIBUTION

1	MS0321	J. S. Peery	01400
1	MS0370	T. G. Trucano	01411
1	MS0384	A. C. Ratzel	01500
1	MS0801	R. W. Leland	09300
1	MS0807	J. P. Noe	09328
1	MS0822	D. B. Karelitz	09326
1	MS0822	L. Ice	09326
1	MS0822	W. A. Scott	09326
1	MS0823	C. J. Pavlakos	09326
1	MS0823	J. D. Zepper	09320
1	MS1323	K. D. Moreland	01424
1	MS1323	D. Rogers	01424
1	MS1323	N. Fabian	01424
1	MS1322	S. S. Dosanjh	01420
1	MS0783	J. Wilke	06475
1	MS0847	S. W. Attaway	01534
1	MS0828	R. G. Hills	01544
1	MS9159	D.C. Thompson	08963
1	MS9159	J. C. Bennett	08963
1	MS0899	Technical Library	09536 (electronic copy)

