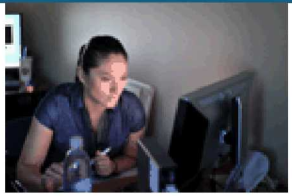
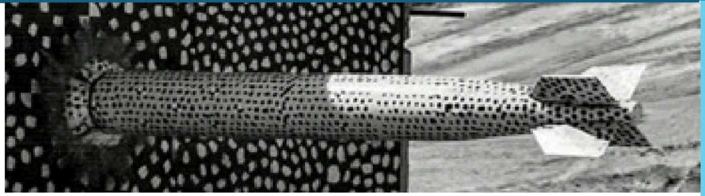
SAND2019-1957C

Towards Scalable Scientific Machine Learning: Motivation and an Approach





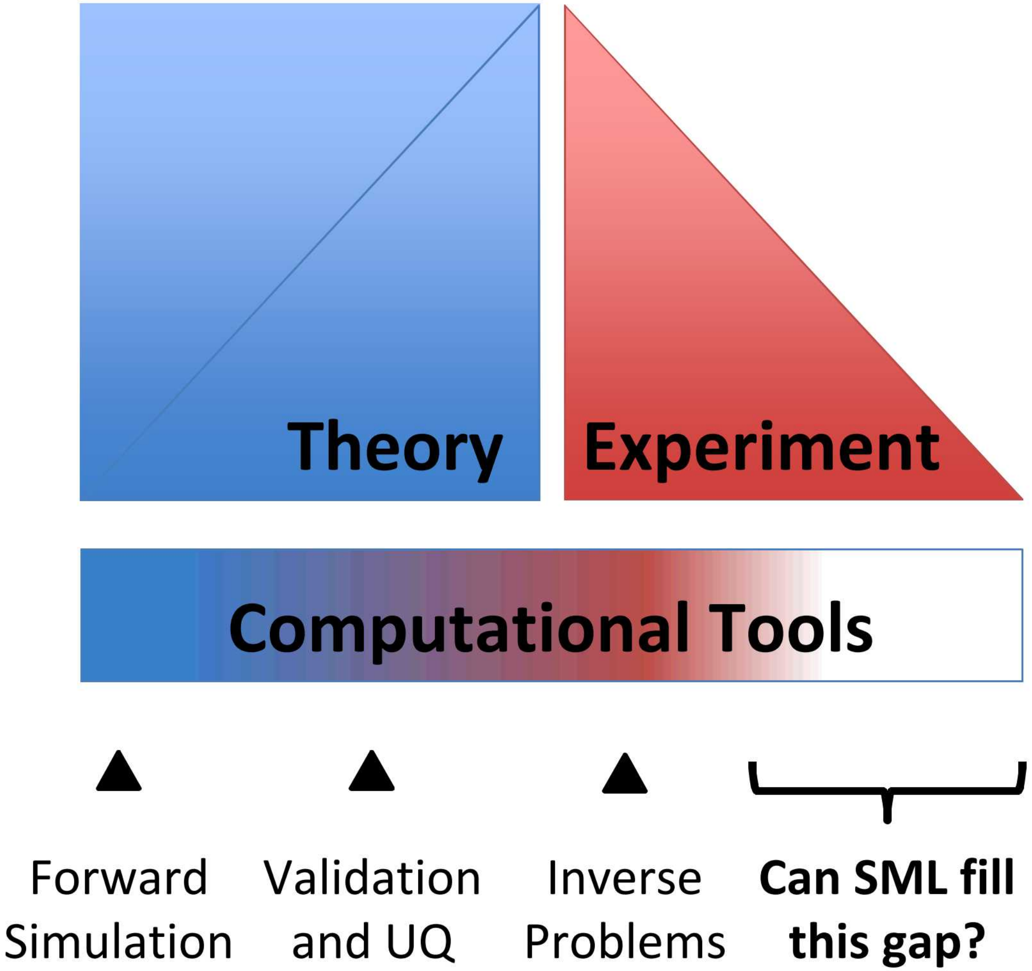
PRESENTED BY

Eric C. Cyr

K. Beckwith, C. Siefert, Pat Knapp (SNL),
P. Sentz, L. Olson (UIUC),
S. Guenther, N. R. Gauger (TU Kaiserslautern),
L. Ruthotto (Emory),
J. B. Schroder (UNM)

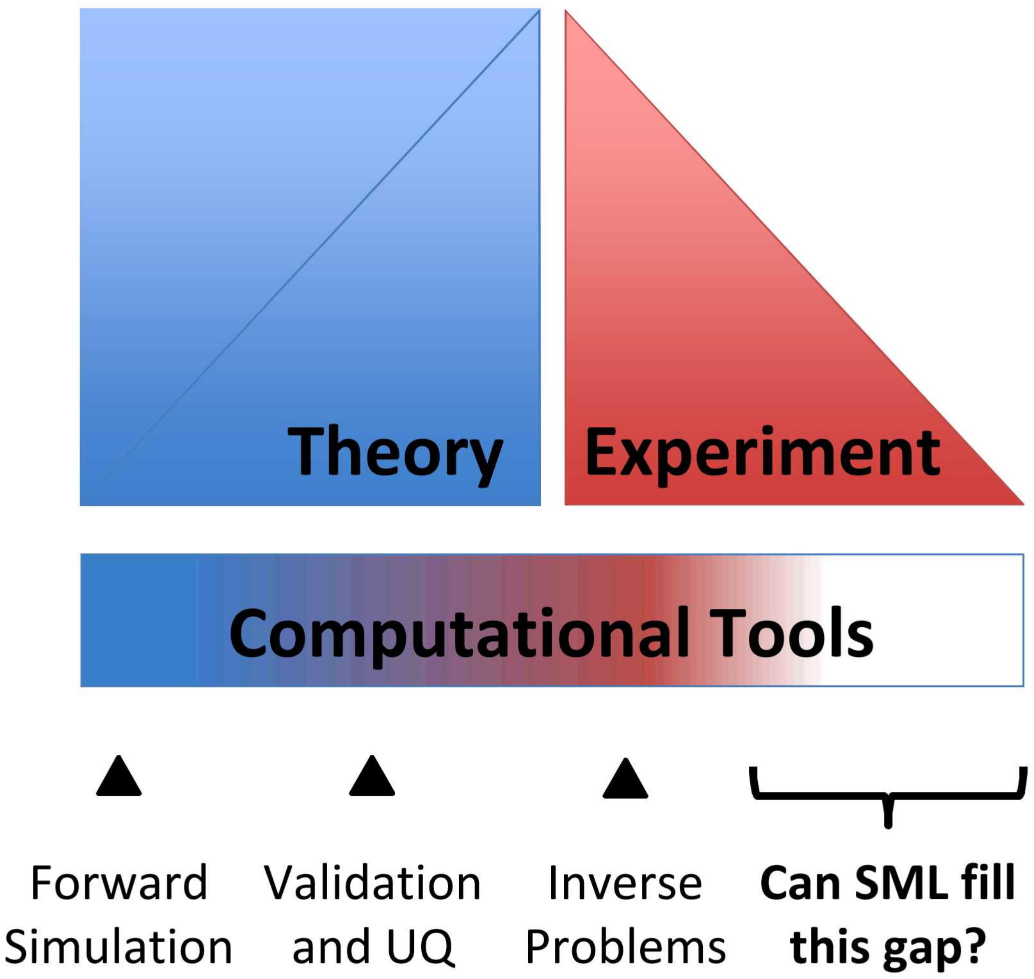
Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

Why Scientific Machine Learning (SML)?



- Large amounts of data from experiment and simulation
- Difficult to use traditional human centered analysis techniques
- Despite success of comp. science, tool gap remains for analyzing data
- Can Scientific Machine Learning (SML) fill the gap?

What do I mean by scalability (in general)?



1. Data set size
   - Peta/Tera byte sized data sets
   - Few independent samples
2. Complexity of DNNs
   - Architectures are complex, not broadly applicable
3. Scalability of training
   - Use DOE computing platforms for training and machine learning
4. Mixed types of data
   - Images, frequency, rate calculations, etc…
5. How do you use DNNs and ML for science apps
   - Response surface construction, inversion

Outline: Two distinct pieces

1) Scientific Machine Learning motivated by Sandia's Z-machine

2) Layer-parallel training of neural networks

Motivating Engineered System: Sandia's Z Machine

"Z compresses energy in time and space to achieve extreme powers and intensities"[1]

- Used to explore fusion concepts and as an x-ray source
- Uses currents of around 26 million amps
- Peak x-ray output of 350 terawatts
- This makes pretty pictures (see right)

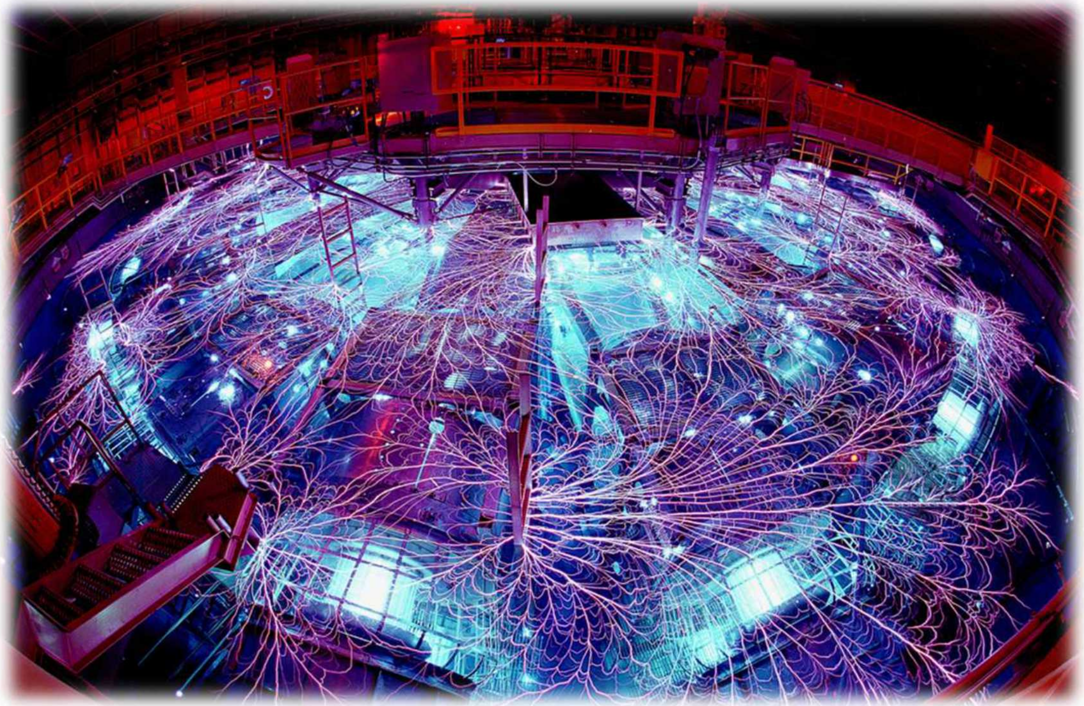
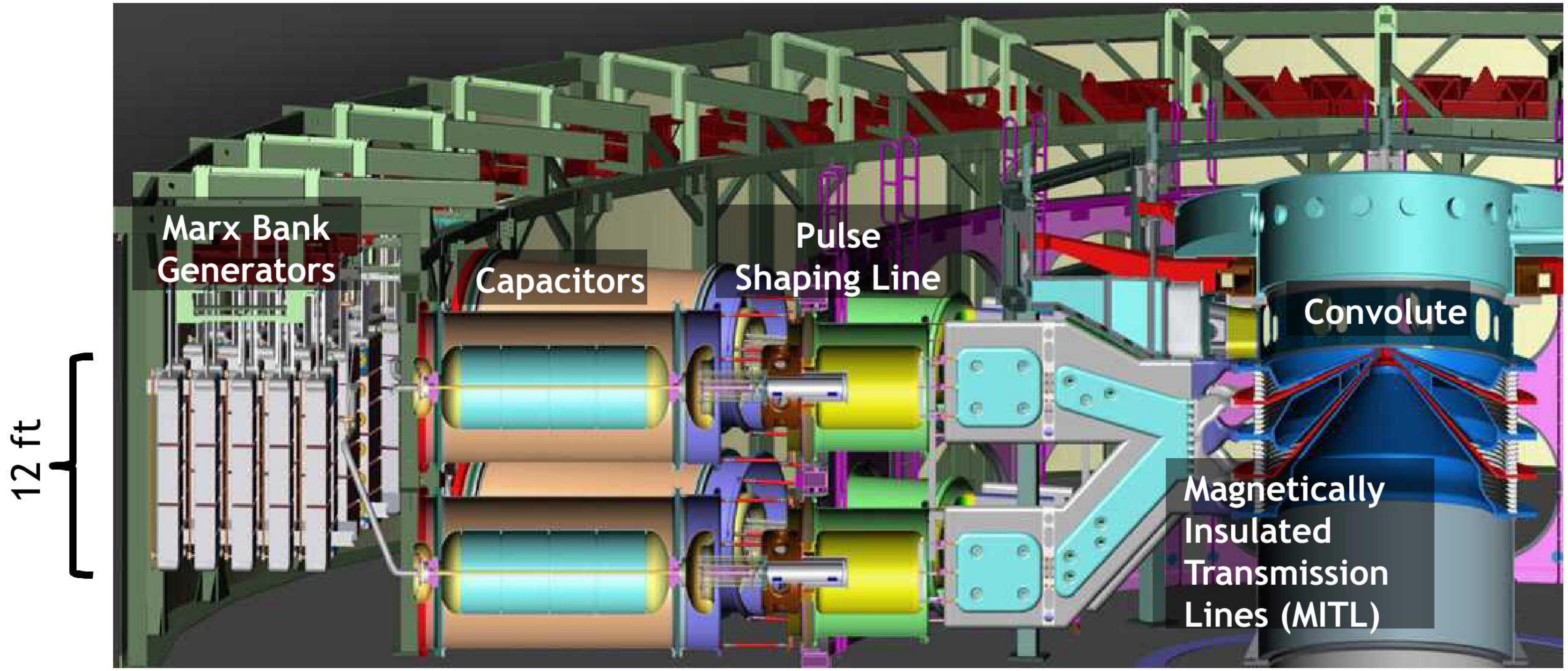
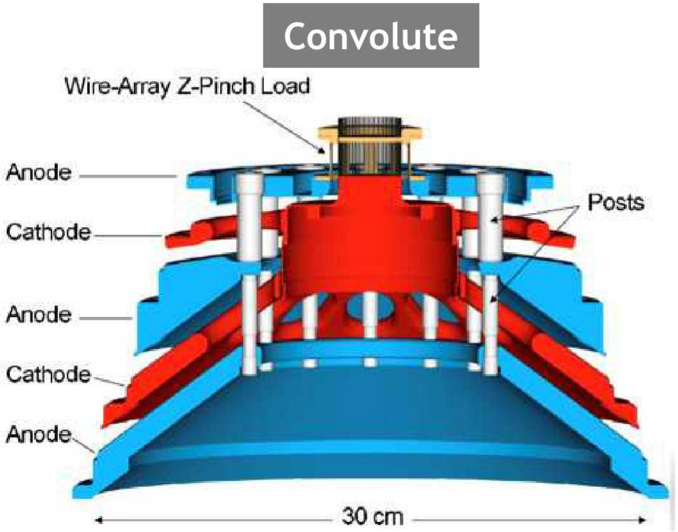


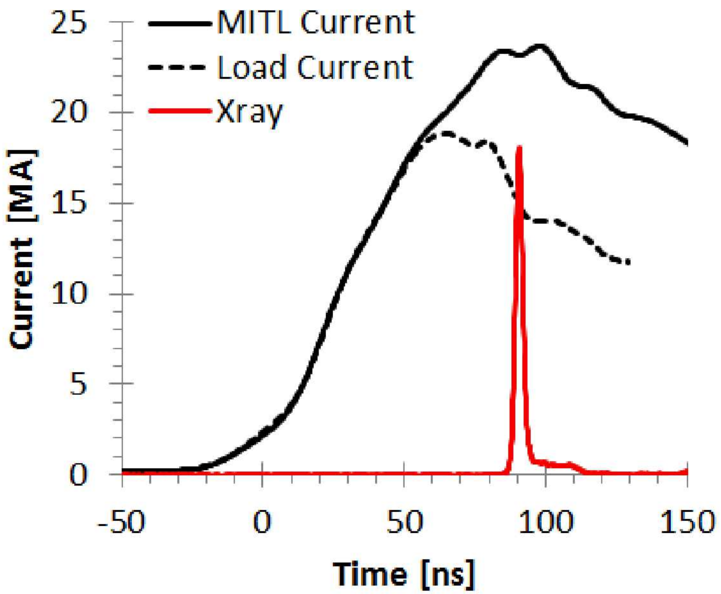
Photo: Randy Montoya

[1]For more about "Z" see https://www.sandia.gov/z-machine/

Motivating Engineered System: How does Z work? (As explained by a computer scientist)







This machine is complex, SML models maybe valuable:

- Simulation is incomplete, inaccurate or expensive
- Interactions can challenge physical intuition and conventional simulation
- “Simple” model used to guide experimental or machine design
- **Ultimately another capability for an engineer or scientist**

Motivating Engineered System: Three potential SML activities for Z

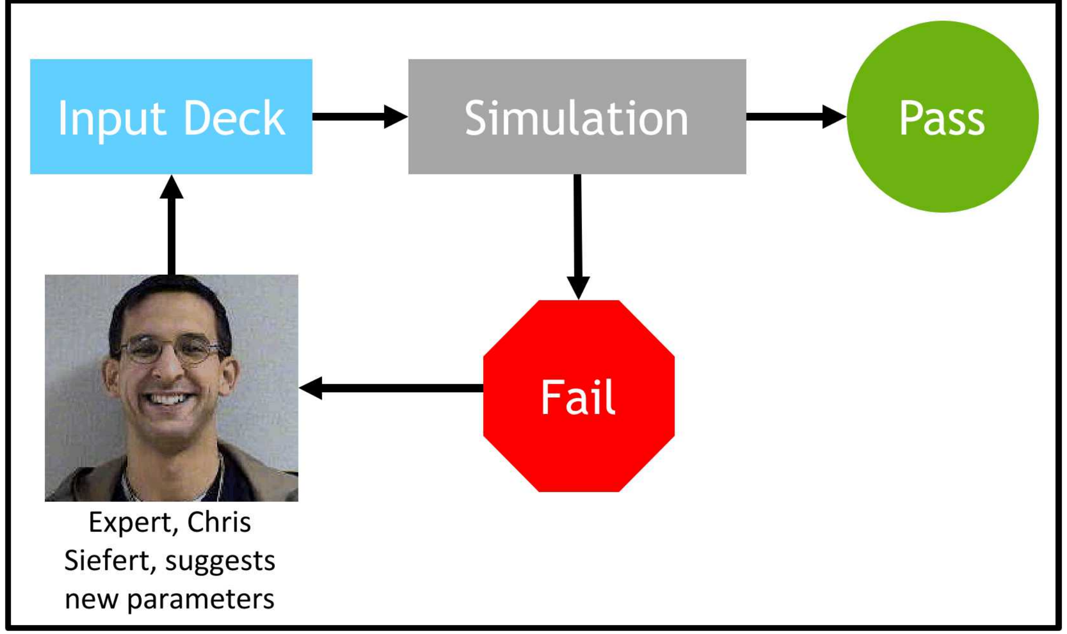
1) Tuning of simulation parameters
2) Learning physical models
3) Guidance in experimental design

Motivating Engineered System: 1) Tuning of simulation parameters

Despite our best efforts, simulation codes and algorithms have many parameters

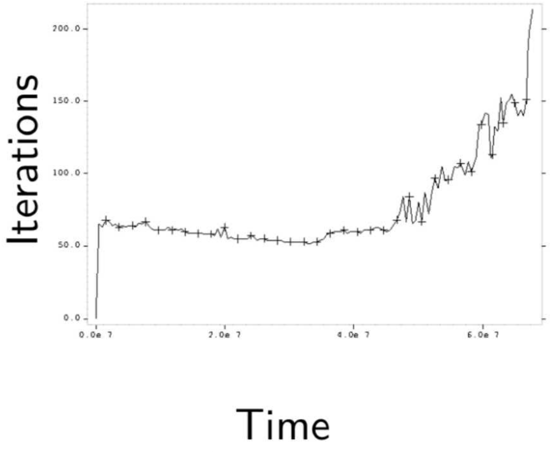
- Multigrid methods and linear solvers are particularly challenging for analysts
- Can we use machine learning to make writing an input deck easier?

Current algorithm for determining best solver parameters

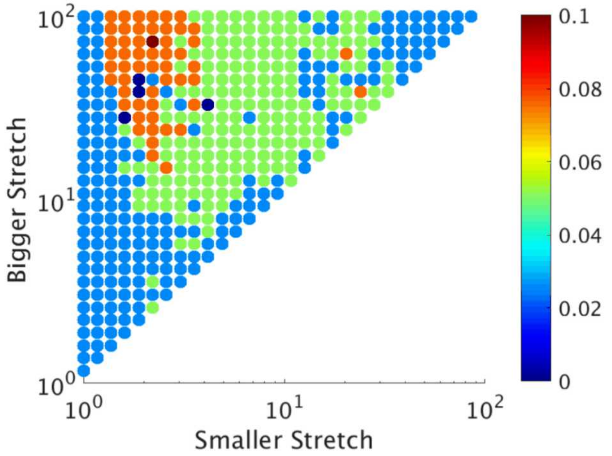


Approach

- Use ML to tune linear solver parameters
- Use “ensemble” training techniques



Good parameters reduce run time

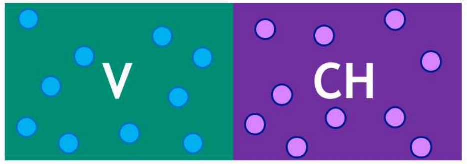


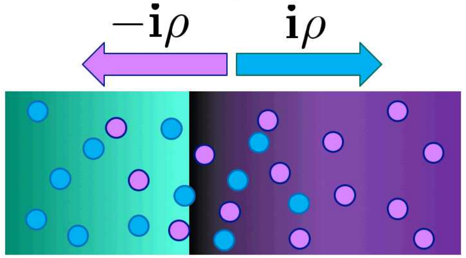
Outcome of good and bad choices of parameters

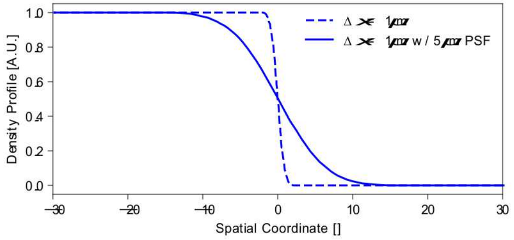
Credit to: Chris Siefert, Mark Hoemmen, John Kaushagen, Ali Pinar, Matthew Peterson, Ron Oldfield, Connor Smith

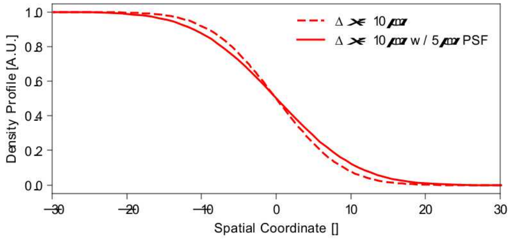
Motivating Engineered System: 2) Tuning of simulation parameters

Measure the blurring of the material boundary:



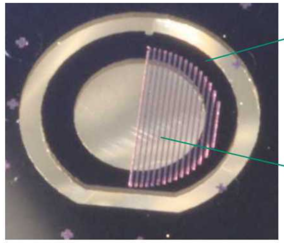


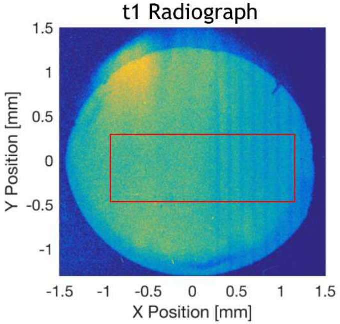


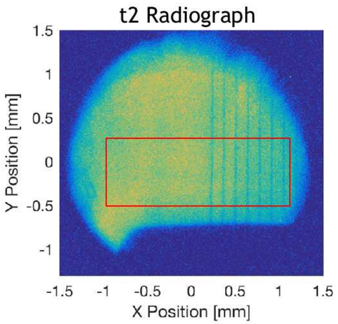


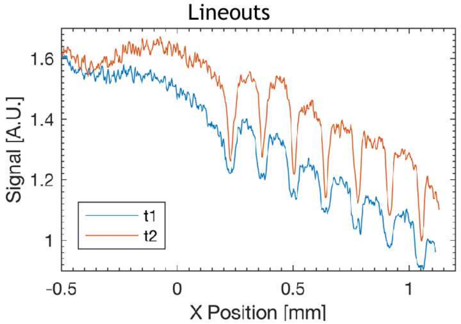
Experimental setup:

- Silicon wafer with "stripes" of material
- Exposed to X-ray's from Z









What do the scientists hope to learn from this data?

- We would like to understand the blurring process
- Kinetic models to develop a moment based macroscopic model model has not been predictive
- Can we determine a macroscopic model?

This leads to math and modeling questions?

- Do we have enough data?
- What more data do we need?
- What critical physical processes do we need to included (e.g. mass conservation)? Does this make the data requirements tractable?
- What machine learning models shall be considered and can be used for useful interpretation?

Credit to: Kris Beckwith and Pat Knapp

Motivating Engineered System: 3) Guidance in experimental design

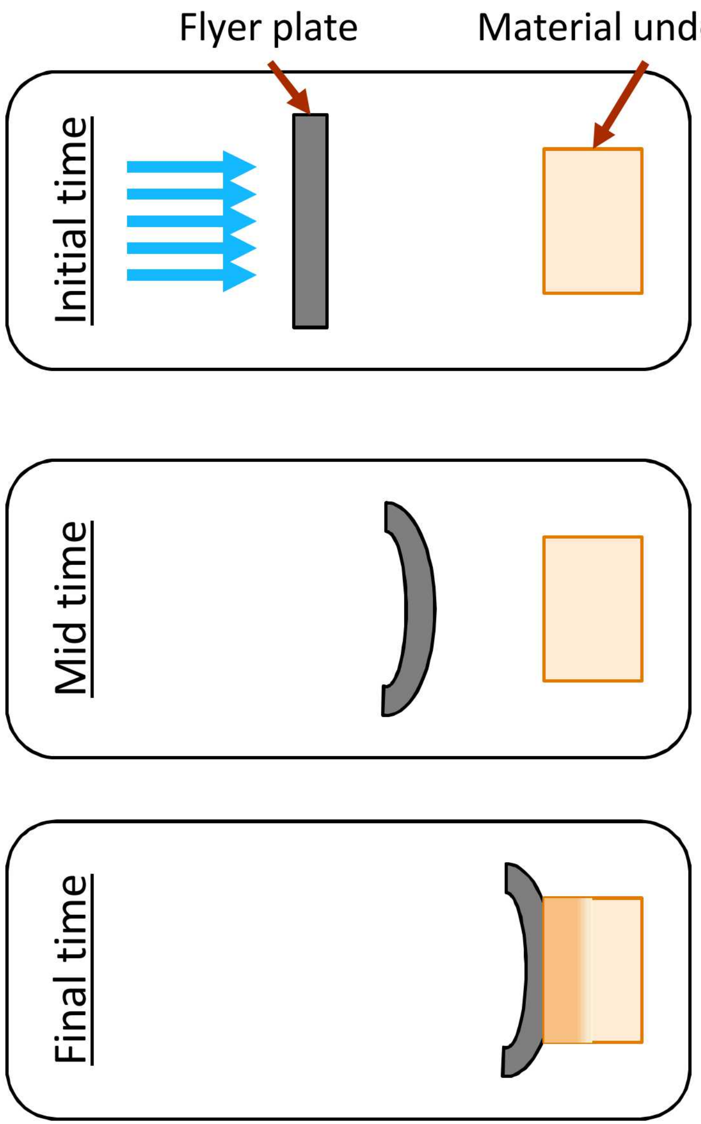
Z is used to determine equations of state (EOS) for materials at extreme pressures

- Flyer plates are launched into material
- VISAR data measures shock, determines EOS (think Riemann problem)

Problem: flyer must be accelerated but must not shock!

- This is controlled by ”pulse shaping”
- Force driving flyer is a function of pulse shape including current losses and impedance interactions

Machine learning question: Using modeling and data can we invert for pulse shape given a target pressure/temperature?



Magnetic field from Z pushes flyer (very fast!)

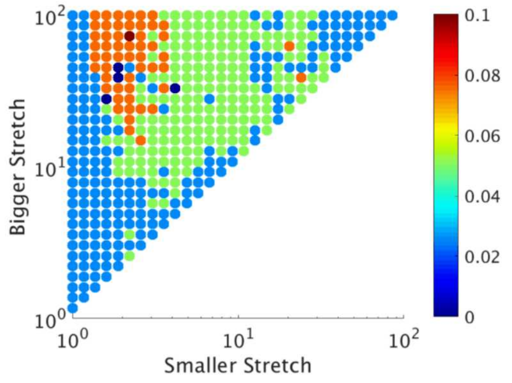
Flyer can deform, but shocks **must not** form

Flyer impacts material, driving shock at specified pressures

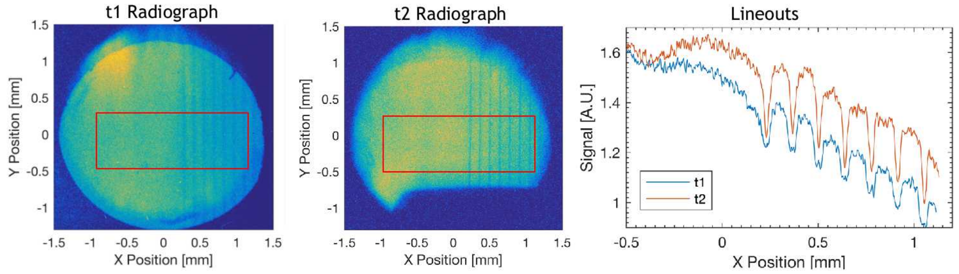
Flyer plate experiment: time sequence

Machine Learning for Z: Rewind

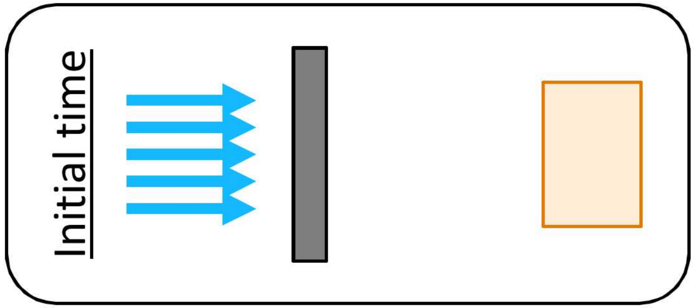
1) Assisting analysts using learned linear solver parameters



2) Learning new models for interface blurring



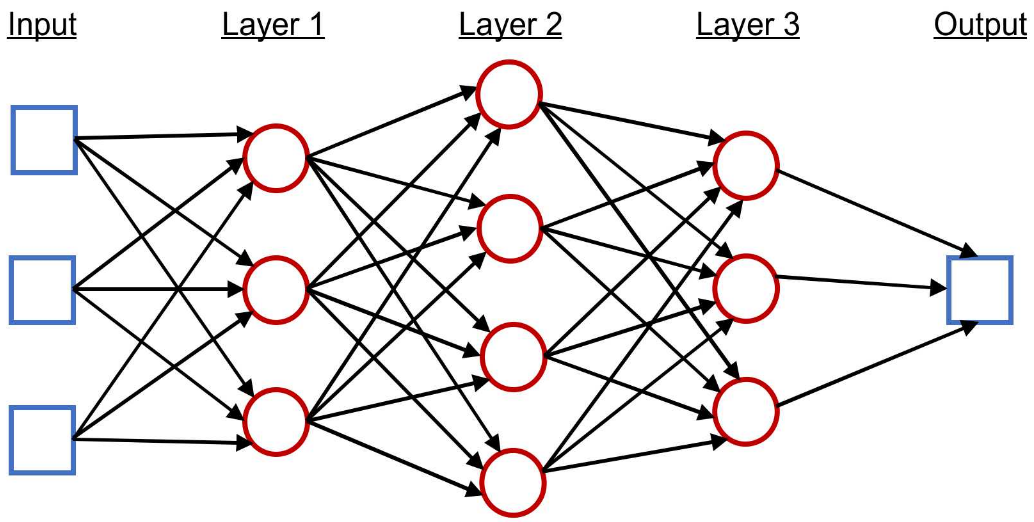
3) Designing pulse profiles for flyer plates



Layer-Parallel Training of Deep Neural Networks

Neural networks are the “hot” thing in machine learning

- Important to realize machine learning is more than neural networks
- Deep neural networks have had an amazing impact on image recognition and other commercial learning applications
- They are often depicted like this:



The data (e.g. image) is contained and ‘x_l’, ‘g’ is a nonlinear activation function, the weight matrix ‘W_l’ and vector ‘bl’ must be “learned” through training

$$x_{l+1} = g(W_l x_l + b_l) \text{ for } l = 1 \ldots L - 1$$

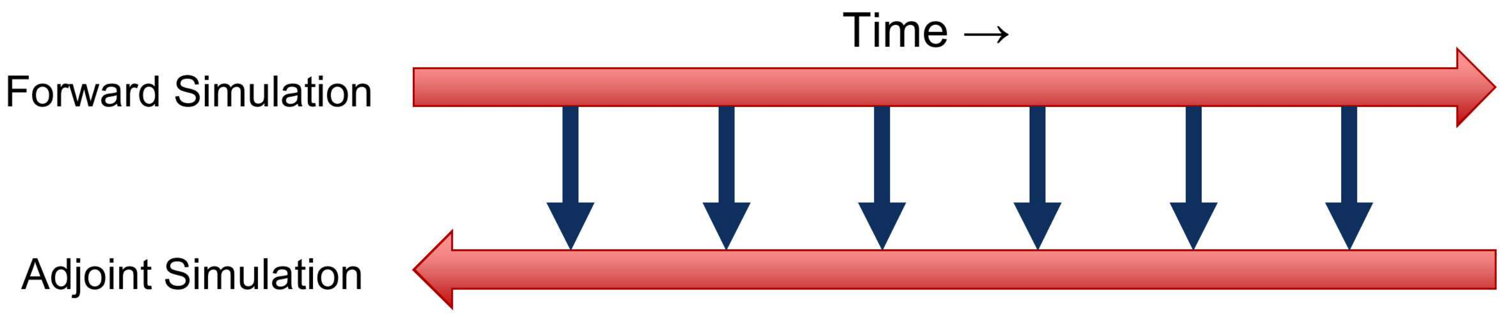
Layer-Parallel DNNs: An Evolutionary Viewpoint

The forward evolution from layer-to-layer looks like time evolution. This motivates the ODE form[1]:

$$x' = g(W(t)x(t) + b(t)) \text{ for } t \in [0, T], x(0) = x_0$$

Training algorithm requires computation of the gradient (gross approx. here):

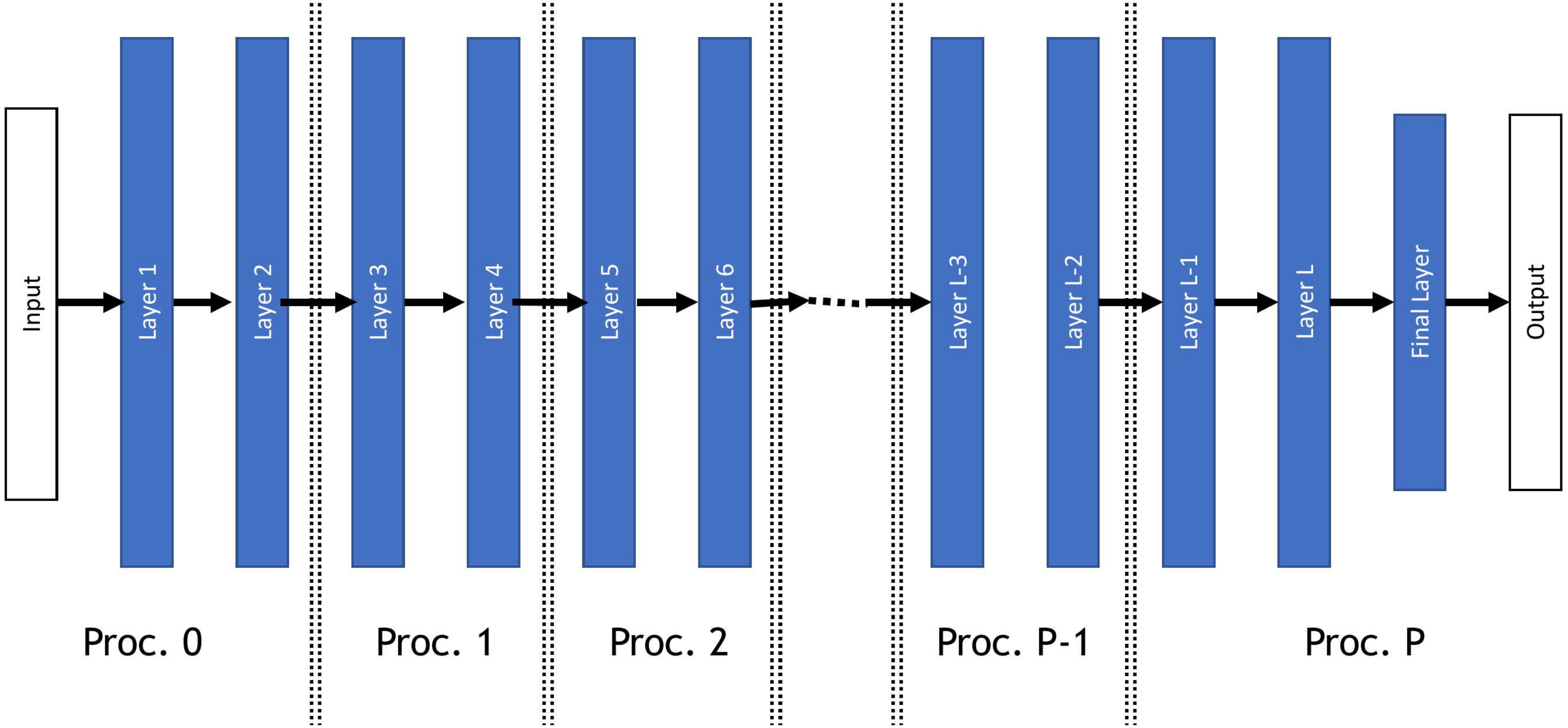
1) Compute forward solution
2) Compute adjoint solution



Gradient computation requires two time evolutions in serial. This limits scalability.

[1] E. Haber and L. Ruthotto. *Stable architectures for deep neural networks*. Inverse Probl., 34:014004, 2017.

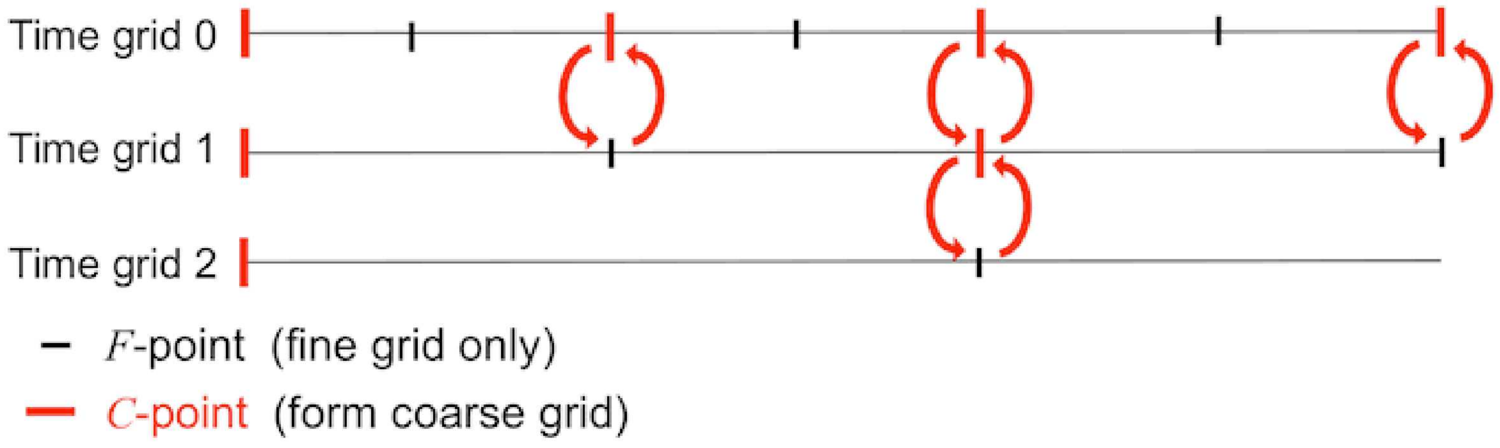
Layer-Parallel Training: What would be cool!



- Many parallelization strategies for DNNs subdivide the layers
- Pursue a complementary direction and also introduce layer parallelism
- Based on ideas in parallel-in-time methods
- How do we do this?

Layer-Parallel Training: Multigrid Reduction in Time (MGRIT)[1,2]

To accelerate the forward and backward solve, we will apply MGRIT[1]



Xbraid library modified by Stefanie Guenther (speaking "parallel-in-time!") to solve the adjoint problem.

1: Perform m_1 state updates:
  for $m = 1, \dots, m_1$: $\mathbf{U}_m \leftarrow \mathrm{MGRIT}(A, \mathbf{U}_{m-1}, \boldsymbol{\theta}, \mathbf{G})$
2: Perform m_2 adjoint updates:
  for $m = 1, \dots, m_2$: $\bar{\mathbf{U}}_m \leftarrow \mathrm{MGRIT}(A_{\mathbf{U}_{m_1}}, \bar{\mathbf{U}}_{m-1}, \boldsymbol{\theta}, \mathbf{G}_{\mathbf{U}_{m_1}})$
3: Assemble reduced gradient $\nabla_{\boldsymbol{\theta}} J, \nabla_{\mathbf{W}} J, \nabla_{\boldsymbol{\mu}} J$
4: Approximate Hessians $\mathbf{B}_{\boldsymbol{\theta}}, \mathbf{B}_{\mathbf{W}}, \mathbf{B}_{\boldsymbol{\mu}}$ and select a stepsize $\alpha > 0$
5: Network control parameter update:
  $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \alpha \mathbf{B}_{\boldsymbol{\theta}}^{-1} \nabla_{\boldsymbol{\theta}} J$
  $\mathbf{W} \leftarrow \mathbf{W} - \alpha \mathbf{B}_{\mathbf{W}}^{-1} \nabla_{\mathbf{W}} J$
  $\boldsymbol{\mu} \leftarrow \boldsymbol{\mu} - \alpha \mathbf{B}_{\boldsymbol{\mu}}^{-1} \nabla_{\boldsymbol{\mu}} J$
6: **If** converged: halt
  Else: go to step 1.

[1] R. D. Falgout, S. Friedhoff, T. V. Kolev, S. P. MacLachlan, and J. B. Schroder. *Parallel time integration with multigrid*. SIAM J. on Sci. Comp., 36, 6, 2014.

[2] XBraid: Parallel multigrid in time. software available at https://github.com/XBraid/xbraid

Layer-Parallel Training: Scalability and Solvers

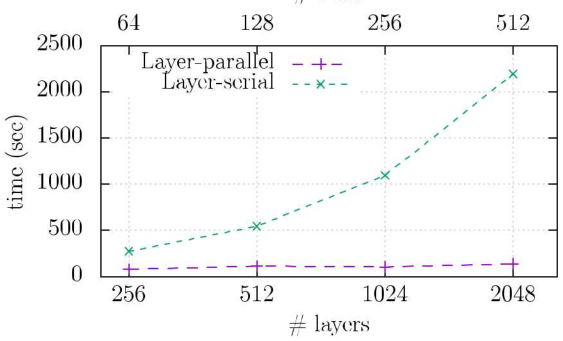
Using "one-shot" optimization to train an ODE Neural Network

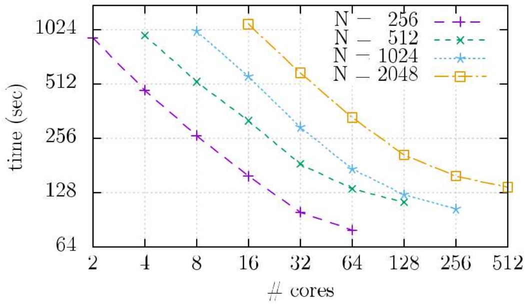
- Hand written NN operators (including convolutional)
- MNIST (hand written digits) and Indian pines (hyperspectral image segmentation) data sets
- Demonstrates good weak/strong scaling

Weak Scaling

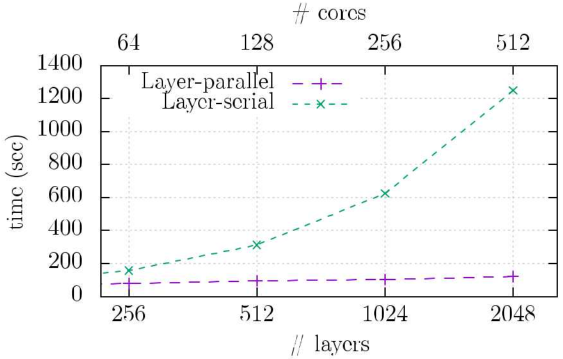
Strong Scaling

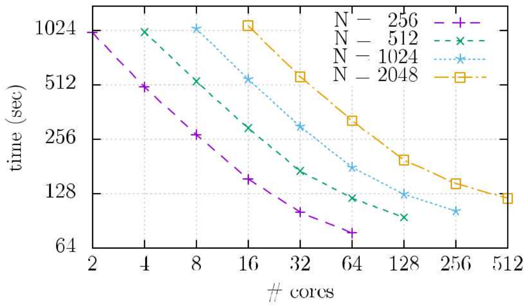
MNIST





Indian Pines





For details, see: S. Guenther, L. Ruthotto, J. B. Schroder, E. C. Cyr, N. R. Gauger, Layer-Parallel Training of Deep Residual Neural Networks, arXiv preprint arXiv:1812.04352, 2018.

Summary

- Motivated scientific machine learning ideas as filling an existing gap in the computational toolset
- Talked through three different types of potential applications associated with Sandia's Z-machine
  1) Determination of solver parameters
  2) Development of a macroscopic model of interface diffusion
  3) Pulse shape design for flying plate experiments
- Changing gears, presented new results demonstrating scalable training of deep neural networks

Thanks to the DOE Office of Science ASCR Early Career Research Program for supporting this work.