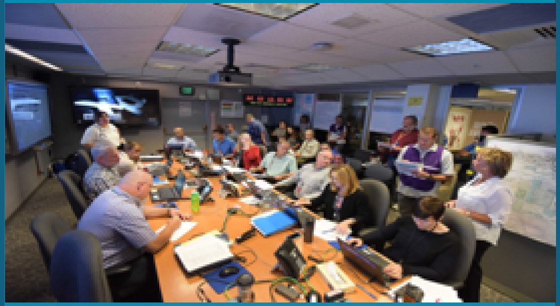
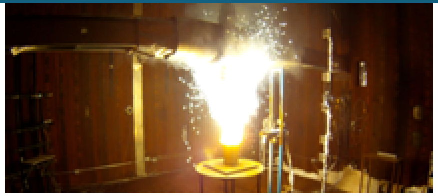
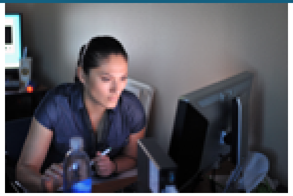
SAND2021-11468C

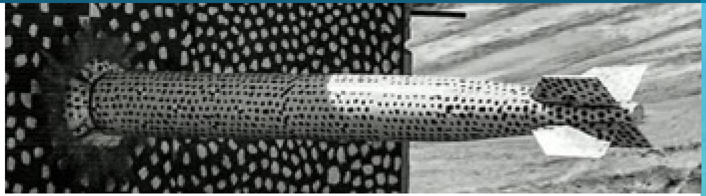
Sandia National Laboratories

A Critique of Optimization Modeling Environments for Complex Engineered Systems









Michael Bynum, **William Hart**, Jordan Jalving, Carl Laird, Bethany Nicholson, John Siirola

Sandia National Laboratories

wehart@sandia.gov

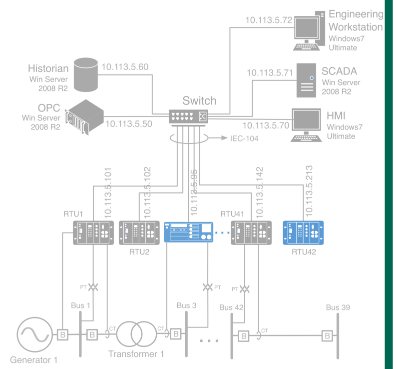
U.S. DEPARTMENT OF ENERGY NNSA

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and 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.

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.

Elements of the solution of complex optimization applications

Application Formulation	Optimization Equations	Modeling Environment	Optimization Solver
Specification of application goals and decision options	**Mathematical representation of application goals**	**General-Purpose tools for expressing models**	**Optimization solvers identify best solutions**



IEEE 42 Bus Power System

Attacker: Maximize Damage/Disruption

Defender: Minimize Losses

$$
\begin{array}{lll}
\max_{\boldsymbol{x}\in\{0,1\}} & F(\boldsymbol{x},\boldsymbol{y}) & \\
\text{s.t.} & G(\boldsymbol{x},\boldsymbol{y}) \le 0 & \\
 & \min_{\boldsymbol{y}\ge 0} & F(\boldsymbol{x},\boldsymbol{y}) = f(\boldsymbol{x},\boldsymbol{y}) \\
 & & g(\boldsymbol{x},\boldsymbol{y}) \ge 0
\end{array}
$$

PYOMO, GAMS, AMPL

COIN|OR, KNITRO, GUROBI OPTIMIZATION

$$\underbrace{\hspace{6cm}}$$

Our focus today

Examples of optimization modeling environments

AMPL

GAMS

julia

JUMP

MATLAB®

PYOMO

GUROBI OPTIMIZATION

PyTorch

TensorFlow

PYOMO DAE

IBM CPLEX

EGRET

Prescient

GRAVITY

POEK
COEK

IDAES
Institute for the Design of Advanced Energy Systems

ExaGO

Impact of optimization modeling environments

1. **Simplify expression of complex applications**
   - Intuitive algebraic expressions
   - Compact mathematical notation
   - Domain-specific problem representations
2. **Automate grungy parts of the computational workflow**
   - Automatic differentiation
   - Application of model transformations
3. **Facilitate transformations between problem representations**
   - Transformations to simplify the problem formulation
   - Transformations to tailor problem to solver requirements

How do we effectively use
general-purpose modeling environments
on emerging computational platforms?

Some perspectives on future modeling environments

1. Performance optimization

2. Application-centric vs Solver-centric

3. SME- vs Data-driven models

Performance Optimization

Can we use the modeling environment to tailor calculations in the optimization workflow?

Idea 1: Use code generation or automatic differentiation to tailor sparse, unstructured derivative calculations to target hardware

Explicit Code

Derivatives are coded by hand.

Pro:
- Can directly tailor calculations to HW

Con:
- Hand coding can be error-prone

Code Generation

Code is generated automatically for derivatives.

Pro:
- Can tailor code generation to HW

Con:
- Complex SW code management

Automatic Differentiation

Derivatives computed numerically with AD

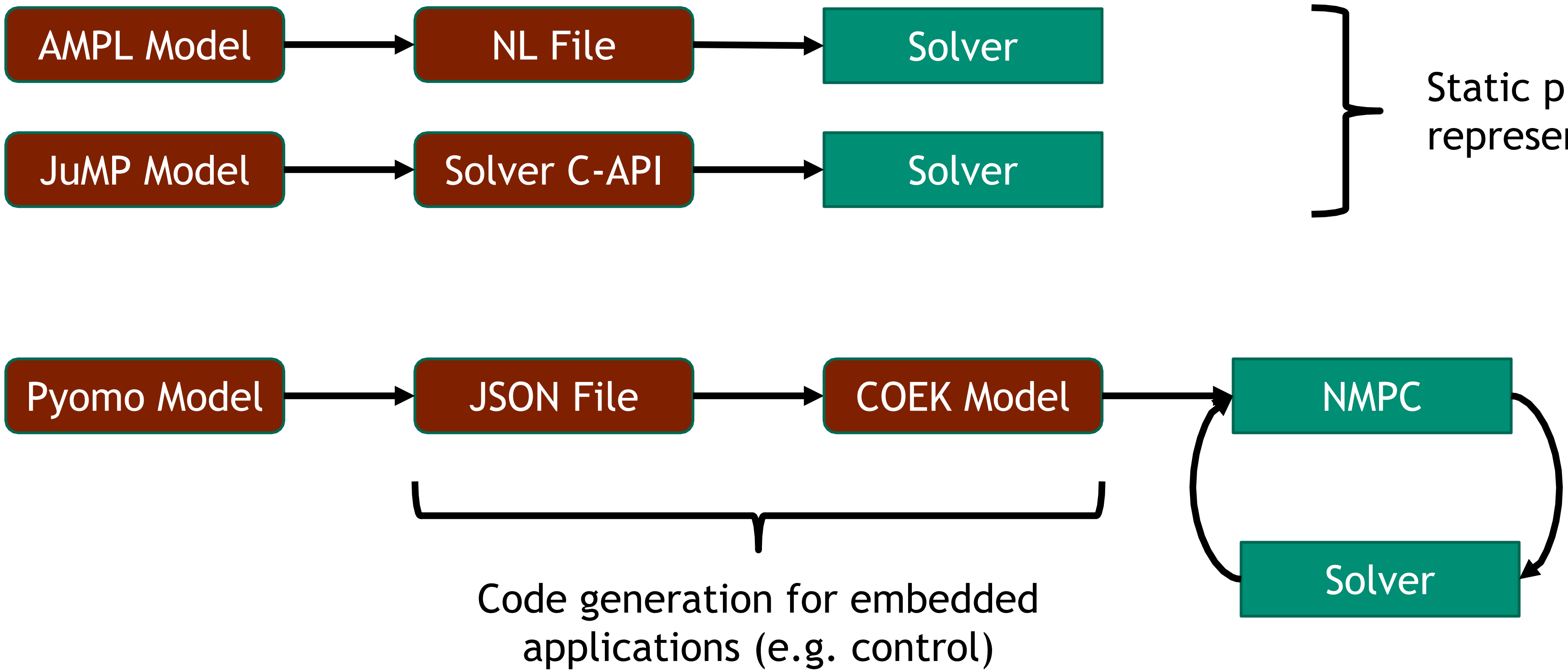
Pro:
- Flexible model for applications

Con:
- Complex mapping of AD to HW

Performance Optimization

Can we use the modeling environment to tailor calculations in the optimization workflow?

Idea 2: Rethink workflow to exploit model structure

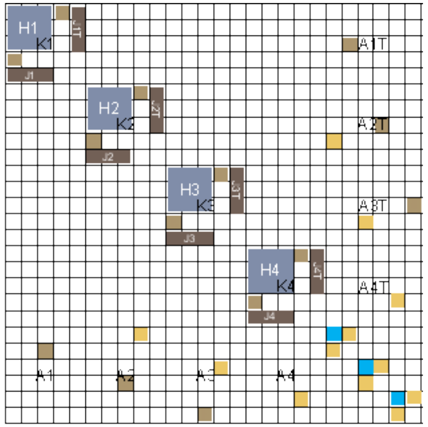


Performance Optimization

Can we use the modeling environment to tailor calculations in the optimization workflow?

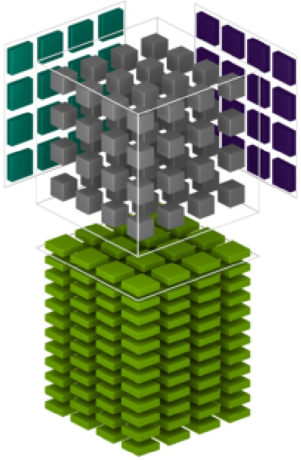
Idea 3: Exploit model structure to parallelize optimization workflow

Coarse-Grain Decomposition



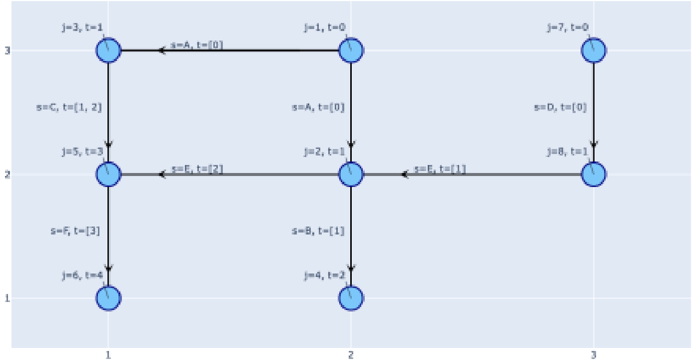
Parapint: Parallel-in-time decomposition

Fine-Grain Mapping of Dense Kernels



Dense matrix-matrix multiplication on tensor cores

Sparse, Irregular Computations?



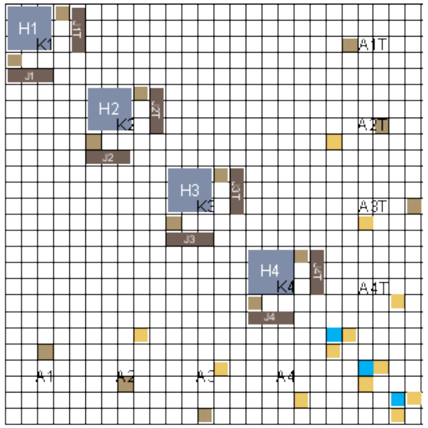
ARIAA: Mapping execution graphs onto data flow architectures

Performance Optimization

Can we use the modeling environment to tailor calculations in the optimization workflow?

Idea 4: Exploit model structure to automatically interface with distributed optimization solvers

Unstructured Coarse-Grain Decomposition



Can we automatically partition model generation across processors?

How do we robustly parallelize model transformations?

How do we coordinate parallel model generation and setup of parallel solver?

Application-centric vs Solver-centric

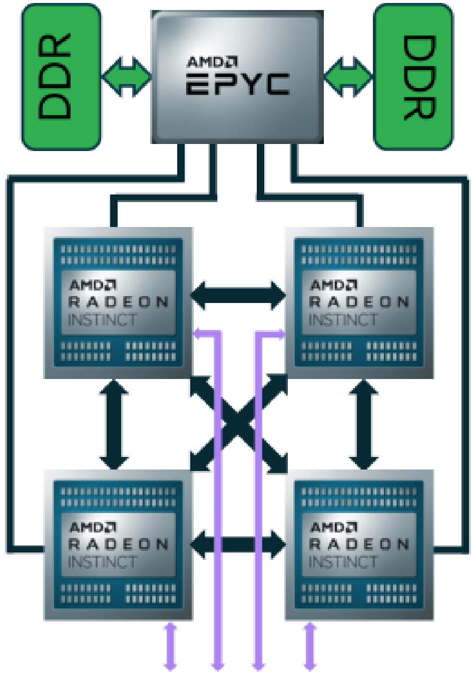
How can we leverage modeling environments to both express problem structure and inform the optimization workflow?

Challenge: Where do we construct the model and apply model transformations in our optimization workflows?

Application interaction is naturally supported by the CPU, but we want to exploit solvers running on GPUs.

Do we represent models on CPUs or GPUs or both?

If "both", then how do we manage multiple representations?



Application-centric vs Solver-centric

How can we leverage modeling environments to both express problem structure and inform the optimization workflow?

Idea 1: Model expansion on GPUs

PYOMO DAE

- Continuous domains
- Ordinary differential equations
- Partial differential equations
- Systems of differential algebraic equations
- Higher order differential equations and mixed partial derivatives

$$\dot{x} = f(x, y, u)$$
$$0 = g(x, y, u)$$

Pyomo DAE:
1. Used to express model dynamics within Pyomo
2. Includes model transformations to discretize the model (e.g. Collocation)

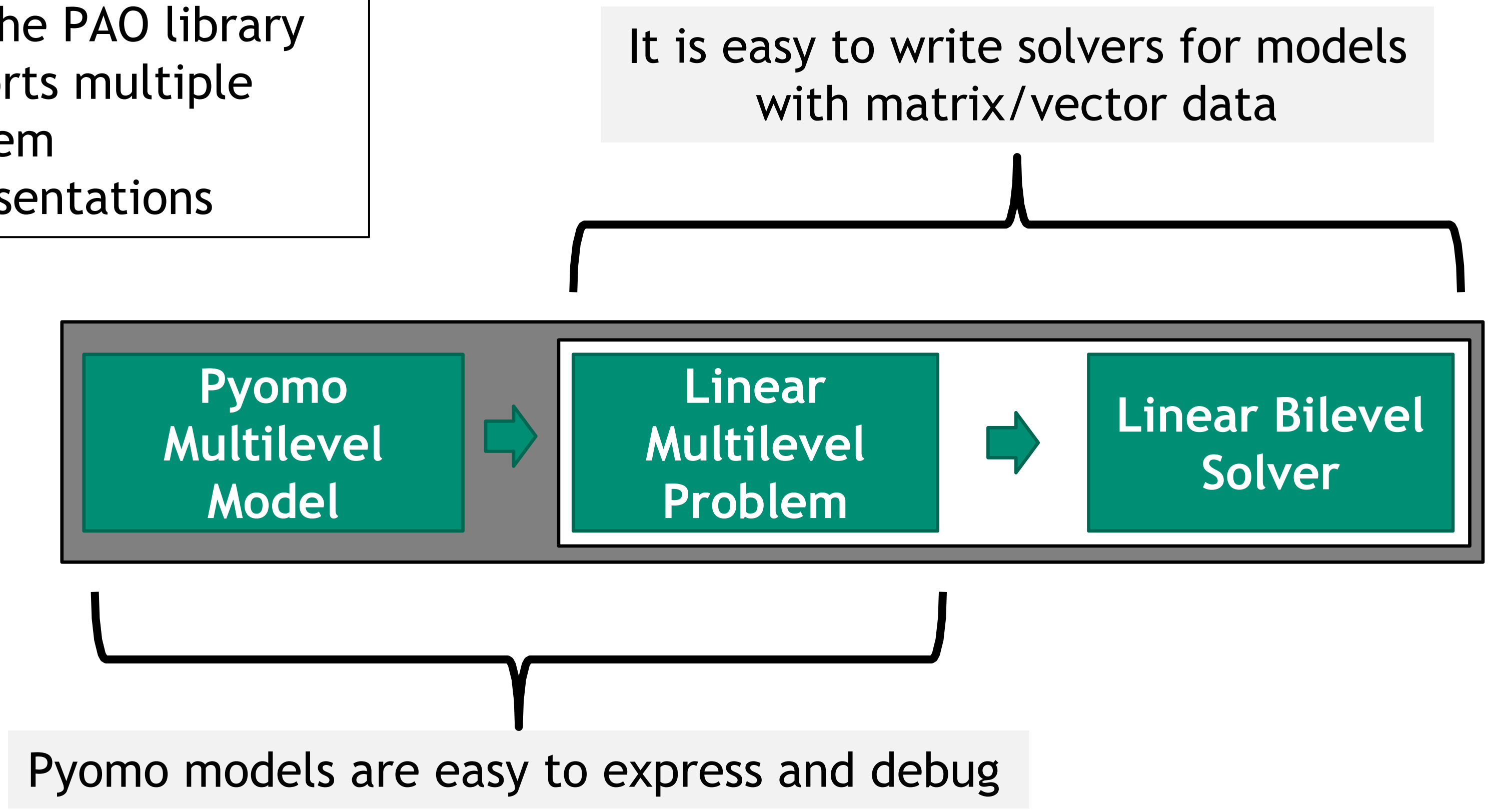
Key Idea:
- Use transformation to expand model directly on GPU
- Can leverage HW-specific features to tailor model calculations

Application-centric vs Solver-centric

How can we leverage modeling environments to both express problem structure and inform the optimization workflow?

Idea 2: Facilitate user- and solver- specific representations

E.g. the PAO library supports multiple problem representations



Key Idea:

- Explicitly support CPU- and GPU-specific model representations

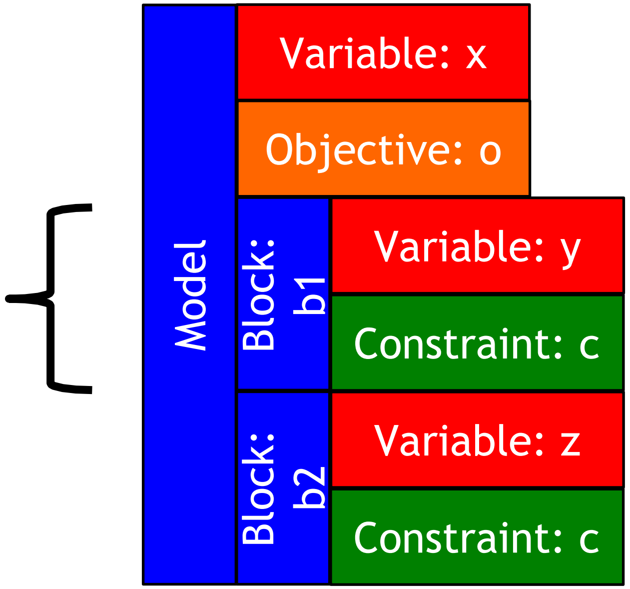
Application-centric vs Solver-centric

How can we leverage modeling environments to both express problem structure and inform the optimization workflow?

Idea 3: Use modeling environments that facilitate the application of model transformations on CPUs, GPUs and between them

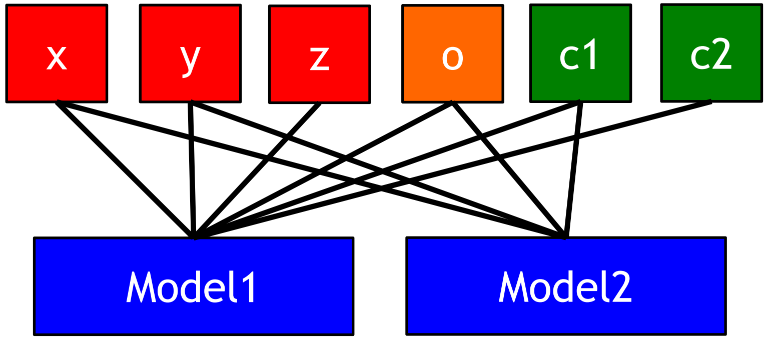
Blocks can be transformed in a modular manner

E.g. locally transform complementarity conditions to big-M representation



Pyomo

VS



AMPL, GAMS, …

SME- vs Data-driven models

Can we develop effective strategies to integrate both SME and data-driven modeling strategies?

Challenge: Evolve our modeling capabilities to support a continuum of application needs

SME-driven Models ⟷ **Data-driven Models**

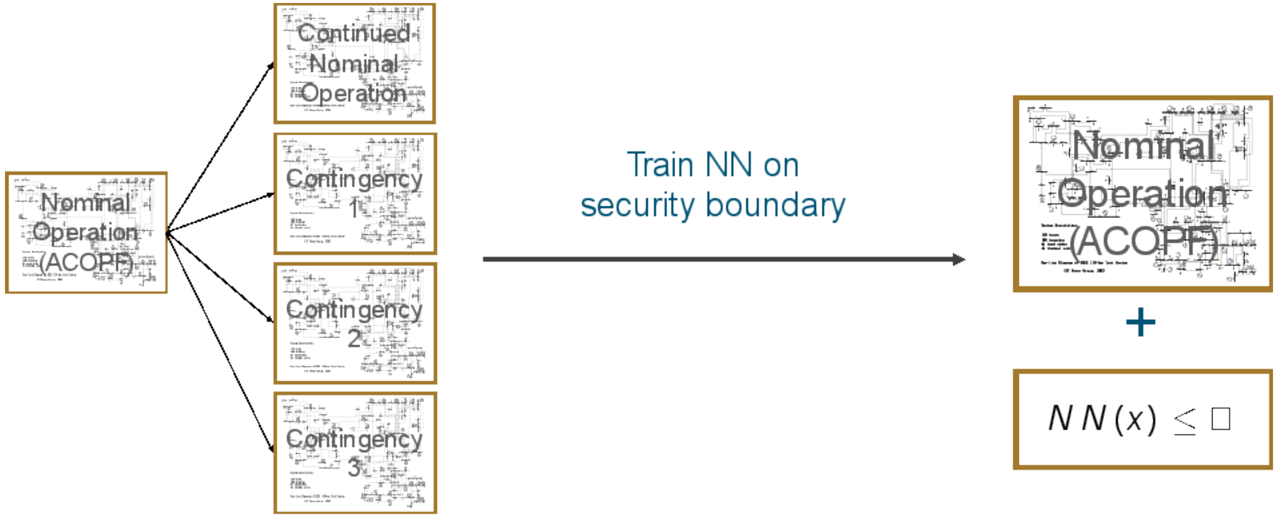
Note: There have been few demonstrations of capabilities "in the middle"

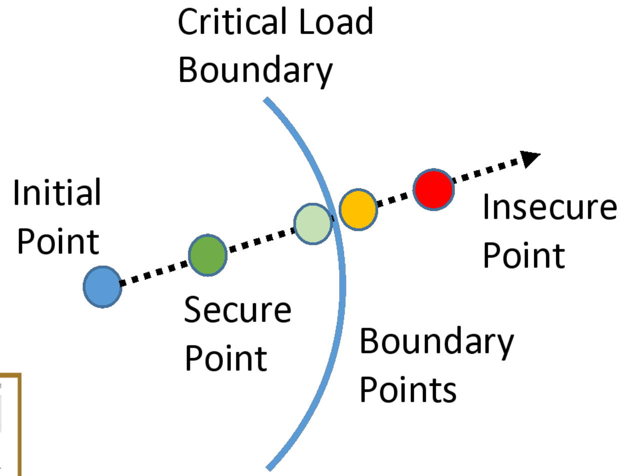
SME- vs Data-driven models

Can we develop effective strategies to integrate both SME and data-driven modeling strategies?

Idea 1: Augment SME models with embedded data-driven models

E.g. SNL Redly project: use neural network to replace contingency constraints





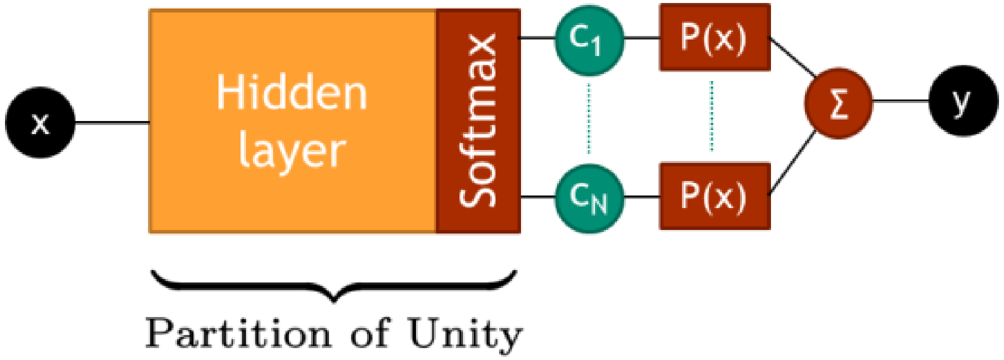
We need optimization modeling environments that can effectively represent large ML models

SME- vs Data-driven models

Can we develop effective strategies to integrate both SME and data-driven modeling strategies?

Idea 2: Data-driven methods tailored for specific application domains

E.g. Partition of unity networks



"Partition of unity networks: deep hp-approximation."

Partition of unity networks mimic the structure of traditional finite elements

Data-driven exterior calculus discovers bilinear form that conserves mass/momentum/energy without knowing underlying physics

A key challenge is optimization methods that can handle general nonlinear constraints

The End