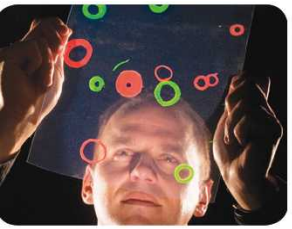
SAND2016-11102PE



Sandia National Laboratories

Exceptional service in the national interest

Pyomo
An Extensible Modeling Language for Mathematical Programming

William Hart
Sandia National Laboratories
wehart@sandia.gov

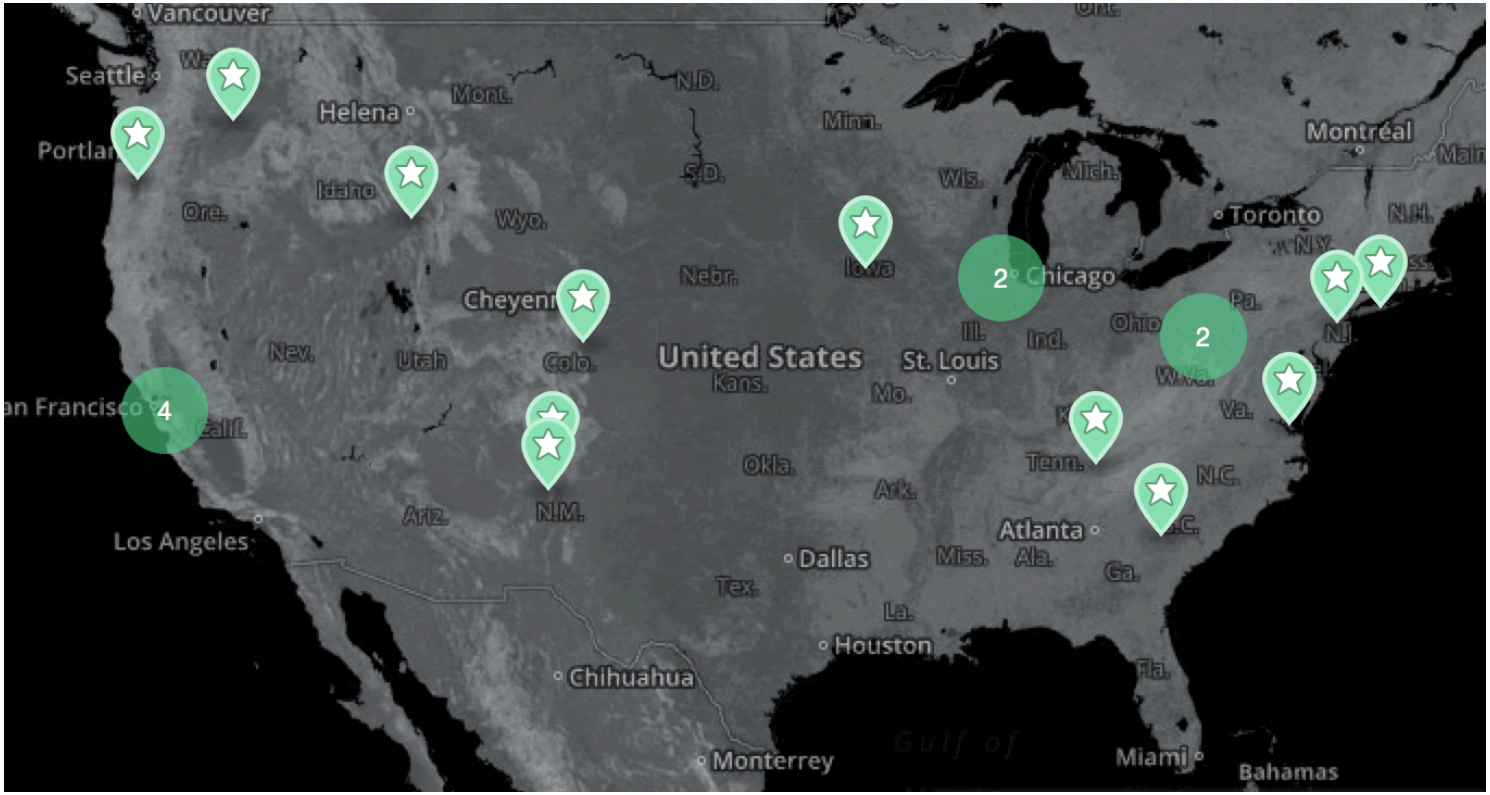
September 27, 2016

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Overview

- About Sandia National Laboratories
- Solving optimization problems with Pyomo
- Pyomo's extensible architecture

DOE National Laboratories



DOE's National Laboratories

DOE National Labs address …

- large scale, complex research and development challenges …
- with a multidisciplinary approach …
- that places an emphasis on translating basic science to innovation.

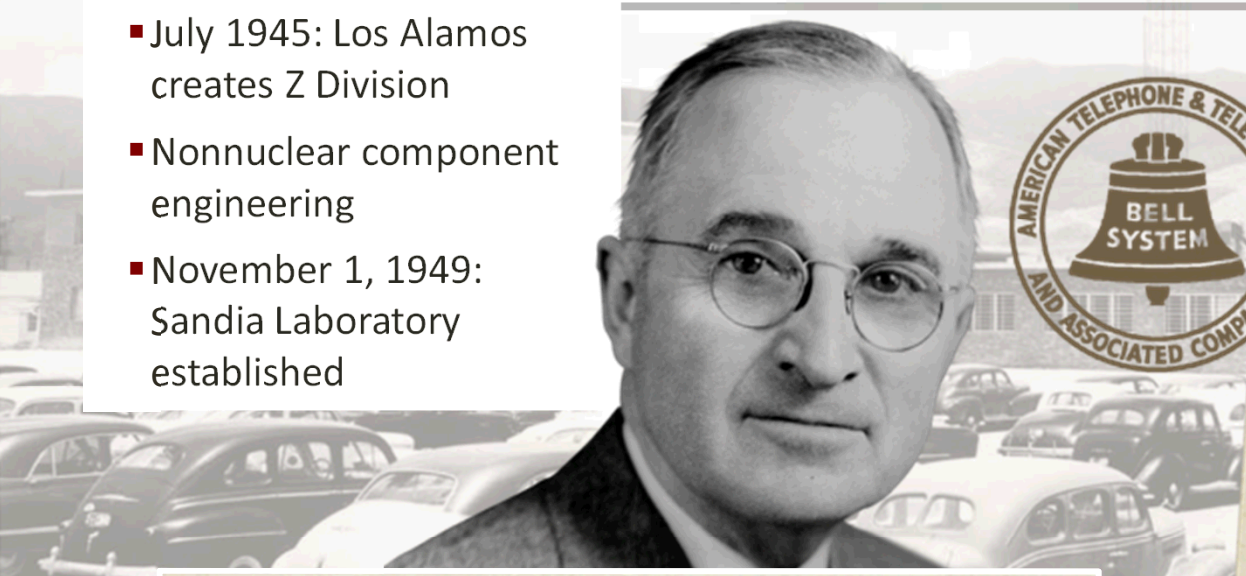
The Energy Department's National Labs tackle critical scientific challenges:

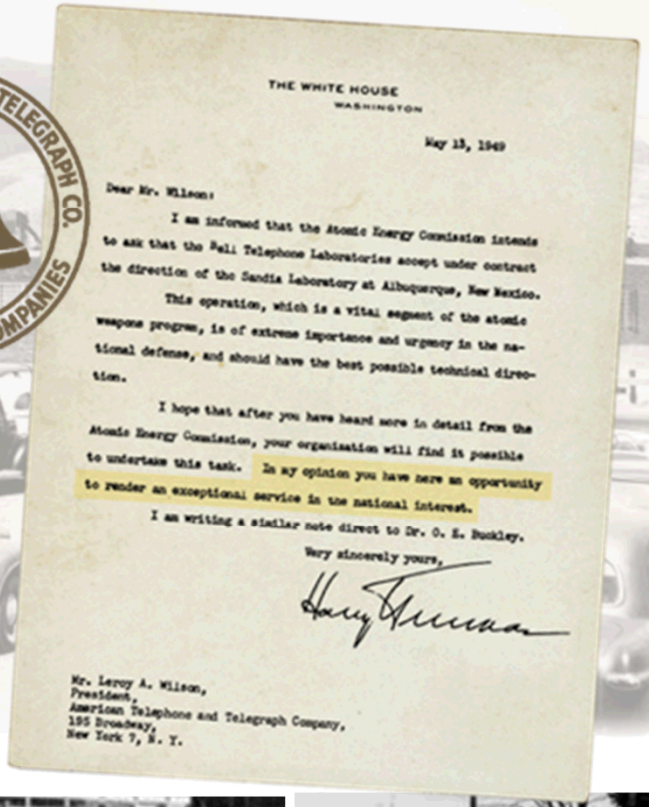
- Conduct research of the highest caliber
- Advance U.S. energy independence and leadership
- Enhance global, national, and homeland security
- Design, build, and operate distinctive scientific instrumentation and facilities

Sandia's History

Exceptional service in the national interest

- July 1945: Los Alamos creates Z Division
- Nonnuclear component engineering
- November 1, 1949: Sandia Laboratory established





THE WHITE HOUSE
WASHINGTON

May 13, 1949

Dear Mr. Wilson:

I am informed that the Atomic Energy Commission intends to ask that the Bell Telephone Laboratories accept under contract the direction of the Sandia Laboratory at Albuquerque, New Mexico.

This operation, which is a vital segment of the atomic weapons program, is of extreme importance and urgency in the national defense, and should have the best possible technical direction.

I hope that after you have heard more in detail from the Atomic Energy Commission, your organization will find it possible to undertake this task. In my opinion you have here an opportunity to render an exceptional service in the national interest.

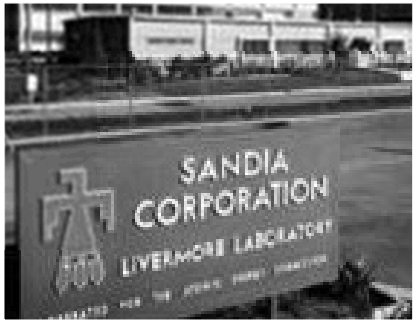
I am writing a similar note direct to Dr. O. E. Buckley.

Very sincerely yours,

Harry Truman

Mr. Leroy A. Wilson,
President,
American Telephone and Telegraph Company,
195 Broadway,
New York 7, N. Y.

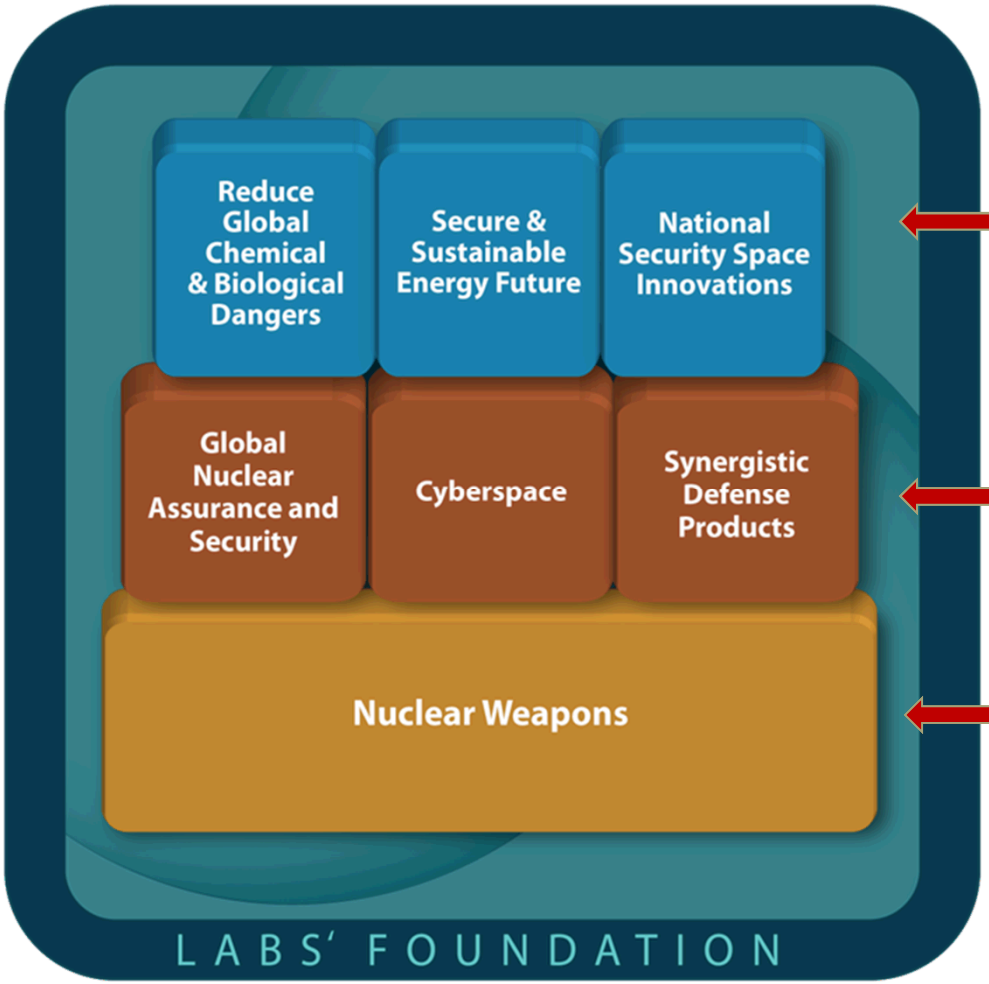
to undertake this task. In my opinion you have here an opportunity to render an exceptional service in the national interest.



Sandia's Evolving Mission

1950s	1960s	1970s	1980s	1990s	2000s	2010s
NW production engineering & manufacturing engineering	Development engineering	Multiprogram laboratory	Missile defense work	Post–Cold War transition	Expanded national security role post 9/11	LEPs Cyber, Biosecurity Proliferation
	Vietnam conflict	Energy crisis	Cold War	Stockpile stewardship		Evolving national security challenges

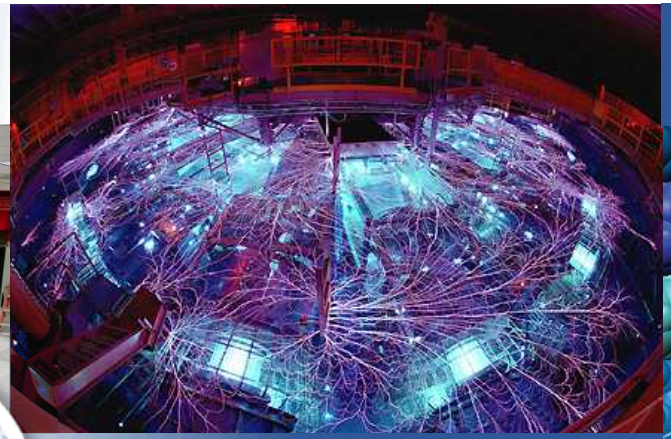
National Security Mission Areas



- These missions are critical to national security, and they leverage and enhance our capabilities.
- These missions are essential to sustaining Sandia's ability to fulfill its NW core mission.
- *Our core mission*: nuclear weapons (NW).

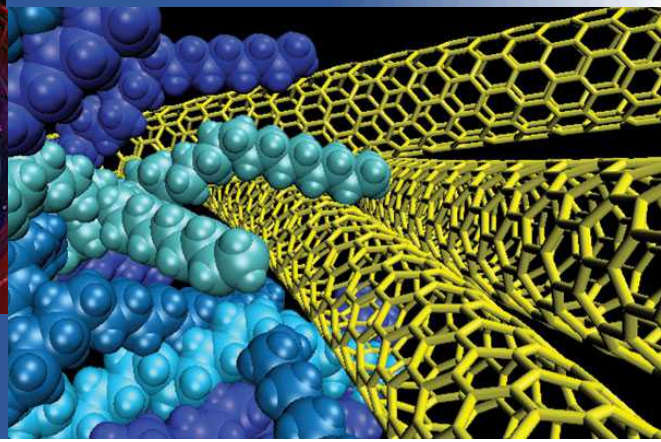
Research Foundations

Computing & Information Sciences

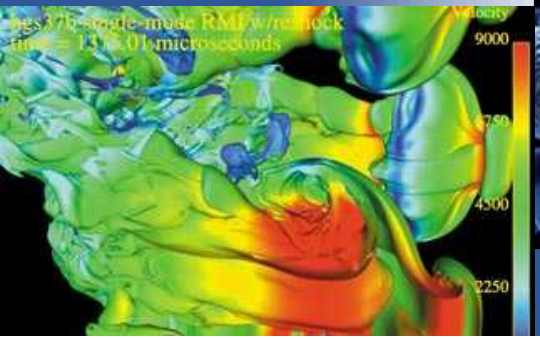


Radiation Effects & High Energy Density Science

Materials Sciences

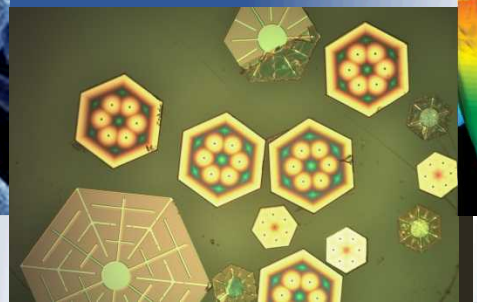


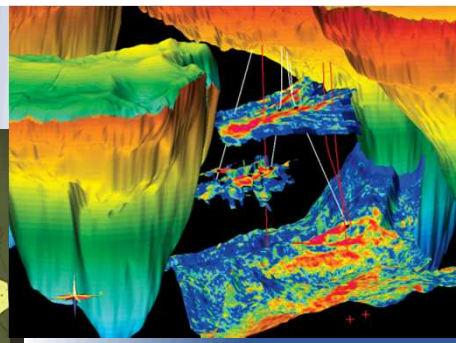
Engineering Sciences



Bioscience

Nanodevices & Microsystems



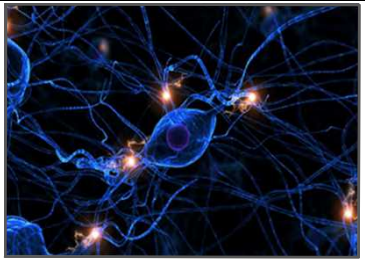


Geoscience

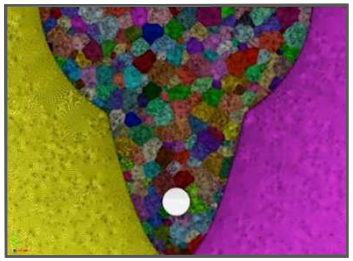
Sandia Research Challenges

Data Science

Detection at the Limit

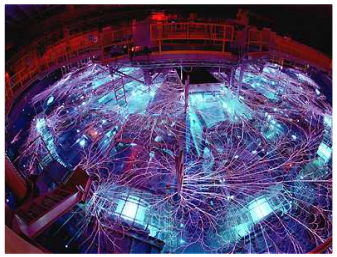


Engineering Abiotic-Biotic Living Systems

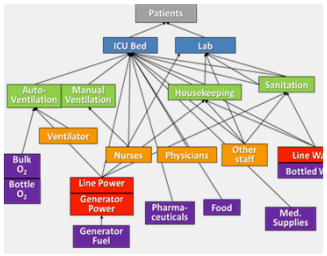


Engineering of Materials Reliability

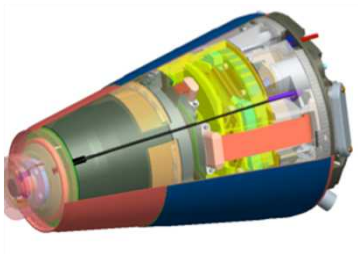
Power on Demand



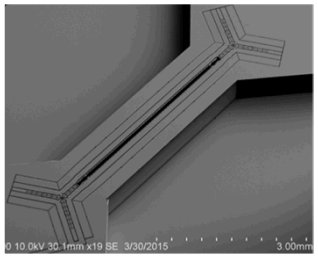
Pulsed Power Opportunities for Weapons & Effects Research



Resiliency in Complex Systems



Revolutionary Approaches to the Stockpile



Science & Engineering of Quantum Information Systems

Trusted Systems & Communication

Engage expertise from fundamental science to technology application

Pursue decade-scale "moon shot" goals guided by roadmaps

Create transformational capabilities that address mission-critical problems

Academic Alliance Program

Vision

- Enhance Sandia's capability base and mission impact by partnering more closely with leading universities to advance science and engineering in support of national security

Strategic Objectives

- Solve significant problems we could not address alone
- Sustain and enrich our talent pipeline
- Accelerate the commercialization and adoption of new technologies

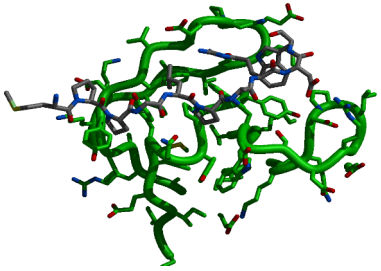
Overview of Academic Alliance at Purdue

- On-site Partnerships Manager
  - Bill Hart (since 1/16)
- Developing Technical Focus Areas
  - Bio, Complex Systems, Cyber, Data, Energy, Nano/Micro, Hypersonics/Propulsion
- Joint Funding
  - Sandia LDRD
  - DARPA, NSF, MDA
- Joint Workshops
  - V&V of Models of Complex Systems
  - Nanoscience and Microsystems Engineering

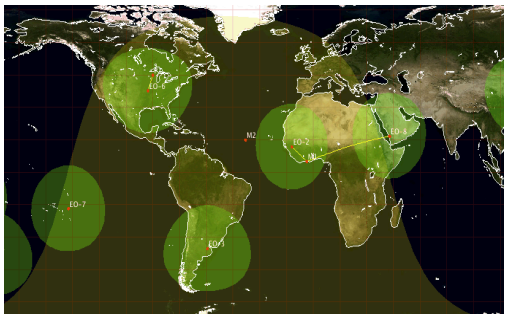
Overview

- About Sandia National Laboratories
- Solving optimization problems with Pyomo
- Pyomo's extensible architecture

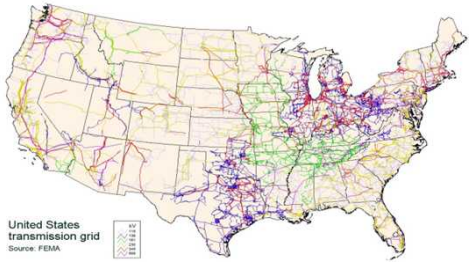
Examples of Optimization Applications



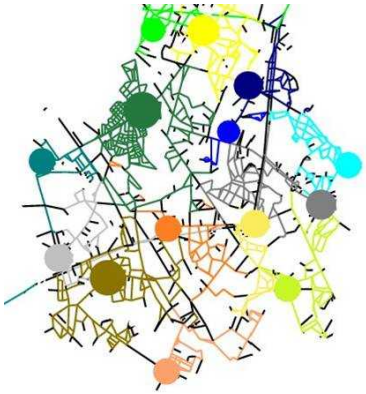
Molecular Docking



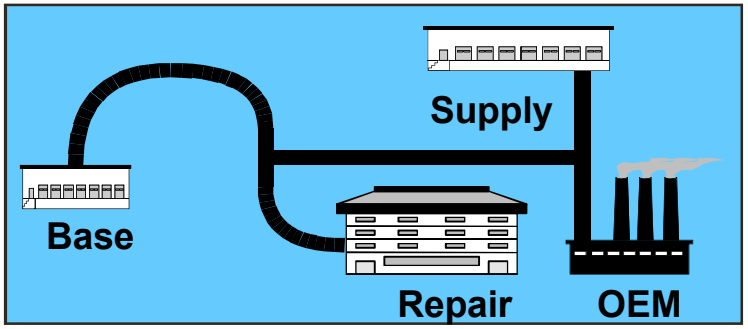
Satellite Scheduling



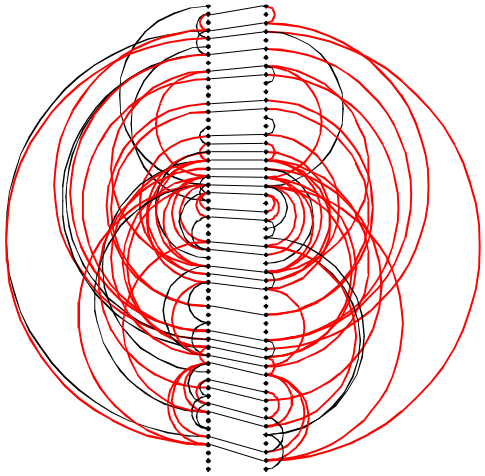
Power Grid Planning



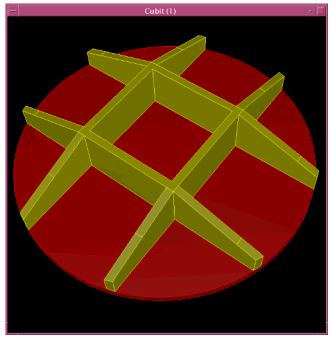
Water Security



Inventory Logistics

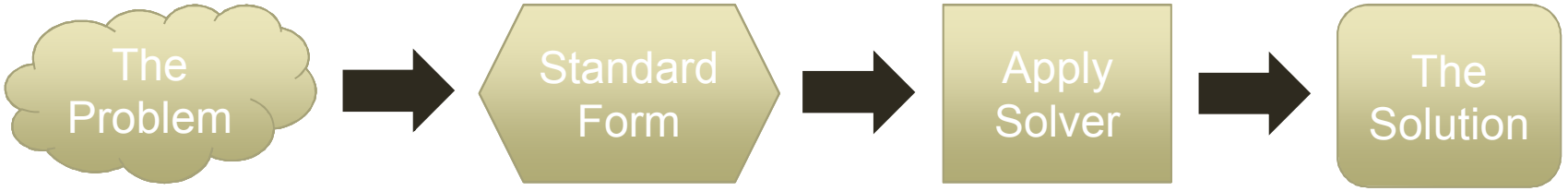


Protein Comparison



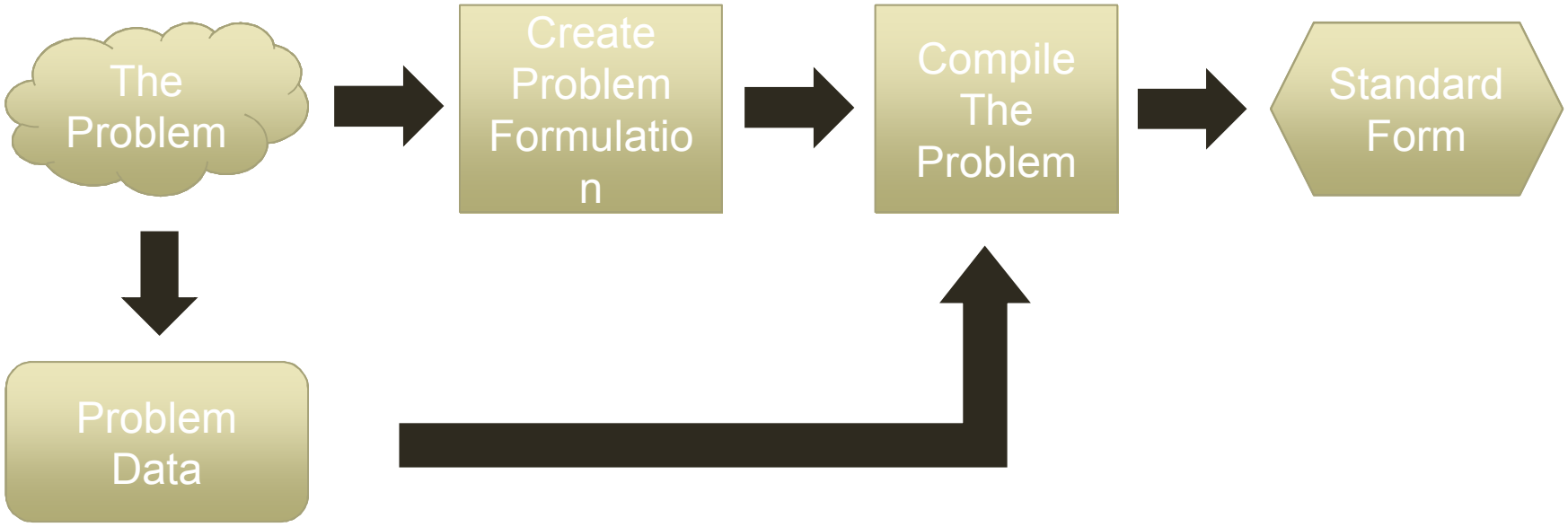
Topology Design

Solving an Optimization Problem



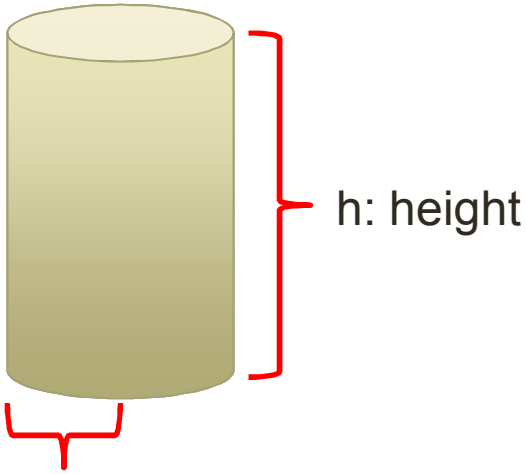
$$\begin{aligned} \min_x \quad & f(x) \\ & g(x) \leq 0 \\ & h(x) = 0 \end{aligned}$$

Generating the Standard Form



Example: Optimizing Soda Cans

Problem: minimize the metal used in a soda can



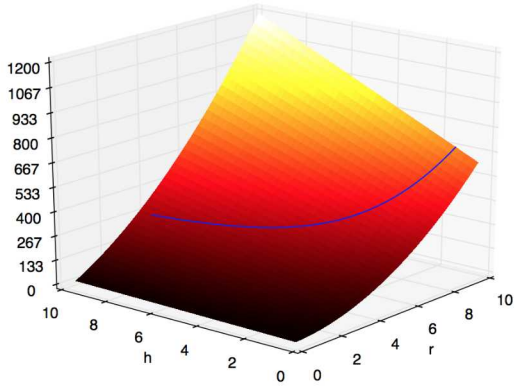
Formulation

- Objective minimize surface area of a soda can
- Use a cylinder to represent an idealized can
- Fix the volume of the can (355 cm^2)

Standard Form

minimize $2 \pi r (r + h)$ *# Surface Area*

with $\pi h r^2 = 355$ *# Volume*



Compiling an Optimization Problem

Note: an optimization problem needs to be compiled into a form that a solver can understand

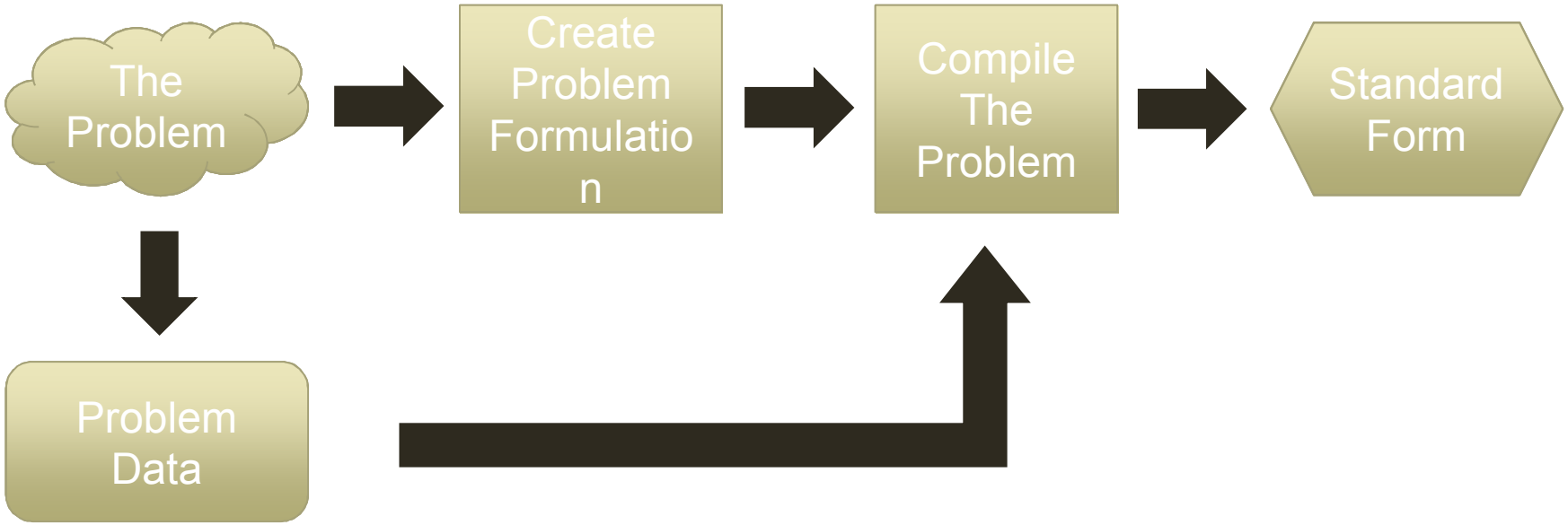
Examples

- A code or script that computes objectives and constraints
  - This includes a subroutine that does this computation
- Matrices and vectors used in a standard form (e.g. for linear programming)

$$\begin{aligned} \min_x \quad & c^T x \\ & Ax = b \\ & x \geq 0 \end{aligned}$$

- An general representation of algebraic forms of the objectives and constraints
  - This could be an XML file, or the standard NL file format

Generating the Standard Form



Optimization Modeling Languages

MLs are commonly used to express and compile optimization models

Features

- Provide a natural syntax to describe mathematical models
- Formulate large models with a concise syntax
- Separate modeling and data declarations
- Enable data import and export in commonly used formats

Impact

- Robustly automate the compilation of models into standard forms
- Simplify the expression of large complex applications
- Integrated support of model analysis
  - E.g. Automatic differentiation of complex nonlinear models

Pyomo: A ML Built in Python

Sandia National Laboratories

Open Source License

- No licensing issues w.r.t. the language itself
- Can extend/refine the language in some cases

Extensibility and Robustness

- Highly stable and well-supported
- Simple model for integrating code developed by a user

Support and Documentation

- Extensive online documentation and several excellent books
- Long-term support for the language is not a factor

Standard Library

- Includes a large number of useful modules.

Scripting

- Language features includes functions, classes, looping, procedural constructs, etc.

Portability

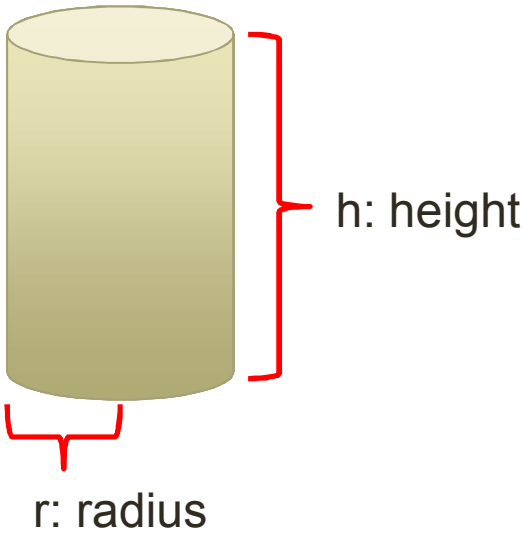
- Widely available on many platforms

Revisiting the Soda Can Example

Formulation

minimize $2 \pi r (r + h)$ *# Surface Area*

with $\pi h r^2 = 355$ *# Volume*



Pyomo Model

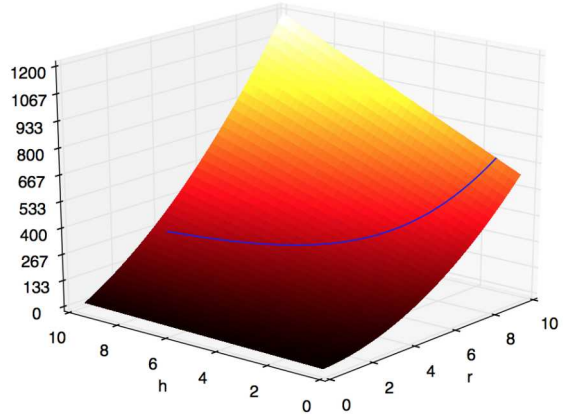
```
# sodacan.py
from pyomo.environ import *
from math import pi

M = ConcreteModel()

M.r = Var(bounds=(0,None))
M.h = Var(bounds=(0,None))

M.o = Objective(expr=  2*pi*M.r*(M.r + M.h))

M.c = Constraint(expr= pi*M.h*M.r**2 == 355)
```



Comparison with Advanced MLs

		Advanced Modeling Capabilities	Programming Language	Object-Oriented Design	Hierarchical Models	Model Transformations
Commercial Software	AIMMS	MPEC, SP, RO, CP	Custom	No	No	No
	AMPL	MPEC, CP	Custom	No	No	No
	GAMS	MPEC, BP, SP, GDP	Custom	No	No	No
	LINGO	SP	Custom	No	No	No
	MPL	MPEC	Custom	No	No	No
Open Source Software	JuMP	SP, RO	Julia	Yes	No	No
	Pyomo	**MPEC, BP SP, GDP, DAE**	**Python**	**Yes**	**Yes**	**Yes**
	YALMIP	MPEC, BP, RO, CP	Matlab	No	No	No

Advanced Modeling Capabilities

- Bilevel Programming (BP), Constraint Programming (CP), Differential Algebraic Equation Models (DAE), Mathematical Programming with Equilibrium Constraints (MPEC), Robust Optimization (RO), Stochastic Programming (SP)

More than just modeling …

Scripting

- Construct models using native Python data
- Iterative analysis of models leveraging Python functionality
- Data analysis and visualization of optimization results

Model transformations (a.k.a. reformulations)

- Automate generation of one model from another
- Leverage Pyomo's object model to apply transformations sequentially
- E.g.: relax integrality, GDP -> Big M

Meta-solvers

- Integrate scripting and/or transformations into optimization solver
- Leverage Python's introspective nature to build "generic" capabilities
- E.g.: progressive hedging, SP extensive form -> MIP

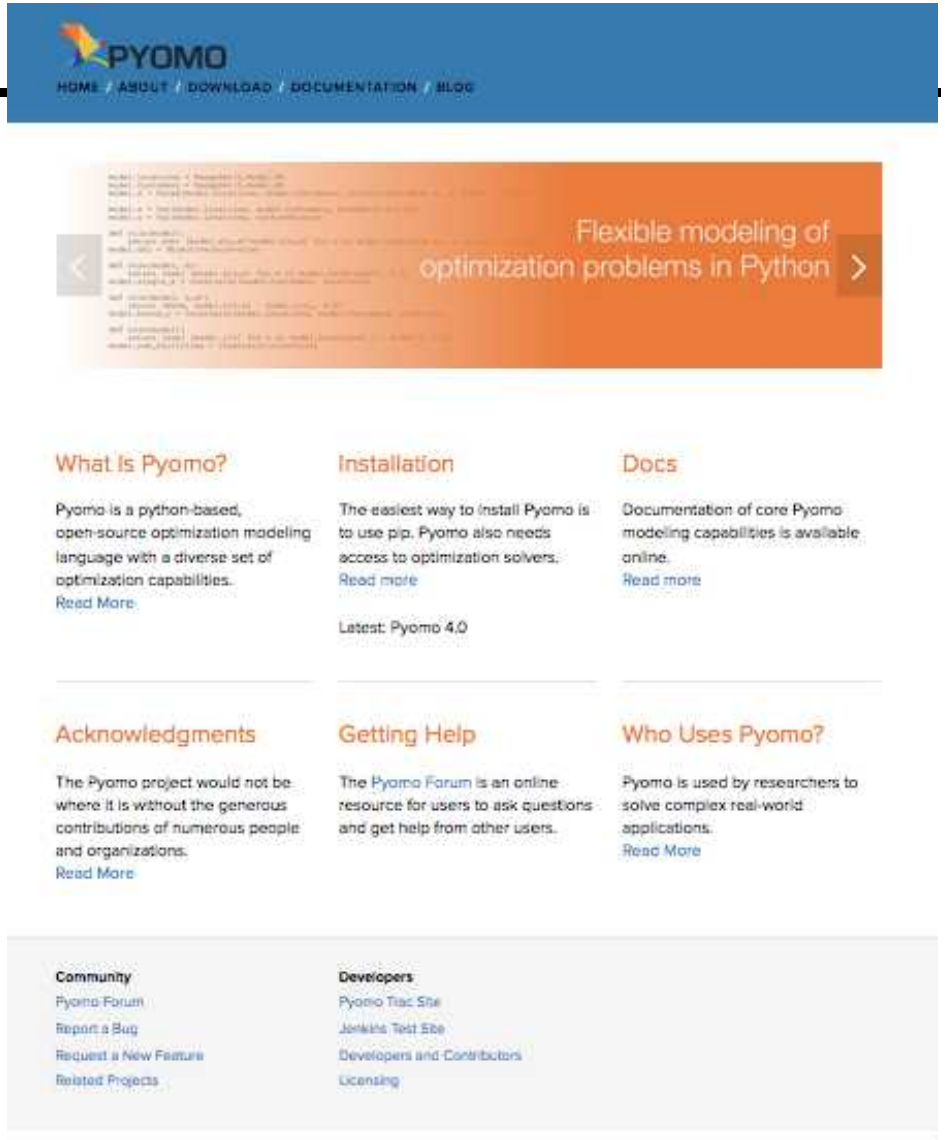
For More Information

See the new Pyomo homepage

- www.pyomo.org

The Pyomo homepage provides a portal for:

- Online documentation
- Installation instructions
- Help information
- Developer links
- A gallery of simple examples



The End?

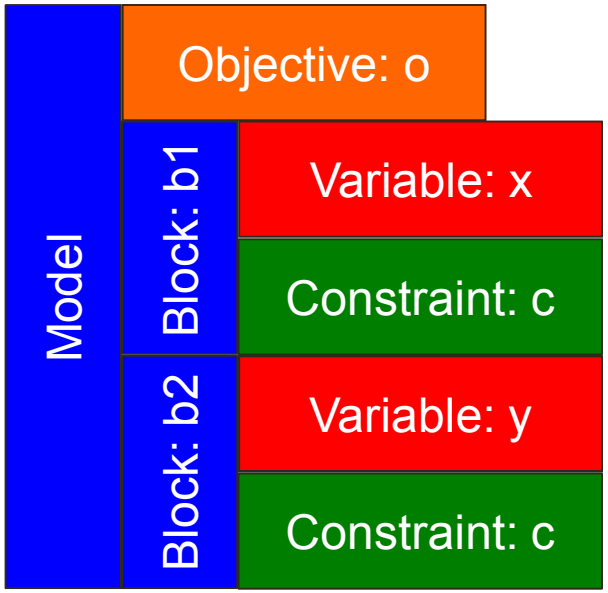
Overview

- About Sandia National Laboratories
- Solving optimization problems with Pyomo
- Pyomo's extensible architecture

Pyomo's Object-Oriented Design

Explicit Object Composition

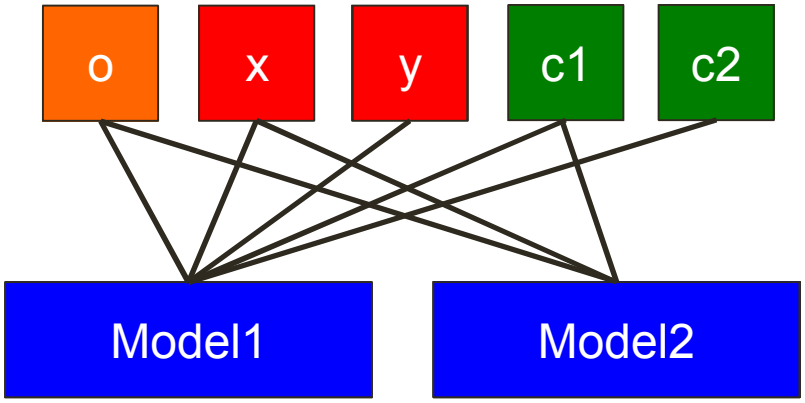
- Models are composed of independent declarations of components
- A model object *contains* component objects as Python attributes
- The Block component is used to express hierarchical models
- Component objects are owned by a parent model or block object



Comparison with Aggregation

Some MLs use the *aggregation* design principle

- Components are declared separately
- The user can select a subset that comprise a model
- Different models can share component declarations



Notes:

- Modeling as an aggregate of components is very flexible way to reuse component declarations
- Hierarchical relationships cannot be easily expressed

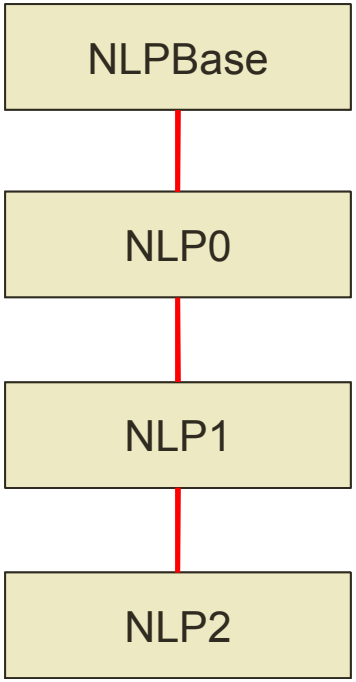
Comparison with Inheritance

Many optimization libraries use the *inheritance* design principle

- Optimization problems are encapsulated in objects
- Class inheritance is used to create new types of optimization problems

Example: OPT++ problem class hierarchy

- NLPX defines the order of the derivative supported by the problem
- Solver classes can be tailored to suitable classes of problems



Notes:

- Inheritance imposes the IS-A semantics, which may be limiting
- A combinatorial explosion of inherited classes arises when

Implications of Pyomo's Design (1)

Claim: Pyomo's design allows it to easily express complex, structure problems

- Can express hierarchically structure problems
- Models can include declarations of disparate components to construct "hybrid" problem formulations
- Model transformations can operate on model components in a modular manner

Note: Pyomo can express a wider range of models than we can solve!

Example: Multilevel Programming (1)

Idea: Represent a hierarchy of decision-makers with optimization sub-models

$$\begin{aligned} \min_{x,y,z} \quad & x - 4y + 2z \\ \text{s.t.} \quad & -x - y \leq -3 \\ & -3x + 2y - z \geq -10 \end{aligned}$$

$$\begin{aligned} \min_{y,z} \quad & x + y - z \\ \text{s.t.} \quad & -2x + y - 2z \leq -1 \\ & 2x + y + 4z \leq 14 \end{aligned}$$

$$\begin{aligned} \min_{z} \quad & x - 2y - 2z \\ \text{s.t.} \quad & 2x - y - z \leq 2 \end{aligned}$$

Example: Multilevel Programming (2)

Pyomo uses SubModel components to express optimization sub-models

```
from pyomo.environ import *
from pyomo.bilevel import *

M = ConcreteModel()
M.x = Var()
M.s = SubModel()
M.s.y = Var()
M.s.s = SubModel()
M.s.s.z = Var()

M.o = Objective(expr=       M.x - 4*M.s.y + 2*M.s.s.z)
M.c1 = Constraint(expr=   - M.x -   M.s.y             <=  -3)
M.c2 = Constraint(expr=  -3*M.x + 2*M.s.y             >= -10)
M.s.o = Objective(expr=     M.x +   M.s.y -   M.s.s.z)
M.s.c1 = Constraint(expr=-2*M.x +   M.s.y - 2*M.s.s.z <=  -1)
M.s.c2 = Constraint(expr= 2*M.x +   M.s.y + 4*M.s.s.z <=  14)
M.s.s.o = Objective(expr=   M.x - 2*M.s.y - 2*M.s.s.z)
M.s.s.c = Constraint(expr=2*M.x -   M.s.y -   M.s.s.z <=   2)
```

Implications of Pyomo's Design (2)

Claim: Pyomo's design is fundamentally more extensible than other MLs

- Pyomo's open source distribution facilitates software extensions
- Researchers can implement new modeling components
- There are no a priori restrictions on how new components are added to Pyomo models

The End!

Pyomo Developer Team

- Gabe Hackebeil
- Bill Hart
- Carl Laird
- Bethany Nicholson
- John Siirola*
- Jean-Paul Watson
- David Woodruff

*Developed logic supporting blocks and block connectors

A Problem with Inheritance

Combinatorial explosion of classes

E.g. (Derivative Order) X (Constraint Type)