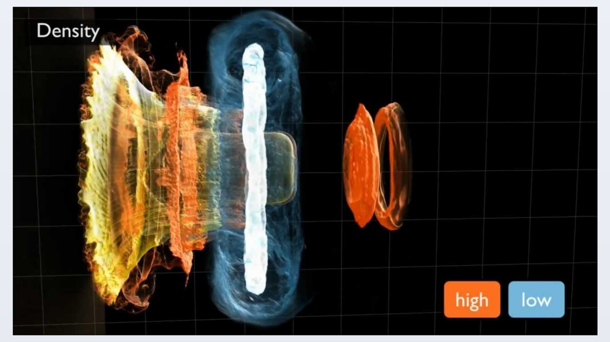
Bubble Shock Simulation Analysis

Rebecca Petersen, Matt Oehler, Andrew Wheeler, Jared Hoff (BYU)
Ming Jiang, Brian Gallagher, Cyrus Harrison (LLNL)

Background

The bubble shock experiment is set up by having a bubble of helium and shooting a shock wave through it to experiment with how it deforms over time. Results of these experiments have helped in the fields of industry (explosions and material integrity), medicine (effects of lasers and ultrasounds), and energy production (Sonoluminescence - the change of acoustic energy to optical energy) to name a few.

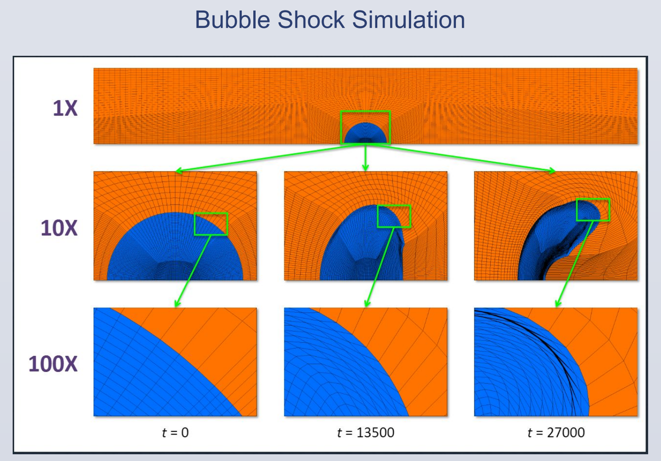


A Helium Bubble hit with a Mach 3 shock wave [1]

Introduction

LLNL carries out long running physics simulations in their high performance computing environment. These simulations are highly complex and sometimes fail, wasting hundreds of hours of work and computer time. LLNL has large amounts of telemetry data from failed simulations and would like to use that data to predict failures before they happen.

Using 560 GB of data from 140 failed bubble shock simulations, our team at BYU worked with LLNL employees to analyse and identify patterns in the data that could be used to predict simulation failure.



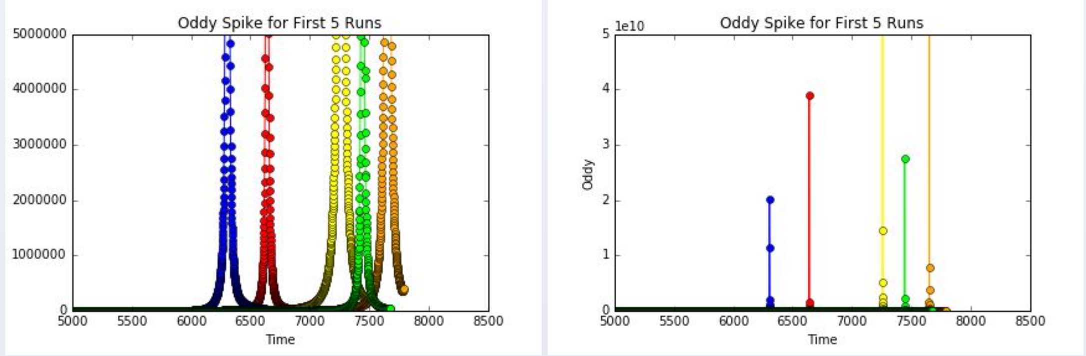
Failure at t=27000 when zones near the border of the shockwave deformed too far, causing the simulation to crash

Objectives

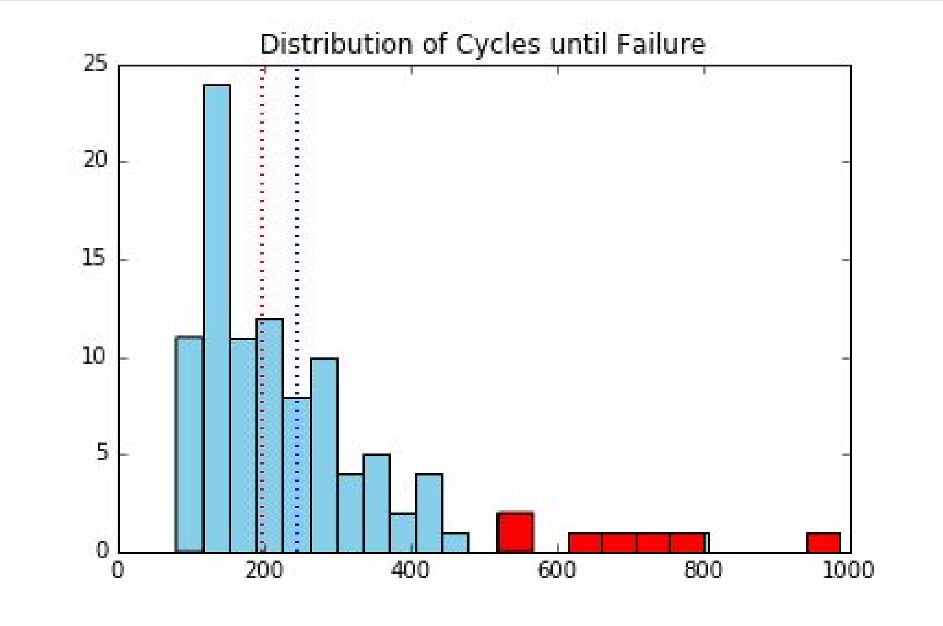
1) Determine factors that cause simulations to fail
   a) Verify the preliminary data found by LLNL and research if it is a viable method of predicting failure
   b) Find patterns in the variables that appear in the failed zones, but not in the passing zones
2) Come up with a reliable way to predict if a simulation is going to fail
   a) If the run is predicted to fail, when will it fail
3) Predict the failure early enough so that the simulation can be stopped and another one started, preventing lost time.

Results - Oddy Analysis

It was discovered in the data for our 140 failed simulations that all of the failed zones had a spike in the "Oddy" variable. The figures below show us the Oddy spikes for the first five failed simulations. As we can see the spikes vary in duration, magnitude and timing. Note: the only difference between the two plots is the scale of the y-axis



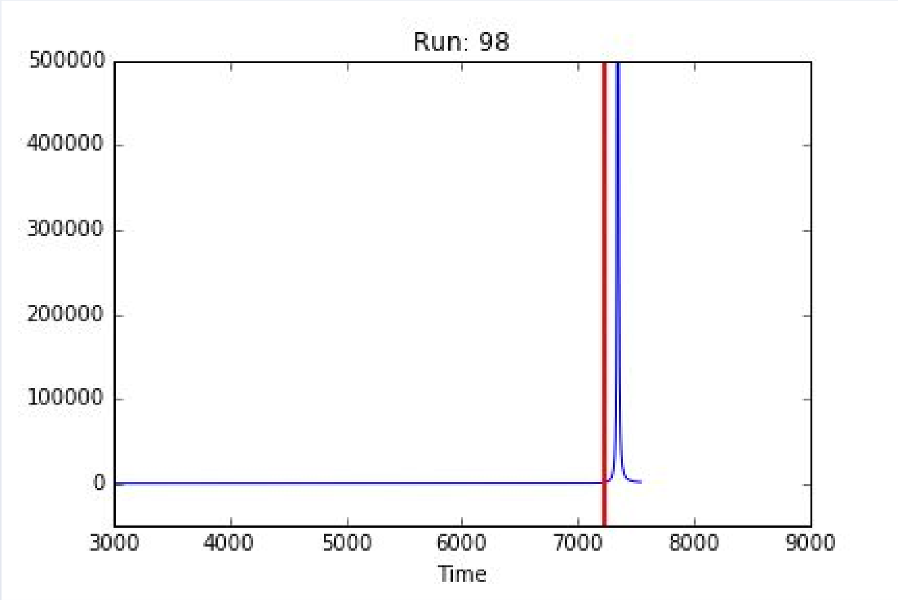
One approach we tried was to see if there were any noticeable patterns between when the peak of the Oddy spike occurred, and when the simulation failed. The distribution of cycles until failure for our failed simulations is shown below. However, this isn't helpful for us to predict anything because it doesn't give us any information about when the simulation will fail until after it happens. (Note: The red sections are considered outliers, the blue line is the mean of the distribution, and the red line is the median of the distribution.)




```
Number of Runs:  99
Outlier Index:  [6, 35, 43, 44, 74, 75, 89]
Outlier Values [720, 551, 673, 794, 521, 637, 987]
Mean:  245.04040404
Median  198.0
Standard Deviation:  158.929113173
Variance:  25258.463014
```

Results - Oddy Analysis

Since we can't use time until failure for predicting, we developed a rule using the standard deviation of an oddy value for a zone given its preceding values. We found that we could identify a spike if the oddy value at cycle t is more than 16 standard deviations away from the mean oddy value for cycles 0 to t - 1 . We assessed the validity of this rule by using it to evaluate 3 samples of 500 zones, and we didn't have any false positives or false negative when it came to predicting a spike.

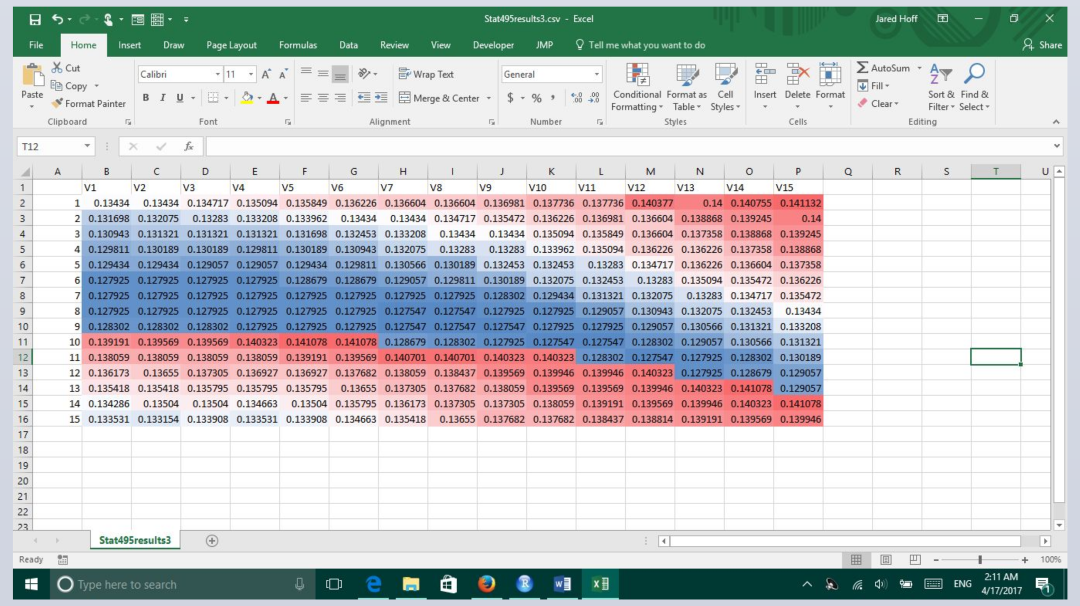


Simulation 98 - the red line is where we determined that the spike starts based off of the standard deviation

We also created a logistical regression of the Oddy variable. The logistical regression gives us the odds of failure occurring within x cycles based off of what is happening y cycles before hand.

This method caught 100% of failed zones and 0% catch any of actually passing zones. It also caught 9% of the zones in failed simulations that we believe were most likely to fail, but hadn't failed yet.

Overall we found that if the average value of the last 400-500 cycles was 150-155 then the simulation was bound to fail within 700-900 cycles.

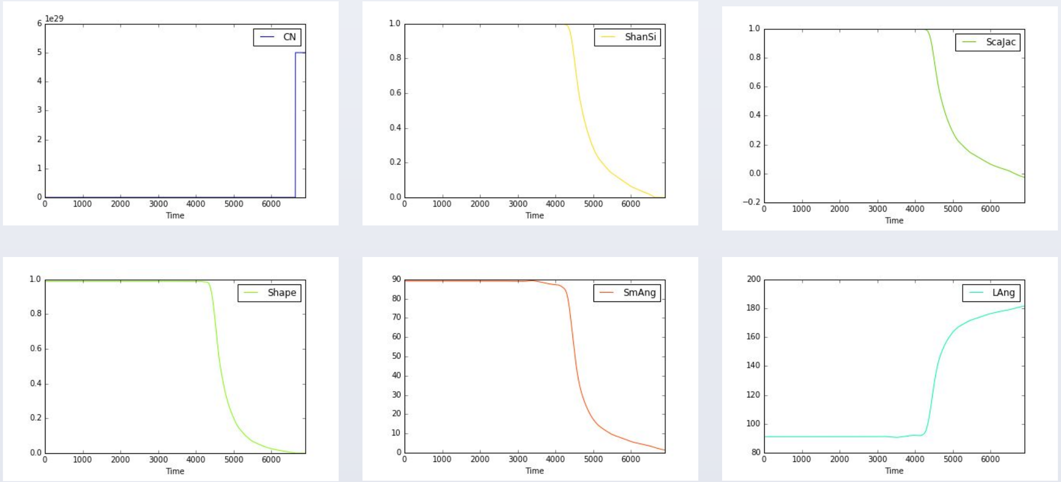


Logistical regression of the Oddy variable. The Blue areas are good predictions and the red areas are poor predictions.

Results - Random Forest

We were able to narrow the failing zones down to two sections of the simulation, and this in turn cut down the number of zones that we analyzed from 11,650 to approximately 1,400 zones.

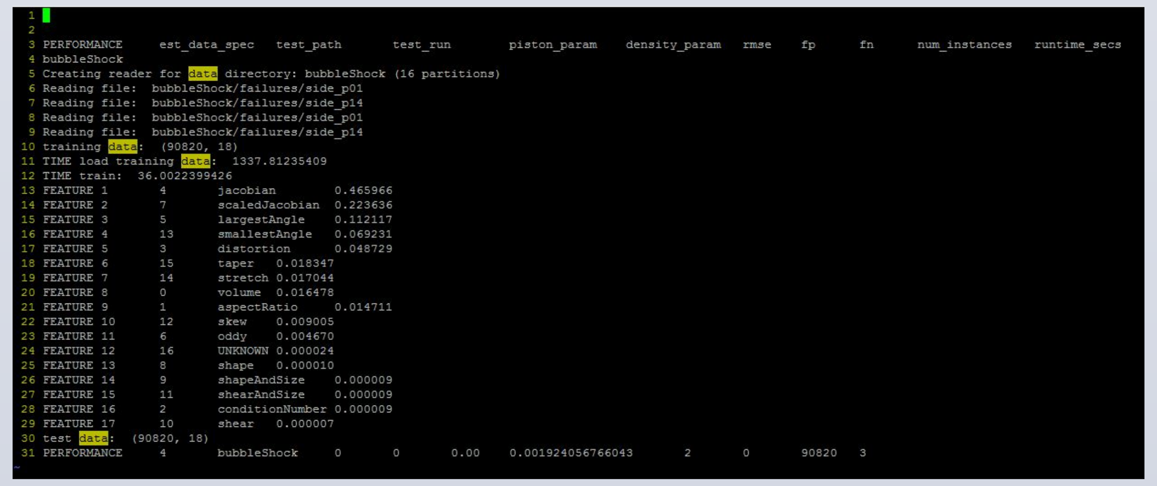
Early on we pulled preliminary data about each of the 16 variables and found trends in the failing zones. For the most part these trends were not present in the passing zones, but they did appear in a couple of the zones that we predicted were on the way to failure and would have failed if the simulation could have continued.



Graphs of the value of 6 variables for one of the failing zones over time.

We integrated our Oddy rule with LLNL's random forest algorithm to find the variables that were most effective in predicting failure. The higher up the list the UNKNOWN variable that we created was in the list, the more effective that it could be used as a predictor.

A lot of the variables had a specific range and in failed zones they would hit the max or the min of that range. We started UNKNOWN variable with 0. If it hit the max/min of the range we added 100 to the UNKNOWN variable. If it wasn't quite there we would add a portion of the 100 points to the UNKNOWN variable depending on how close it was to the max/min.



Output from RandomForest. Our new variable shows up 12th on the list

Although the UNKNOWN variable has lower overall predictive power than Oddy alone, it can give an accurate prediction of failure much sooner. With more time we would have liked to try different ratios and values that were added to the total for the UNKNOWN variable. We would have also liked to work with different combinations of the trending variables and find a way to use the other variables that we weren't able to easily quantify.

Conclusions

- Verified LLNL's preliminary analysis: found spikes in variables in all failed simulations that were not present in passing simulations
- By monitoring the Oddy values during a simulation and using our developed rule, we should be able to preemptively stop and adjust a simulation prior to failure
- Logistic regression is helpful to classify zones that are likely to fail
- When it comes to Big Data projects, narrowing the scope of the project yields better results than trying to solve too many problems at once

References / Materials

Python- a programming language that it very intuitive and useful for data analysis

Scikit learn- a python library commonly used to implement machine learning algorithms such as random forests.

Numpy- a python library used for performing mathematical operations

Matplotlib- a python library used to create graphic representations of data
R- a widely used statistical programming language

[1]B. Hejazialhosseini, D. Rossinelli and P. Kumoutsakos, "Mach 3 Bubble Shockwaves", *YouTube*, 2012. [Online]. Available: https://www.youtube.com/watch?v=gnhbhpwTx1c. [Accessed: 17- Apr- 2017].

Contact

Address and Phone Numbers
Lawrence Livermore National Laboratory
7000 East Ave., Livermore, CA 94550-9234
P.O. Box 808, Livermore, CA 94551-0808 (Mail)
Main Operator 925-422-1100