Internship Presentations

Paul Crittenden
Mohamed Gadou
Trokon Johnson
Yash Mehta

Contact Front Instability and Code Verification

Lawrence Livermore National Laboratory
Spring 2017

Paul Crittenden
PhD Student
Mechanical and Aerospace Engineering Department
University of Florida
Introduction

- Mentor Dr. Kambiz Salari (Sam Schofield, Tim Dunn)
- Worked on analytical stability results
- Ran shock tube test cases using ALE3D (Eulerian-Eulerian)
- ALE3D is a multi-physics numerical simulation software tool
- Main goal of the work was verification of my multiphase code

Single-phase Instability

- Assumptions about contact (Epstein)
  - Spherical Harmonics
  - Spatially constant density
  - Irrotational base and perturbed flow
  - Equal compression rates (removed)
  - Incompressible perturbations
Two-Phase Verification-ALE3D

Particle Volume Fraction

Pressure

Gas Density

Particle Velocity

Gas Velocity

Gas Temperature

Particle Temperature

N=2540
N=5080
N=10160

Two-Phase Verification-AUSM1D

Particle Volume Fraction

Pressure

Gas Density

Particle Velocity

Gas Velocity

Gas Temperature

Particle Temperature

N=2540
N=5080
N=10160
Activities at LLNL

- Seminars
- End of internship presentation

Bay to Breakers!
My Internship at LANL

Mohamed Gadou
Computer and Information Science and Engineering
University of Florida

- Mentors:
  - Galen Shipman
  - David Daniel
  - Ryan Braithwaite

Interning at The Lab

- Amazing environment for learning
- Organized workspace and excellent mentorship
- Opportunity for interacting with best minds in various Science fields
- Great Friendly and Helpful coworkers
- Aligned Values to Personal Values (Integrity, Safety, Security, Respect, ..)
### Architecture Description

**Measuring Performance, Power, and Energy**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| IBM Power 8 + P100    | - IBM power8 server with 2 sockets  
|                       | - Reported TDP of 190 watts per socket  
|                       | - NVLink to reduce data transfer overhead between CPUs/GPUs  
|                       | - 2 Nvidia Tesla P100 GPUs |
| Intel Broadwell – 2 sockets | - Intel Broadwell family E5 with 2 sockets  
|                        | - TDP of 135 watts per socket |
| Intel Haswell + K80   | - Intel Haswell family E5 with 2 sockets  
|                        | - TDP of 120 watts per socket  
|                        | - 2 of ½ NVIDIA Tesla K80 GPUs |
| Intel KNL             | - Intel Knights Landing with 72 cores  
|                        | - High Bandwidth memory is available (MCDRAM)  
|                        | - Multiple MCDRAM Configurations  
|                        | - MCDRAM size is 16 GB |

---

### Life at Los Alamos

- [Image: Los Alamos Landscape](image1)
- [Image: Los Alamos Night Sky](image2)
Overview:
Parallel Computing Summer Research Institute

- 10 Week Program
  - Crash Course of HPC Topics
    - MPI
    - Vectorization
    - Scrum
    - Quantum Computing
  - Special Projects
    - Small group projects for the duration
  - Contacts
    - Experts in different HPC areas
Project: SPORCO Acceleration

- **Goal:**
  - Sparse Coding: efficiently represent signals using dictionaries
    - can be used for denoising, feature identification, etc
  - Dictionary Learning: creates dictionary elements by training on signals

- **Contribution**
  - Parallelized existing convolutional dictionary learning library (SPORCO)
    - mpi4py

- **Impact**
  - Significant reduction in time to create trained dictionary elements
  - Allows for largest convolutional dictionary problem solved, to our knowledge

**Sparse Coding**

\[
\arg\min_{\{x_m\}} \frac{1}{2} \sum_m d_m * x_m - s_m \|_2^2 + \lambda \sum_m \|x_m\|_1
\]

**Dictionary Update**

\[
\arg\min_{\{d_m\}} \frac{1}{2} \sum_k \sum_m d_m * x_{k,m} - b_k \|_2^2
\]

---

**Results**

- Great weak scaling results
- Allowed for much larger tests
  - 256x256 -> 1024x1024
  - 100 Images
Excursions

Excursions, pt. 2
Introduction

- Mentor Dr. Kambiz Salari
- Worked on analyzing data from ALE3D simulations
- ALE3D is a multi-physics numerical simulation software tool using arbitrary Lagrangian-Eulerian (ALE) techniques
- Main goal of the work was to understand motion of multiple particles under detonation conditions
Shock Interacting with Moving Particle(s)

\[ M_x = 3.00 \; \varphi = 10\% \; \rho_p/\rho_f = 530 \]

Drag single particle; \( M_x = 3.00 \)

Area fraction; \( M_x = 3.00 \; \varphi = 10\% \)

Pressure through domain; \( M_x = 3.00 \; \varphi = 10\% \)

Activities at LLNL

- NIF tour
- Summer Interns barbecue
- Summer seminar series (various programs)
- Computation program – “Apple Time”
- Computation seminar – end of internship presentation
Life around Livermore

Do you have any questions?
Surrogate Modeling of the JWL Equation of State

Frederick Ouellet
PhD Student

Goal – Handling Cells with Mixed E.O.S.

- A short time after detonation, there are both pure explosive products and pure air
- At later time, pure species still exist but a mixture region will form where two equations of state must be satisfied
- Goal is to find a way to simplify and speed up the gas computations in this zone
**Standard Approaches for Mixed Cells**

<table>
<thead>
<tr>
<th>Method of Tackling</th>
<th>Pros</th>
<th>Cons</th>
</tr>
</thead>
</table>
| Mass fraction weighted averaging   | • Algebraic equations  
• Small data storage                                               | • Does not place species     |
| Tabulated data                     | • Interpolation is only operation  
• Can place species equilibrium requirement                        | • Must store the tabulated data  
• Interpolation error                                                  |
| Polynomial regressions/curve fits  | • Algebraic equations  
• Small data storage  
• Can place species equilibrium requirement                        | • Curve fit errors           |
| Iterative schemes                  | • Small data storage  
• Can place species equilibrium requirement                        | • Computationally expensive |

- Rocflu uses a Broyden’s method iterative solver to enforce pressure and temperature equilibrium between air and the explosive products.

**The Iterative Solver**

1. Loop over all cells in domain
2. IF $Y_{exp} < 0.001$ THEN Ideal Gas Equations ELSE JWL Equations
3. Broyden’s Method Iteration
   - Inputs: $\rho_m, e_m, Y_{exp}$
4. Calculate Initial guesses, values for $f_1$ to $f_4$ and Jacobian matrix (4 exponential operations)
5. Invert Jacobian matrix
6. Update Solution (1 matrix multiply)
7. IF converged THEN Break, Calculate P and T ELSE Update Inverse Jacobian (2 matrix multiplies, 4 exponential operations)
8. ELSE Update Inverse Jacobian (2 matrix multiplies, 4 exponential operations)
What is an iterative scheme really doing?

Inputs:
\[ \rho m, e', V_{exp} \]

Outputs:
1) Common pressure/temperature for both EoS
2) Internal energy attributed to the air and the products
3) Density attributed to the air and the products

If we compute the common pressures and temperatures outside of the code, a curve fit can be applied.

The Jones-Wilkins-Lee (JWL) equations of state are used to predict the pressures of high energy substances and are:

\[
\begin{align*}
P_{JWL}(\rho, \epsilon) &= A \left(1 - \frac{\epsilon}{\rho c_s^2}\right)e^{-\frac{\epsilon}{\rho c_s^2}} + B \left(1 - \frac{\epsilon}{\rho c_s^2}\right)e^{-\frac{\epsilon}{\rho c_s^2}} + \omega \\
T_{JWL}(P, \rho) &= \frac{1}{\rho u c_s} \left( P - A e^{-\frac{\epsilon}{\rho c_s^2}} - B e^{-\frac{\epsilon}{\rho c_s^2}} \right)
\end{align*}
\]

where \( V = \frac{\rho u}{\rho} \) and \( \rho, A, B, C, R_1, R_2 \) and \( \omega \) are parameters for the substance.

<table>
<thead>
<tr>
<th>Advantages</th>
<th>Iterative Method</th>
<th>One Equation Model</th>
<th>Multi-Fidelity Surrogate</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Accuracy</td>
<td>- Speed</td>
<td>- Speed</td>
<td>- Speed</td>
</tr>
<tr>
<td>- Problem independent</td>
<td>- Problem</td>
<td>- Algebraic</td>
<td>- Accounts for species</td>
</tr>
<tr>
<td>- Speed</td>
<td>- Independent</td>
<td>- equation</td>
<td>- equation</td>
</tr>
<tr>
<td>- Algebraic equation</td>
<td>- Speed</td>
<td>- accounts</td>
<td>- accounts</td>
</tr>
<tr>
<td>- Speed</td>
<td>- Independent</td>
<td>- delay</td>
<td>- delay</td>
</tr>
</tbody>
</table>

| Disadvantages                  | Run models       | Uncertainty and   | Uncertainty               |
|                                | of energy        | JWL + ideal gas   | Equation of State         |
|                                | inefficient      | case only         | specific (problem specific) |
|                                | May be slow to   | JWL + ideal gas   | Uncertainty               |
|                                | converge         | case only         | Equation of State         |
|                                | Computationally  | JWL + ideal gas   | specific (problem specific) |
|                                | expensive        | case only         | specific (problem specific) |

| Used In                        | HR 2             | HR 3/4/5           | Future HRs               |
Kriging method used to generate two models for mixed cell pressure and temperature in terms of $\rho_m$, $e_m$ and $Y_{exp}$ using 50-800 sampling points.

Domain Space: $\rho_m \rightarrow (1, 1.770)$
$e_m \rightarrow (1.25e5, 1.5e7)$
$Y_{exp} \rightarrow (0, 1)$

Average relative errors (for 200 point model):
- Pressure: 1.298%
- Temperature: 0.132%
Surrogate Model Results – Timing

**Standalone Surrogate**
- Data taken from calculations of pressure/temperatures of 1000 mixture states
- Surrogate model with 200 points performed calculations 40.5 times faster than the iterative solver

<table>
<thead>
<tr>
<th>Model</th>
<th>Computing Time(s)</th>
<th>Speed-up</th>
</tr>
</thead>
<tbody>
<tr>
<td>50 Pts</td>
<td>0.0087</td>
<td>111.81</td>
</tr>
<tr>
<td>100 Pts</td>
<td>0.0127</td>
<td>76.61</td>
</tr>
<tr>
<td>200 Pts</td>
<td><strong>0.0240</strong></td>
<td><strong>40.51</strong></td>
</tr>
<tr>
<td>400 Pts</td>
<td>0.0396</td>
<td>24.60</td>
</tr>
<tr>
<td>800 Pts</td>
<td>0.0766</td>
<td>12.70</td>
</tr>
<tr>
<td>Iterative</td>
<td>0.9733</td>
<td>1.00</td>
</tr>
</tbody>
</table>

Surrogate Model Results – Accuracy

**Standalone Surrogate**
- Data taken from same calculations of 1000 mixture pressure/temperatures
- The 200 point model is the smallest model with 99% correlation for both pressure and temperature

<table>
<thead>
<tr>
<th>Sampling Pts</th>
<th>$R_{\text{Pressure}}$</th>
<th>$R_{\text{Temperature}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>0.9044</td>
<td>0.9981</td>
</tr>
<tr>
<td>100</td>
<td>0.9247</td>
<td>0.9984</td>
</tr>
<tr>
<td>200</td>
<td><strong>0.9905</strong></td>
<td><strong>0.9986</strong></td>
</tr>
<tr>
<td>400</td>
<td>0.9952</td>
<td>0.9987</td>
</tr>
<tr>
<td>800</td>
<td>0.9965</td>
<td>0.9992</td>
</tr>
</tbody>
</table>
Simulation Results – Timing

- Simulation is run until first appearance of a species mixture cell.
- 1000 time steps are run for each EoS model to gather timing data.
- Process is repeated five times and averaged.
- Notable code speed-up factor:
  - 200 points: 2.63

Simulation Description
- 2D, Cylindrical grid
- Outer radius = 0.3 m
- 400,000 cells
- Gas-Only

Simulation Results – Accuracy

- Maximum errors in blast wave location:
  - 50 points: 6.85%
  - 200 points: 0.83%
  - 800 points: 0.35%
- Maximum errors in peak pressure:
  - 50 points: 14.06%
  - 200 points: 8.52%
  - 800 points: 3.41%
Conclusions and Future Work

Conclusions:
- Surrogate models allow for faster computations of the equation of state in mixture cells
- The models produce results that approach the iterative scheme given a sufficient number of sampling points in the input space

Future Work:
- **Multiple Species:**
  - Using an iterative solver, this means going from a $4 \times 4$ system of equations to an $2n \times 2n$ system of equations
  - As long as the same number of sampling points are used to create a new surrogate, the computation time would remain fixed
- **Multiphase Flows:**
  - Tests to verify that shifting to the surrogate models does not impact the particle metrics

Do you have any questions?

This work was supported by the U.S. Department of Energy, National Nuclear Security Administration, Advanced Simulation and Computing Program, as a Cooperative Agreement under the Predictive Science Academic Alliance Program, under Contract No. DE-NA0002378.
Surrogate-Based Optimization for Exploring the Physics in Explosive Dispersal of Particles

M. Giselle Fernández-Godino
PhD Student (UB-Physics)

Departure from Axisymmetry

- Relatively axisymmetric
- Markedly non-axisymmetric

Highly axisymmetric

<table>
<thead>
<tr>
<th>Image</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_0$</td>
<td>Relatively axisymmetric</td>
</tr>
<tr>
<td>$t_1$</td>
<td>Relatively axisymmetric</td>
</tr>
<tr>
<td>$t_2$</td>
<td>Markedly non-axisymmetric</td>
</tr>
<tr>
<td>$t'_1$</td>
<td>Relatively axisymmetric</td>
</tr>
<tr>
<td>$t'_2$</td>
<td>Markedly non-axisymmetric</td>
</tr>
</tbody>
</table>
Questions

- What is a good metric of departure from axisymmetry?
  - Difference between the Particle Volume Fraction (PVF) of the sector with most particles and the one with least.

- Which initial disturbances amplify most the departure?
  - Multimodal initial PVF perturbations
  - Optimization
  - Surrogate models

Parametrization of Initial PVF Perturbation

$$\varphi^P(\theta) = \varphi^P_0 [1 + A_1 \cos(k_1 \theta) + A_2 \cos(k_2 \theta + \Phi_{12}) + A_3 \cos(k_3 \theta + \Phi_{13})]$$
**Parametrization of Initial PVF Perturbation**

\[ \varphi_P(\theta) = \varphi_0^P \left[ 1 + A_1 \cos(k_1 \theta) + A_2 \cos(k_2 \theta + \Phi_{12}) + A_3 \cos(k_3 \theta + \Phi_{13}) \right] \]

- **Effective** \( k = \frac{\sum_{i=1}^n k_i}{n} \)
- **Effective** \( A = \sqrt{\sum_{i=1}^n A_i^2} = 0.1 \sqrt{2} \)

**Wavenumber**

- \( k = 4 \)
- \( k = 6 \)
- \( k = 8 \)
- \( k = 16 \)

**Amplitude**

- \( A_1 \)
- \( A_2 \)

\( A_1 > A_2 \)

**Phase Shift**

- \( \Phi_{12} \)

**Illustration from Simulations**

- Single mode, \( k = 10 \), \( A = 0.1 \sqrt{2} \)

- Computational particles on top of density contours
  - Single mode, \( k = 10 \), \( A = 0.1 \sqrt{2} \)
- Initial sinusoidal angular particle volume fraction (PVF) perturbations translate in finger like structures at later times
- Particles travel faster in sectors where PVF is initially lower
Departure from Axisymmetry

\[ \xi(t) = \frac{\text{Norm. Max. PVF Diff}(t)}{|\text{Max}(\text{PVF}(\theta)) - \text{Min}(\text{PVF}(\theta))|(t)} \]

- PVF per sector = \( \frac{\text{Volume of particles}}{\text{Volume of the sector}} \)
- PVF of sector with most particles minus PVF of sector with least particles, divided by this difference at \( t=0 \)
- This metric does not take into account radial variations

Computational domain is divided in as many sectors as cells in \( \theta \)

Runs: Summary

<table>
<thead>
<tr>
<th># Runs LF/HF</th>
<th>Machine</th>
<th>Case</th>
<th>Individual Time LF/HF</th>
</tr>
</thead>
<tbody>
<tr>
<td>14/6</td>
<td>HPG</td>
<td>1 parameter</td>
<td>15h/24h</td>
</tr>
<tr>
<td>40/40</td>
<td>HPG</td>
<td>2 parameters</td>
<td>6h/24h</td>
</tr>
<tr>
<td>80/20</td>
<td>Quartz</td>
<td>3 parameters</td>
<td>6h/24h</td>
</tr>
<tr>
<td>100/0</td>
<td>Quartz</td>
<td>5 parameters</td>
<td>6h</td>
</tr>
<tr>
<td>1600/800</td>
<td>Quartz</td>
<td>7 parameters</td>
<td>6h/24h</td>
</tr>
<tr>
<td>TOTAL</td>
<td>1834/866</td>
<td></td>
<td>11,130h/20,784h</td>
</tr>
</tbody>
</table>
Results

$A_i = \text{equal} \quad \varphi_{ij} = 0$

![Graph showing results with parameters for different times and values of $A_i$, $k$, and $\varphi$.]
**Surrogate based Optimization**

\[ \varphi^P(\theta) = \varphi^P_0 [1 + A_1 \cos(k_1 \theta) + A_2 \cos(k_2 \theta + \Phi_{12}) + A_3 \cos(k_3 \theta + \Phi_{13})] \]

\[ \text{maximize } \text{Norm. Max. PVF Diff.} \]

\[ A_1, A_2, A_3, k_1, k_2, k_3, \Phi_{12}, \Phi_{13} \]

\[ \text{Subject to } \sqrt{A_1^2 + A_2^2 + A_3^2} = \text{constant} \]

- LF (low-fidelity) is a reduced grid of about 0.25HF (high-fidelity) cost
- HFS (high-fidelity surrogate) fits HF alone, MFS (multi-fidelity surrogate) fits both together

**Single Parameter Study (tri-modal pert.)**

- Tri-modal perturbation
- One variable, \( k_1 \)
- \( k_1 = 15, \ k_2 = 25, \ A_1 = A_2 = A_3 = 0.1633, \ \Phi_{12} = \Phi_{13} = 0 \) remain constant
- Surrogate model obtained using 14 LF points, 3 HF points and 3 validation points
- The model is able to predict the validation points quite well just with a few HF points
- General trend: NMPVFD increases as \( k \) increases
- \( k_1 = 5 \) shows a lower growth while \( k_1 = 10 \) a higher one. Important triadic interaction between modes!
3 Parameters Study (tri-modal pert.)

- Simplified problem: Three variables, \( k_1, k_2, \) and \( k_3 \)
- \( A_1=A_2=A_3=0.1, \Phi_{12}, \Phi_{13}=0 \) remain constant
- General trend: NMPVFD increases with the effective \( k \)
- Preliminary Bayesian MFS is used to fit the MFS
- Mirror points reduce the CV error a 77%
- Mirror points are points with the same output. i.e. In this case wave numbers \( (2,4,3) \) has mirror points \( (2,3,4) (3,2,4) (3,4,2) (4,2,3) (4,3,2) \)

<table>
<thead>
<tr>
<th>CV not including mirror points</th>
<th>CV error including mirror points</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.186147</td>
<td>0.042238</td>
</tr>
</tbody>
</table>

Conclusions and Future Work

- The future and main goal is **optimization** in 7 variables to find the initial disturbance producing **maximum departure from axisymmetry**
- We have obtained encouraging results using **multi-fidelity surrogate models** This will allow a reduced cost optimization
- We are finding interesting **interactions between the modes** imposed as a perturbation in the particle volume fraction at initial time
Thank you!

Do you have any questions?
Pairwise Interaction Extended Point-Particle (PIEP) Modeling

W. Chandler Moore

Outline

- Neighbors matter due to fluid-mediated Particle-Particle interactions
- Previously formulated PIEP model provides a rational approach to accounting for this fluid-mediated Particle-Particle interaction at low volume fractions
- The addition of a data-driven term greatly enhances the PIEP model at high volume fractions
**Euler-Lagrange Approach**

- Orders of magnitude faster than fully resolved simulations
  - EL & EE only viable approaches for practical problems
  - Point-particle models are needed for EL

**Fully Resolved Simulations**

- Immersed boundary method, Grid = $(490)^3$, $d / \Delta x = 60$

\[
\phi = 44\% , \text{Re} = 20 , \text{N} = 459
\]
Exact Location of Neighbors

View as seen by Incoming Flow

- Local volume fraction cannot explain the variations
- Upstream, downstream, lateral neighbors have different influence

Akiki, Balachandar, JCP, 307, 34-59 (2016)
Akiki, Jackson, Balachandar, Phys Rev Fluids, 1, 044202 (2016)

Current PIEP Model

\[
F_{PIEP} = F_{qs} + F_{un} + F_{am} + F_{vu} + F_L + F_c
\]

\[
T_{PIEP} = T_{qs} + T_{un} + T_c
\]

where, using perturbation maps resulting from direct numerical simulations (DNS),

\[
F_{PIEP} \equiv F_{PIEP}(Re, \phi, r_1, r_2, \ldots, r_N, v_1, v_2, \ldots, v_N)
\]

\[
T_{PIEP} \equiv T_{PIEP}(Re, \phi, r_1, r_2, \ldots, r_N, v_1, v_2, \ldots, v_N)
\]
Example: Quasi-Steady Force

\[ F_{qs,i} = F_{rot} \left( \phi_i, Re_i \right) + \left\{ 3\pi \propto d_i \left( \sum_{j=x}^{N} u_{j-x} \right)^{2} \right\} + 1+0.15(Re_i)^{0.687} \]

Data-Driven Volume Fraction Correction

\[ R^2 = 1 - \frac{\sum_{n=1}^{N_{p}} \left[ F_{DNS}(n) - F_{PIE}(n) \right]^2}{\sum_{n=1}^{N_{p}} \left[ F_{DNS}(n) - \bar{F}_{DNS} \right]^2} \]

<table>
<thead>
<tr>
<th>Case</th>
<th>Lateral Force R² (DNS vs PIEP)</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \phi = 0.1, Re = 40 )</td>
<td>0.68</td>
</tr>
<tr>
<td>( \phi = 0.2, Re = 16 )</td>
<td>0.34</td>
</tr>
<tr>
<td>( \phi = 0.45, Re = 21 )</td>
<td>0.09</td>
</tr>
</tbody>
</table>

\[ F_{PIE, Data-Driven} = F_{PIE} + F_{\phi,C} \]

\[ T_{PIE, Data-Driven} = T_{PIE} + T_{\phi,C} \]

where

\[ F_{\phi,C} \equiv F_{\phi,C}(Re, \phi, r_1, r_2, \ldots, r_N) \]

\[ T_{\phi,C} \equiv T_{\phi,C}(Re, \phi, r_1, r_2, \ldots, r_N) \]
Nonlinear Regression Modeling

For a given Re and volume fraction:

Response Variable (output)

\[ F_{\phi,c} \equiv \text{function of } \{ r_1, r_2, \ldots, r_N \} \]

Predictor Variables (inputs)

\[ T_{\phi,c} \equiv \text{function of } \{ r_1, r_2, \ldots, r_N \} \]

Function Form and Parameters

Let us define a set of scalar invariants \( I \) that fully define the configuration formed by the \( N \) neighbors.

- \( x \) – Separation \( : r_{x,1}, r_{x,2}, \ldots, r_{x,N} \)
- \( y \) – Separation \( : r_{y,1}, r_{y,2}, \ldots, r_{y,N} \)
- \( z \) – Separation \( : r_{z,1}, r_{z,2}, \ldots, r_{z,N} \)

Regression Modeling (drag)

For a given Re and volume fraction:

Cost (error) evaluation and gradient decent

Parameter array

\[ \beta \]

Predictor Variables (inputs)

\[ r_1, r_2, \ldots, r_N \]

Postulated Functional Form

\[ \Sigma \]

Response Variable (output)

\[ \beta_{\text{drag}}, \beta_{\text{drag}} \]

\[ \frac{\partial \beta_{\text{drag}}}{\partial \beta} \]
Postulating the Functional Form

\[ D(I) = \sum_{i=1}^{N} \left[ \sum_{m=1}^{L} \sum_{l=0}^{N} a_{l,m} \bar{i}(k_{j,l,m}, r_i) Y_{l,0}(\theta_i) + \sum_{m=1}^{L} \sum_{l=0}^{N} b_{l,m} r_i(k_{j,l,m}, r_i) Y_{l,0}(\theta_i) \right] H(r_i) \]

\[ L_I(I) = \sum_{i=1}^{N} \left[ \sum_{m=1}^{L} \sum_{l=0}^{N} a_{l,m} \bar{i}(k_{j,l,m}, r_i) Y_{l,0}(\theta_i) + \sum_{m=1}^{L} \sum_{l=0}^{N} b_{l,m} r_i(k_{j,l,m}, r_i) Y_{l,0}(\theta_i) \right] \bar{r}_i \cdot e_y H(r_i) \]

\[ T_s(I) = \sum_{i=1}^{N} \left[ \sum_{m=1}^{L} \sum_{l=0}^{N} a_{l,m} \bar{i}(k_{j,l,m}, r_i) Y_{l,0}(\theta_i) + \sum_{m=1}^{L} \sum_{l=0}^{N} b_{l,m} r_i(k_{j,l,m}, r_i) Y_{l,0}(\theta_i) \right] \bar{r}_i \cdot e_y H(r_i) \]

where

\[ Y_{l,0}(\theta) = \sqrt{2l + 1} \pi p(\cos \theta) \quad \frac{d\bar{i}(k_{j,l,m}, r_{min})}{dr} = 0 \quad \text{and} \quad \frac{d\bar{i}_i(k_{j,l,m}, r_{min})}{dr} = 0. \]

\[ a_{l,m} \text{ and } b_{l,m} \text{ make up the array of parameters, } \beta, \text{ to be determined by regression} \]

\[ N, M, \text{ and } L \text{ are the hyper parameters that must be optimized using a second data set} \]

Coefficient of Determination (R²) Results

<table>
<thead>
<tr>
<th>φ</th>
<th>Re</th>
<th>Drag</th>
<th>Lift</th>
<th>Torque</th>
<th>Data-Driven Drag</th>
<th>Data-Driven Lift</th>
<th>Data-Driven Torque</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>40</td>
<td>0.70</td>
<td>0.68</td>
<td>0.75</td>
<td>0.72</td>
<td>0.75</td>
<td>0.79</td>
</tr>
<tr>
<td>0.1</td>
<td>70</td>
<td>0.65</td>
<td>0.68</td>
<td>0.65</td>
<td>0.67</td>
<td>0.72</td>
<td>0.72</td>
</tr>
<tr>
<td>0.1</td>
<td>173</td>
<td>0.35</td>
<td>0.59</td>
<td>0.45</td>
<td>0.44</td>
<td>0.60</td>
<td>0.58</td>
</tr>
<tr>
<td>0.2</td>
<td>16</td>
<td>0.38</td>
<td>0.34</td>
<td>0.48</td>
<td>0.66</td>
<td>0.74</td>
<td>0.71</td>
</tr>
<tr>
<td>0.2</td>
<td>89</td>
<td>0.50</td>
<td>0.48</td>
<td>0.72</td>
<td>0.62</td>
<td>0.63</td>
<td>0.77</td>
</tr>
<tr>
<td>0.45</td>
<td>21</td>
<td>0.01</td>
<td>0.09</td>
<td>0.47</td>
<td>0.55</td>
<td>0.59</td>
<td>0.76</td>
</tr>
<tr>
<td>0.45</td>
<td>115</td>
<td>0.21</td>
<td>0.19</td>
<td>0.51</td>
<td>0.63</td>
<td>0.57</td>
<td>0.65</td>
</tr>
</tbody>
</table>

\[ R^2 = 1 - \frac{\sum_{n=1}^{N} \left( F_{DNS}(n) - F_{PIE}(n) \right)^2}{\sum_{n=1}^{N} \left( F_{DNS}(n) - \bar{F}_{DNS} \right)^2} \]
Drag Regression Results ($\phi = 0.45$ & $Re = 21$)

Previous PIEP Drag Force Map

Regression Drag Correction Map

Lift Regression Results ($\phi = 0.45$ & $Re = 21$)

Previous PIEP Lift Force

Regression Lift Correction Map
**Torque Regression Results ($\phi = 0.45$ & $Re = 21$)**

Previous PIEP Torque Force Map  
Regression Torque Correction Map

---

**Main Message**

- Neighboring particle locations matter
  - Local volume fractions is not sufficient for force modeling
- Fluid-mediated Particle-Particle interactions are strong
- Previously formulated PIEP model provides accurate forces/torque predictions at low volume fractions
- Implementing a data driven approach allows the PIEP model to predict forces/torques at high volume fractions
Thank you!
Questions?

Acknowledgment:
This material is based upon work supported in part by National Science Foundation Graduate Research Fellowship Program under Grant No. DGE-1315138 and in part by the the U.S. Department of Energy, National Nuclear Security Administration, Advanced Simulation and Computing Program, as a Cooperative Agreement under the Predictive Science Academic Alliance Program, under Contract No. DE-NA0002378.

Extra Slides:
Ideal Point Particle Model

If the model is EXACT

Recovers Fully-Resolved physics

Postulating the Functional Form

Substitute a general functions for neighbor specific functions \(F_x, F_y, \text{ and } T\).

\[
\mathbf{F}_{\text{VF}} = \sum_{i=1}^{N} \left( F_x (I_i) \mathbf{e}_x + F_y (I_i) \mathbf{e}_y + F_z (I_i) \mathbf{e}_z \right)
\]

\[
\mathbf{T}_{\text{VF}} = \sum_{i=1}^{N} \left( T_y (I_i) \mathbf{e}_x \times \mathbf{e}_y \right)
\]

Individual component functions:

\[
\mathbf{F}_{\text{VF}} = D(I) \mathbf{e}_x + L_y (I) \mathbf{e}_y + L_z (I) \mathbf{e}_z
\]

\[
\mathbf{T}_{\text{VF}} = T_x (I) \mathbf{e}_x + T_y (I) \mathbf{e}_y + T_z (I) \mathbf{e}_z
\]

Combine the above formulations:
Postulating the Functional Form

The function is only defined within radius of influence \((r_{\text{max}})\)

\[
H(r) = \begin{cases} 
0 & r < d \\
1 & d \leq r \leq r_{\text{max}} \\
0 & r > r_{\text{max}} 
\end{cases}
\]

where \(r_{\text{max}}\) can be found by root finding the following:

\[
\int_{V_d}^{V_{r_{\text{max}}}} g(r)\rho dV - N_{\text{train}} = 0
\]

Re & VF vs. Symmetry for Drag

<table>
<thead>
<tr>
<th>VF</th>
<th>Low Re</th>
<th>Medium Re</th>
<th>High Re</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.10</td>
<td>0.34</td>
<td>0.38</td>
<td>0.43</td>
</tr>
<tr>
<td>0.20</td>
<td>0.63</td>
<td>0.58</td>
<td>0.57</td>
</tr>
<tr>
<td>0.45</td>
<td>0.91</td>
<td>0.90</td>
<td>0.85</td>
</tr>
</tbody>
</table>

\[\frac{||M_{\text{symmetric}}|| - ||M_{\text{asymmetric}}||}{||M||}\]
### Re & VF vs. Symmetry for Lift

<table>
<thead>
<tr>
<th>VF</th>
<th>Low Re</th>
<th>Medium Re</th>
<th>High Re</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.10</td>
<td>0.10</td>
<td>0.11</td>
<td>0.24</td>
</tr>
<tr>
<td>0.20</td>
<td>-0.44</td>
<td>-0.11</td>
<td>-0.03</td>
</tr>
<tr>
<td>0.45</td>
<td>-0.91</td>
<td>-0.71</td>
<td>-0.56</td>
</tr>
</tbody>
</table>

**CCMT**

\[
\frac{||M_{symmetric}||}{||M_{asymmetric}||} / ||M||
\]

---

### Re & VF vs. Symmetry for Torque

<table>
<thead>
<tr>
<th>VF</th>
<th>Low Re</th>
<th>Medium Re</th>
<th>High Re</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.10</td>
<td>0.33</td>
<td>0.27</td>
<td>0.11</td>
</tr>
<tr>
<td>0.20</td>
<td>0.44</td>
<td>0.29</td>
<td>0.39*</td>
</tr>
<tr>
<td>0.45</td>
<td>0.68</td>
<td>0.63</td>
<td>0.34</td>
</tr>
</tbody>
</table>

**CCMT**

\[
\frac{||M_{symmetric}||}{||M_{asymmetric}||} / ||M||
\]

*Purely Statistical Model*
Microscale Shock and Contact Simulations

Brandon Osborne

Motivation

<table>
<thead>
<tr>
<th>PM-1: Blast Wave Location</th>
<th>PM-2: Particle Front Location</th>
<th>PM-3: Number of Instability Waves</th>
<th>PM-4: Amplitude of Instability Waves</th>
</tr>
</thead>
</table>

[Images and diagrams related to blast wave location, particle front location, and number/amplitude of instabilities.]
Simulation Overview

- Three sets simulations:
  - Close: Shock and contact reach first particle nearly simultaneously
  - Intermediate: Shock is 2 particle diameters downstream of first particle before the contact reaches its leading edge
  - Shock only: The simulation is absent a contact discontinuity

<table>
<thead>
<tr>
<th>ϕ</th>
<th>$M_{ct}$</th>
<th>$M_s$</th>
<th>$\rho_1$ (kg/m$^3$)</th>
<th>$\rho_2$ (kg/m$^3$)</th>
<th>$\rho_3$ (kg/m$^3$)</th>
<th>$\rho_2/\rho_1$</th>
<th>$\rho_3/\rho_1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>5-40%</td>
<td>0.31</td>
<td>1.22</td>
<td>1.20</td>
<td>1.65</td>
<td>2.15</td>
<td>1.37</td>
<td>1.78</td>
</tr>
<tr>
<td></td>
<td>0.90</td>
<td>1.90</td>
<td>1.20</td>
<td>3.03</td>
<td>8.20</td>
<td>2.52</td>
<td>6.81</td>
</tr>
<tr>
<td></td>
<td>1.26</td>
<td>2.69</td>
<td>1.20</td>
<td>4.27</td>
<td>26.35</td>
<td>3.54</td>
<td>21.87</td>
</tr>
</tbody>
</table>

Contours: $\phi = 10\%$, $M_{ct} = 0.31$, Close
Case Comparison: $\varphi = 10\%, M_{ct} = 0.31$

- Shock + Contact, Close
- Shock + Contact, Intermediate
- Shock Only

Contours: $\varphi = 10\%, M_{ct} = 0.90$, Close

- Density (kg/m$^3$): 2.06, 5.03, 8.05, 11.08, 14.10, 17.13
- Density Gradient Magnitude: 0.00x10$^{-16}$, 3.62x10$^{-16}$, 7.54x10$^{-16}$, 1.54x10$^{-15}$, 2.92x10$^{-15}$, 5.80x10$^{-15}$, 1.16x10$^{-14}$
- Pressure (Pa): 1.80x10$^{15}$, 3.22x10$^{15}$, 6.53x10$^{15}$, 7.04x10$^{15}$, 7.07x10$^{15}$, 1.11x10$^{16}$
- Mach Number: 0.86, 0.43, 0.28, 0.20, 0.15, 0.10
Case Comparison: $\varphi = 10\%, M_{ct} = 0.90$

Difference Plots: $\varphi = 10\%, M_{ct} = 0.90$
**Contours: \( \varphi = 10\%, M_{ct} = 1.26 \), Close**

- Density (kg/m³): 0.00, 20.14, 40.25, 60.43, 80.58, 100.72, 120.89
- Density Gradient Magnitude: 2.68x10⁻³, 2.35x10⁻³, 4.67x10⁻³, 1.00x10⁻³
- Pressure (Pa): 5.00x10⁻⁶, 2.52x10⁻⁵, 6.54x10⁻⁵, 6.99x10⁻⁵, 8.54x10⁻⁵
- Mach Number: 0.53, 0.64, 0.77, 0.91, 1.05, 1.24

---

**Case Comparison: \( \varphi = 10\%, M_{ct} = 1.26 \)**

- **Shock + Contact, Close**
- **Shock + Contact, Intermediate**
- **Shock Only**
Difference Plots: $\varphi = 10\%, \, M_{ct} = 1.26$

Shock + Contact, Close

Shock + Contact, Intermediate

$\varphi$ Comparison: $M_{ct} = 0.31$, Close

$\varphi = 5\%$

$\varphi = 10\%$

$\varphi = 20\%$

$\varphi = 40\%$
**Comparison:** $M_{ct} = 0.90$, Close

- $\phi = 5\%$
- $\phi = 10\%$
- $\phi = 20\%$
- $\phi = 40\%$

**Comparison:** $M_{ct} = 1.26$, Close

- $\phi = 5\%$
- $\phi = 10\%$
- $\phi = 20\%$
- $\phi = 40\%$
Summary

- Subsonic post-shock Mach number
  - Formation of a wake as the contact passes the particles that results in a “stretching” of the contact interface
  - Force oscillations created by reflected waves are dampened by the contact
  - Oscillations of greater magnitude as volume fraction increases and a decrease in peak force for subsequent particles for higher volume fractions

- Near sonic post-shock Mach number
  - Rapid mixing of the contact interface
  - Formation of compression waves traveling upstream and shocks extending from particle surface to wall after sonic flow is reached
  - Increasing force on the first particle and decreasing force on subsequent particles as volume fraction increases

- Supersonic post-shock Mach number
  - Rapid mixing of the contact interface
  - Formation of bow shocks and shocklets
  - High particle forces and the formation of a compression wave traveling upstream for $\varphi = 40\%$

Future Work

- Shock and contact interaction with an FCC array of particles
  - Characterize the effects a shock and contact have on force history and flow field compare to current 1D array data

- Shock and contact interaction with a random arrangement of particles
  - Apply techniques and models developed from 1D and FCC arrays to a random arrangement of particles

- PIEP model application to shock-contact-particle interaction
Thank you!  
Do you have any questions?
BE Simulations of CMT-nek

Sai Chenna (BE)

Outline: BE-simulation of CMT-nek

- Particle-solver
- Modelling approach
- Particle-distribution tool
- DSE of CMT-nek using BE
CMT-nek DSE: Motivation & Approach

- **Motivation**
  - CMT-nek has huge design space
  - Various architectural options exist for Exascale systems

- **Goal**
  - Use BE methods & tools to perform DSE
  - **Approach**
    - Perform BE Simulations of CMT-nek (baseline)
    - Identify optimization candidates (i.e. most expensive subroutines)
    - Create and validate models of algorithms for these subroutines
    - Use BE to predict performance

---

CMT-nek: Particle Solver subroutine

- **Particle solver** – *expensive kernel in CMT-nek*
  - Calculates the particle properties at each time-step
  - **Assumptions**
    - No particle to particle interaction
    - No two-way coupling
  - **Parameters**
    - $N$ – element size
    - $nelt$ – elements per processor
    - $\alpha$ – particles/gridpoint
    - $N_p$ – # particles = $\alpha N^3 nelt$

- **Workflow**
  - **Update particle location**
  - **Move particles**
  - **Interpolate fluid properties**
  - **Calculate fluid forces**
  - **Update particle position and velocity**
Modelling Approach

- **Symbolic Regression**
  - Tool to generate multi-parameter performance models for computation kernels
  - No prior knowledge of kernel required
  - Captures machine-specific performance behaviours

- **Trace-driven simulation**
  - Building a tool which determines particle distribution from a trace file based on user-specified parameters
  - Determines # particles residing and moving in each processor at given time-step
  - Improves accuracy of BE-simulation

Modelling Approach: Trace-driven simulation

- **Particle solver kernel**
  - Workload per processor depends on # particles
  - # particles/processor is dynamic
    - varies among processors – depends on total # of processors
    - varies at each timestep – based on fluid forces on particles
  - Need a trace to perform simulations

![Particle-distribution across 4k ranks](image-url)
Modelling Approach: Trace-driven simulation

- Particle-distribution tool
  - Input: trace data containing particle location at each time-step
  - Output
    - # particles residing in each rank @ every timestep
    - # particles moving across each rank @ every timestep
  - Particle movement doesn’t depend on # processors
    - Single trace for a given problem size is sufficient to predict particle movement for any # of processors

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
</tr>
<tr>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
</tr>
</tbody>
</table>

Particle-distribution tool: Results

Element size – 11, # of elements ~ 262144, # particles - 512000

![Graphs showing particle distribution and execution time across different ranks.](image)
Particle solver: Strong Scaling

- Element size – 11, # of elements – 262144, # particles – 512000

- No speedup obtained by increasing # of cores from 4k to 8k
- Increase in processor-count results in
  - Huge load imbalance
  - Poor resource utilization
- Need for better mapping algorithm – load-balancing algorithm
- Trace-driven simulation provides optimal configuration by identifying
  - Computation cost
  - Communication cost
  - Resource utilization

Validation of Particle Solver Model

- CMT-nek particle solver (baseline version) - Validation results on Vulcan

- Observed an average error of 4% in simulations
Particle Solver: Algorithmic options

- Evaluate performance of various algorithmic options available for particle-solver kernel

- **Time Integration** - solve differential equations to calculate particle properties
  - Current: Runge-Kutta 3 (rk3) – 3 stage time integration
  - Alternate: Backward Differentiation Formula (bdf) – single stage time integration

- **Particle Interpolation** - interpolates fluid properties acting on particles
  - Current: Barycentric Interpolation
  - Alternate: reduced Barycentric Interpolation, Tri-linear Interpolation

---

Simulator Predictions: Time Integration

Simulation predictions on Vulcan

- Both algorithms vary
  - **non-linearly** w.r.t element-size(N) and elements-per-processor(nelt)
  - **linearly** w.r.t particles/gridpoint(α)

- bdf time integration provides 3x speedup over rk3
Simulator Predictions: Interpolation

Simulation predictions on Vulcan

- Algorithms vary
  - non-linearly w.r.t \( \text{element-size}(N) \) and \( \text{elements-per-processor}(nelt) \)
  - linearly w.r.t \( \text{particles/gridpoint}(\alpha) \)
- Reduced barycentric interpolation provides 5x speedup
- Tri-linear interpolation provides 164x speedup

Conclusions & Future work

- Conclusions
  - BE Methods
    - Symbolic Regression
      - accurate, doesn’t require prior knowledge of kernel
    - Trace-driven simulation
      - improves BE-simulation accuracy
      - provides optimal configuration for a given problem
    - BE Simulations of particle-solver:
      - Validation – average error of 4%
    - DSE
      - bdf time-integration algorithm provides 3x speedup
      - Reduced barycentric interpolation provides 5x speedup
      - Tri-linear provides 164x speedup
  - Future work
    - BE Simulations of CMT-nek
      - modelling Gas-solver kernel
    - Adding two-way coupling into particle-solver AppBEO
    - CMT-nek DSE
      - BE Simulations of CMT-nek on Notional Architectures
Do you have any questions?