APMonitor Modeling Language

John Hedengren
Brigham Young University

Advanced Process Solutions, LLC
http://apmonitor.com

Overview of APM

- Software as a service accessible through:
  - MATLAB, Python, Web-browser interface
  - Linux / Windows / Mac OS / Android platforms
- Solvers
  - APOPT[1], BPOPT[1], IPOPT[2], SNOPT[3], MINOS[3]
- Problem characteristics:
  - Large-scale
  - Nonlinear Programming (NLP)
  - Mixed Integer NLP (MINLP)
  - Multi-objective
  - Real-time systems
  - Differential Algebraic Equations (DAEs)

$$
\begin{aligned}
&\min J(x,y,u,z) \\
&s.t.\ 0 = f\left(\frac{\partial x}{\partial t}, x, y, u, z\right) \\
&\quad 0 = g(x,y,u,z) \\
&\quad 0 < h(x,y,u,z) \\
&\quad x, y \in \Re^n \ u \in \Re^m \ z \in I^m
\end{aligned}
$$

1 – APS, LLC
2 – EPL
3 – SBS, Inc.

Overview of APM

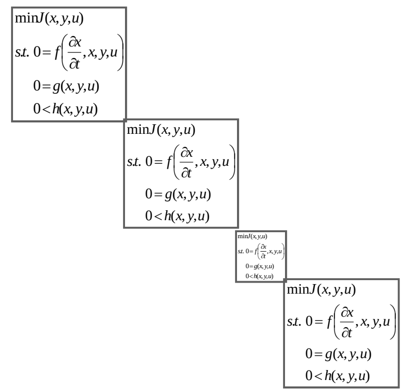
- Vector / matrix algebra with set notation
- Automatic Differentiation
  - Exact 1st and 2nd Derivatives
- Large-scale, sparse systems of equations
- Object-oriented access
  - Thermo-physical properties
  - Database of preprogrammed models
- Parallel processing
- Optimization with uncertain parameters
- Custom solver or model connections

Unique Features of APM

- Initialization with nonlinear presolve

$$\begin{aligned} &\min J(x,y,u) \\ &s.t.\ 0 = f\left(\frac{\partial x}{\partial t}, x, y, u\right) \\ &\quad 0 = g(x,y,u) \\ &\quad 0 < h(x,y,u) \end{aligned}$$

→



- Explicit variable substitution every function call

$$\begin{aligned} &\min J(x,y,u) \\ &s.t.\ 0 = f\left(\frac{\partial x}{\partial t}, x, y, u\right) \\ &\quad 0 = g(x,y,u) \\ &\quad 0 < h(x,y,u) \end{aligned}$$

→

$$\begin{aligned} &\min J(x,y,u) \\ &s.t.\ y_1 = g_1(x,u) \\ &\quad 0 = g_2(x, y_1, y_2, u) \\ &\quad 0 = f\left(\frac{\partial x}{\partial t}, x, y_1, y_2, u\right) \\ &\quad 0 < h(x, y_1, y_2, u) \end{aligned}$$

← Intermediate Variables

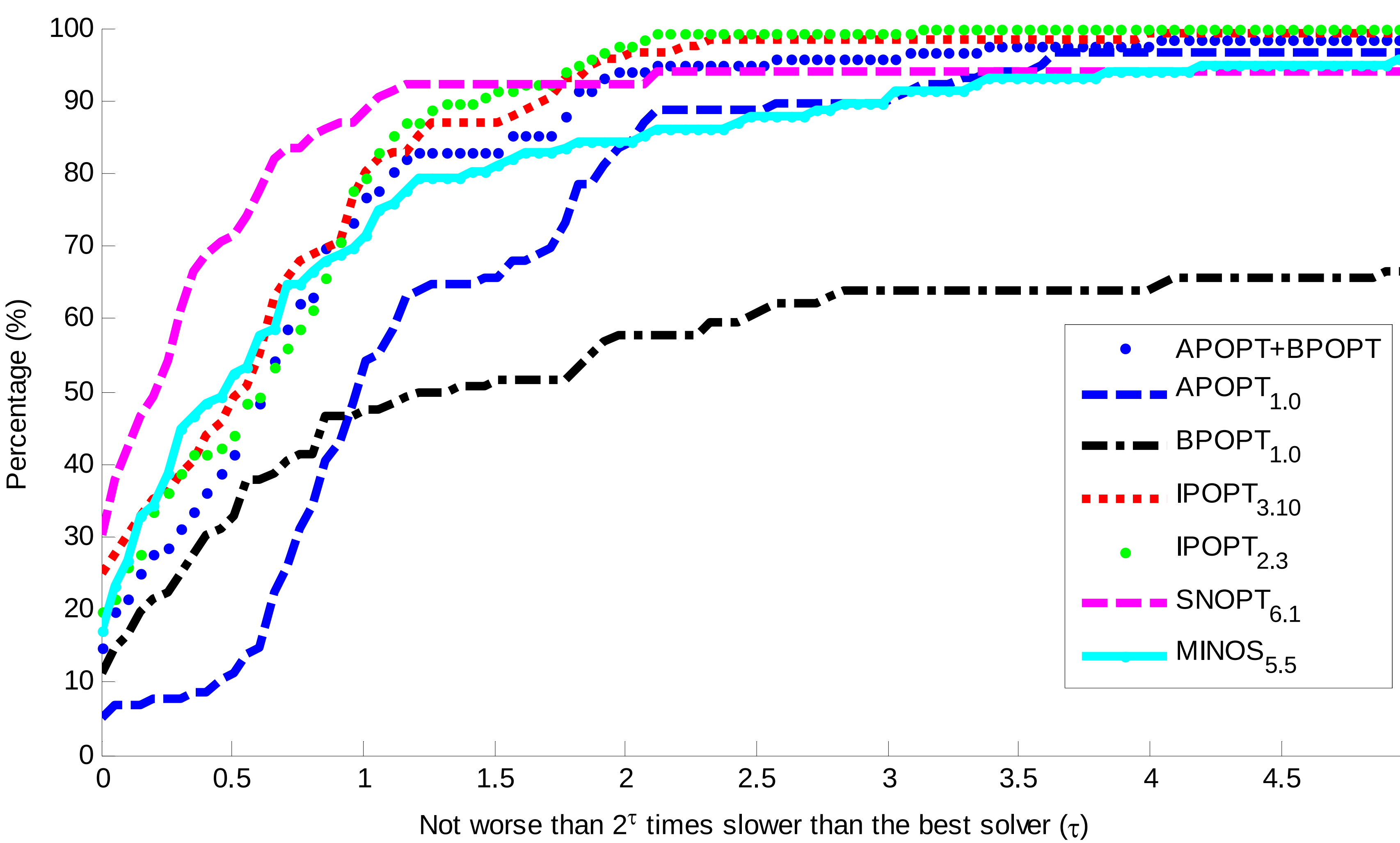
Unique Features of APM

- Model development workflow

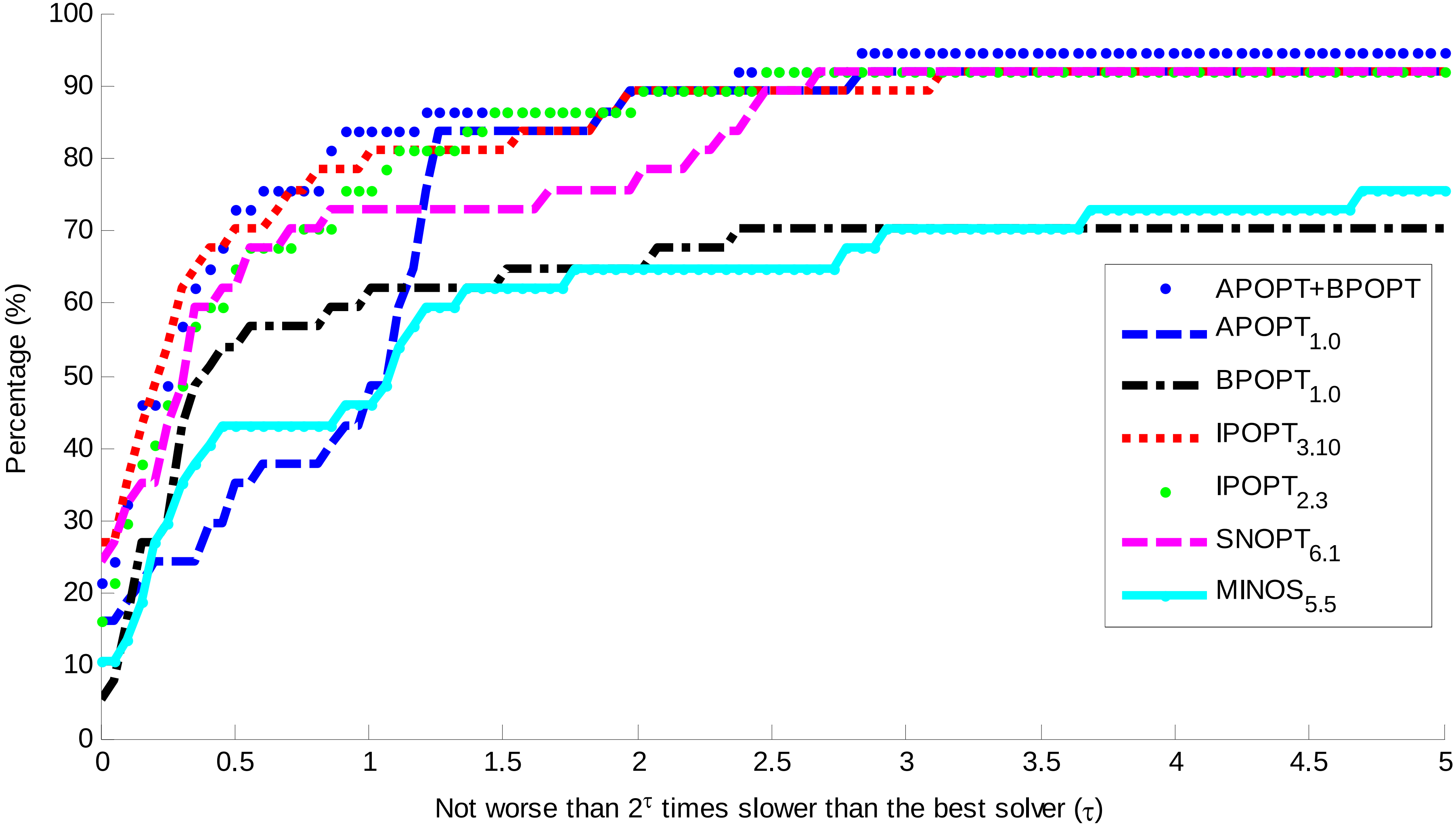
	Steady State	Dynamic	Sequential
Simulate	1	4	7
Estimate	2	5	-
Optimize	3	6	-

- Solve higher index DAEs (Index 3+ with APM)
  - Index-1 only (e.g. MATLAB ode15s)
  - Index-1 + Index-2 Hessenberg (e.g. DASPK)
- Classes of problems
  - LP, QP, NLP, ***DAE***
  - MILP, MIQP, MINLP, ***MIDAE***

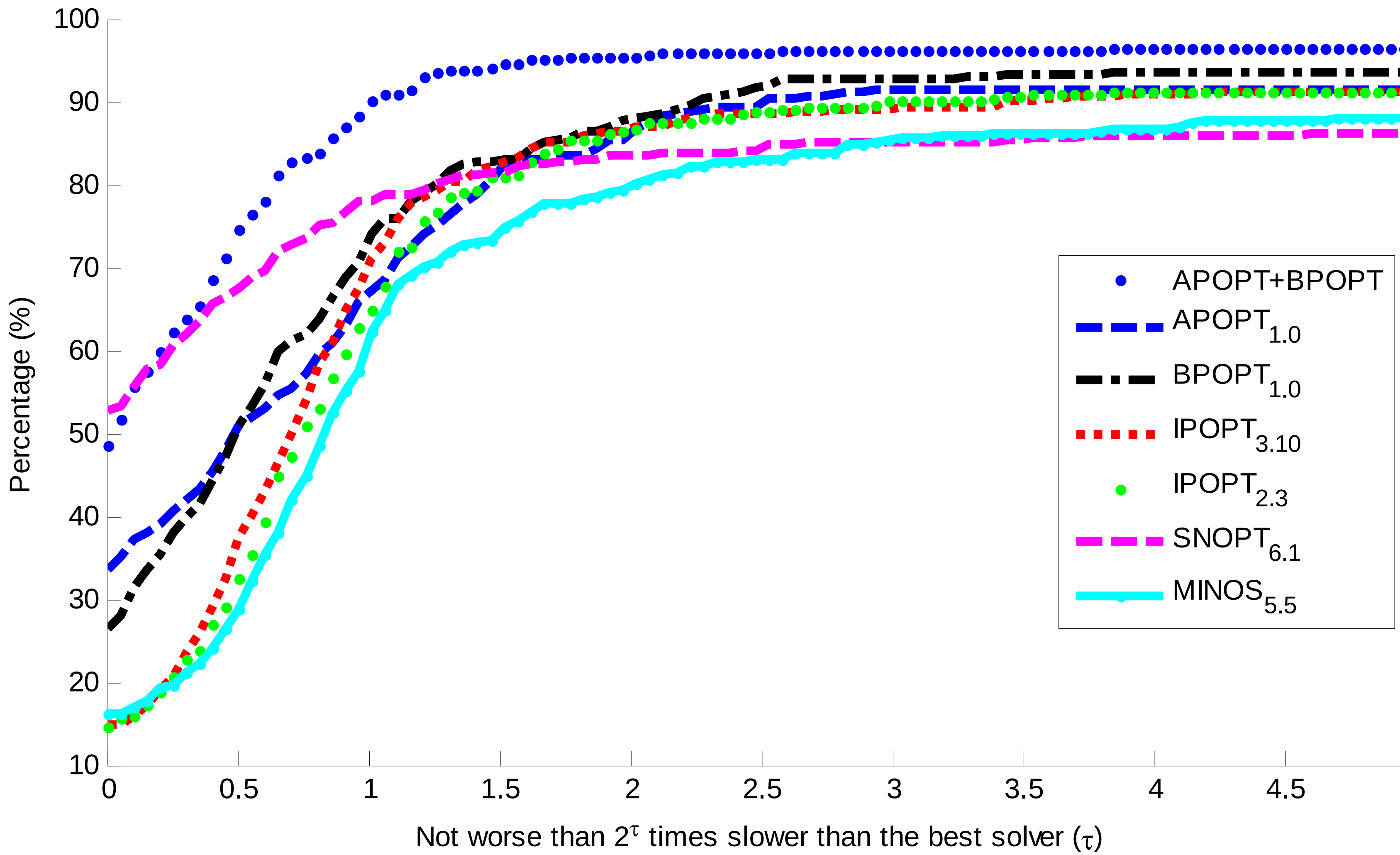
Solver Benchmarking – Hock-Schittkowski (116)



Solver Benchmarking - Dynamic Optimization (37)

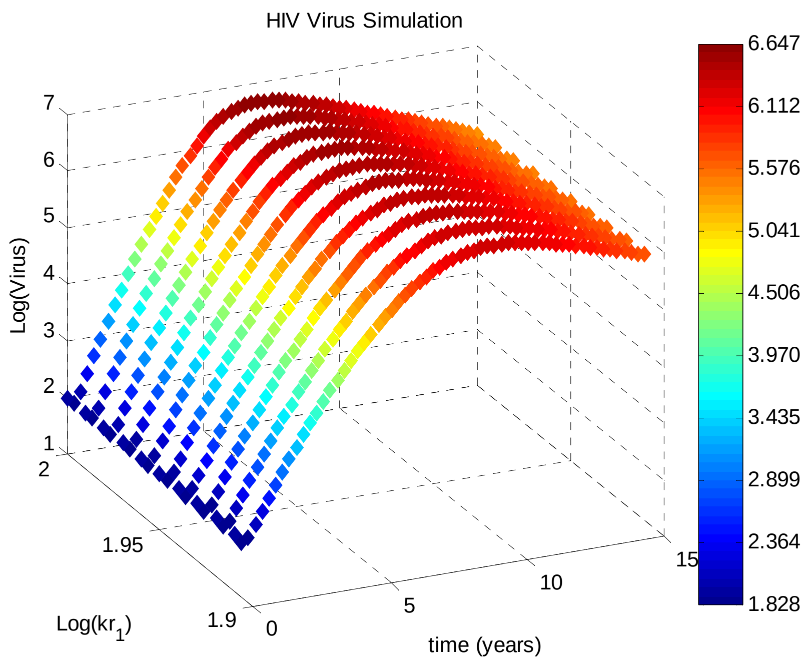


Solver Benchmarking – SBML (341)



Computational Biology

- Drug treatment and discovery – large-scale models

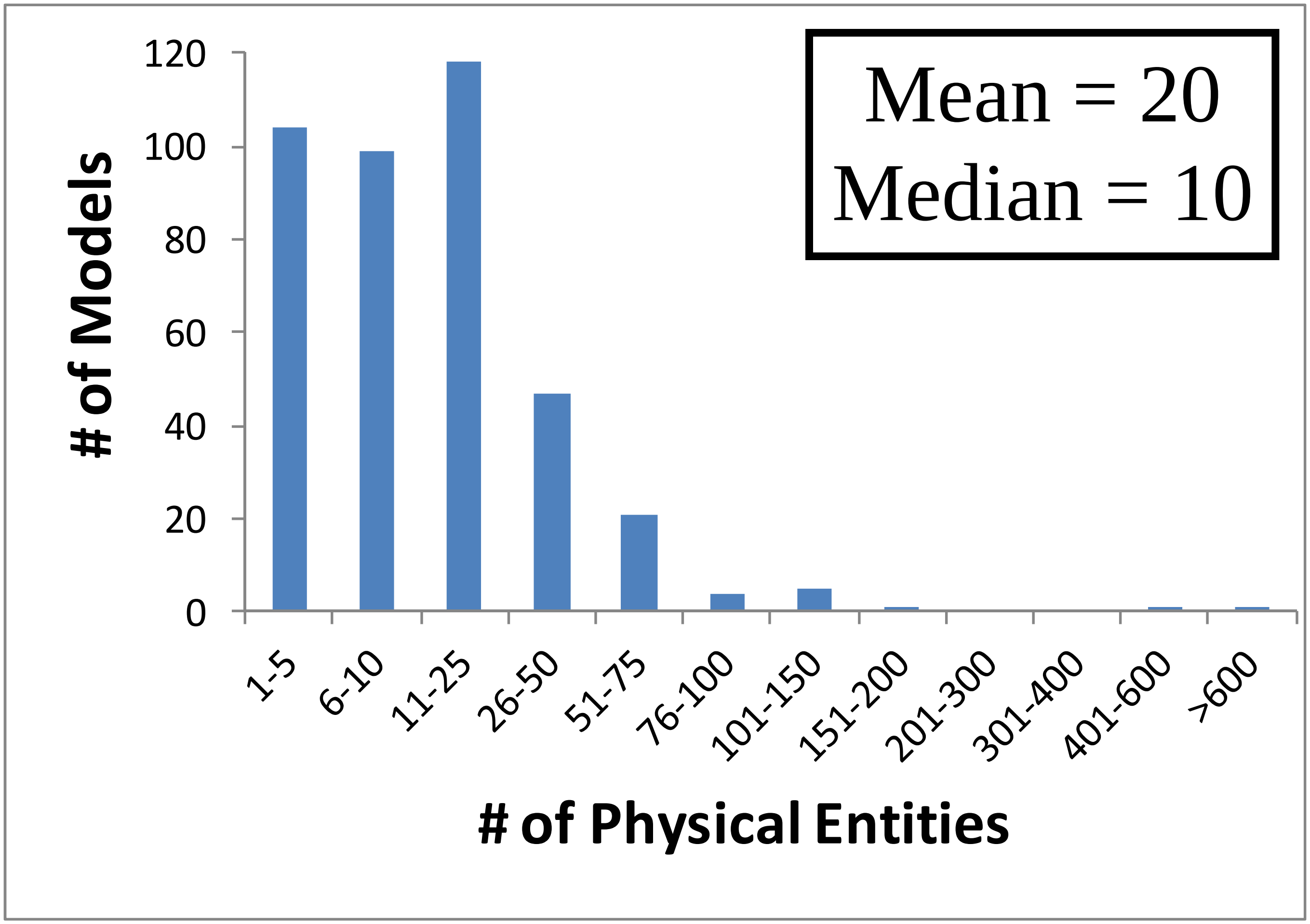



```
Model
  Parameters
    kr1 = 1e5
    kr2 = 0.1
    kr3 = 2e-7
    kr4 = 0.5
    kr5 = 5
    kr6 = 100
  End Parameters
  Variables
    H = 1e6      ! healthy cells
    V = 1e2      ! virus
    I = 0        ! infected cells
  End Variables
  Equations
    $H = kr1 - kr2*H - kr3*H*V
    $I = kr3*H*V - kr4*I
    $V = -kr3*H*V - kr5*V + kr6*I
  End Equations
End Model
```

SBML.org The Systems Biology Markup Language

VERTEX

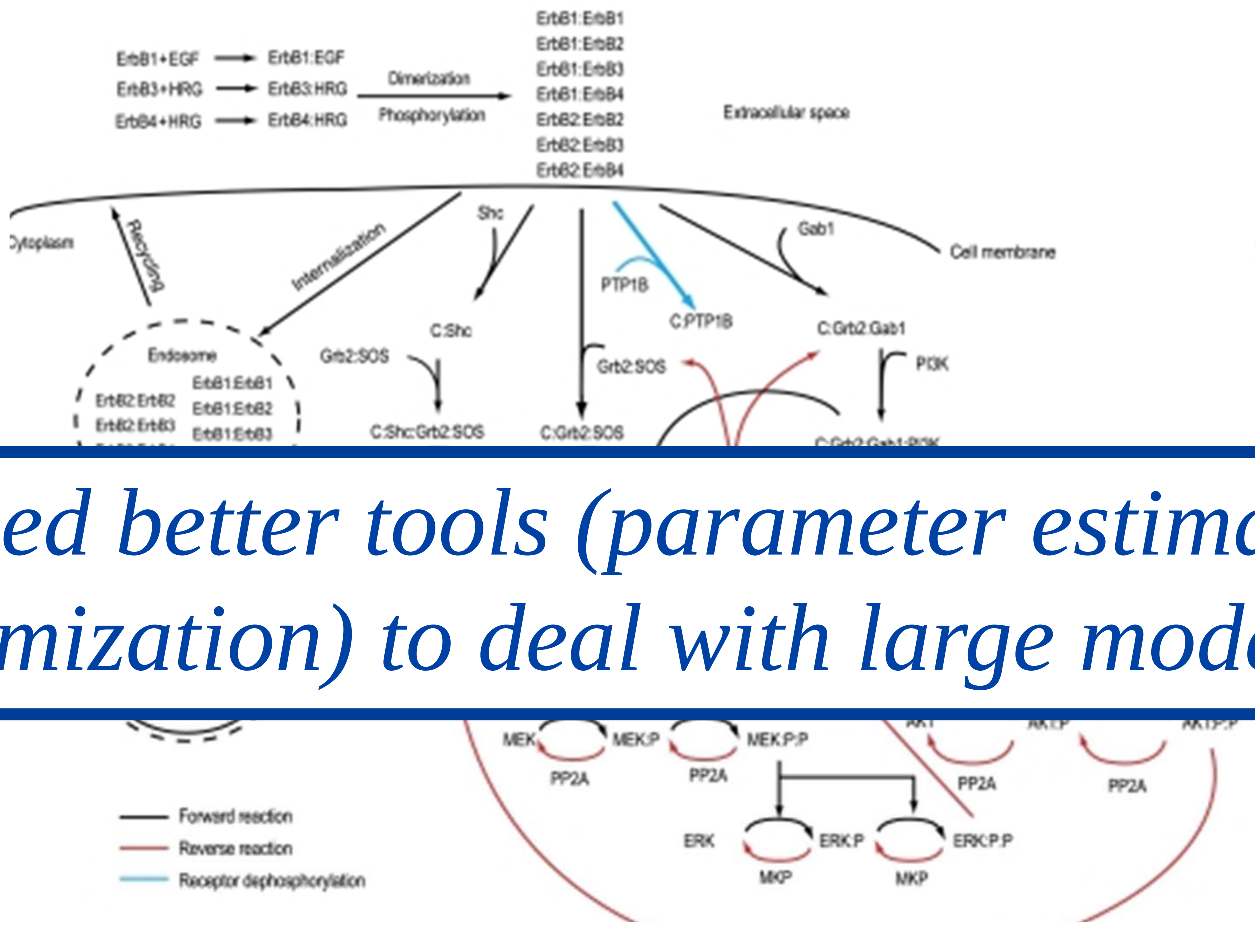
Biological Kinetic Models Modestly Sized



SBML.org The Systems Biology Markup Language

Model sizes from 409 curated models in the Biomodels repository (http://www.ebi.ac.uk/biomodels-main/)

Model Size Limited by Tools

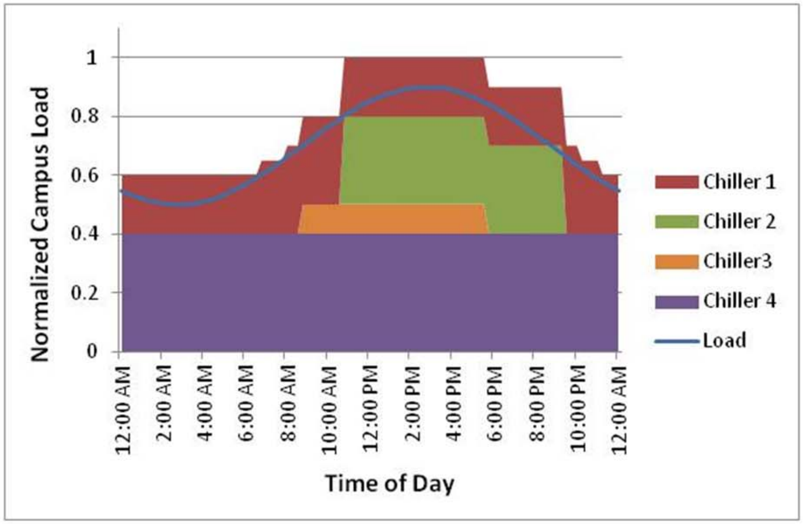


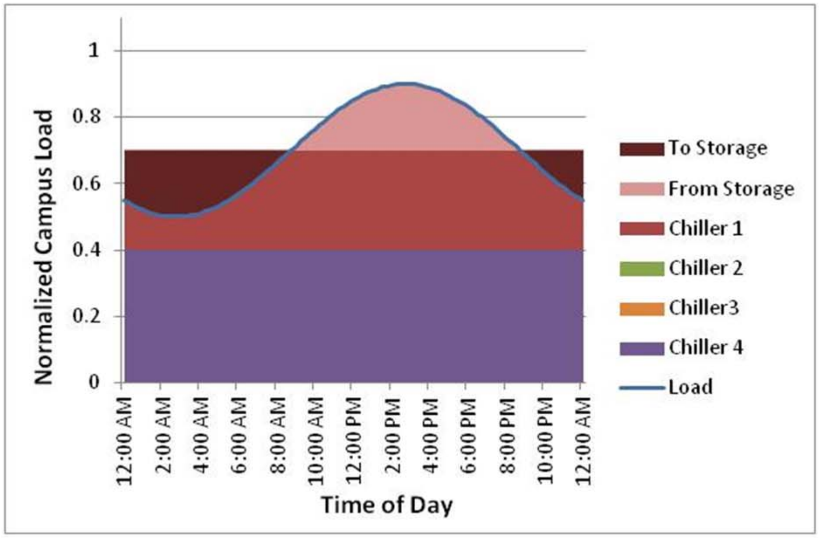
We need better tools (parameter estimation, optimization) to deal with large models!

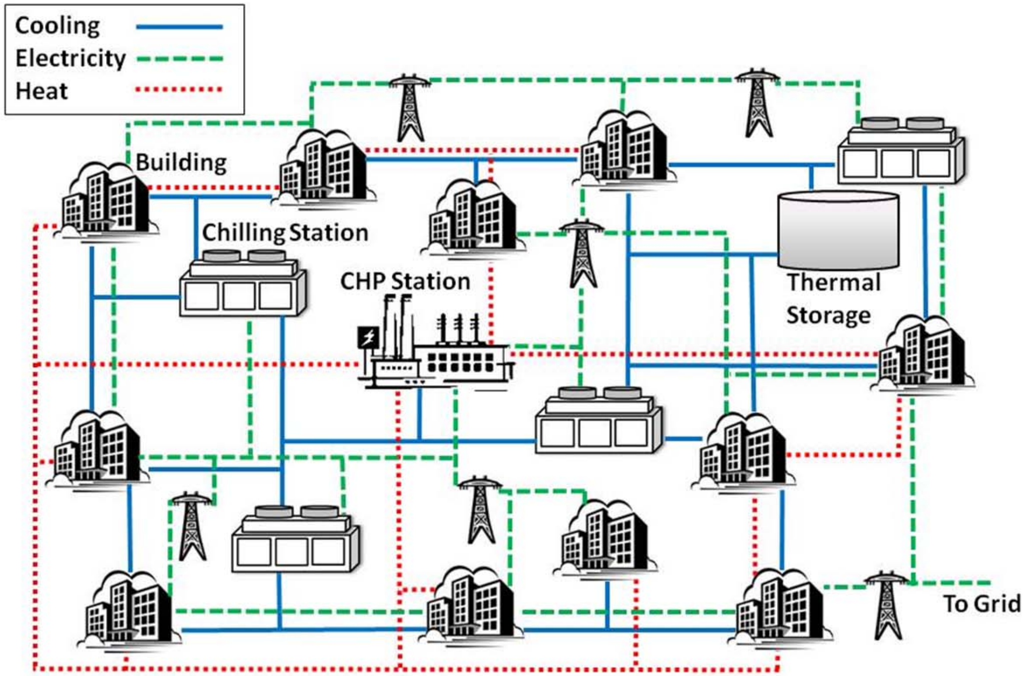
- Large ErbB signalling model (~504 physical entities)*
- Parameter estimation (simulated annealing) took “24 hours on a 100-node cluster computer”

*Chen et al. Mol Syst Biol. 2009;5:239 .

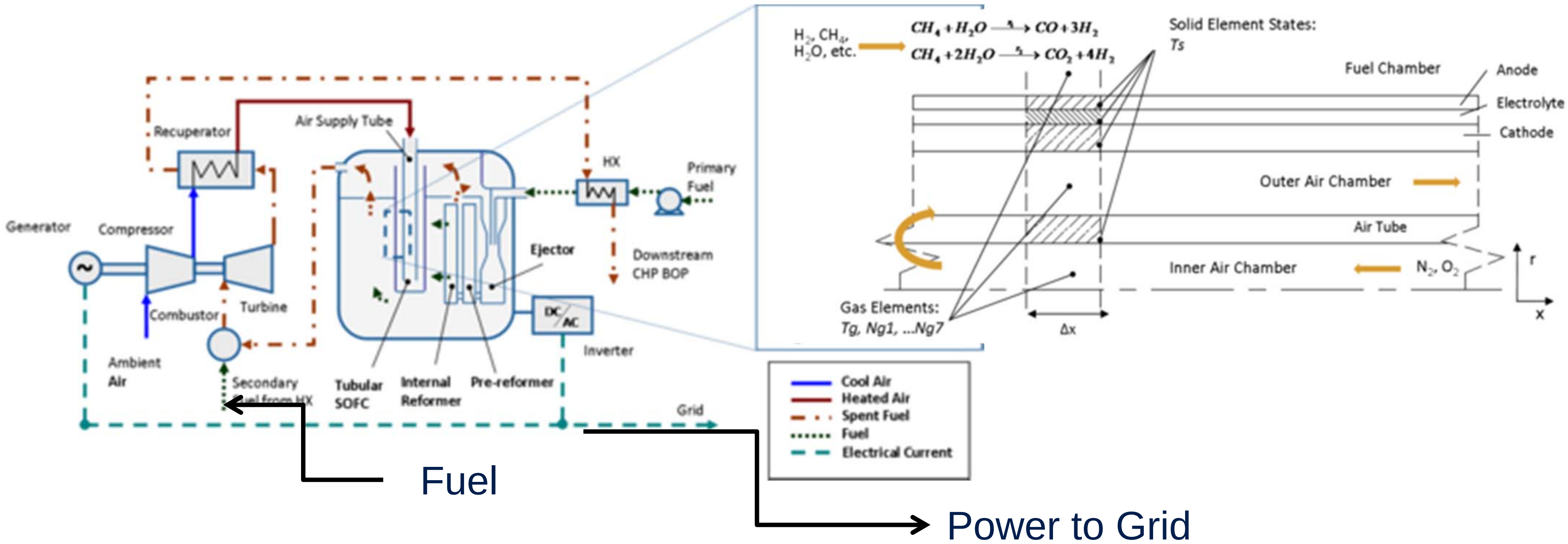
Smart Grid Energy Systems

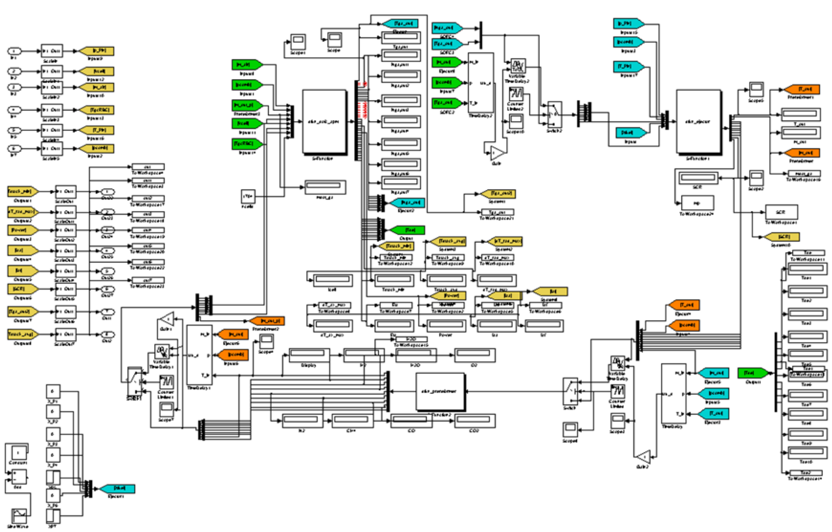






Solid Oxide Fuel Cells





Pacific Northwest NATIONAL LABORATORY

Bloomenergy

Flow Assurance for Oil and Gas Industry

- Fouling and Plugging largest loss category
  - Billions $$$ per year in lost revenue
- Predictive Analytics
  - Real-time or Off-line Monitoring Solution
  - Empirical and First Principles Models

+

=

Safe Operations
Reliability Targets
Regulatory Reports
Maximize Economics
Training Simulators

Engineering in Remote Locations

Pressure Overview

All Pressures (psig) last 24 hrs

All Pressures (psig) last month

Temperature Overview

All Temperatures (°F) last 24 hrs

All Temperatures (°F) last month

FLMT #2 (18 miles)

T °F

Temperature (°F) last 24 hrs

Temperature (°F) last month

P psig

Pressure (psig) last 24 hrs

Pressure (psig) last month

FLMT #1 (36 miles)

T °F

Temperature (°F) last 24 hrs

Temperature (°F) last month

P psig

Pressure (psig) last 24 hrs

Pressure (psig) last month

FLET (57 miles)

T °F

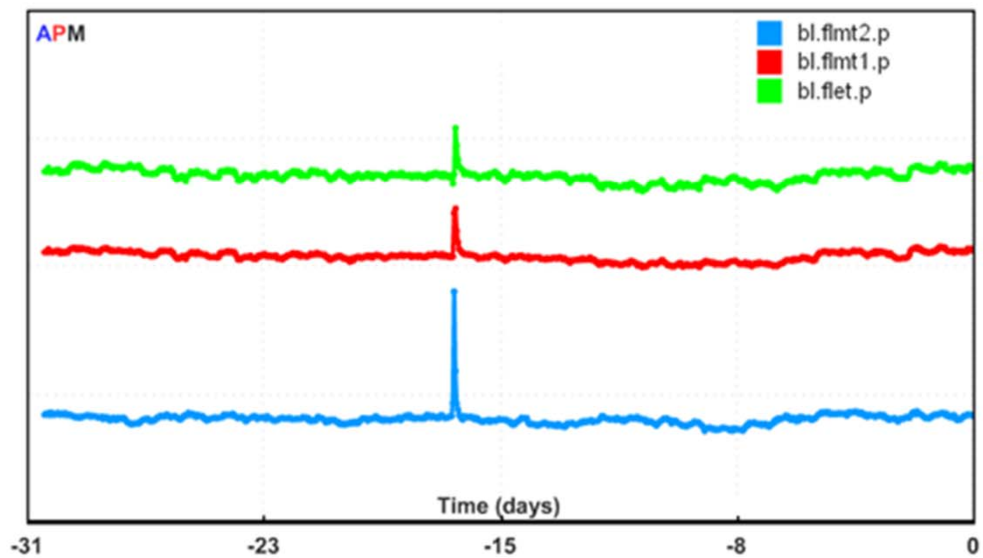
Temperature (°F) last 24 hrs

Temperature (°F) last month

P psig

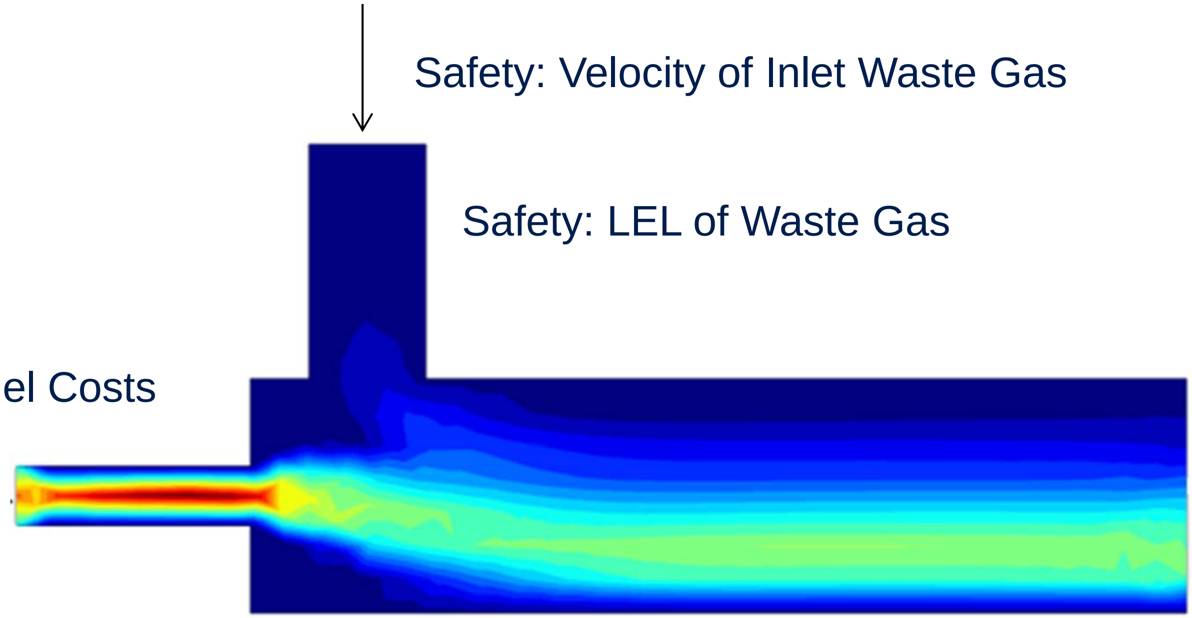
Pressure (psig) last 24 hrs

Pressure (psig) last month

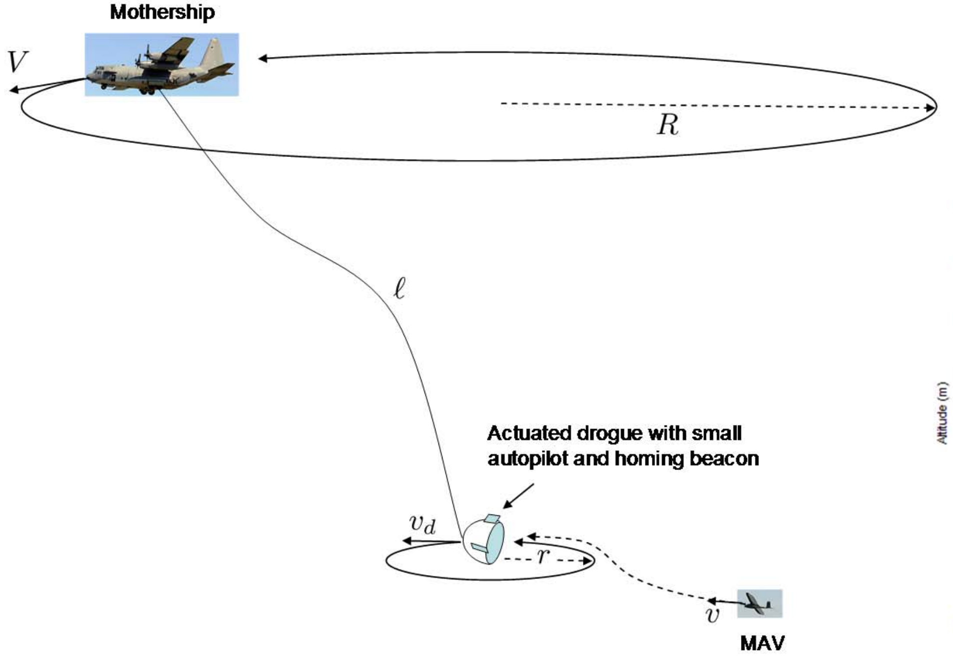


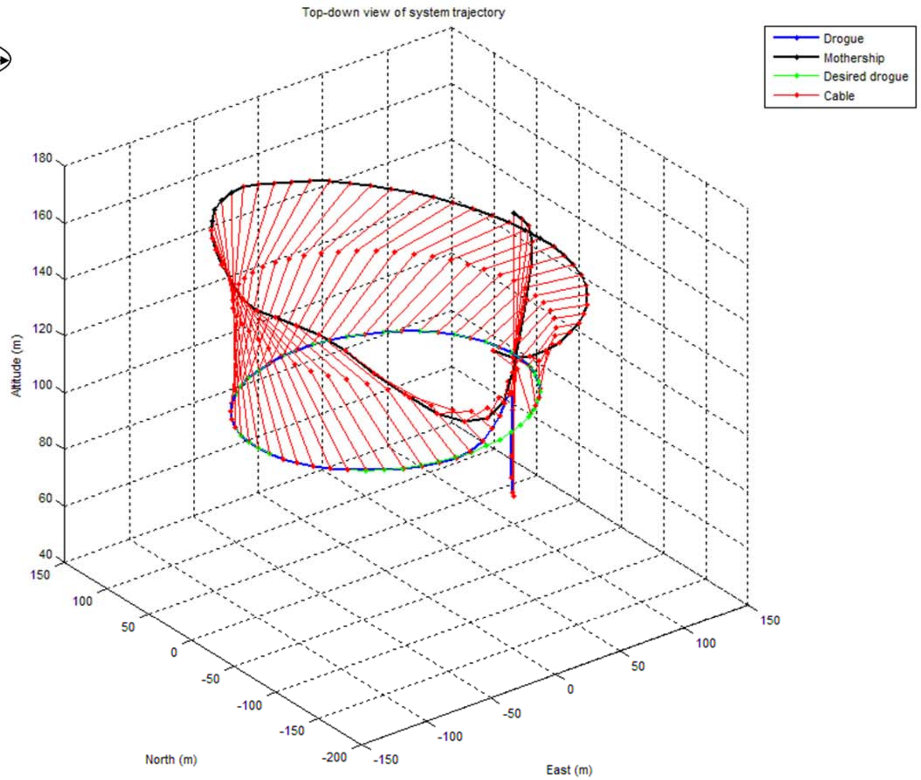
Environmental Impact

- Safe, environmentally friendly, and economic operations



Unmanned Aerial Systems





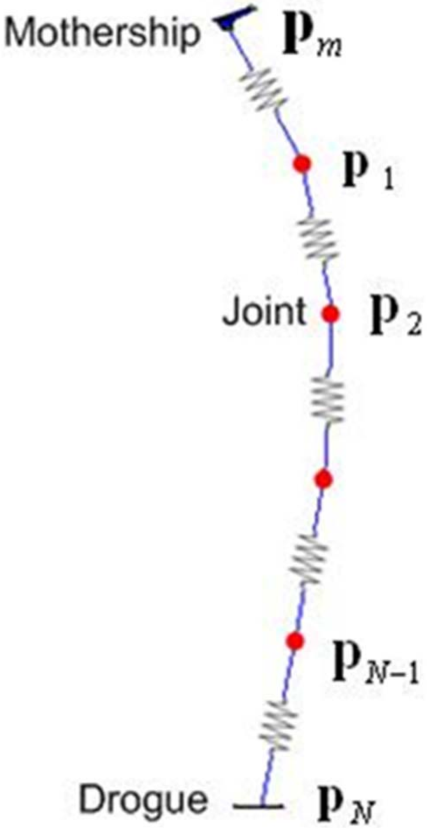
UAS System Dynamics

- Cable-drogue dynamics using Newton 2nd law

$$
\begin{aligned}
m_N \ddot{\mathbf{p}}_N &= \mathbf{T}_N + \Omega_N \\
\Omega_N &= \mathbf{G}_N + \mathbf{D}_N + \mathbf{L}_N,
\end{aligned}
$$

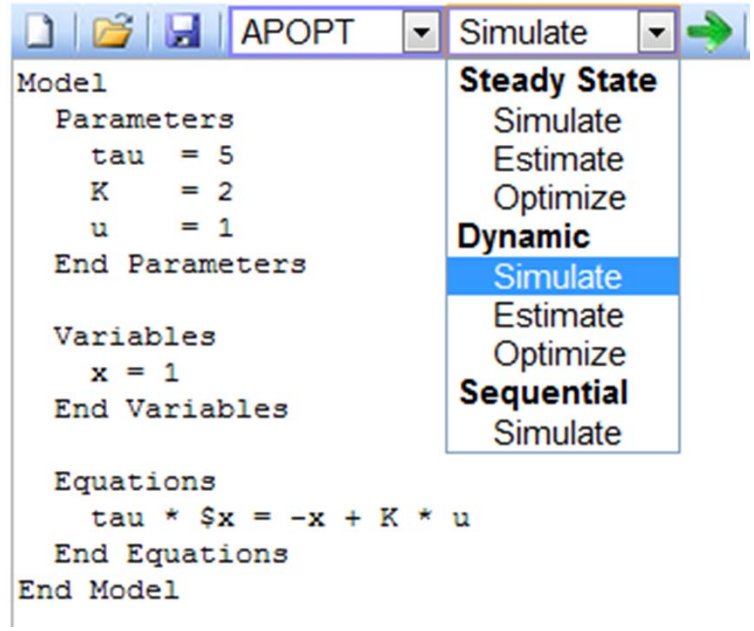
$$
\begin{aligned}
m_{j-1} \ddot{\mathbf{p}}_{j-1} &= \mathbf{T}_{j-1} + \Omega_{j-1} - \mathbf{T}_j \\
\Omega_{j-1} &= \mathbf{G}_{j-1} + \mathbf{D}_{j-1} + \mathbf{L}_{j-1} \\
j &= 2, 3, \cdots, N,
\end{aligned}
$$

$$
\begin{aligned}
\mathbf{T}_j &= \frac{EA}{\ell_0} \left(\|\mathbf{p}_{j-1} - \mathbf{p}_j\| - \ell_0 \right) \frac{\mathbf{p}_{j-1} - \mathbf{p}_j}{\|\mathbf{p}_{j-1} - \mathbf{p}_j\|}, \\
j &= 1, 2, \cdots, N,
\end{aligned}
$$



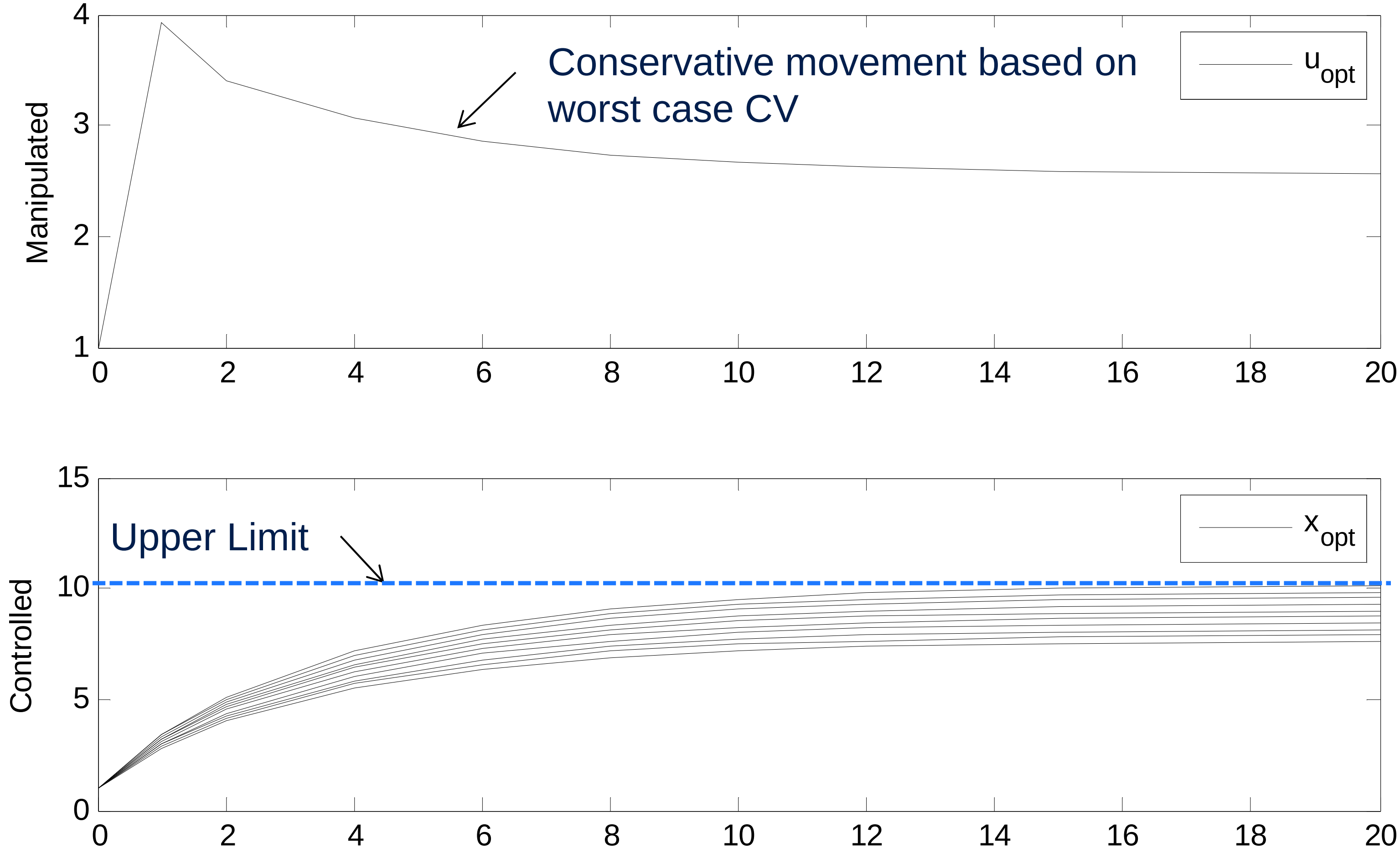
Dynamic System Example

```
Model
  Parameters
    ! time constant
    tau  = 5
    ! gain
    K    = 2
    ! manipulated variable
    u    = 1
  End Parameters
  Variables
    ! output or controlled variable
    x = 1
  End Variables
  Equations
    ! first order differential equation
    tau * $x = -x + K * u
  End Equations
End Model
```



Name	Lower	Value	Upper
p(1).n(1).tau	---	5.0000E+00	---
p(1).n(1).k	---	2.0000E+00	---
p(1).n(1).u	---	1.0000E+00	---
p(1).n(1).x	---	1.0000E+00	---
p(1).n(2).tau	---	5.0000E+00	---
p(1).n(2).k	---	2.0000E+00	---
p(1).n(2).u	---	1.0000E+00	---
p(1).n(2).x	---	1.1667E+00	---

Optimization Under Uncertainty



Selecting a Model for Predictive Control

- Many model forms
  - Linear vs. Non-linear
  - Steady state vs. Dynamic
  - Empirical vs. First Principles
- Select the simplest model
  - Accuracy requirements
    - Steady State Gain
    - Dynamics – Time to Steady State
  - Speed requirements
    - PID < Linear MPC < Nonlinear MPC

Continuous Form (SS_c)

$$\dot{x} = Ax + Bu$$

$$y = Cx + Du$$

Discrete Form (SS_d)

$$x[k+1] = A_d x[k] + B_d u[k]$$

$$y[k] = C_d x[k] + D_d u[k]$$

Nonlinear Model

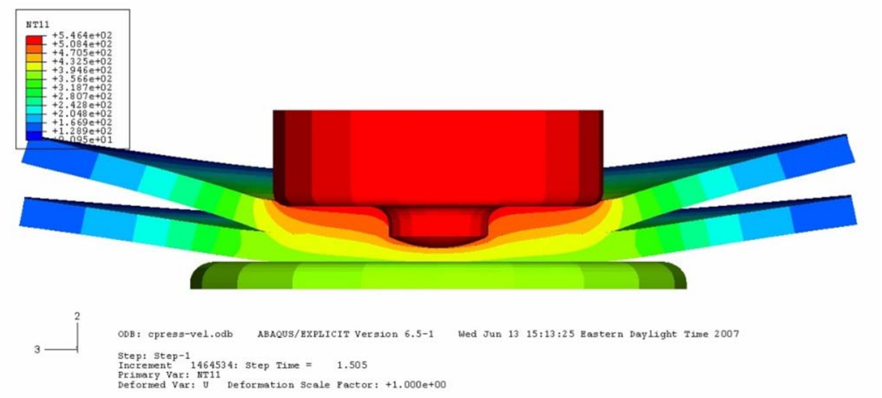
$$0 = f\left(\dot{x}, x, u, p, d\right)$$

$$0 = g(x, u, p, d)$$

$$0 \le h(x, u, p, d)$$

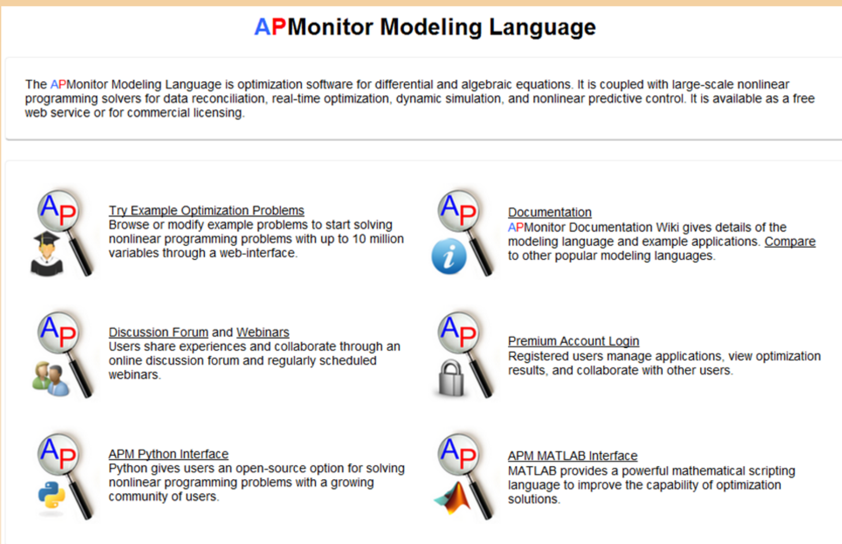
Friction Stir Welding

- A rotating tool creates heat and plasticizes the metal. This allows the metal to be “stirred” together



Getting Started with APM

Download Software at APMonitor.com



APMonitor Modeling Language

The APMonitor Modeling Language is optimization software for differential and algebraic equations. It is coupled with large-scale nonlinear programming solvers for data reconciliation, real-time optimization, dynamic simulation, and nonlinear predictive control. It is available as a free web service or for commercial licensing.

Try Example Optimization Problems
Browse or modify example problems to start solving nonlinear programming problems with up to 10 million variables through a web-interface.

Documentation
APMonitor Documentation Wiki gives details of the modeling language and example applications. Compare to other popular modeling languages.

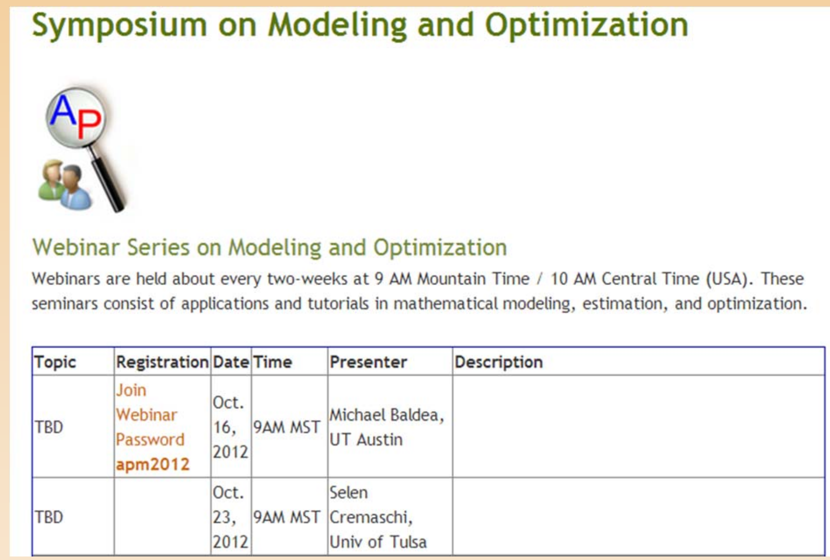
Discussion Forum and Webinars
Users share experiences and collaborate through an online discussion forum and regularly scheduled webinars.

Premium Account Login
Registered users manage applications, view optimization results, and collaborate with other users.

APM Python Interface
Python gives users an open-source option for solving nonlinear programming problems with a growing community of users.

APM MATLAB Interface
MATLAB provides a powerful mathematical scripting language to improve the capability of optimization solutions.

Bi-weekly Webinars



Symposium on Modeling and Optimization

Webinar Series on Modeling and Optimization

Webinars are held about every two-weeks at 9 AM Mountain Time / 10 AM Central Time (USA). These seminars consist of applications and tutorials in mathematical modeling, estimation, and optimization.

Topic	Registration	Date	Time	Presenter	Description
TBD	Join Webinar Password **apm2012**	Oct. 16, 2012	9AM MST	Michael Baldea, UT Austin	
TBD		Oct. 23, 2012	9AM MST	Selen Cremaschi, Univ of Tulsa	

Applications Deployed for Real-time Systems

Future Development Plans

- APM Modeling Language
  - MI-DAE systems
- Active Development Efforts
  - Mixed Integer solvers that exploit DAE structure
  - Interfaces to other scripting languages
- Industrial and Academic Collaborators
- APOPT and BPOPT MINLP solver development
  - Additional information at INFORMS session WC04