High Performance Systems Biology Tools

CScADS Summer Workshop, Lake Tahoe
July 27-30, 2009

Peter Graf, Christopher Chang, David Alber, Monte Lunacek, Ambarish Nag, Kwiseon Kim, Michael Siebert

National Renewable Energy Laboratory

Dave Biagioni, David Bortz

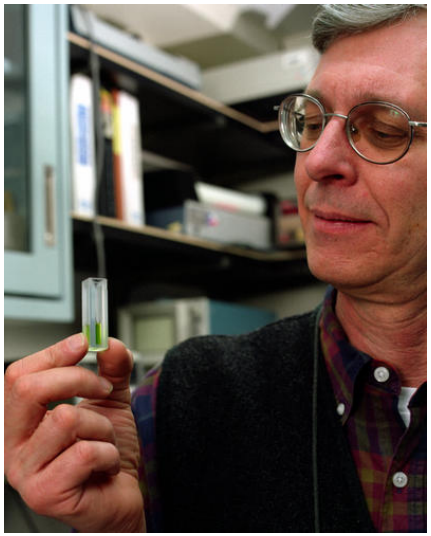
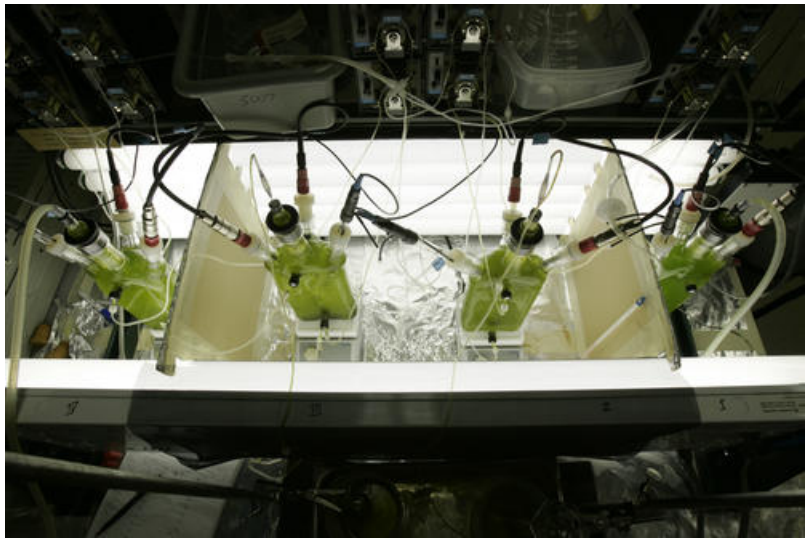
University of Colorado, Boulder

Glenn Murray

Colorado School of Mines

Outline and goals:

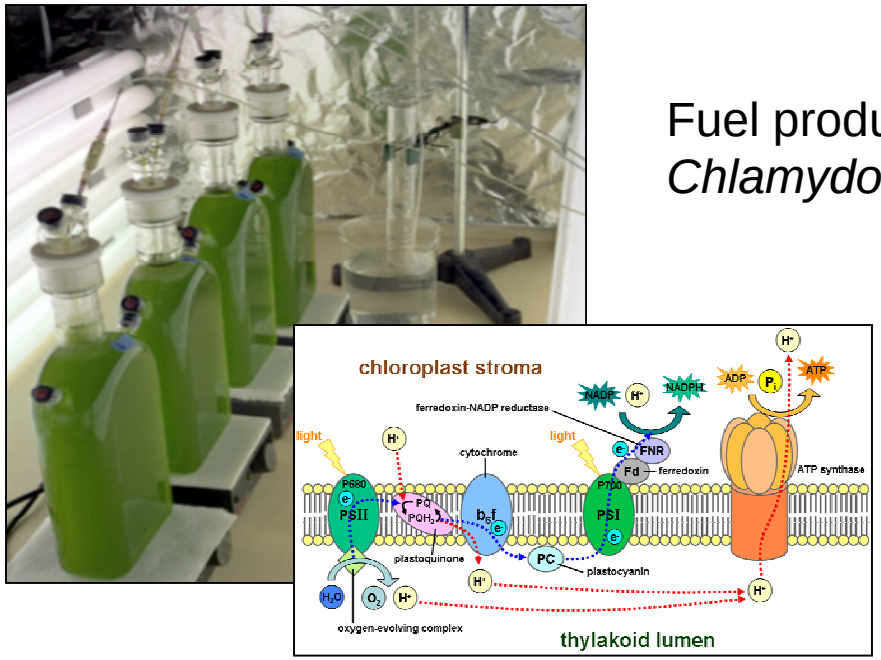
- (Brief) NREL introduction
- (Brief) Metabolic modeling/Systems Biology introduction
- (Main) Report ongoing case study--application of HPC to metabolic modeling at kinetic level--of many links in chain necessary to formulate mathematically and solve numerically real science problems
- (Brief) Simulation ~~Optimization~~ Exploration


National Renewable Energy Laboratory

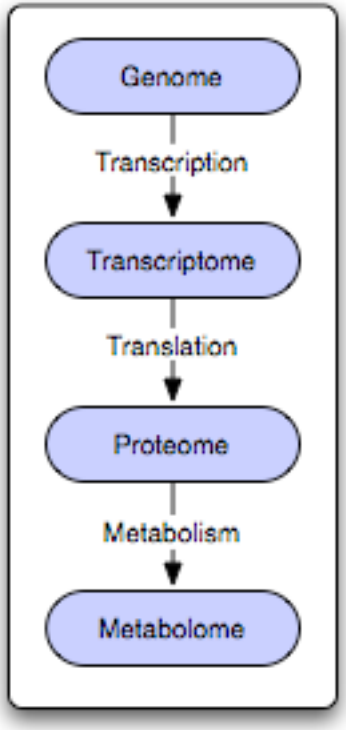
- A DOE national laboratory dedicated to renewable energy and energy efficiency R&D
- Fundamental science to technology
- Collaboration with industry and university partners
- Aspirations to market relevance

High Performance Systems Biology: Whole Cell Metabolic Modeling of Fuel Producing Algae



Fuel producing green algae
Chlamydomonas reinhardtii

Systems biology: system level understanding enabled by in-depth knowledge of the molecular nature of biological systems



Project goals:

- Build high performance tools for resolving model uncertainties in large biological models
- Fill knowledge gaps and develop full metabolic model of *C. reinhardtii*.

HPC applied to Systems Biology - existing tools

Grid Cellware - Grid based simulation and parameter estimation.
SBaddon - Extension package for Systems Biology Toolbox within MatLab, includes compiled simulation functions.
HiBi09 - Workshop planned for Fall 09, "a forum to link researchers in the areas of parallel computing and computational systems biology."
Systems Biology Workbench (SBW) - Plug in interface, model translation/simulation tools.
...

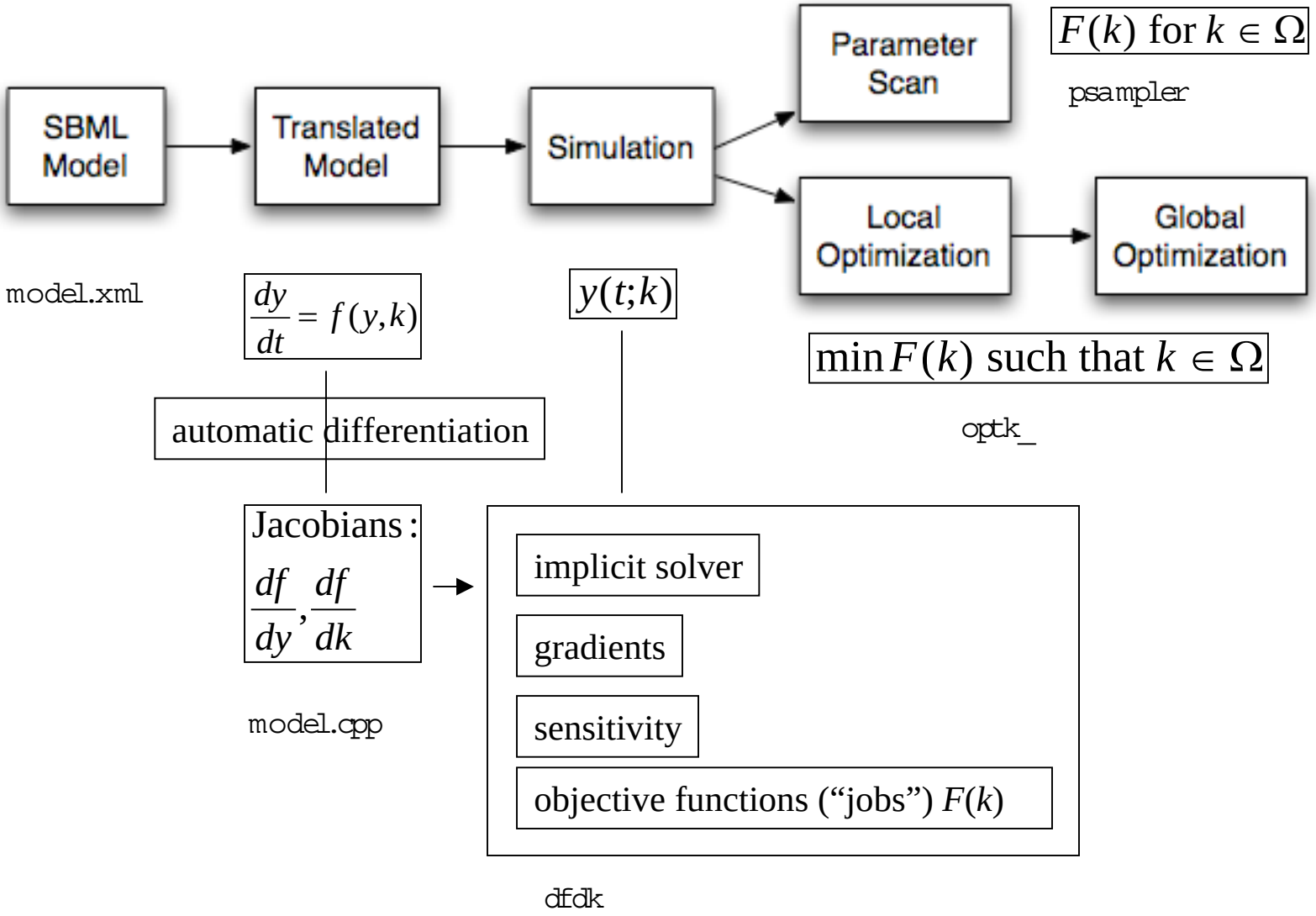
No existing truly HPC Systems Biology tools.

Our SciDAC project:

HiPerSBTK - High Performance Systems Biology Toolkit

•high performance simulation and sensitivity,
•(hierarchically) parallel optimization and parameter scanning.

Summary of steps in HPC kinetic level metabolic simulation and optimization



Model formulation in SBML

A hierarchical approach

Nothing fully known. Proceed in order:

- *C. rheinhardtii*
- Analogs with other species
  - in order of *genetic* similarity
- High performance computing

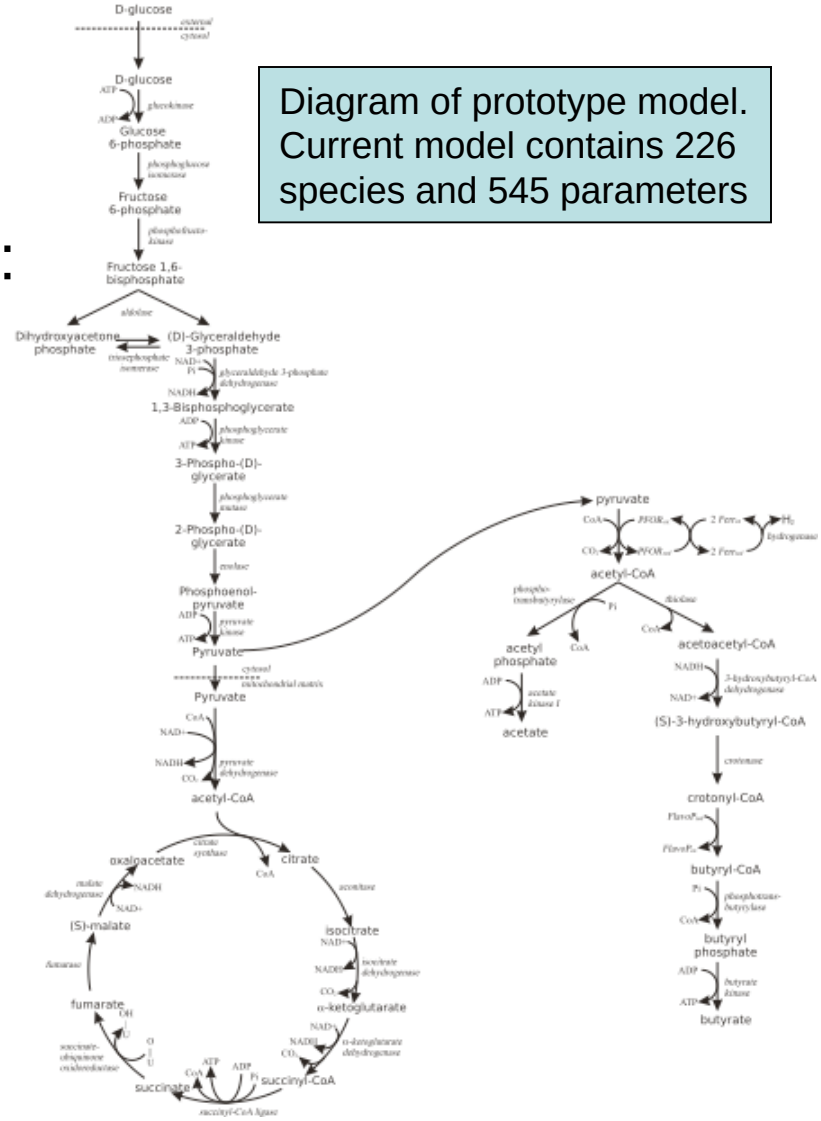


Diagram of prototype model. Current model contains 226 species and 545 parameters

Sources of uncertainty:

- form of equations
- constants
- transferability of analogs

Systems Biology Markup Language (SBML**):**

- An xml variant/schema
- Metabolism consists (roughly) of *species, parameters, reactions*

Model equations

Michaelis-Menten Kinetics:

$$\frac{d[DGP]}{dt} = \frac{K1_{kcat}\,[GK][ATP][D]}{K1_{KiaKB} + K1_{ATP}[D] + K1_{D}[ATP] + [ATP][D]} - \frac{K2_{kcat}\,[GPI][DGP]}{K2_{KM} + [DGP]}$$

$$\frac{dy}{dt} = f(y,k,E)$$

- The vector *y* contains *species concentrations*
- *k* contains *kinetic parameters*
- *E* are *enzyme levels*

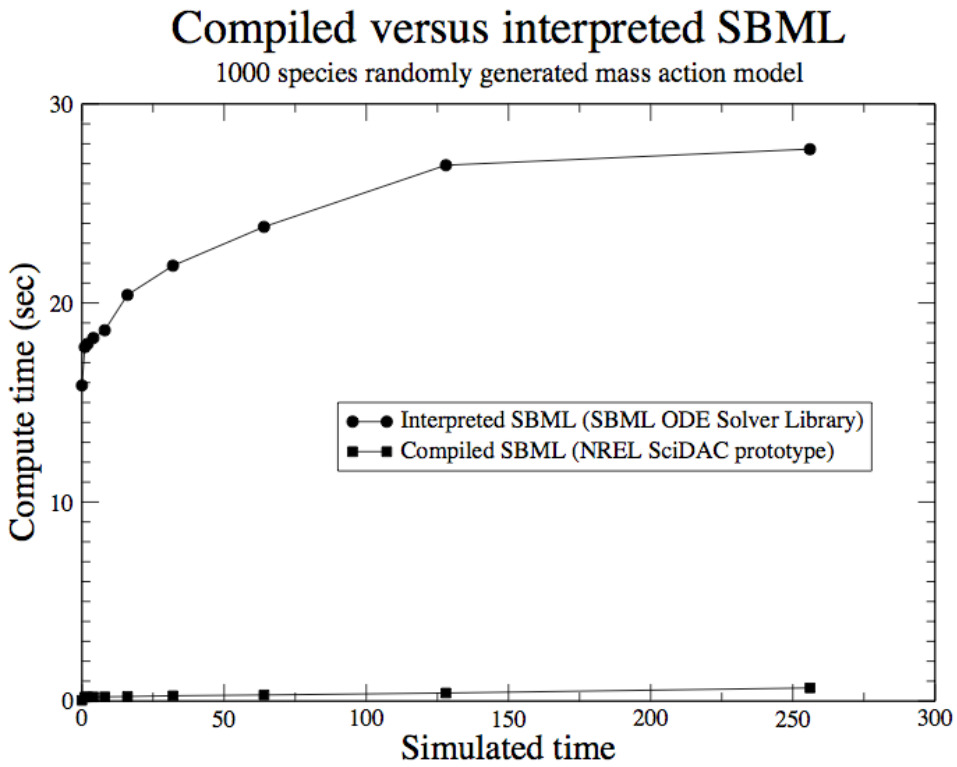
Vectors have *O*(100) elements today, *O*(1000) in future

Goals:

- Resolve uncertainties in *k*
- Optimize properties w.r.t. *E*

Model translation

The need for translating SBML to a compiled language:



`sbml2cpp`:

- `Model.cpp/h` generated from `model.xml`.

- Efficient implementation (e.g. hash tables) avoids O(species X *p*arameters X reactions) ODE construction.

- Jacobians enabled through automatic differentiation.

Simulation and Sensitivity: dfdk

•A *model-specific* library that performs core simulation and sensitivity calculations.

•Uses SUNDIALS (LLNL) suite of ODE tools, esp. CVODES

•Command line and script oriented tools, e.g. text file input:

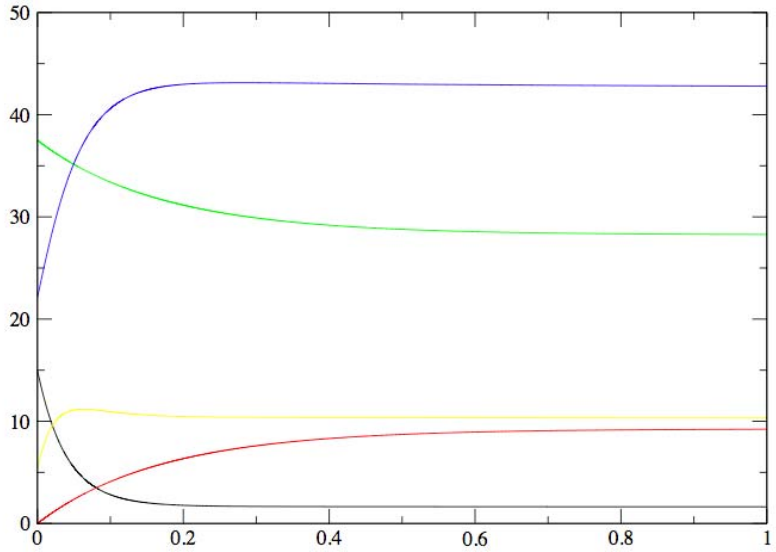
```
# defines objective/what to calculate:
job=s
# time to which we simulate:
time=0.001
# species we care about this run: acetate, butyrate, hydrogen
activey_ids = cpd_C00033 cpd_C00246 cpd_C00282
# species target values
activey_expt_vals = 2 10 30
# parameters to optimize
activek_ids = R01196_kcat R00230_kcat R01061_kcat Rpyr_cyto2mito_V R00238_kcat R01512_kcat R00200_kcat R01196_KA
# bounds on selected params (for optimizer)
R01196_kcat_bounds = 0.000000 1000.000000
R00230_kcat_bounds = 0.000000 1000.000000
R01061_kcat_bounds = 0.000000 1000.000000
Rpyr_cyto2mito_V_bounds = 0.00000 1000.000000
R00238_kcat_bounds = 0.000000 1000.000000
```

•Basic tasks enabled

•Provides foundation for HPC model characterization/optimization

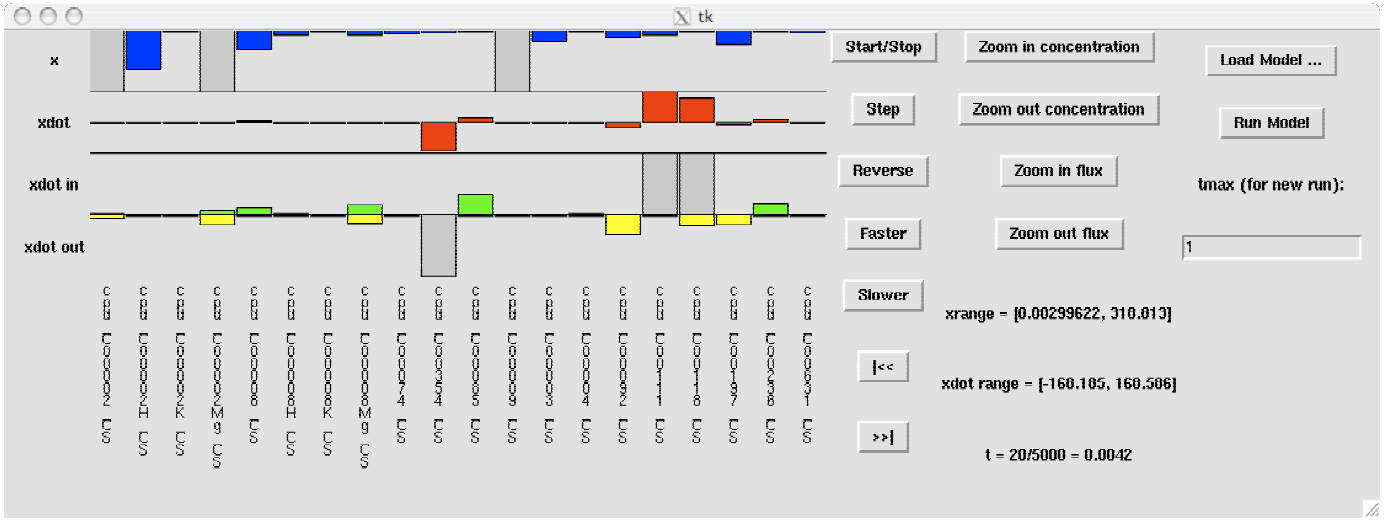
dfdk Examples

```
./dfdk --time=1 --job=f --details=1 | grep "^x:" | awk '{for (i=2; i<=NF; i++) printf "%.12f ", $i; printf "\n"}' |
xmgrace -nxy -
```




```
./dfdk --job=y; python extract_minimal.py dydk.out
effects of params, in order:
[111, 'R01061CS_KiP', 6903571412765018.0]
[107, 'R01061CS_KiC1', 65996370854907.25]
[83, 'R00756CS_p', 143.12400446157102]
[142, 'ec_4_2_1_11_CS', 5.8202960624527815]
...
effects on species, in order:
[11, 'cpd_C00009_CS', 3039405874122647.0]
[12, 'cpd_C00003_CS', 3039405874122647.0]
[13, 'cpd_C00004_CS', 3039344871365413.5]
[17, 'cpd_C00197_CS', 2532305704157037.0]
...
```

```
python flux_viewer.py
```



Model optimization / parameter estimation

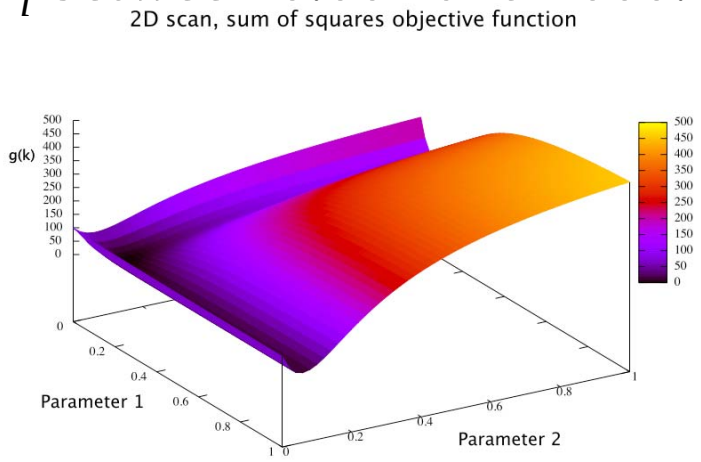
Criteria for determining k: e.g. fit to experimental data

$$g(k) = \sum_{i=1}^{N} | y_i - y_i^{\exp t} |^2$$

With this (or any similar) criterion in hand, **our model building has been formulated as the global optimization problem**

$$\min g(k) \qquad \text{for} \quad k_i^{\min} \le k_i \le k_i^{\max}$$

The bounds on k are roughly six orders of magnitude, e.g. k_i between 0.001 and 1000.



A hard problem:

- narrow troughs and flat plateaus
- multiple local minima
- multiple “funnels”
- unknown global structure

Gradients

Observation: We can calculate the gradient of least squares objective $g(k)$:

$$\frac{\partial g}{\partial k_j} = 2\sum_{i=1}^{N}(y_i - y_i^{\exp t})\frac{\partial y_i}{\partial k_j}$$

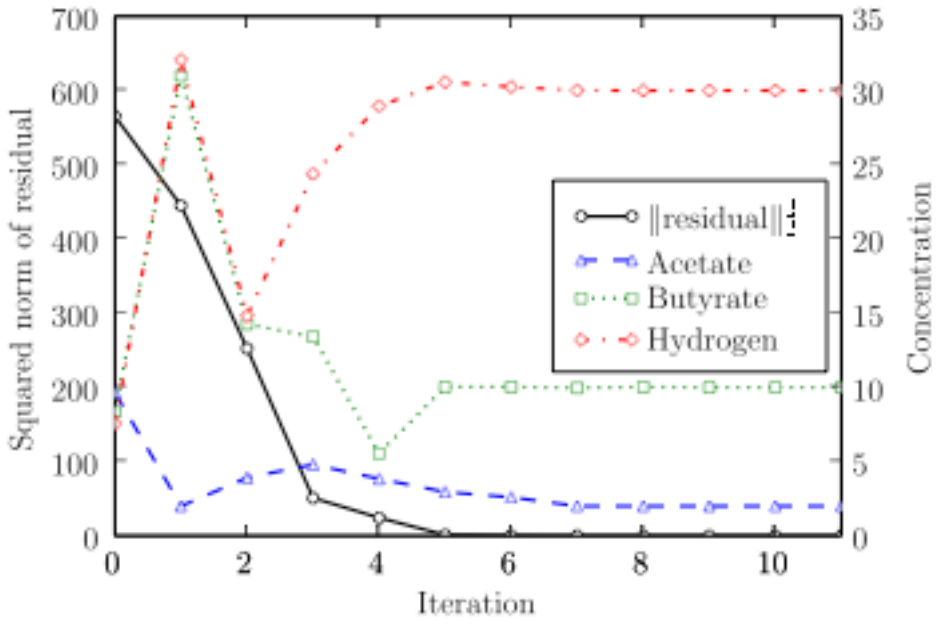
The quantity $s = \frac{\partial y}{\partial k}$ is the *sensitivity:*

$$\frac{ds}{dt} = \frac{d}{dt}\frac{dy}{dk} = \frac{d}{dk}\frac{dy}{dt} = \frac{d}{dk}f(y,k) = \frac{df}{dy}\frac{dy}{dk} + \frac{df}{dk} = \frac{df}{dy}s + \frac{df}{dk}$$

In fact, we never explicitly compute it, but instead use *adjoint sensitivity analysis* to directly compute ∇g.
This functionality is built into the CVODES.

Local Optimization

Gradients enable efficient local search. We use TAO's BLMVM method.



Issues:

- constraints
- extreme sensitivity of gradients
- scaling-large range of spatial and temporal scales
- rootfinding / quasi-steady states
- underdetermination

Global Optimization - Parallel Scatter Search

Local search is not enough. We need *global* search.

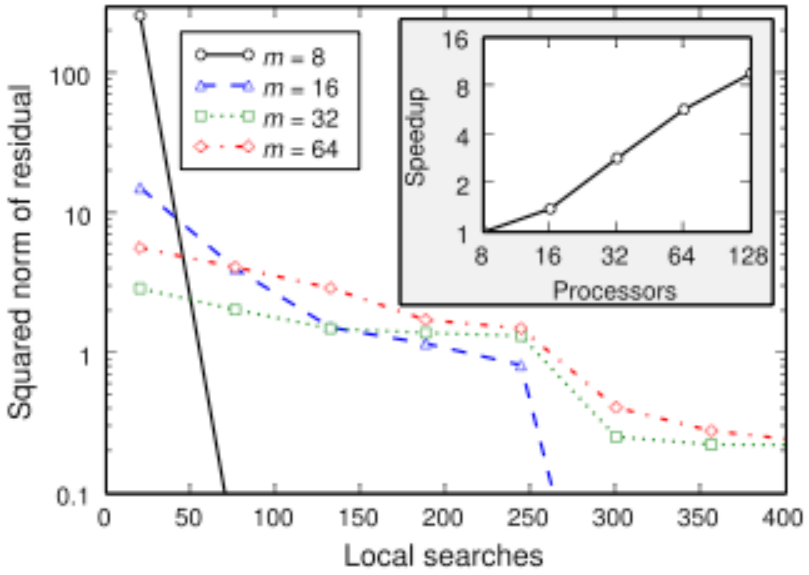
Scatter Search:

•Population based.

•Diversity systematically maintained.

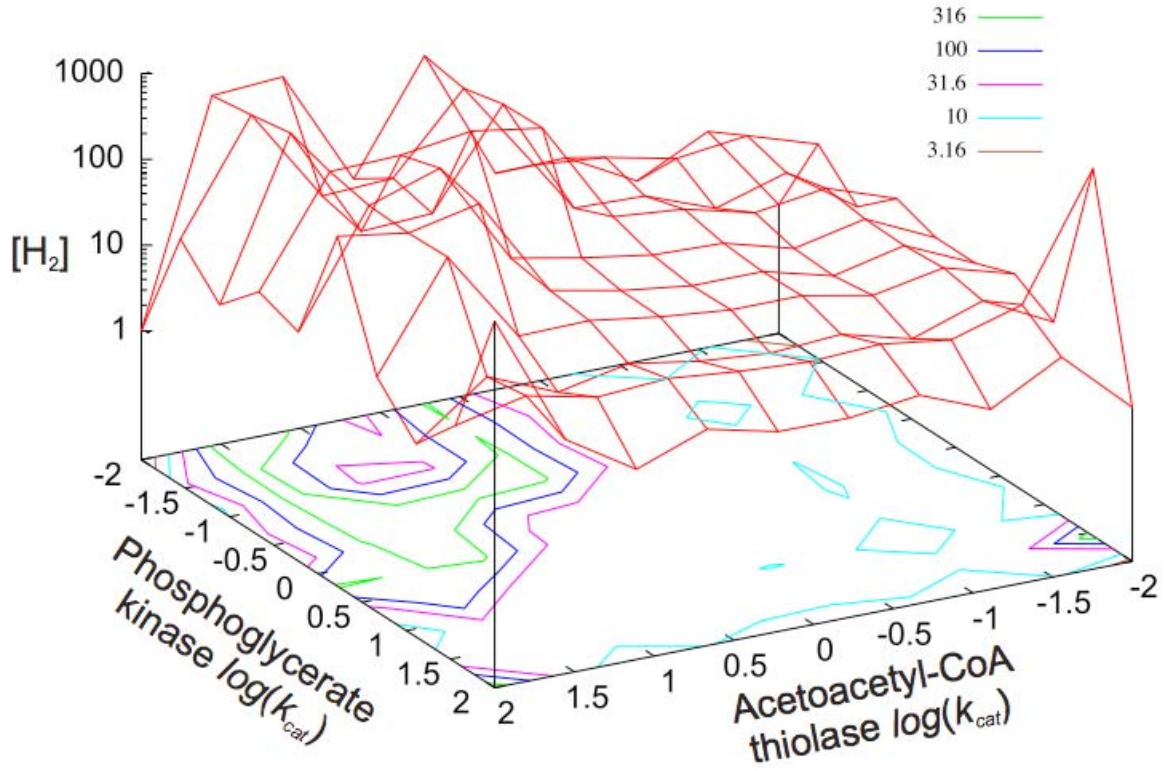
•Generation of new solutions by interpolation and extrapolation between old ones.

•Includes local search phase--we utilize our existing TAO-based local optimizer



Initial results: This method solves 64 variable problems with 4 orders of magnitude bounds in a matter of minutes, and scales well.

Mixing Tasks / Workflow



Mixing sampling and fitting. Test model of *C. reinhardtii*.
Scan a 2D grid of values of acetoacetyl-CoA thiolase k_{cat} and phosphoglycerate kinase k_{cat}: for each value,
Fit the rest of the kinetic parameters to synthetic experimental values of acetate and butyrate
Plot the resulting hydrogen flux. x, y units are log (base 10) of parameter values. z-axis (hydrogen flux) units are relative.

The App/Job matrix

Formulation of concepts of interest so far utilizes the following:

•Species concentrations $y(t)$

•Sensitivity dy/dk and its squared Frobenius norm $||dy/dk||_F^2$

•Fit to experimental data $|y^c - y^e|^2$

Tasks we might want to do include:

•Simulate the network for any one of the above properties (or, later, functions of them).

•Scan the behavior of the network over a range of kinetic parameters ${k}$.

•Optimize the network with respect to k, where the objective function involves the above

•quantities.

Thus the following matrix of functionality:

app \ job	fwd	sens	fit
simulate (dfdk)	$y(t; k, y_0, E)$	dy/dk	$\|y^c - y^e\|^2$
scan (psampler)	+	-	-
optimize (optk_tao, optk_ss)	-	+	+

+ = existing functionality, - = in development

In addition, we will implement *combinations* of these basic tasks, including: treating some of them as constraints, and some of them as objectives; nested combinations of scanning and optimization, over different parameter subsets.

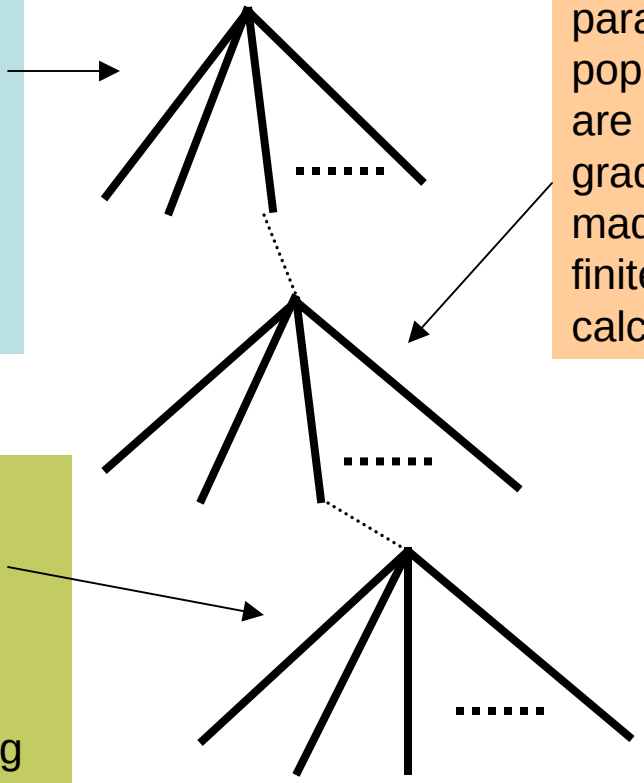
Utilizing petascale resources: hierarchical parallelism

parallel (multi) starts

Both global and local, deterministic and stochastic, optimizers benefit from multiple starts from different initial conditions.

parallel optimizer

parameter scans and population based algorithms are “embarrassingly parallel;” gradient based optimizers are made parallel through parallel finite difference derivative calculations



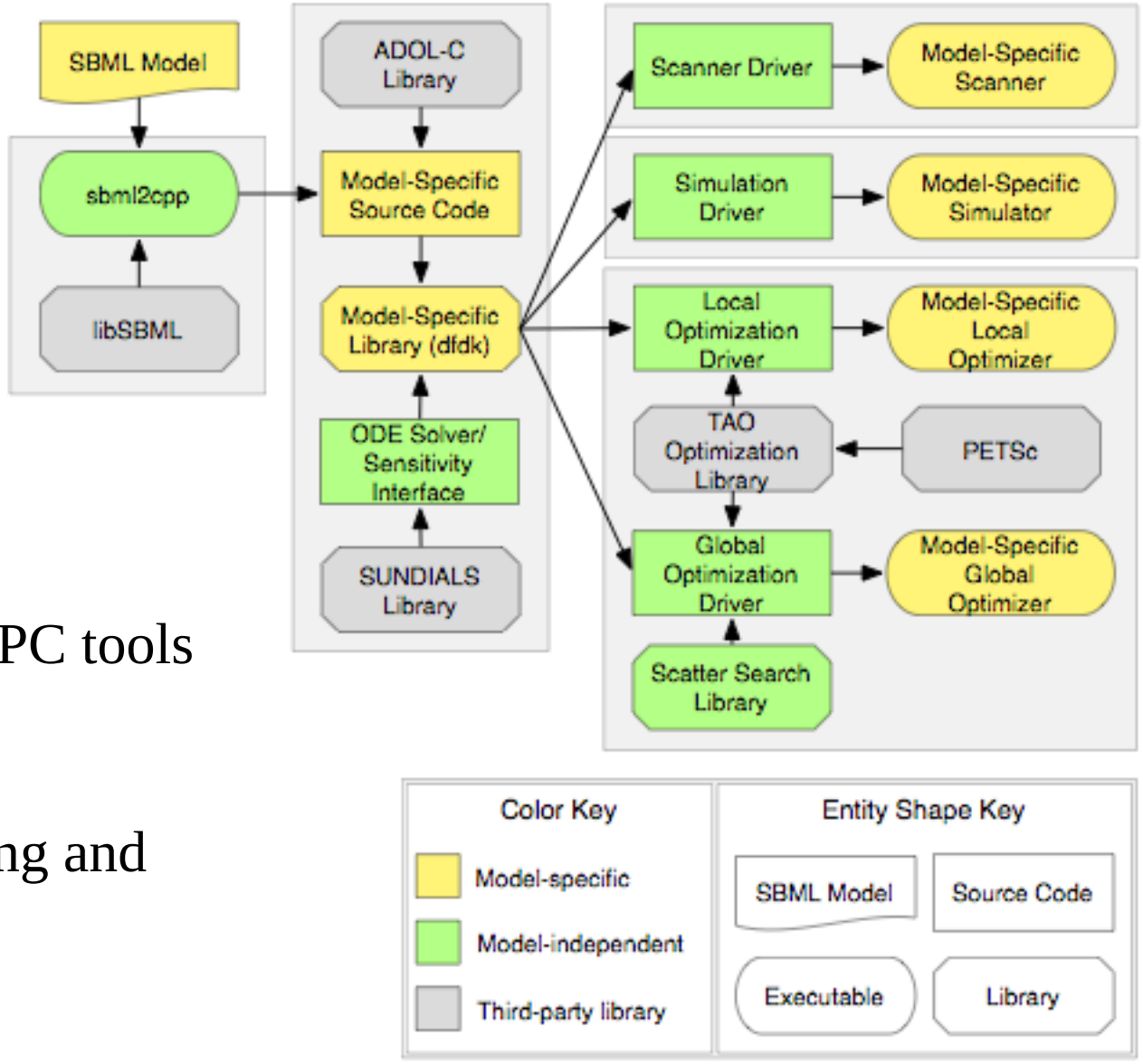
parallel ODE solver

Parallelization achieved by parallelization of the linear solver, accomplished by parallelization of the underlying vectors and matrices

example,
(work in progress):
for k in K1:
  fit k in K2
1000 X 100 = 10^5 processors

100-way (mostly *embarrassing parallel*) parallelism at each level is possible, could potentially utilize more than 1,000,000 processors (?)

Summary of the HiPerSBTK



- high performance
- utilizing existing HPC tools (SUNDIALS, etc.)
- building blocks
- prototypical scanning and optimization tasks

Simulation *Exploration*

We think we are doing *simulation optimization.* In fact, users want *simulation exploration*.

•Unknown objective functions.
•Unclear parameter spaces.
•More than just *optimum* is scientifically interesting.
•Experimentalists, theorists, and mathematicians formulate problems w.r.t. different variables.

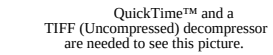
Examples
•Systems Biology
"Scan some parameters, fit w.r.t some others, apply selective constraints, then optimize (or maybe just report), certain values."

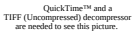
•Inverse Material Design
"What is the PDF of an alloy configuration space?"
"What is the best $A_xB_yC_{1-x-y}$ alloy?"

Simulation exploration in Inverse Material Design

"What is the PDF of an alloy configuration space?"

"What is the best $A_xB_yC_{1-x-y}$ alloy?"





(Compare to Systems Biology case)

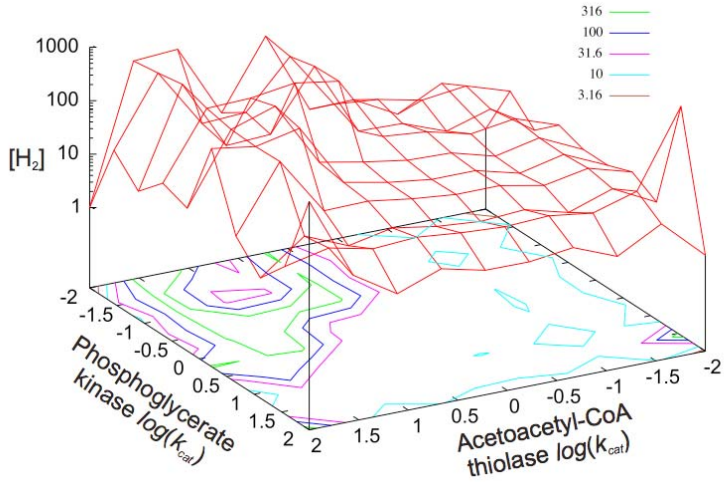
Summary

•NREL is the hub of renewable energy research in the United States.
•Large scale metabolic modeling using HPC is possible.
•Scanning and optimization approach to quantifying and resolving uncertainty.
•Evolution in thinking toward HPC exploration/characterization tools, with optimization as an important sub-component

Outlook

•Testing of 200 species 500 parameter model.
•Use of data versus “physics-based” modeling.
•Rootfinding, explicit calculation of (quasi-)steady states
•More formal treatment of uncertainty
•Allow for logarithmic treatment of all kinetic parameters.
•**Parallel I/O**. Choose output data format and use it (e.g. parallel HDF5 (?)).
•Implement all possible app/job combinations.
•Constrained optimization. Required to support, for example, "Optimize hydrogen subject to butyrate and acetate fitting experimental values.”
•Take flux seriously. Take distinction between "internal" and "boundary" species seriously. With constrained optimization, required to support, for example, "Optimize hydrogen subject to flux of pyruvate not exceeding *X*.”
•**Plug in approach: Combine simulation, scanning, fitting, optimizing [apps], different objective functions [jobs], treatment of them as objectives or constraints, all in user configurable *and* scalably parallel way.**
•Find some users and do what they suggest.

Problems of the Week:



Problem 1:
For $k_1 \in K_1$
Solve:
min $F(k_2;k_1)$ for $k_2 \in K_2$
Report $g(k_2^*;k_1)$

e.g. F = fit to experiment, g = particular chemical species of interest

Problem 2:
For $k_1 \in K_1$
Solve:
max $g(k_2;k_1)$ s.t. $F(k_2;k_1) = 0$, $k_2 \in K_2$

e.g. Explore (scan, optimize, ...) parameter space K_1 (e.g. uncertain model parameters), for each point maximizing flux of chemical fuel subject to constraint that model fit experimental data, over parameter space K_2 (e.g. enzymes).

Goal:

Modular hierarchically* parallel simulation *exploration