Cornell University

PSERC

*Project 21: "Development and Testing of New Tools"*

# MATPOWER Optimal Scheduling Tool (MOST) and Future Directions for MATPOWER

Ray Zimmerman, Carlos Murillo-Sánchez,
Haeyong (David) Shin, Tim Mount, Bob Thomas

CERTS Review, Cornell University
June 9-10, 2016

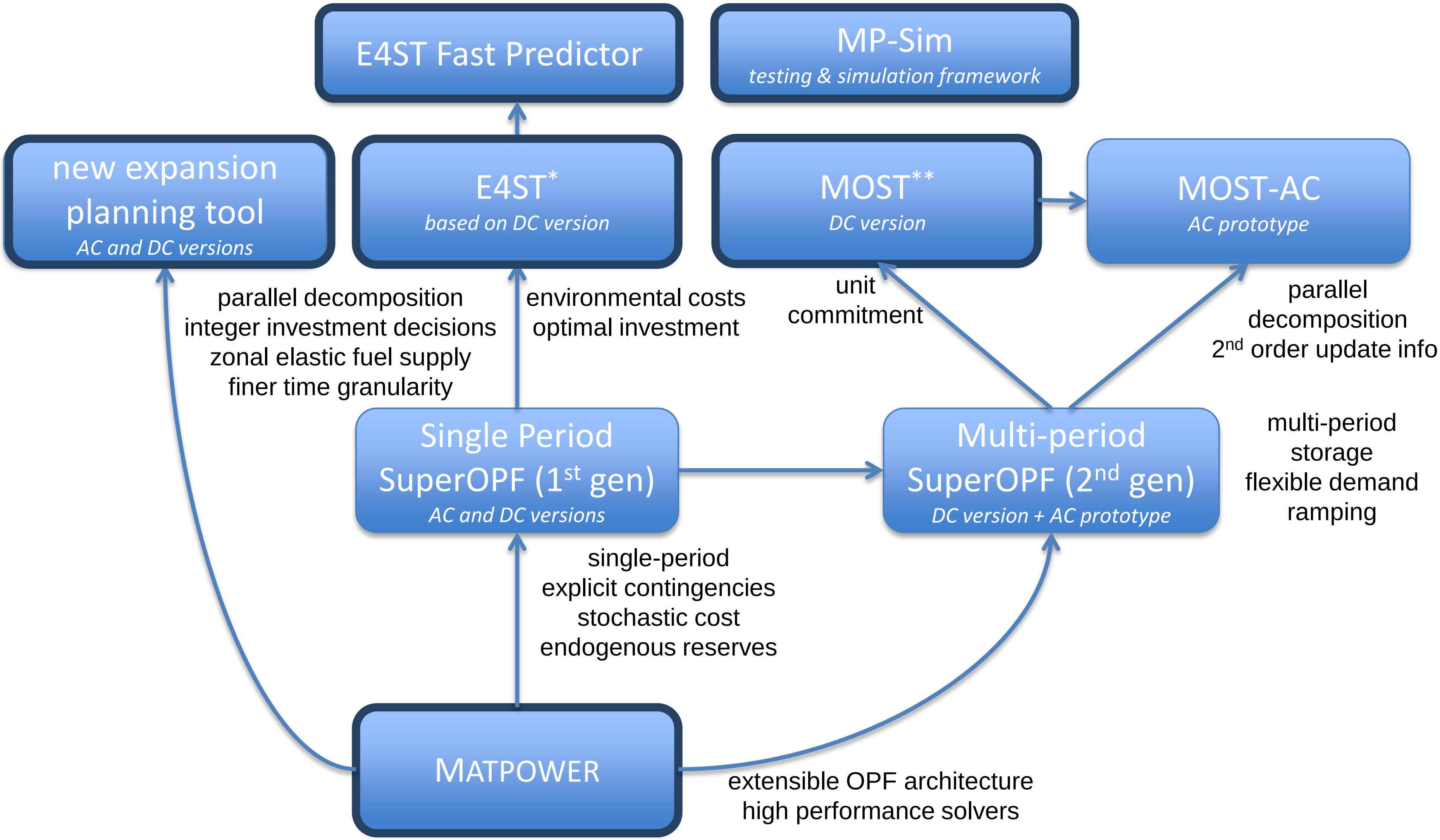
U.S. DEPARTMENT OF ENERGY

CERTS
Consortium for Electric Reliability Technology Solutions

# Outline

- Overview
- MATPOWER 6 and MOST
- Testing MOST
- Future Directions for MATPOWER

# Tools Overview




* **E4ST** – **E**ngineering, **E**conomic, **E**nvironmental **E**lectricity **S**imulation **T**ool, *formerly SuperOPF Planning Tool.*

** **MOST** – **M**ATPOWER **O**ptimal **S**cheduling **T**ool, *based on Multi-period SuperOPF with Unit Commitment (3rd generation)*

# Status of the Tools

- **E4ST** – **E**ngineering, **E**conomic, **E**nvironmental **E**lectricity **S**imulation **T**ool
  - v1.0b2 publicly available now[1]
- **E4ST Fast Predictor** – web-based, graphical predictor of E4ST results
  - v1.0b1 publicly accessible soon[1]
- **new expansion planning tool** – time-linked binary invest/retire, AC netwrk
  - prototype available upon request
- **MATPOWER**
  - v6.0b1 publicly available now[2]
- **MOST** – **M**ATPOWER **O**ptimal **S**cheduling **T**ool
  - v1.0b1 publicly available now[2], bundled with MATPOWER 6
- **MP-Sim** – simulation framework
  - prototype available upon request

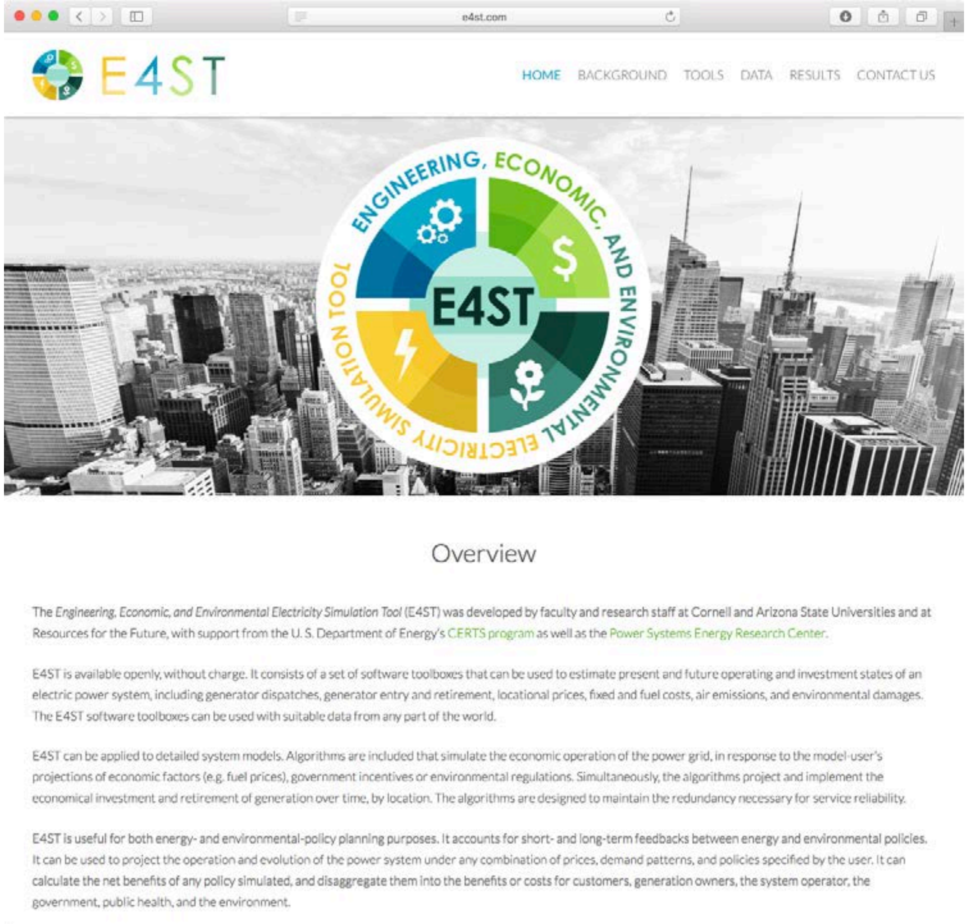
---

[1] E4ST website: http://e4st.com/

[2] MATPOWER website: http://www.pserc.cornell.edu/matpower/

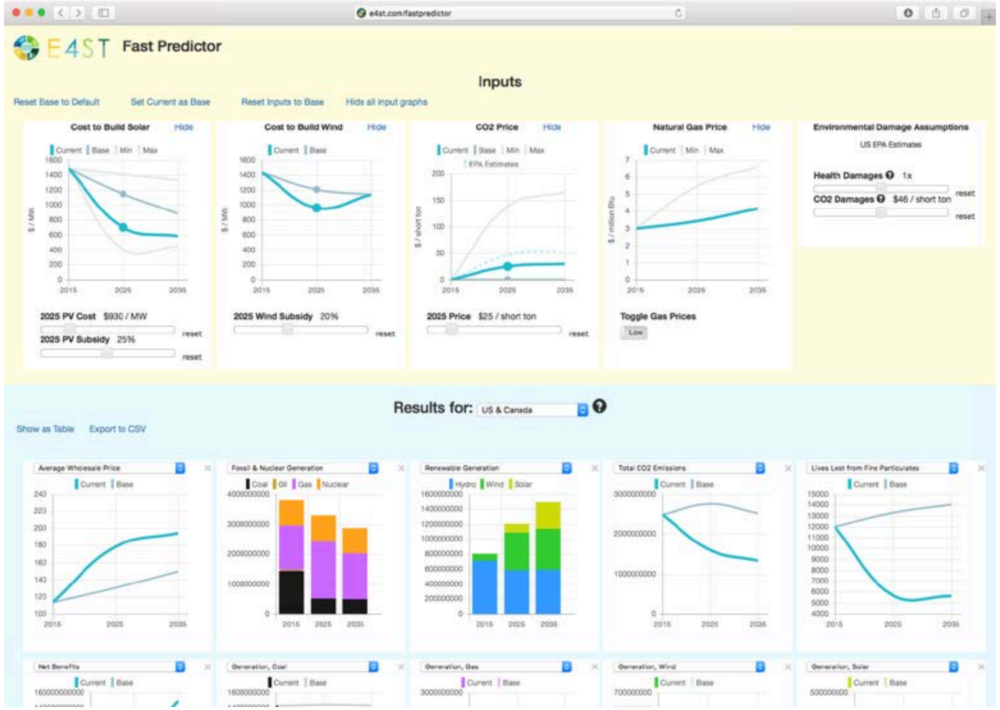
# Current Planning Tools – E4ST

- E4ST
  - public releases of v1.0b1 and b2 available at [http://e4st.com/](http://e4st.com/)
  - open-source BSD license (same as MATPOWER)
  - used by Altenex and researchers from China
- E4ST Data NA
  - v1.0 available to paying EV customers
  - data for continental North America



# New Planning Tool – Fast Predictor

- E4ST Fast Predictor
  - imminent public release of v1.0b1, accessible at http://e4st.com/fp
  - functions fitted from regressions using results of full E4ST model runs
  - web interface built in JavaScript by Cornell undergrad, Haeyong (David) Shin



# New Expansion Planning Tool

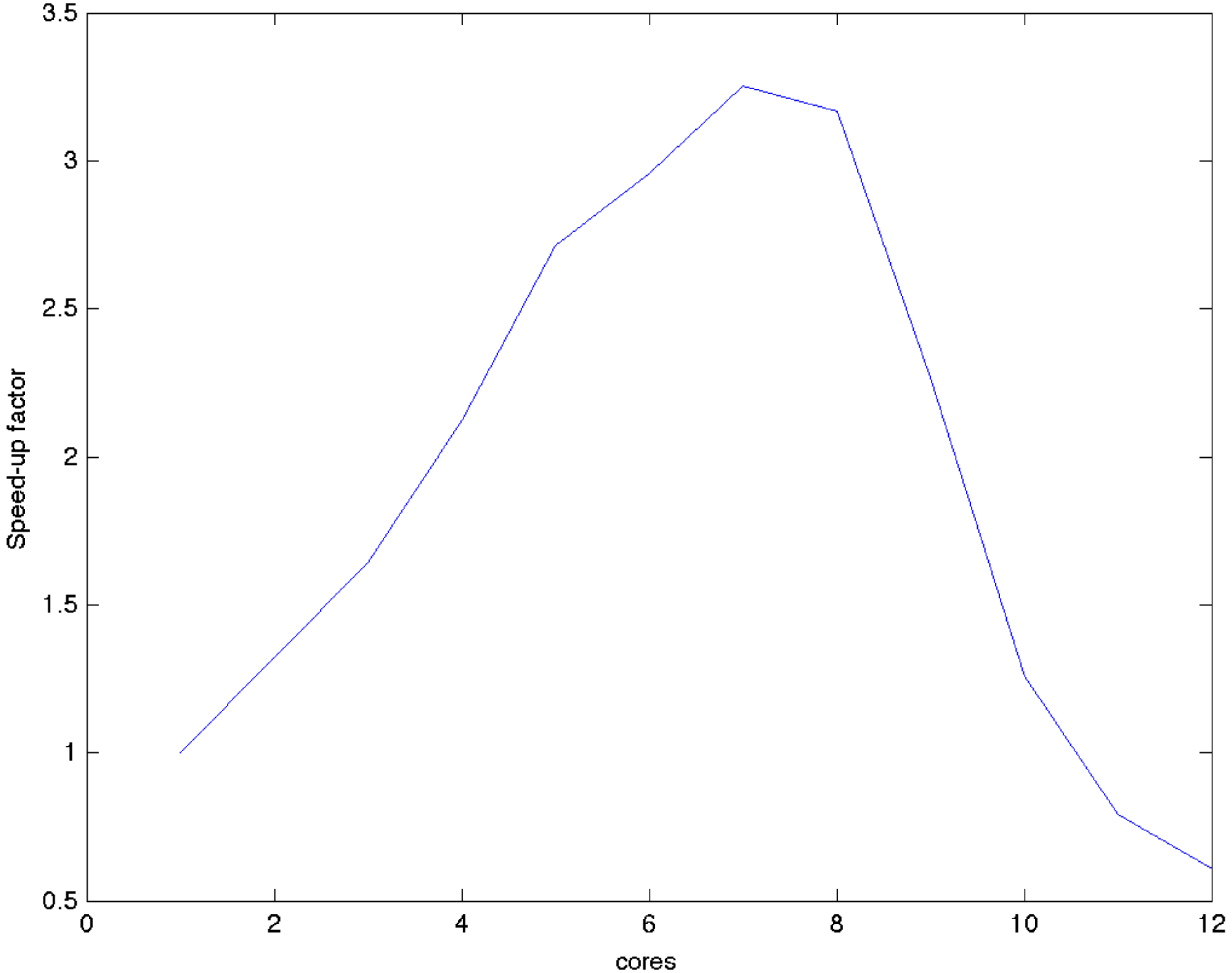
| New Formulation | Current E4ST |
| --- | --- |
| AC or DC network model | DC network model only |
| binary variables for investment/retirement decisions | continuous variables for investment/retirement decisions |
| single time-linked optimization for entire horizon | independent sequential optimizations |
| temporally co-optimized investment/retirement decisions | independent sequential investment/retirement decisions |
| fine time-granularity on investment decisions, yearly steps | coarse time-granularity on investment decisions, decade steps |
| technology-specific invest-to-deploy delays | uniform invest-to-deploy delay *(granularity of investment cycles)* |
| explicit zonal operating reserves | availability factors as proxy for operating reserves, etc. |
| linear elastic zonal fuel supply functions, possibly with delay | exogenous fuel prices |
| *potential to include ramping and UC via typical trajectories** | operations consists of single independent hours |
| iterative solution of model decomposition | direct solution of single large model** |
| highly parallelizable | limited opportunities for parallel computation |
| approximate solution with small non-zero duality gap | exact solution** |
| *explicit hydro constraints, etc. not yet implemented** | total output constraints for hydro, emissions, RPS |

* future enhancement
** for each independent investment cycle

# New Expansion Planning Tool

- Status
  - Completed final report.
  - First usable implementation for other users is now available upon request
  - Capable of parallel execution of OPFs using Mathworks' Parallel Toolbox and Distributed Computing Server (successful tests on a 24-core machine)
    - Speed-up in a 7-core test: 3.25x (one core is the master and does not compute OPFs). 6-core: 2.96x
    - OPFs are memory-intensive tasks and cache hit rate suffers in a multi-core machine, as opposed to a cluster.

# Speed-up factor



# Outline

- Overview
- MATPOWER 6 and MOST
- Testing MOST
- Future Directions for MATPOWER

# MATPOWER Nostalgic Reminiscing

PSERC

POWERWEB

Welcome | Information | Login | Logout | Credits

## Welcome to PowerWeb!

**What is PowerWeb?**

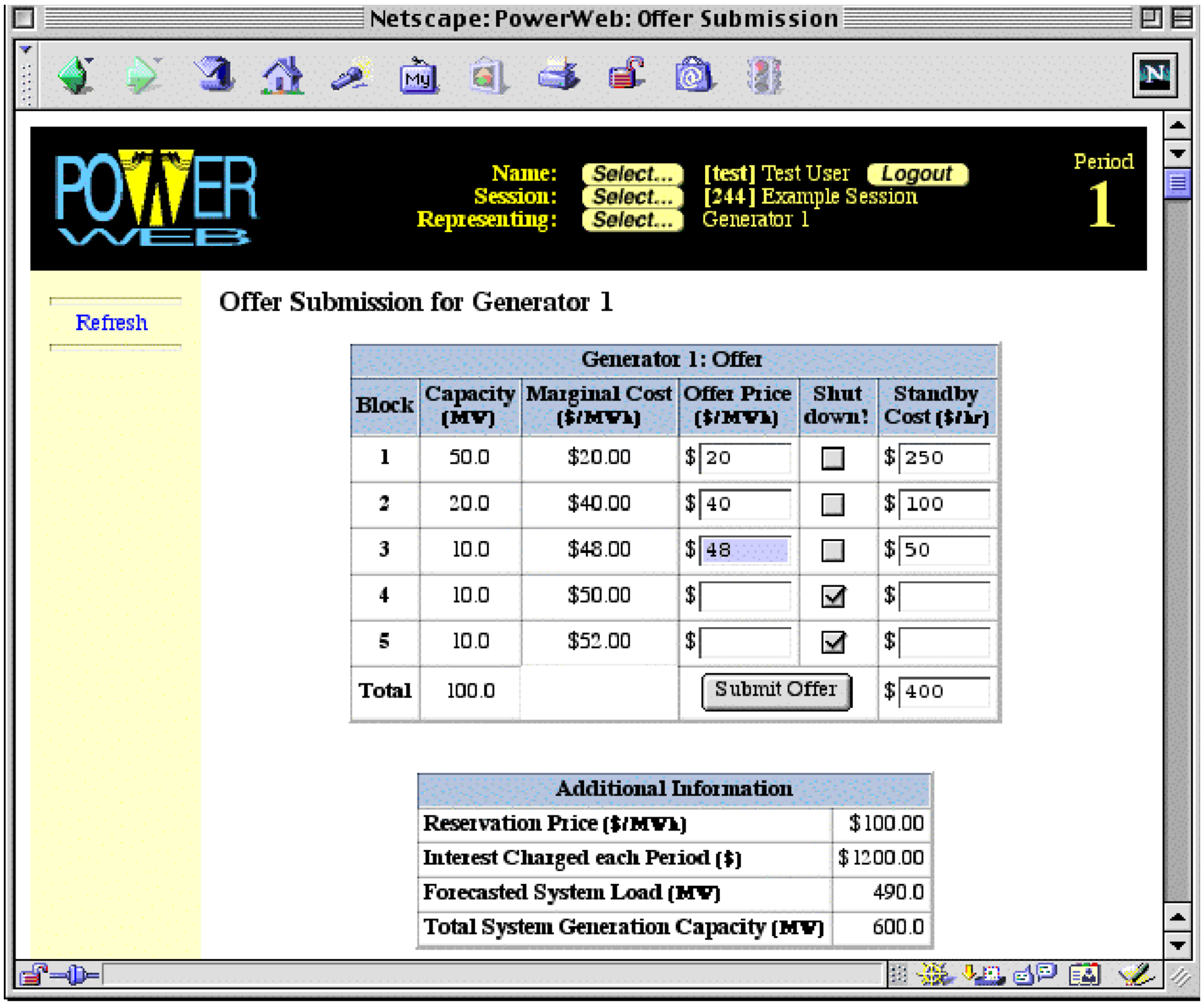
PowerWeb is a web-based, interactive, distributed, real-time electric power system simulator. This "proof-of-concept" prototype is a simple simulation of the operation of a small power system under a PoolCo model. Each generator in the system is controlled by via bids submitted by the generator's operator to the PoolCo. The PoolCo responds to the set of bids from the generators with real power settings for each generator. These real powers are used to run a power flow and the results are displayed as the current state of the system.

**How do I use it?**

In order to access the system you must have a password. Send e-mail to the address below if you need a password. Select "Login" above to go to the login page. Enter your name and password. Select which school you are representing (to be used for a future competition), and whether you are a participant or a spectator. The selection of school determines which generator is "yours". Participants may submit bids for their generator, while spectators are only able to view the data. Only one participant is allowed at a given time for a given school. Then choose the system you would like to work with and hit the login button.

*Problems or feedback?*
*contact:* Ray Zimmerman (rz10@cornell.edu)
*Last Updated: Fri, Feb 2, 1996*

# PowerWeb Input Screen

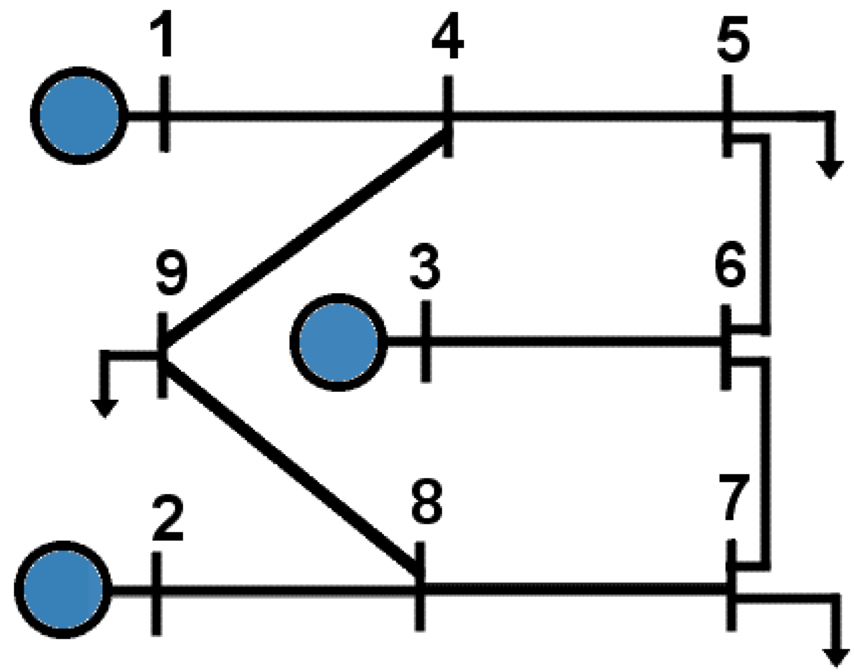



| Generator 1: Offer | | | | | |
|---|---|---|---|---|---|
| Block | Capacity (MW) | Marginal Cost ($/MWh) | Offer Price ($/MWh) | Shut down? | Standby Cost ($/hr) |
| 1 | 50.0 | $20.00 | $ 20 | ☐ | $ 250 |
| 2 | 20.0 | $40.00 | $ 40 | ☐ | $ 100 |
| 3 | 10.0 | $48.00 | $ 48 | ☐ | $ 50 |
| 4 | 10.0 | $50.00 | $ | ☑ | $ |
| 5 | 10.0 | $52.00 | $ | ☑ | $ |
| Total | 100.0 | | Submit Offer | | $ 400 |

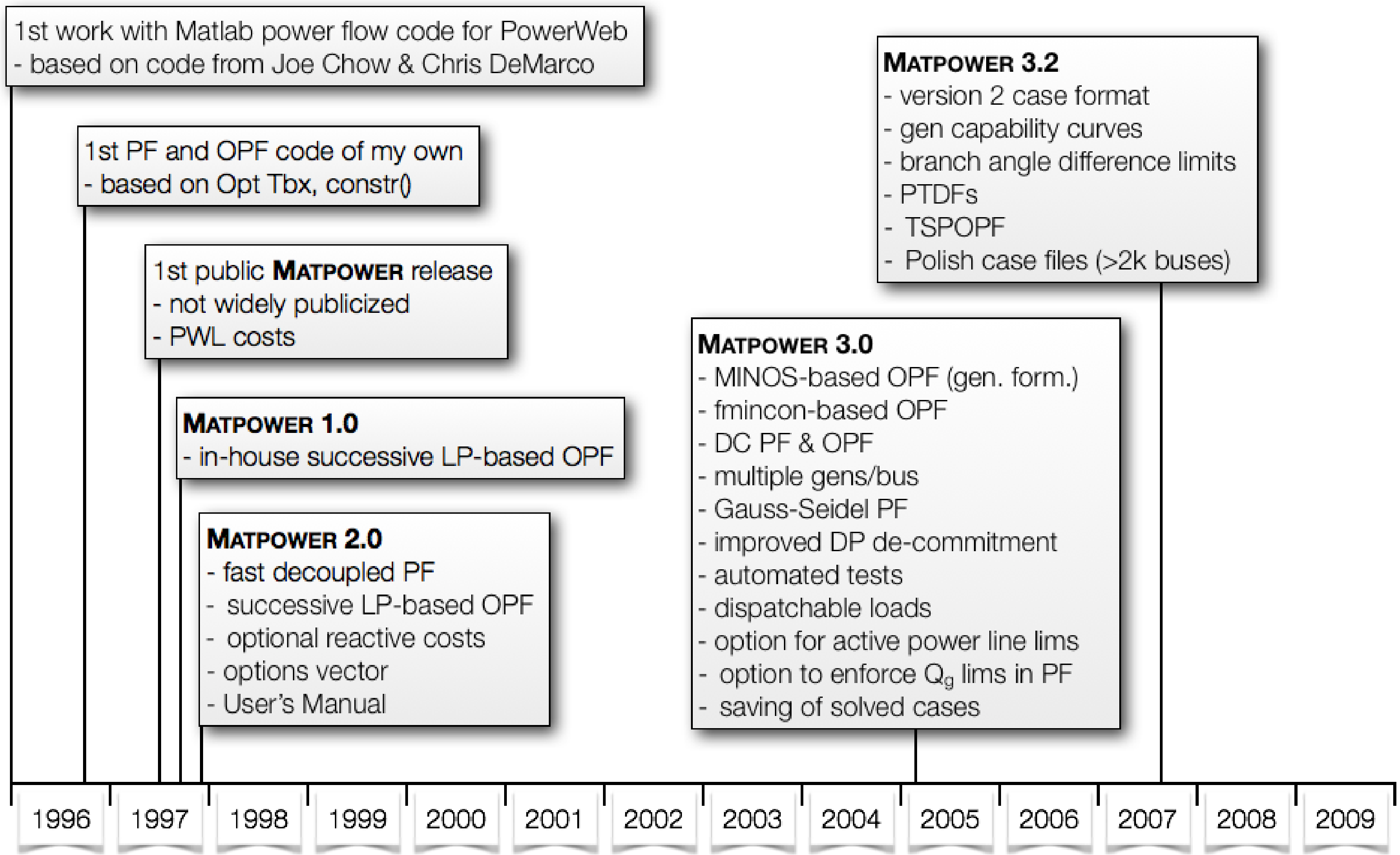
| Additional Information | |
|---|---|
| Reservation Price ($/MWh) | $100.00 |
| Interest Charged each Period ($) | $1200.00 |
| Forecasted System Load (MW) | 490.0 |
| Total System Generation Capacity (MW) | 600.0 |

# Initial PowerWeb OPF

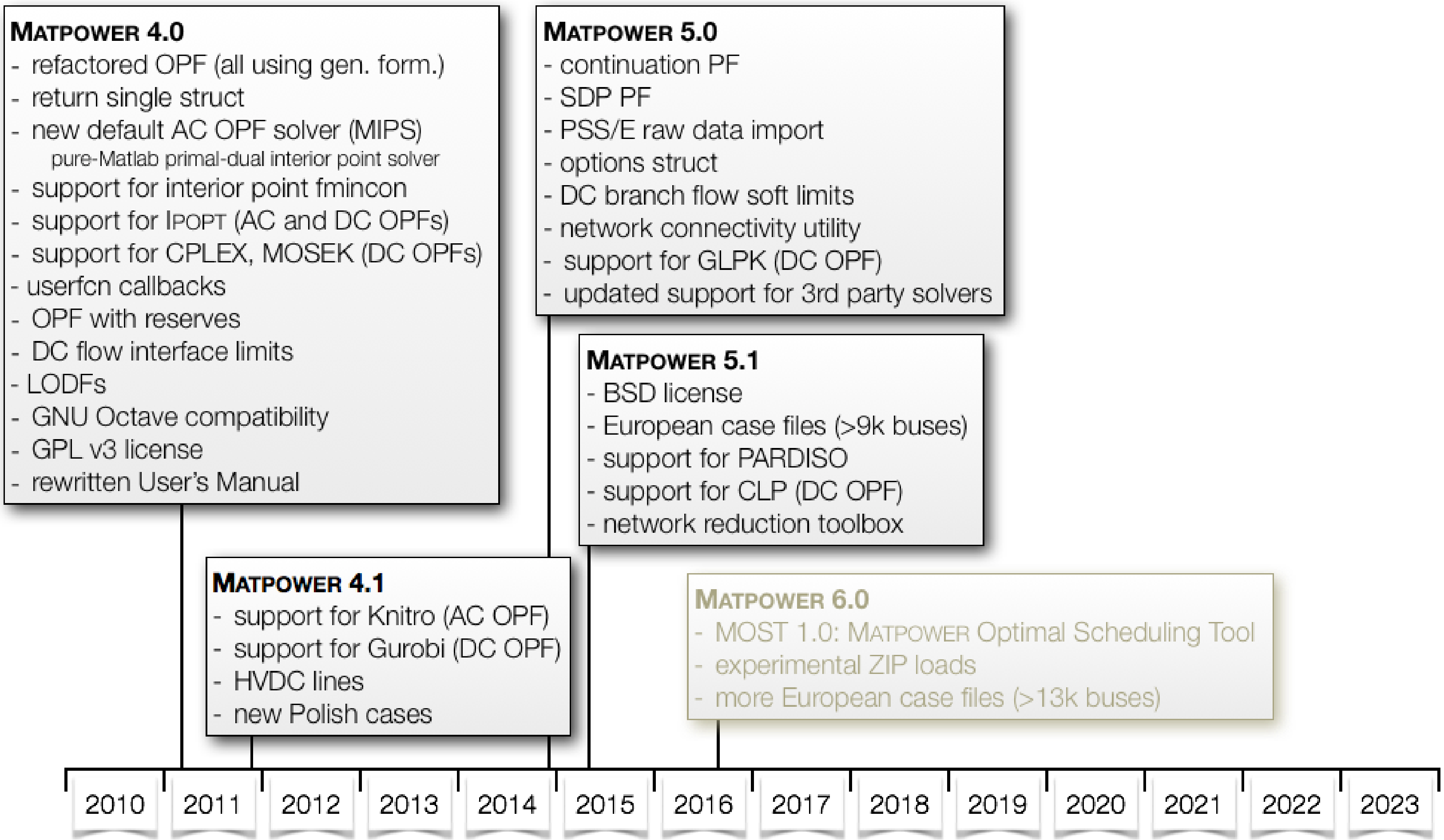
- 9 bus, 3 generator network
- OPF problem is essentially a constrained minimization problem over a 2-dimensional search space.
  - independent choice of dispatch for gens 2 and 3 (gen 1 is slack)
- brute force approach
  - discretize search space, 2-d grid
  - solve PF for each point in grid
  - eliminate infeasible points (violated line constraints or slack gen limits)
  - given set of offers (i.e. gen costs), compute cost of all feasible points, pick least cost solution



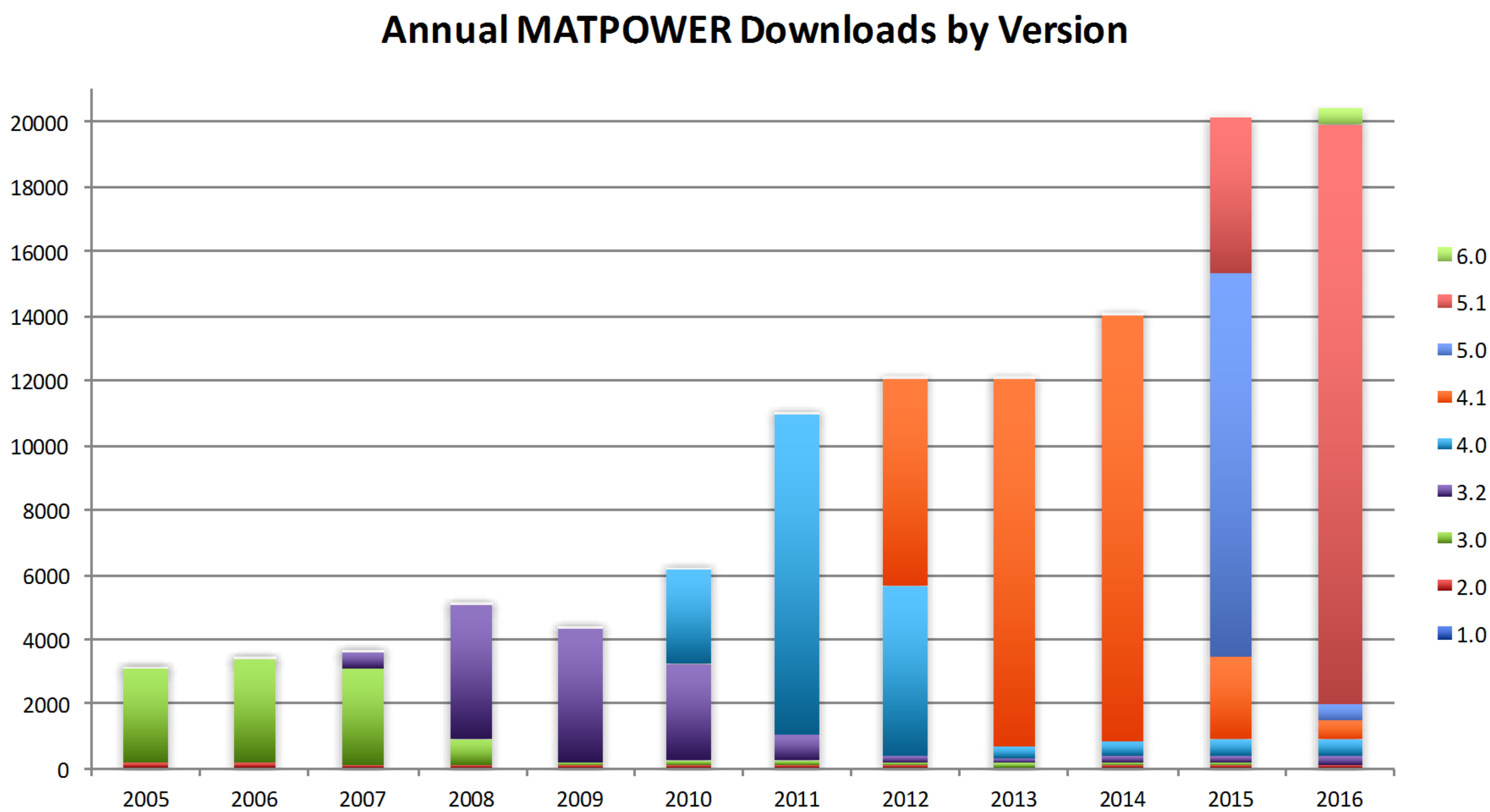
# MATPOWER History



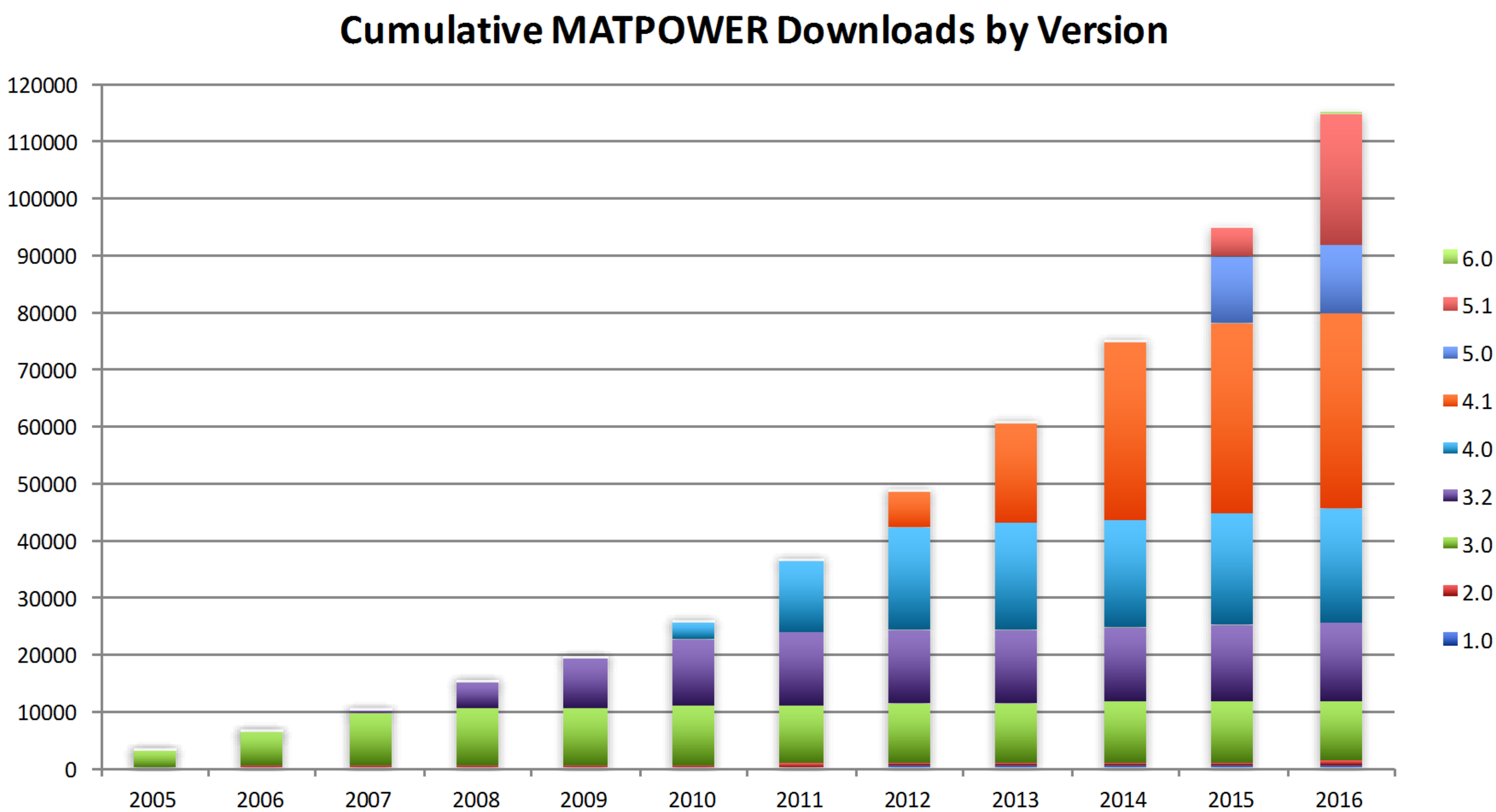
# MATPOWER History



# Annual MATPOWER Downloads



# Cumulative MATPOWER Downloads

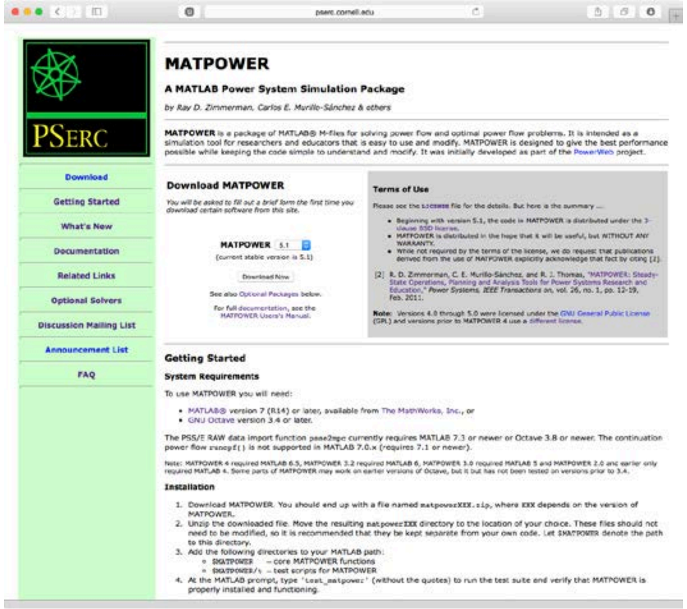


# MATPOWER

Free, open-source power system simulation environment with PF, Continuation PF, extensible OPF, stochastic unit commitment and interfaces to state-of-the-art solvers.

http://www.pserc.cornell.edu/matpower/

- used worldwide in teaching, research, industry
- momentum & impact continues to grow
  - **1315 citations** of 2011 MATPOWER paper*
  - **>1000 citations** of software/manual/other*
- serves as foundation for all tools in this project




R. D. Zimmerman, C. E. Murillo-Sánchez, and R. J. Thomas, "MATPOWER Steady-State Operations, Planning and Analysis Tools for Power Systems Research and Education," *Power Systems, IEEE Transactions on*, vol. 26, no. 1, pp. 12-19, Feb. 2011.

* *Google Scholar, 6/9/16*

# Keys to MATPOWER Success

- open code
  - easy to understand
  - easy to modify
  - customizable architecture
  - scalable high performance algorithms
- open-source license
  - free, no need even for Matlab license with GNU Octave compatibility
  - explicit permission to modify source code, port to other languages (e.g. PyPower)
  - guarantee that it won't "go away"
- case data
  - combination of high quality solvers and ready-to-use cases created a de facto benchmark platform (for optimization & power systems research)
  - thanks to Roman Korab for his work on the Polish cases

# MATPOWER 6

- **version 6.0b1 released** on June 1, 2016
  - 600 downloads in first week
- new features
  - **MOST** – **M**ATPOWER **O**ptimal **S**cheduling **T**ool
  - significant performance improvements when running many small problems
  - general mechanism for applying changes to existing MATPOWER case
  - experimental foundation for ZIP load models for PF, CPF and OPF
  - contributed code
    - plot electrically meaningful drawings of MATPOWER case, *from Paul Cuffe*
    - find max loadability limit via OPF-based method, *from Camille Hamon*
    - create QCQP representation of AC OPF problem, *from Cédric Josz and friends*
- new case files, including 13,659 bus European case
  - *from Cédric Josz and friends from the French Transmission Operator*
- bug fixes

v6.0b2, projected for release this summer, will also include handling of generator limits in CPF

# Preparation for MOST Release

- code cleanup
  - GNU Octave compatibility
  - splitting program options from input data structures
  - adding UC to supporting code for data input, auto-generation of sensible default data
- automated tests
- tutorial examples
- documentation
  - help text for each function in code
  - 109 page *MOST User's Manual*[1]

[1] http://www.pserc.cornell.edu/matpower/MOST-manual.pdf

# MOST References


1. Ray D. Zimmerman, Carlos E. Murillo-Sánchez, **"MATPOWER Optimal Scheduling Tool (MOST) User's Manual,"** 2016.
   - Available: http://www.pserc.cornell.edu/matpower/MOST-manual.pdf
2. Carlos E. Murillo-Sánchez, Ray D. Zimmerman, C. Lindsay Anderson and Robert J. Thomas, **"Secure Planning and Operations of Systems with Stochastic Sources, Energy Storage and Active Demand"**, *Smart Grid, IEEE Transactions on*, vol. 4, no. 4, pp. 2220–2229, Dec. 2013.
   - Available: http://dx.doi.org/10.1109/TSG.2013.2281001
3. A. J. Lamadrid, D. Shawhan, C. E. Murillo-Sánchez, R. D. Zimmerman, Y. Zhu, D. Tylavsky, A. Kindle, and Z. Dar, **"Stochastically Optimized, Carbon- Reducing Dispatch of Storage, Generation, and Loads,"** *Power Systems, IEEE Transactions on*, vol. 30, no. 2, pp. 1064–1075, Mar. 2015.
   - Available: http://dx.doi.org/10.1109/TPWRS.2014.2388214
4. A. J. Lamadrid, D. Muñoz-Álvarez, C. E. Murillo-Sánchez, R. D. Zimmerman, and R. J. Thomas, **"Scheduling of Commitment, Energy and Reserves Under Uncertainty in a Two-Settlement Framework,"** *Power Systems, IEEE Transactions on*.
   - To be submitted any day now.

# Outline

- Overview
- MATPOWER 6 and MOST
- Testing MOST
  - Stochastic vs. Deterministic
  - MP-Sim
- Future Directions for MATPOWER

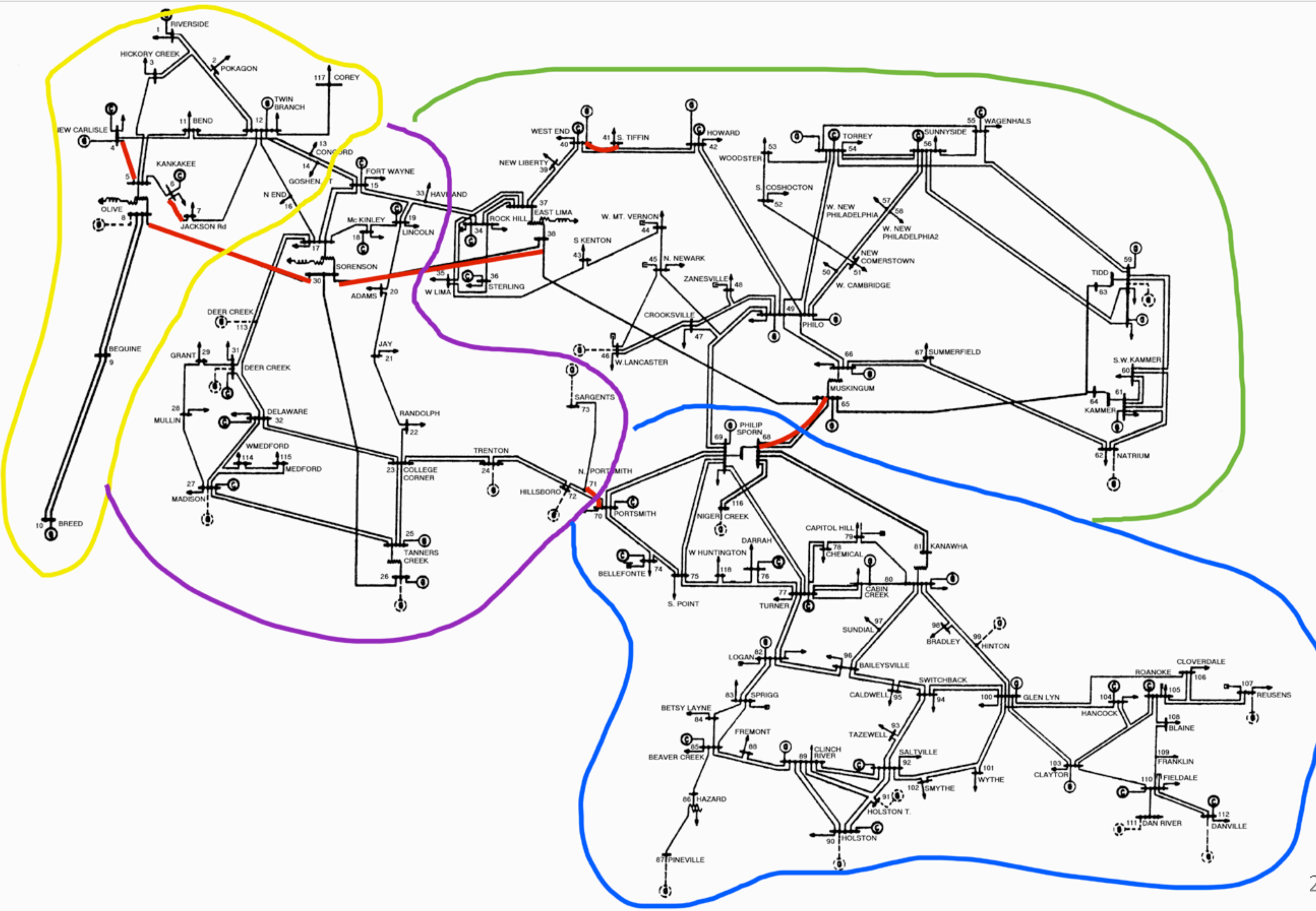
# Stochastic vs. Deterministic Comparison

- 1st settlement
  - solves a multi-period plan resulting in day-ahead commitment decisions and reserve allocations
- 2nd settlement
  - solves single-period problem to determine energy dispatch and contingency reserve allocation subject to
    - UC decisions from 1st settlement
    - dispatch from previous period 2nd settlement
    - newly revealed uncertainty
  - currently using 2nd settlement to approximate actual operation

# Testing Structure

- Given:
  - historical temp, wind, demand up to operating day *(any selected day of interest)*
  - ARIMA model of temp, wind, demand that can generate potential realizations of the operating day
- For each approach:
  1. Solve 1st settlement problem for the day (based on uncertainty predicted by the ARIMA model).
  2. Select *N* realizations of the day generated by ARIMA model, for each solve 2nd settlement problems sequentially for each hour, subject to 1st settlement.

# 118-bus Test System



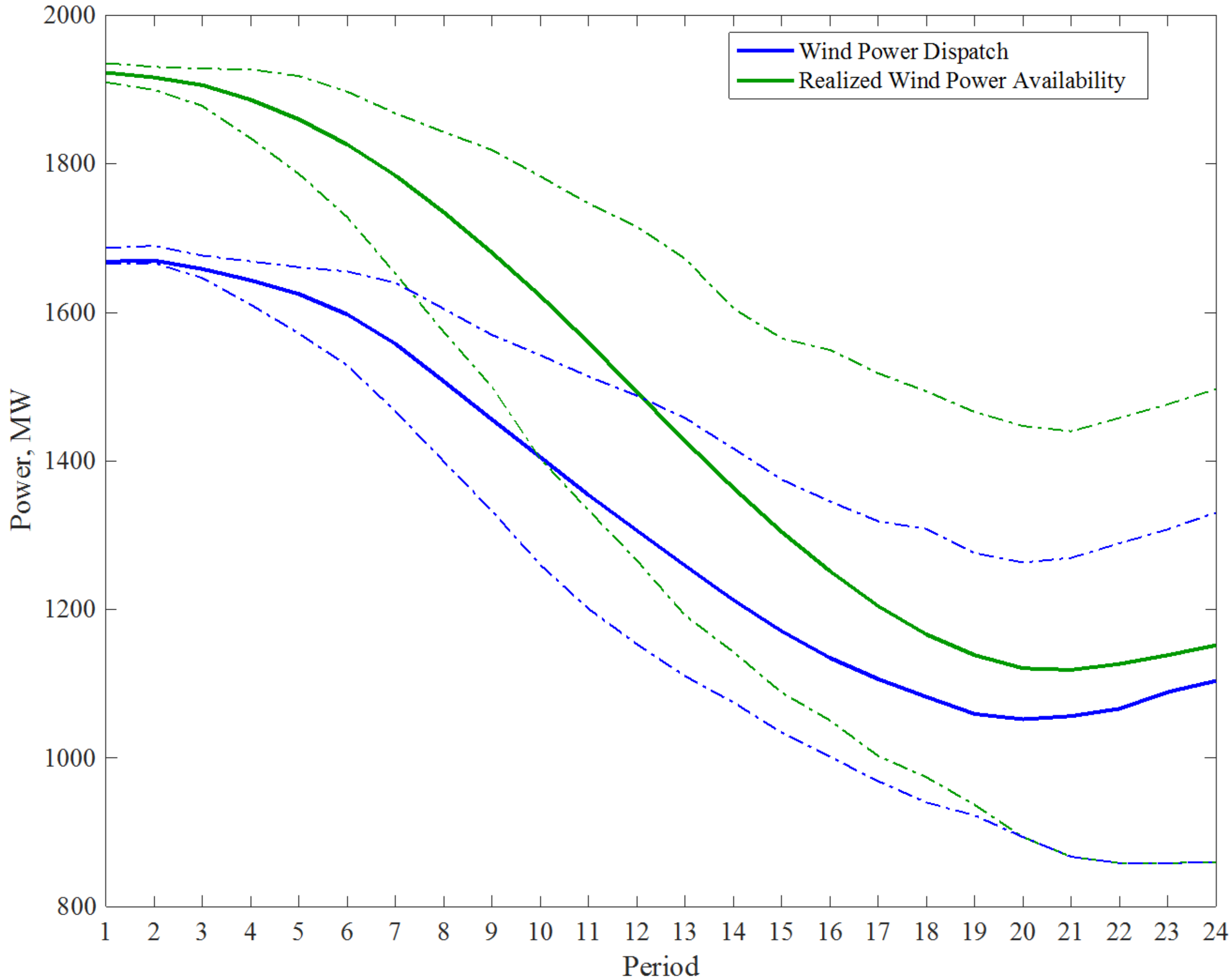
# UC Problem Dimensions

| number of … | |
|---|---|
| buses | 118 |
| conventional generators | 42 |
| wind farms | 12 |
| grid-level storage units | 0 |
| curtailable loads | 99 |
| periods in horizon, $\|T\|$ | 24 |
| scenarios per period, $\|J^t\|$ | 5 |
| contingencies per scenario, $\|K^{tj}\| - 1$ | 7 |
| variables in resulting MILP | 582,990 |
| constraints in resulting MILP | 1,536,006 |

# Updates from Previous Results

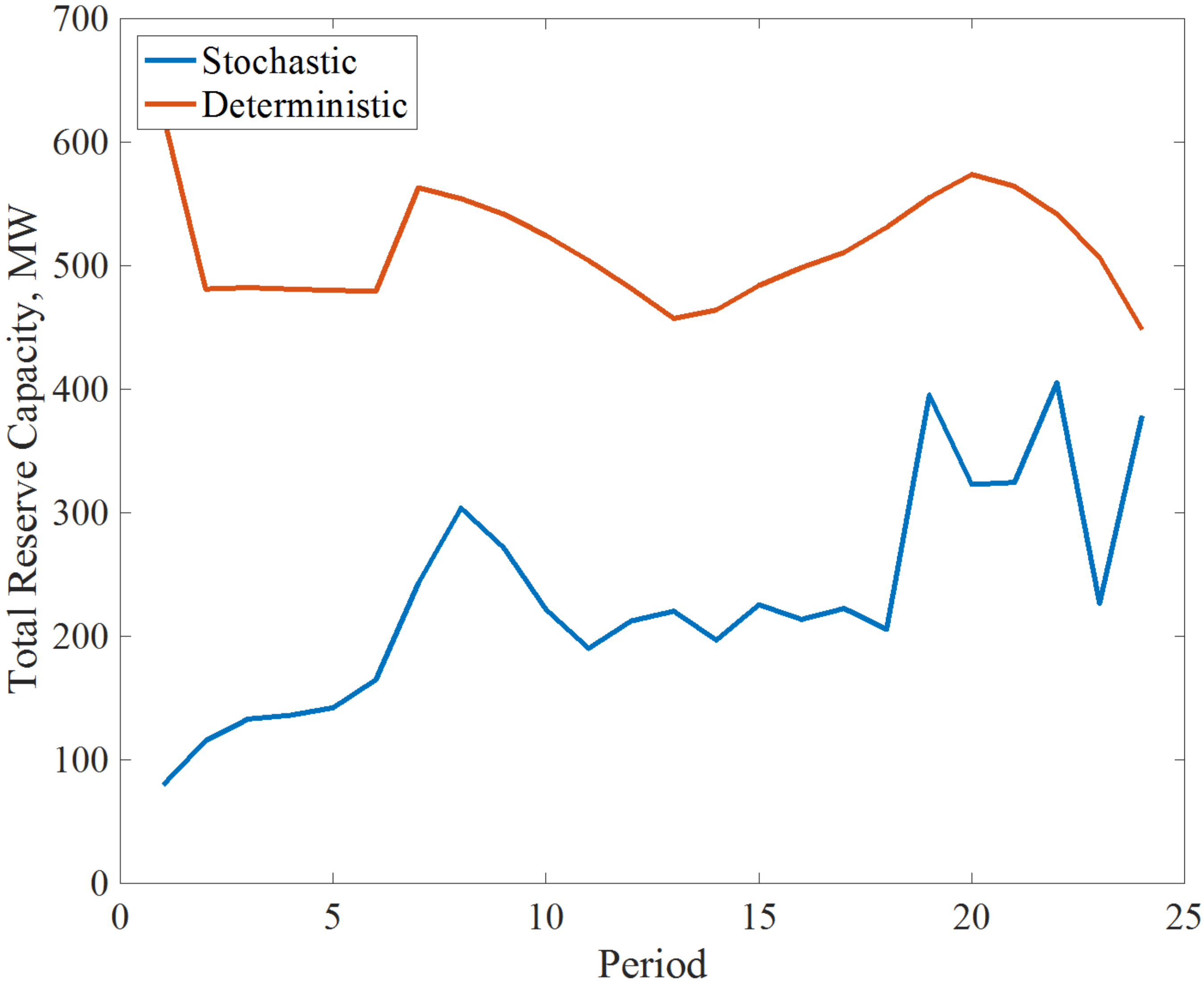
- Increased number of realized trajectories to 500
- Ran many different cases, to understand sensitivities
  - varying reserve levels for deterministic cases
  - varying value of lost load for both stochastic and deterministic
- Updated visualizations
- Wrote the paper
  - ready for submission any day now

# Total Wind Power Availability & Dispatch



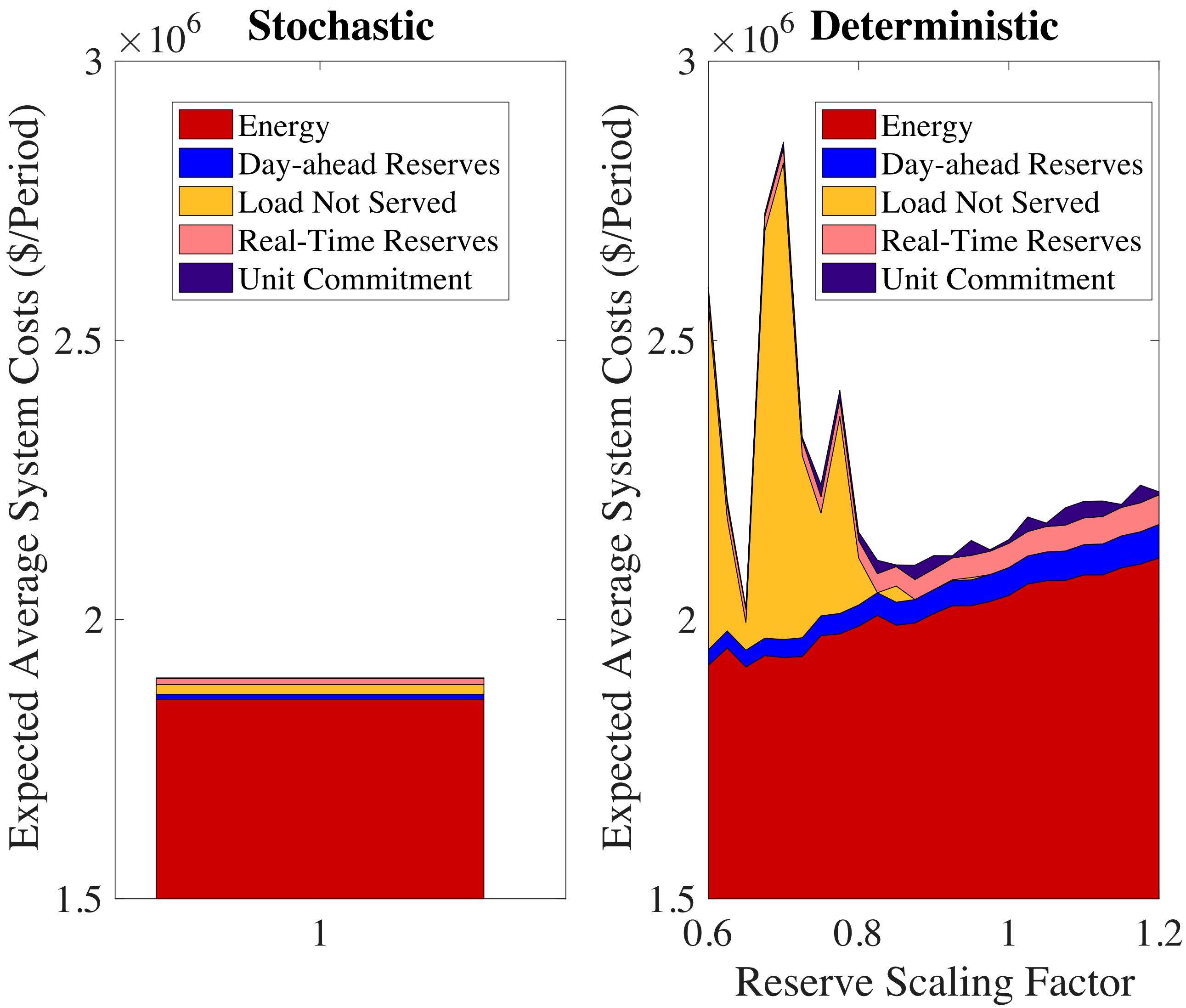
# Reserve Comparison

*by period*



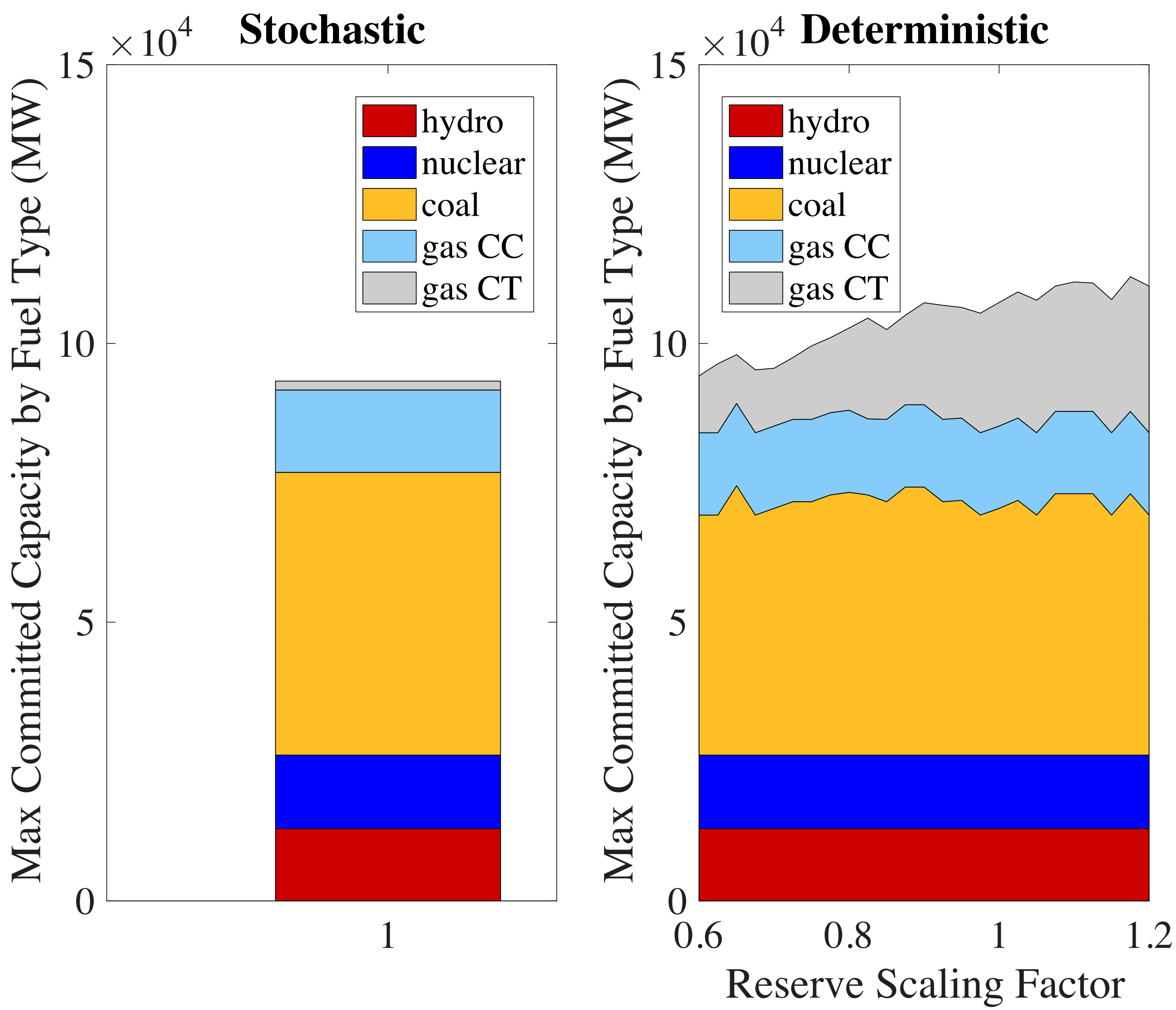
# Expected Average System Costs

*by fuel type*



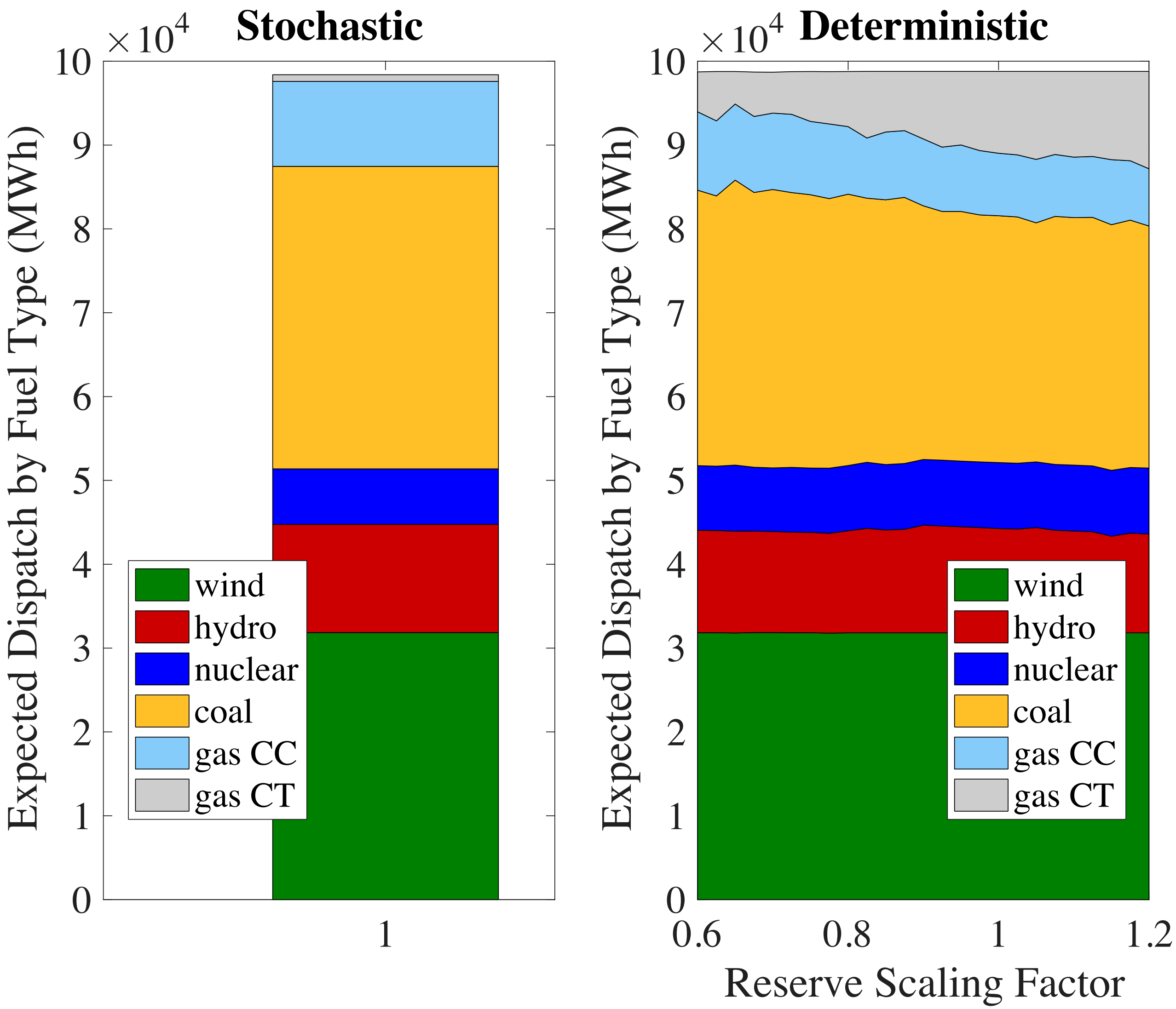
# Maximum Committed Capacity

*by fuel type*



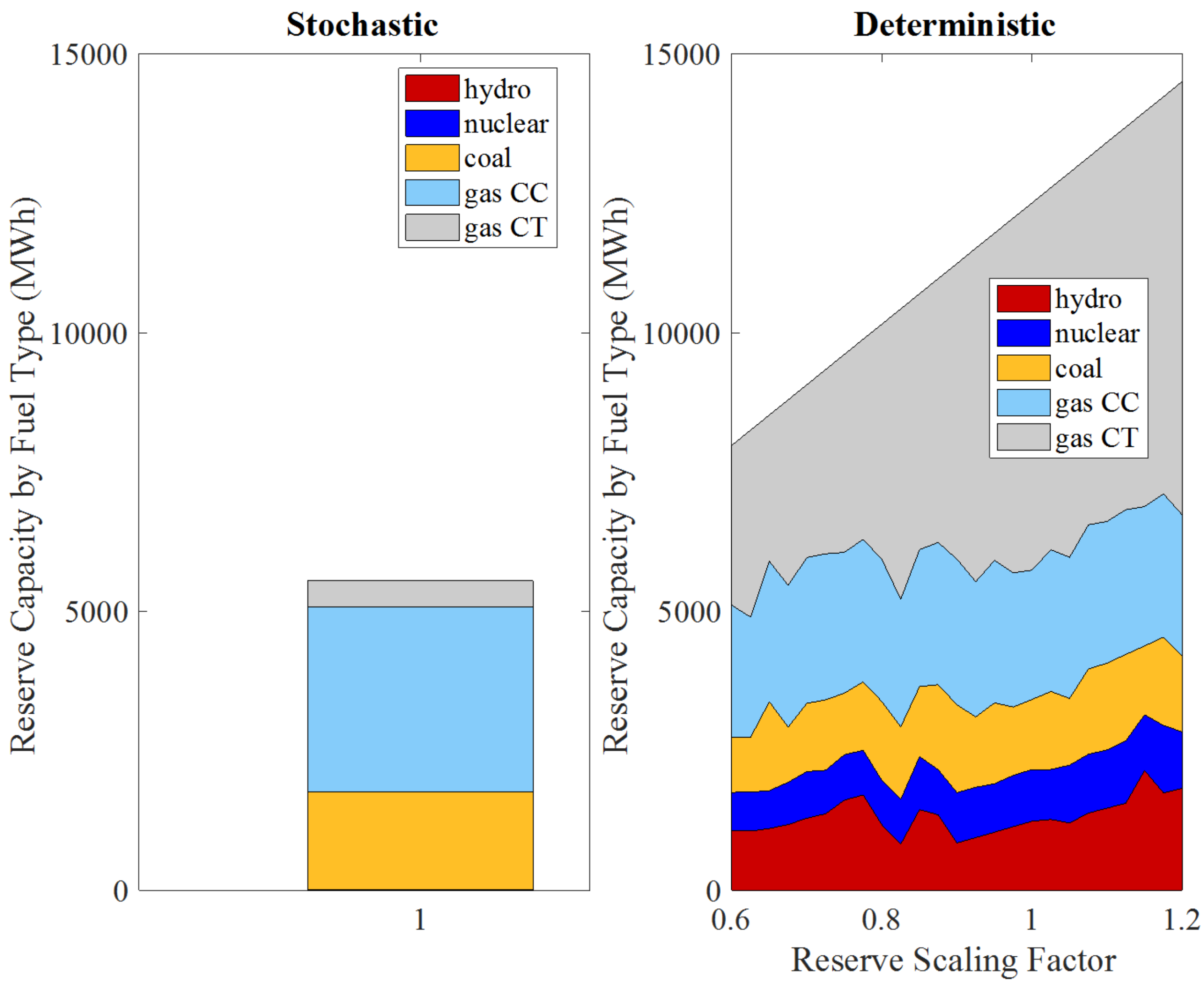
# Expected Dispatch

*by fuel type*

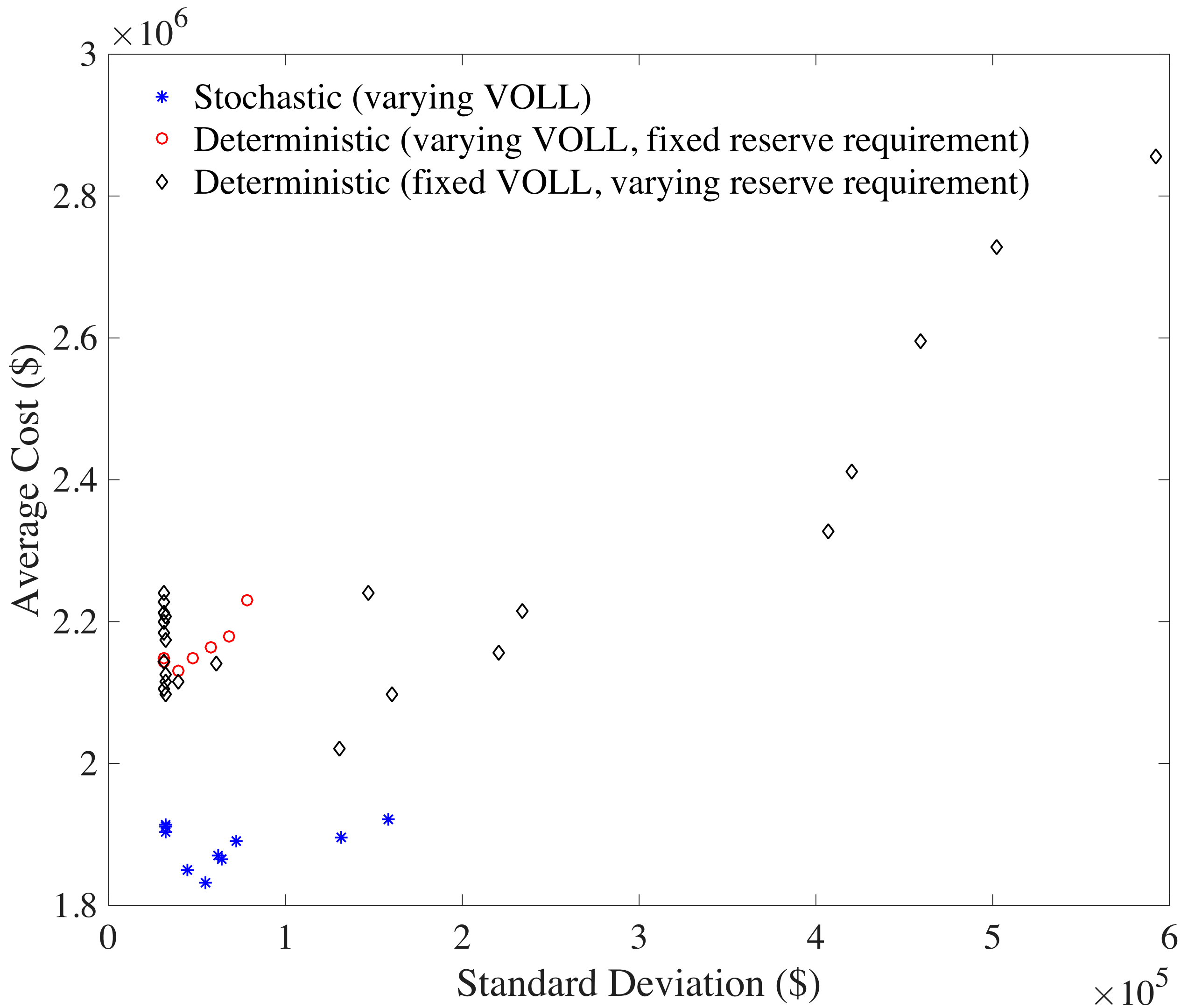


# Reserve Comparison

*by fuel type*



# Total Cost Statistics



# Outline

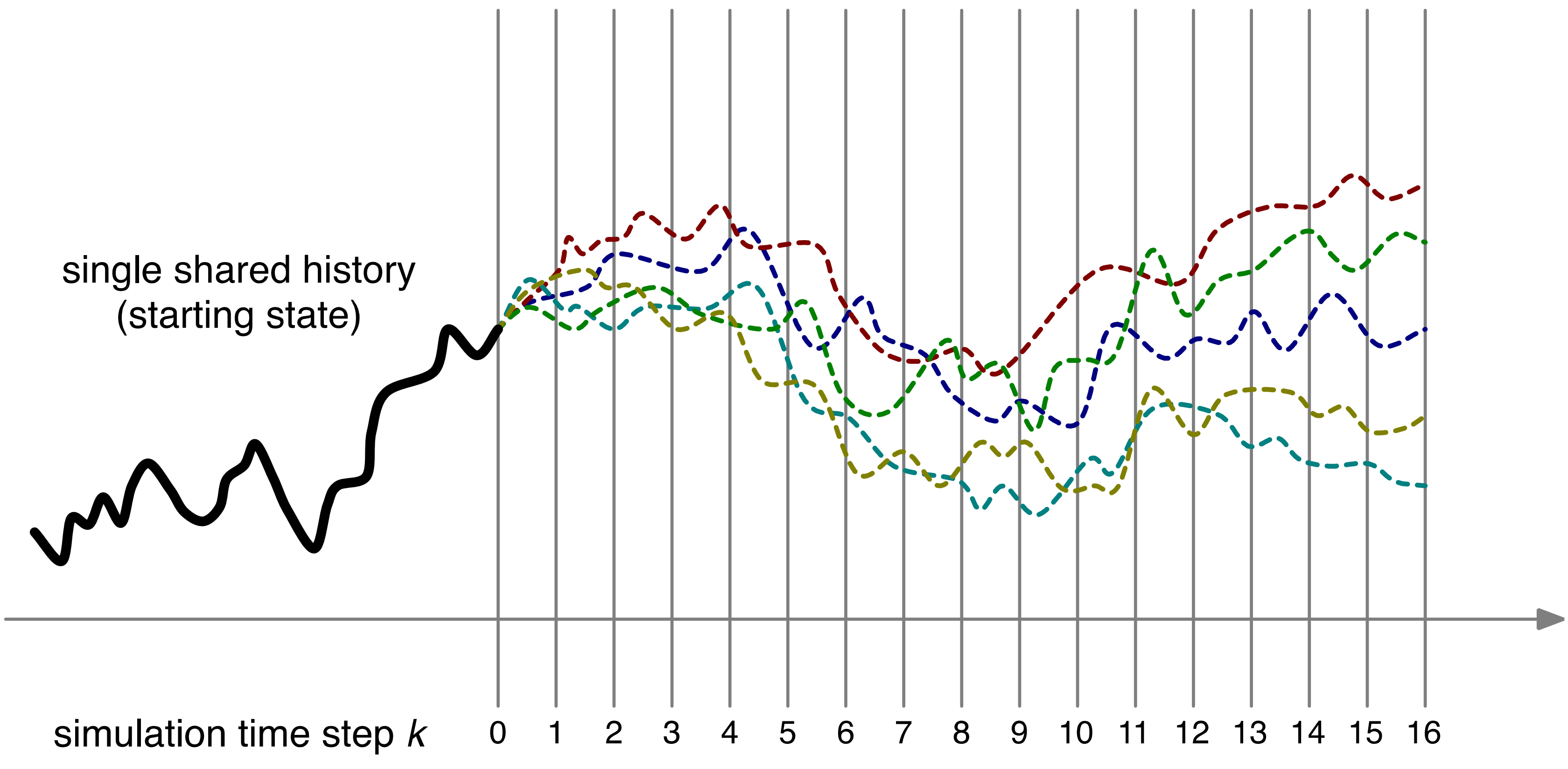
- Overview
- MATPOWER 6 and MOST
- Testing MOST
  - Stochastic vs. Deterministic
  - MP-Sim
- Future Directions for MATPOWER

# MP-Sim

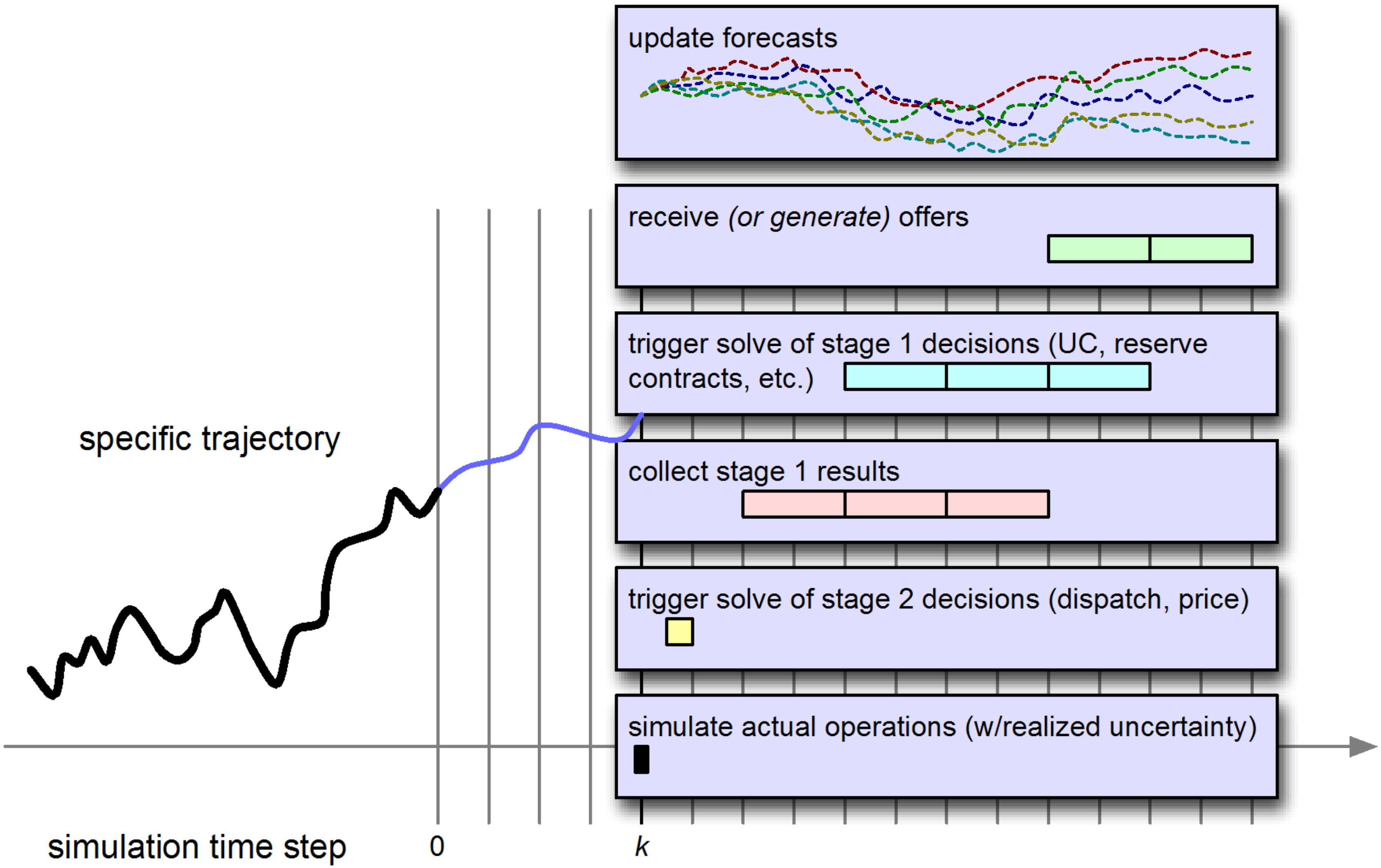
- Goal: build a generic simulation framework that can be used to test sequences of complex inter-related processes, especially scheduling problems:
  - day-ahead markets and real-time operations
  - receding horizon markets and operations
  - demand and renewable forecasting
- an object-oriented generic simulator
  - Matlab language
  - GNU Octave compatible

# Simulation of Multiple Trajectories

*(potentially in parallel)*



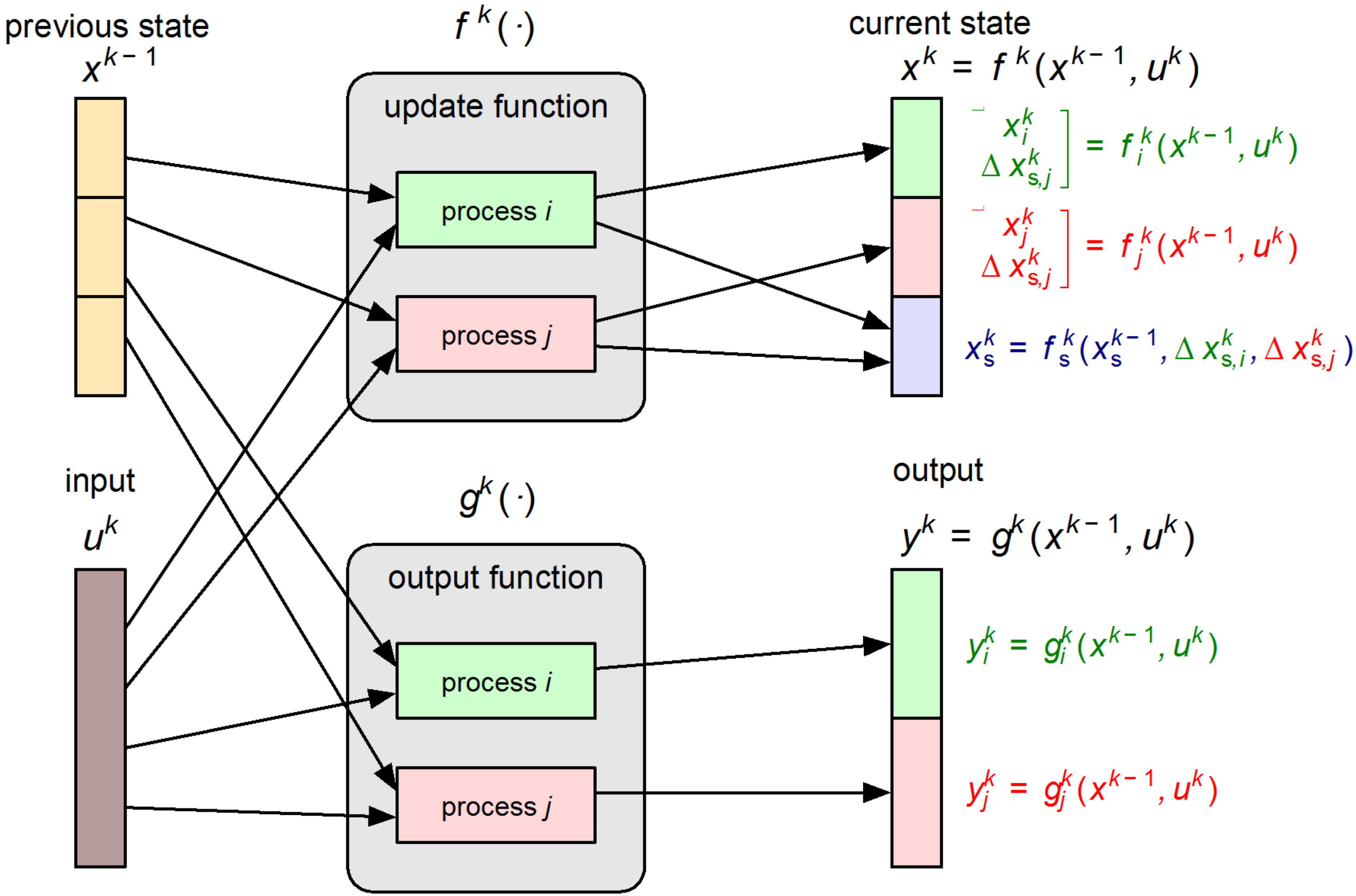
# Step *k* of Given Trajectory



# Global Update Equations

- State at time step $k$ is $x^k$
- Input that becomes available at $k$ is $u^k$
- State update function
$$x^k = f^k(x^{k-1}, u^k)$$
- Output function
$$y^k = g^k(x^{k-1}, u^k)$$
- Simulator initializes $x^0$, then begins stepping through $k = 1..N$, evaluating $f^k$ and $g^k$

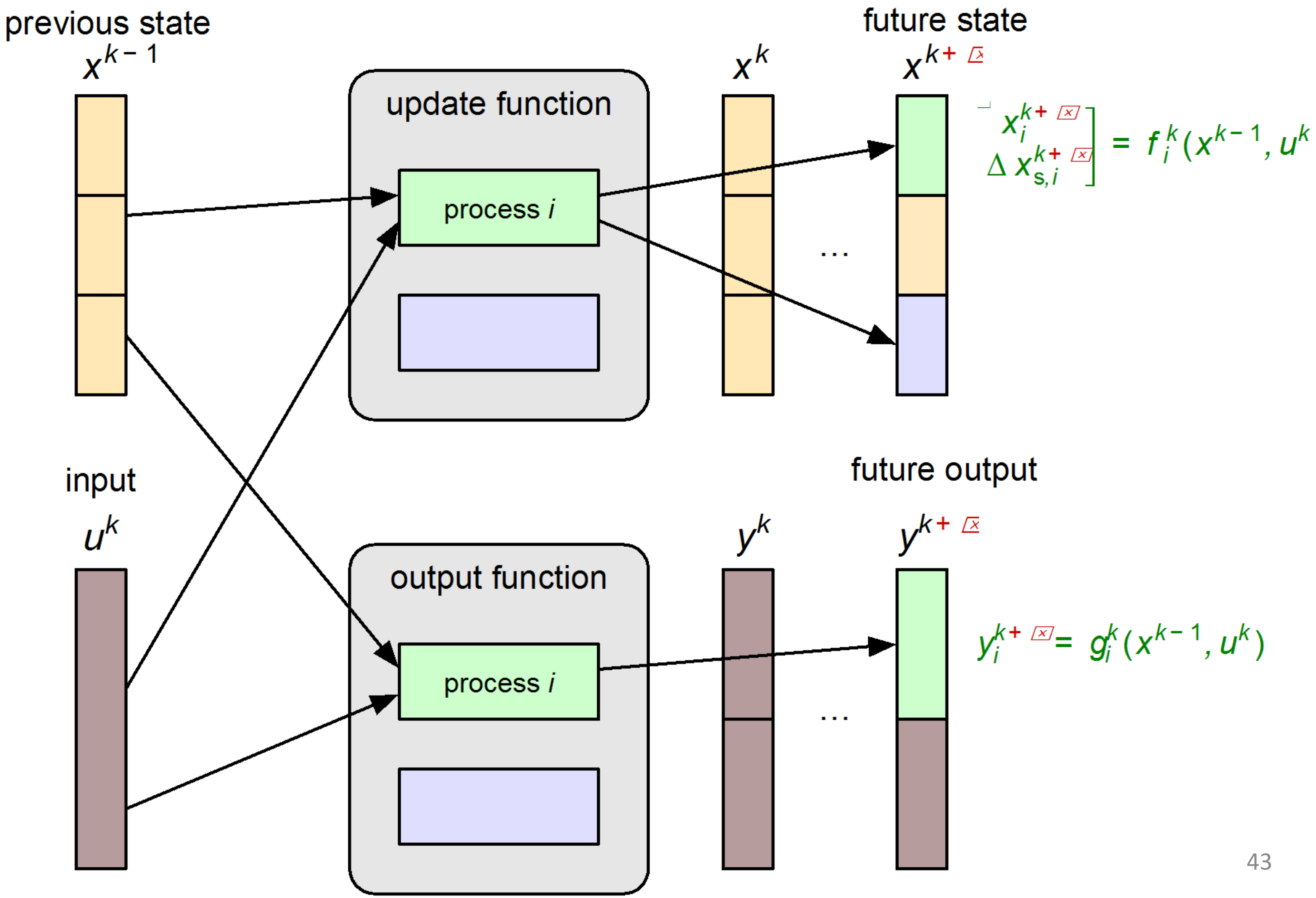
# Update Diagram



# Except That Processes Take Time

- Each process runs (*triggers*) with a specified frequency …
  - UC once every 24 hours
  - dispatch once every 5 minutes
- … and requires a certain number $\tau$ of simulation time steps to complete (*finalize*).
- So the update function $f_i$ and output function $g_i$ for a process $i$ that *triggers* in step $k$ will have access to $x^{k-1}$ and $u^k$ but actually *finalizes* in step $k+\tau$, affecting state $x^{k+\tau}$ and output $y^{k+\tau}$
  - *trigger* time determines what information is available to the process
  - *finalize* time determines when results are available for other processes

# Update Diagram with Delays



# Creating a Simulation

- Override “shared state” class to implement shared state information.
- Override “process” class for each process, defining its portion of the state, and implementing its update and output functions.
- Override the “simulator” class, implementing an initialization function that registers the shared state and process objects, specifying timing parameters for processes.
- Implement post-processing methods to process the results.
- Run it!

# MP-Sim Status

- Undergrad research assistant, Haeyong (David) Shin
  - implemented working prototype
  - wrote draft manual
  - created example simulations
    - tutorial example: burger shop
    - sequential MATPOWER OPF example
    - two-settlement, stochastic vs. deterministic UC and dispatch using MOST
- Next steps *(David will continue next acadmic year)*
  - currently undergoing some redesign
  - receding horizon tests using MOST
  - public release on GitHub under BSD license

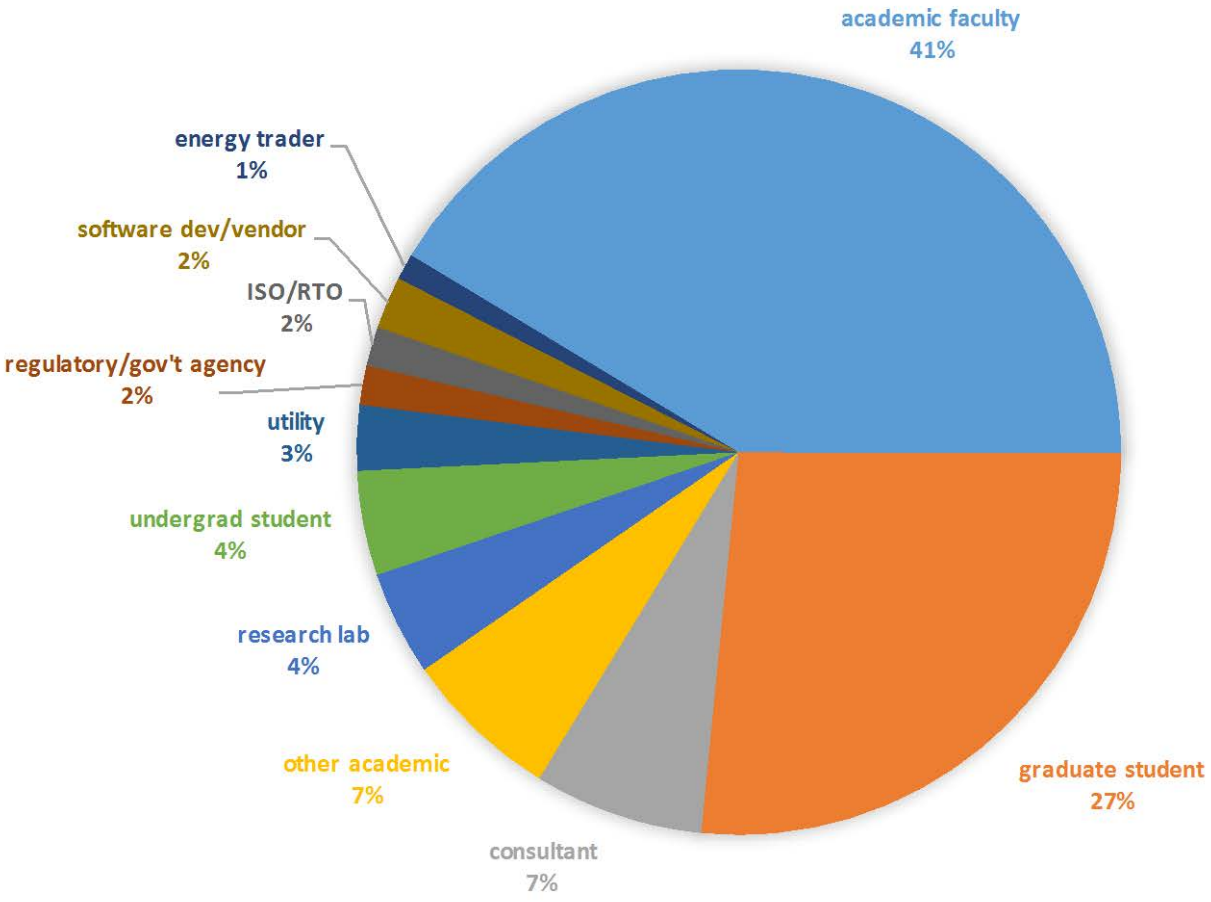
# Outline

- Overview
- MATPOWER 6 and MOST
- Testing MOST
- Future Directions for MATPOWER
  - MATPOWER Survey Results
  - NSF Proposal

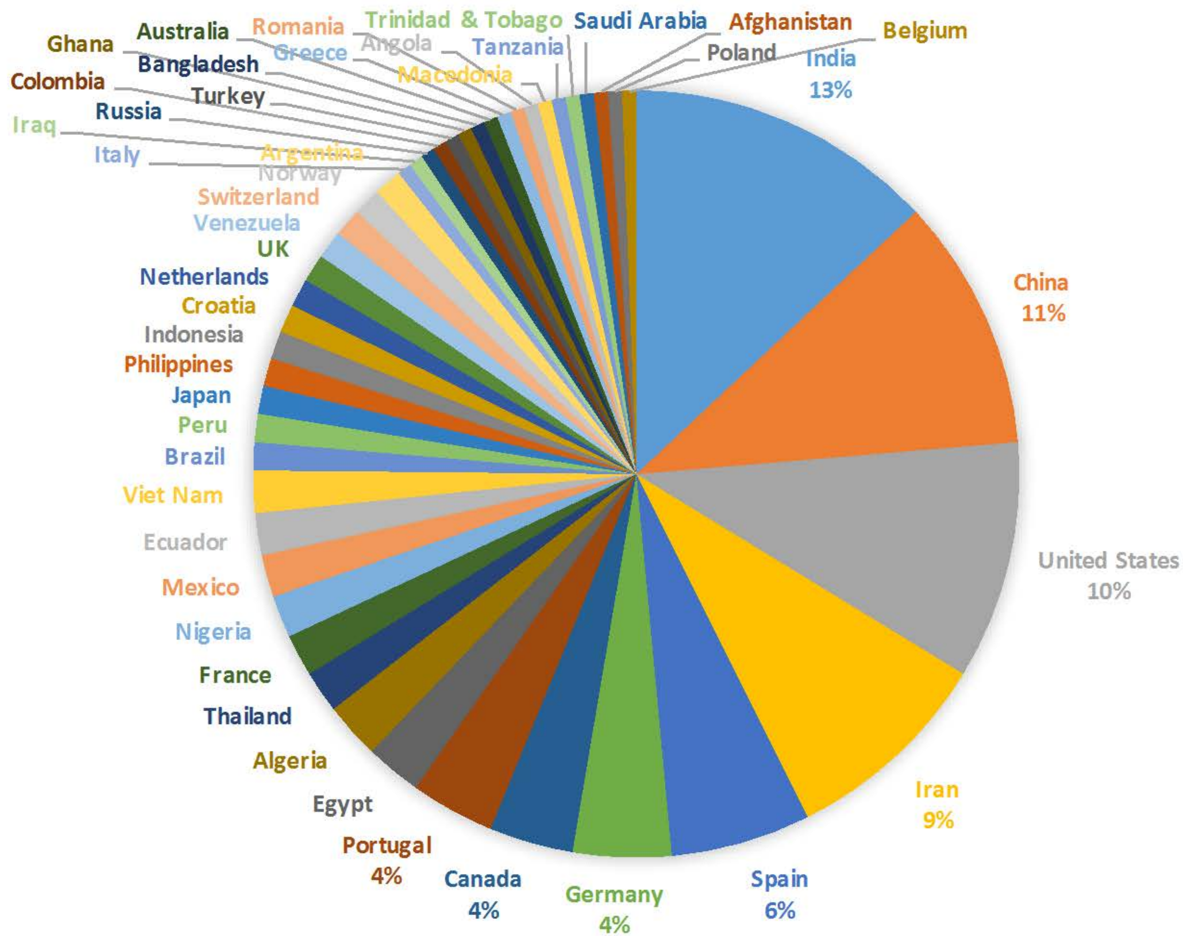
# MATPOWER User Survey

- first MATPOWER user survey, to determine …
  - who is using MATPOWER
  - what they are using it for
  - what enhancements to the software and community would be most valuable
- sent to MATPOWER discussion and announcement e-mail lists
- 183 respondents
  - extremely useful information

# Who is using MATPOWER?

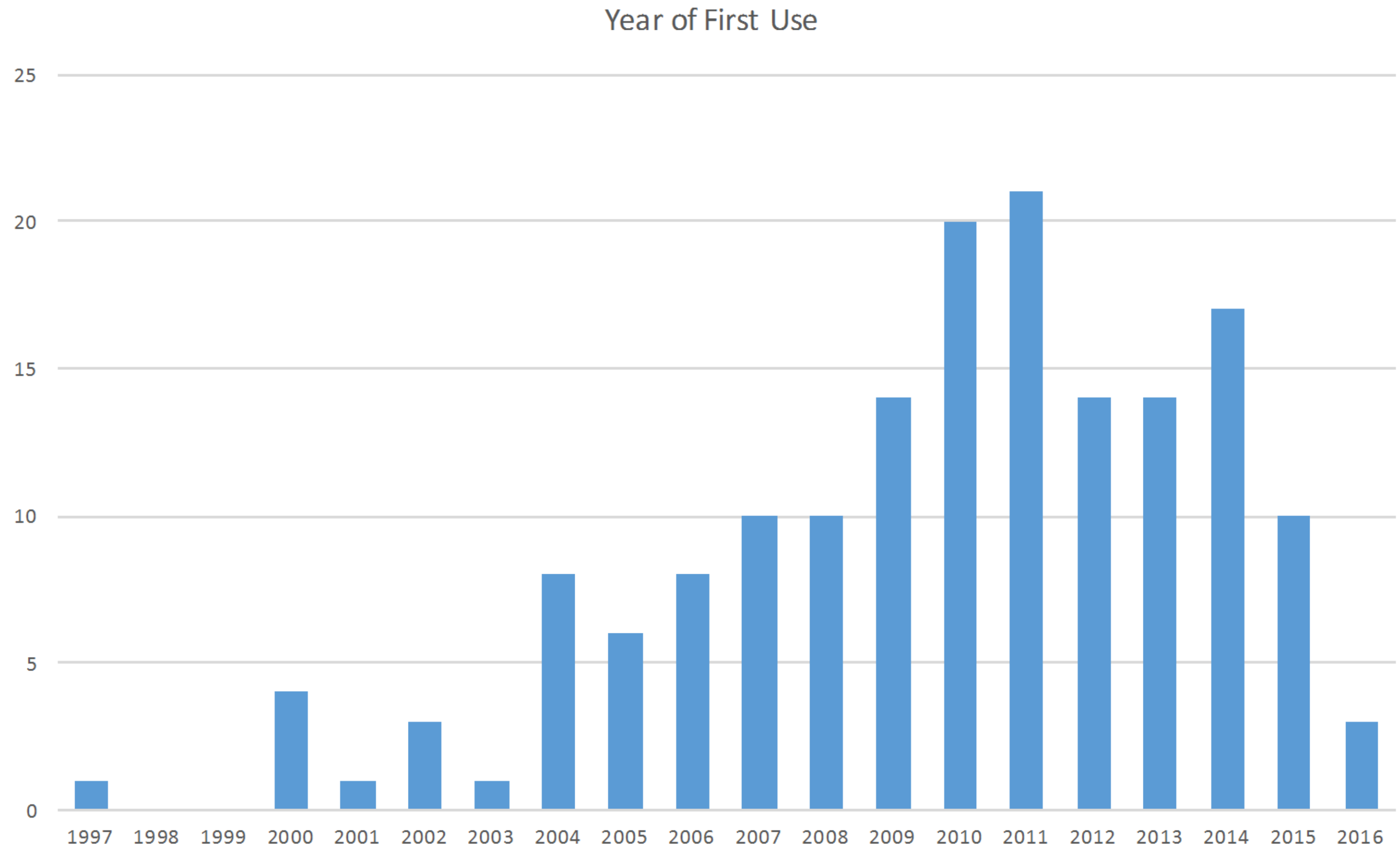


# Countries



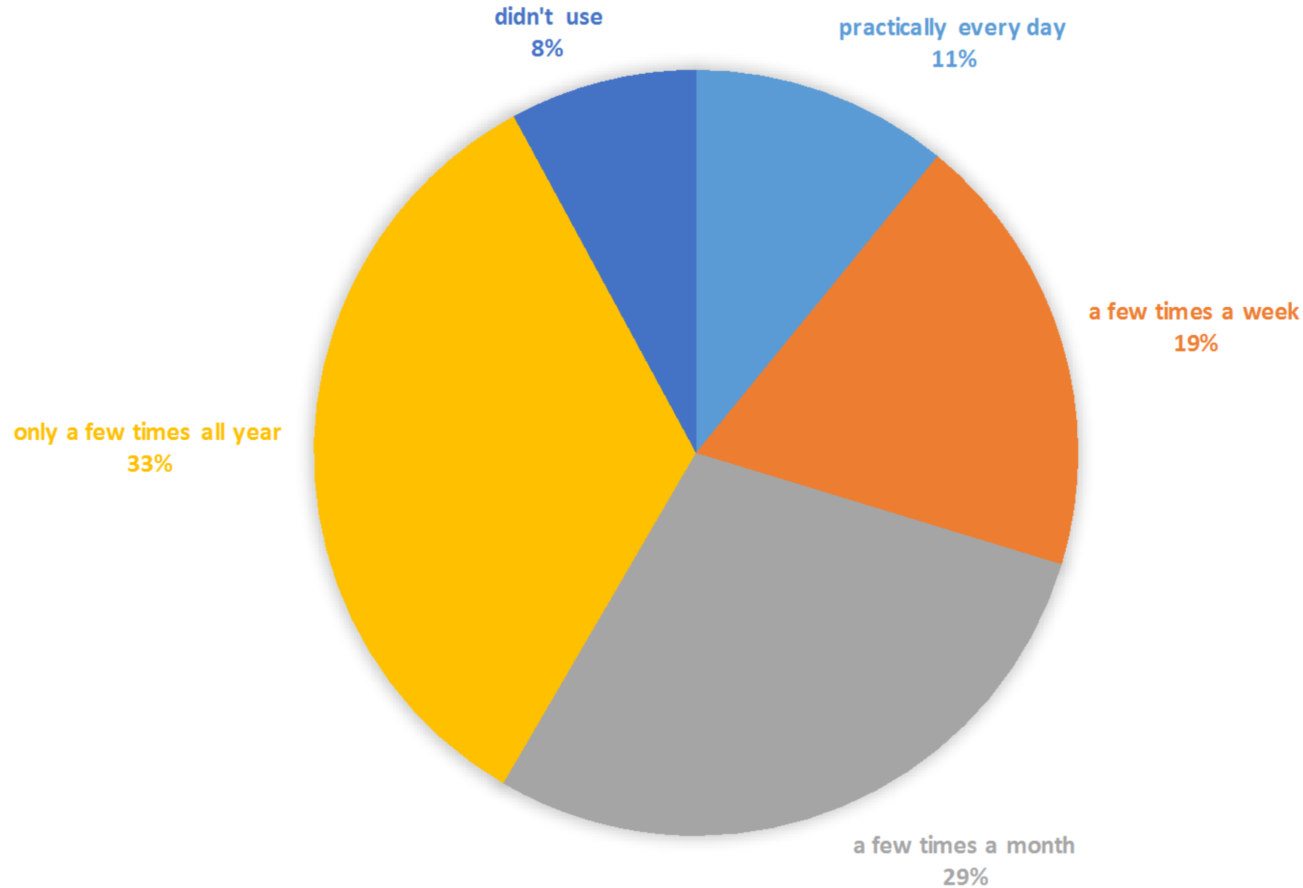
Ghana
Australia
Romania
Trinidad & Tobago
Saudi Arabia
Afghanistan
Belgium
Greece
Angola
Tanzania
Poland
India
13%
Bangladesh
Colombia
Macedonia
Turkey
Russia
Iraq
Argentina
Italy
Norway
Switzerland
Venezuela
UK
Netherlands
Croatia
Indonesia
Philippines
Japan
Peru
Brazil
Viet Nam
Ecuador
Mexico
Nigeria
France
Thailand
Algeria
Egypt
Portugal
4%
Canada
4%
Germany
4%
Spain
6%
Iran
9%
United States
10%
China
11%

# Year of First MATPOWER Use

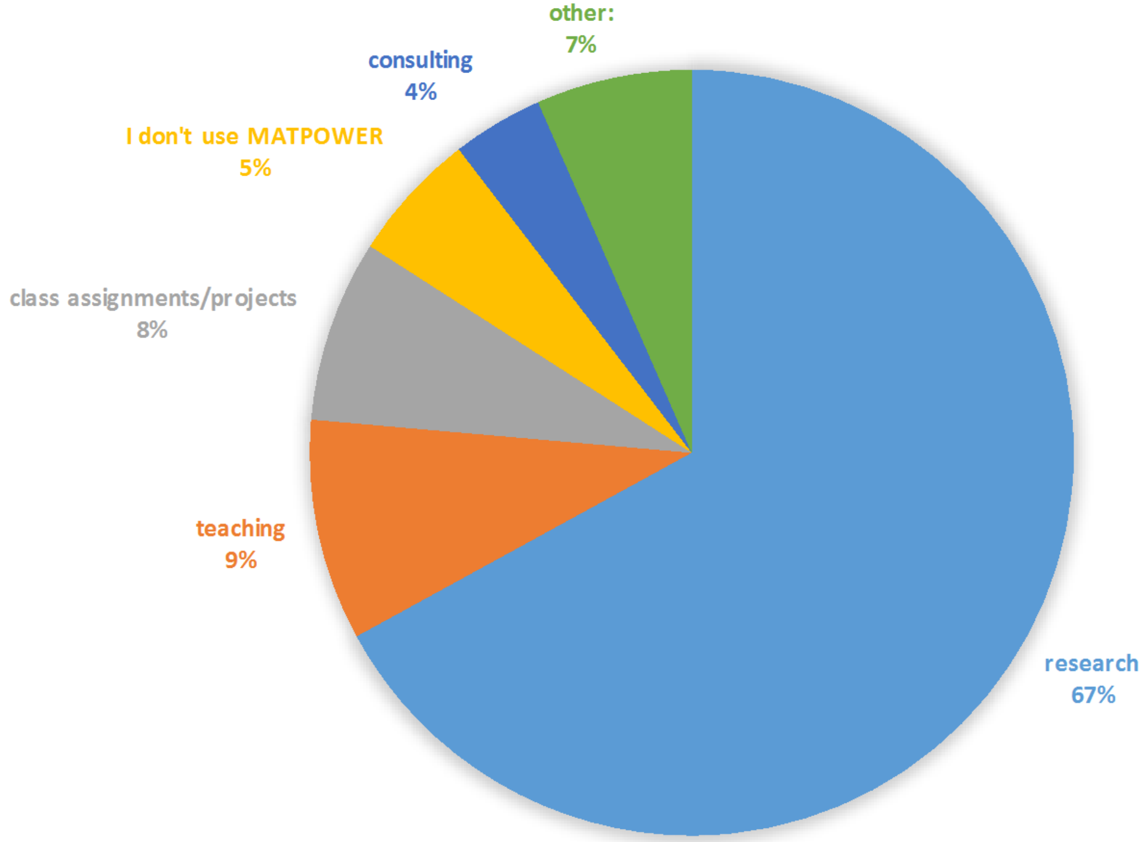


# How often do you use MATPOWER?

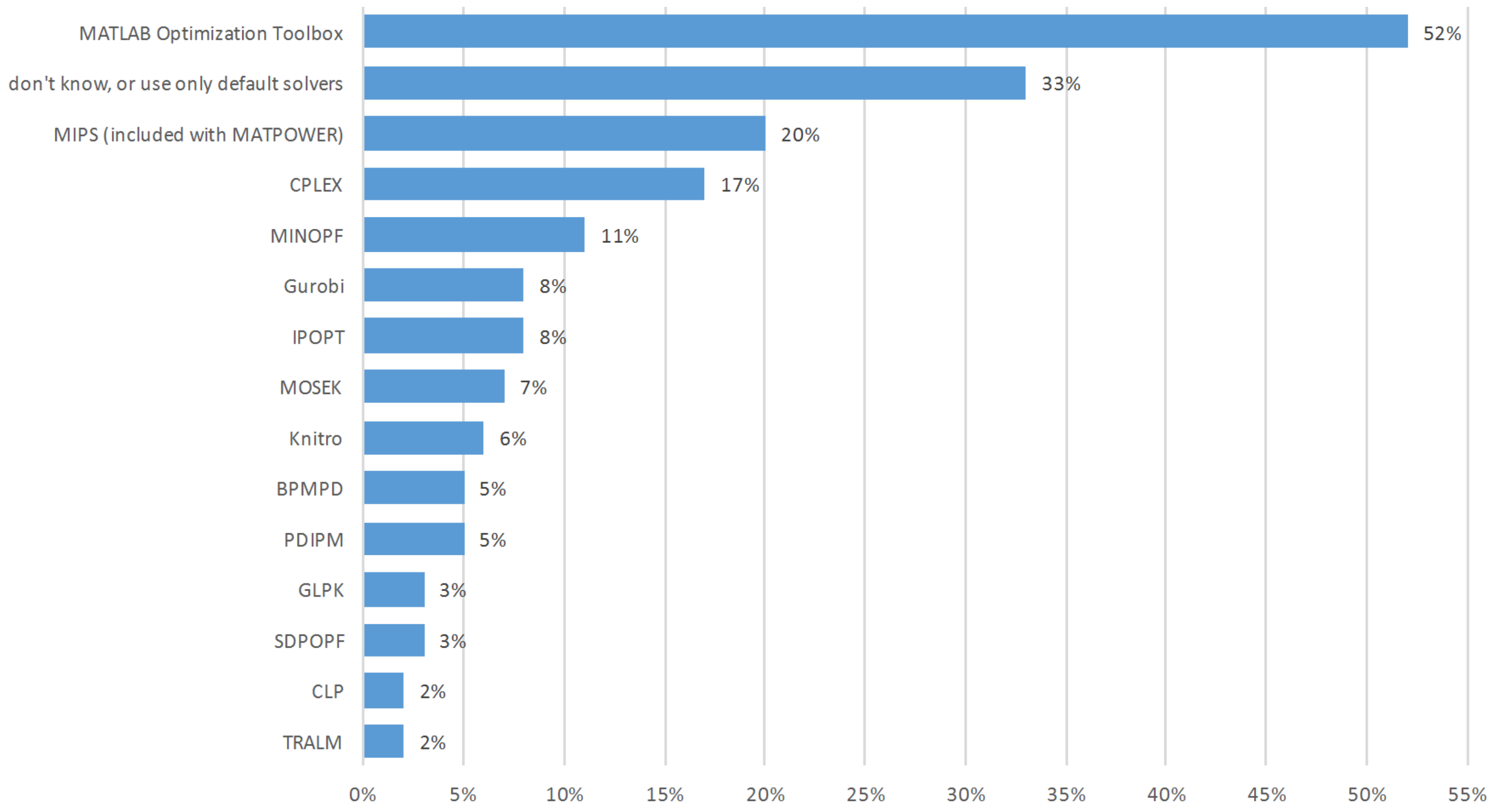
*on average in the last year*



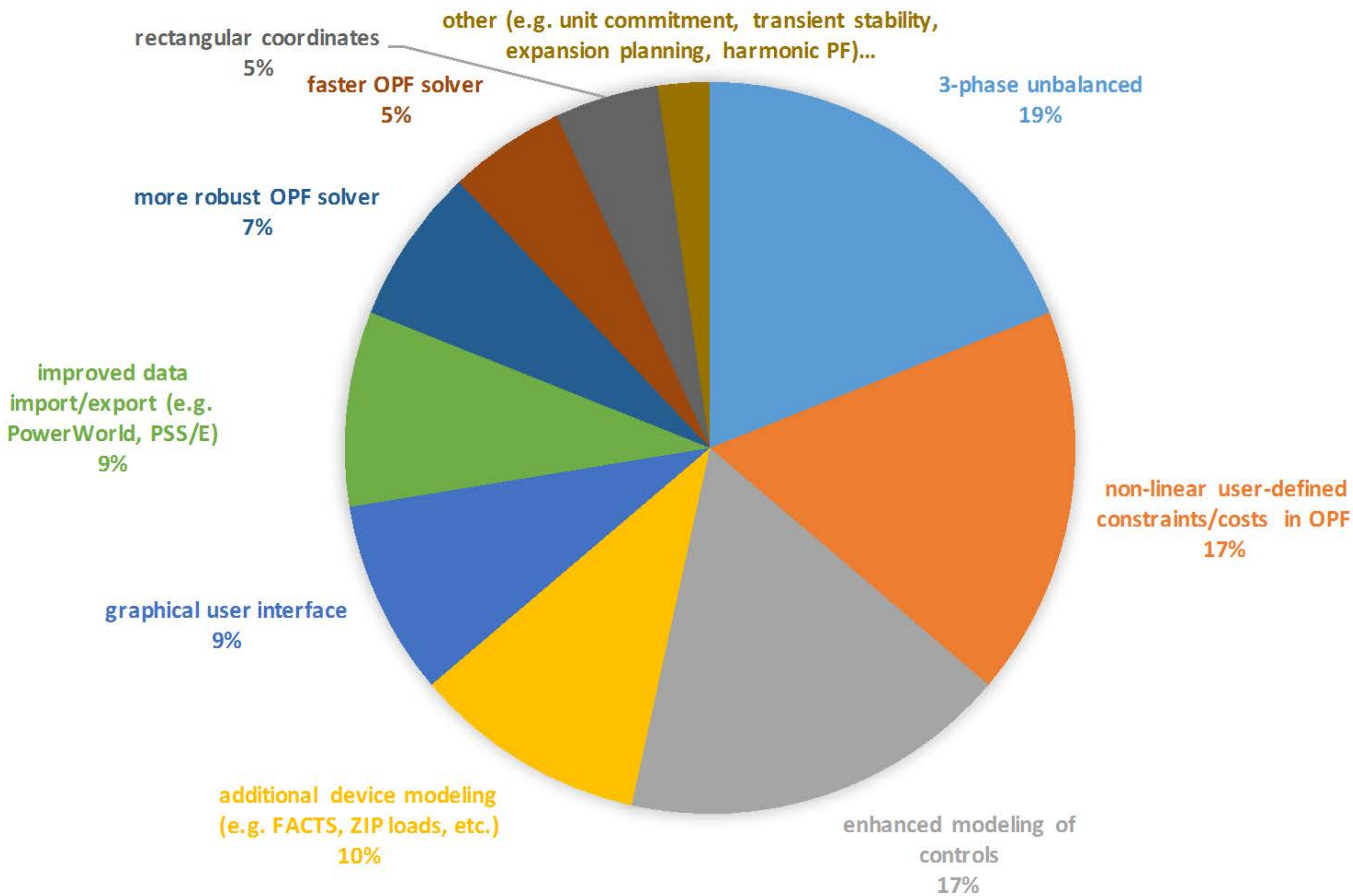
# Primary Use of MATPOWER



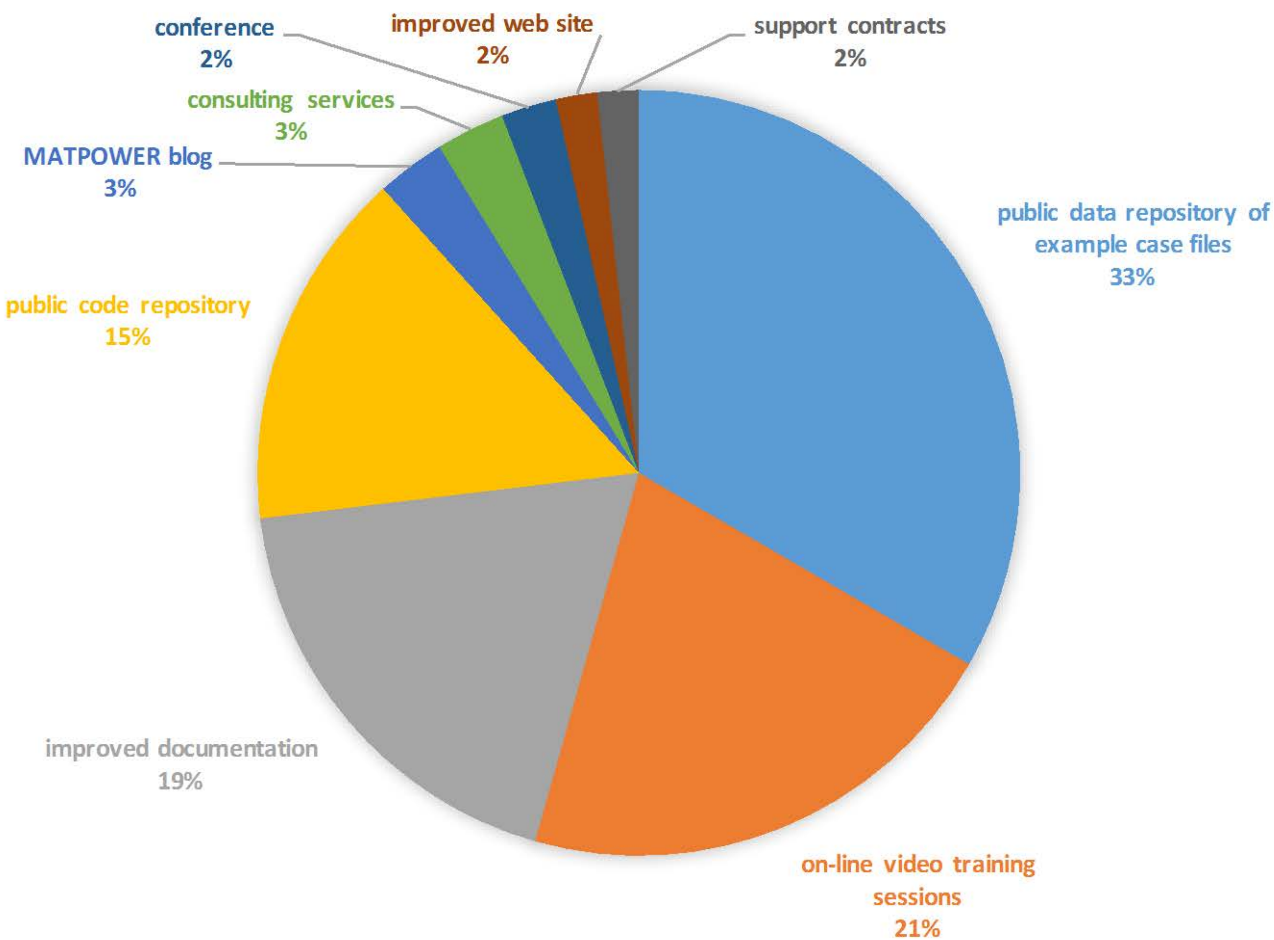
# Which solvers have you used?



# Most Valuable **Software** Enhancement



# Most Valuable **Community** Enhancement



# Collaboration

- ~75 respondents indicated interest in contributing to MATPOWER project in one of the following:
  - MATPOWER software development
  - MATPOWER community development
  - part of team to help shape direction of project/community

# Quotes

*"It allows student to see, explore and modify well-written code for power flow analysis and OPF."*

*"MATPOWER has been an integral part of my work over the last 10 years. It provided me the building blocks (power flow, optimal power flow) for my application codes that drastically reduced the development time. Its ease of use (well-defined and designed API), optimized implementation, and fast user support are MATPOWER attractive features."*

*"We don't have to reinvent the wheel with OPF … just do our stuff."*

*"MATPOWER's greatest contribution was making it available as Open Source."*

# Outline

- Overview
- MATPOWER 6 and MOST
- Testing MOST
- Future Directions for MATPOWER
  - MATPOWER Survey Results
  - NSF Proposal

# Future MATPOWER Directions

- Synthetic Data for Power Grid R&D
  - ARPA-E GRID DATA grant with UIUC, ASU, VCU *(awarded)*
  - incorporate code to generate realistic cases of any size on the fly
- Foundations for MATPOWER as an Extensible Tool for Power Systems Research and Education
  - NSF $SI^2$ (Software Infrastructure for Sustained Innovation) proposal
    - *submitted in April 2016*
  - transition MATPOWER to open collaborative development paradigm
    - public code repository, multiple committers
    - public bug tracking facility, user/developer forums, improved web-site
    - core project documents defining project goals, policies, how to contribute
    - public "wish list" of well-defined project descriptions for would-be contributors
  - redesign MATPOWER internals as foundation for:
    - fully user-customizable non-linear constraints, costs
    - fully modular construction of power flow equations
    - new device models (FACTS, three-winding transformers, etc.)
    - new controls (transformer taps, phase shifters, switched shunts, remote V reg)
    - 3-phase unbalanced modeling for distribution systems

# Current Architecture

- Current power balance equations

$$g(x) = g\left(\begin{bmatrix} x_v \\ x_p \end{bmatrix}\right) = g_y(x_v) + g_p(x_p) + g_d = 0$$

  - nonlinear function of bus voltages $x_v$
  - trivial linear function of generator power injections $x_p$, accumulating injections by bus
  - constant power loads
- Limitations
  - currently $g_y(x_v)$, $g_p(x_p)$ and derivatives are hard-coded, with no facility to modify
  - no facility to add non-linear user-defined constraints

# Proposed Modular Architecture

- Proposed power balance equations

$$g(x) = \sum_k g_k(x) = 0$$

  - each type of network element provides its own function $g_k(x)$ and derivatives
  - standard network elements and user-defined network elements enter formulation identically
  - similar structure for inequality constraints and costs
  - allows arbitrary user-defined customizations to PF and OPF problem

# Proposed Timeline

- Year 1
  - set up public code repository, public issue tracker
  - begin forming core team of developers from survey respondents indicating interest
  - add user-defined non-linear constraints, costs to OPF
  - adapt standard constraints to new mechanism
- Year 2
  - establish public projects page
  - create developer guide
  - unified, modular architecture for modeling & customizing PF, CPF, OPF, including general handling of control variables
  - begin creating public project descriptions for new device types, new controls
- Year 3
  - extension of modular architecture to three-phase unbalanced modeling
  - public project descriptions for unbalanced three-phase devices and controls

Questions?