New Software Stack for Power Systems Modeling, Optimization, and Analysis

Adam Wigington
Sr. Project Scientist

FERC Technical Conference

to Increase Real-Time and Day-Ahead Market Efficiency Through Improved Software

Washington DC

June 26, 2018

Contents

My Goal for the talk is have a few of you clone, fork the OS packages

Overview

Reliability and Efficiency Demands
Good Computational Software

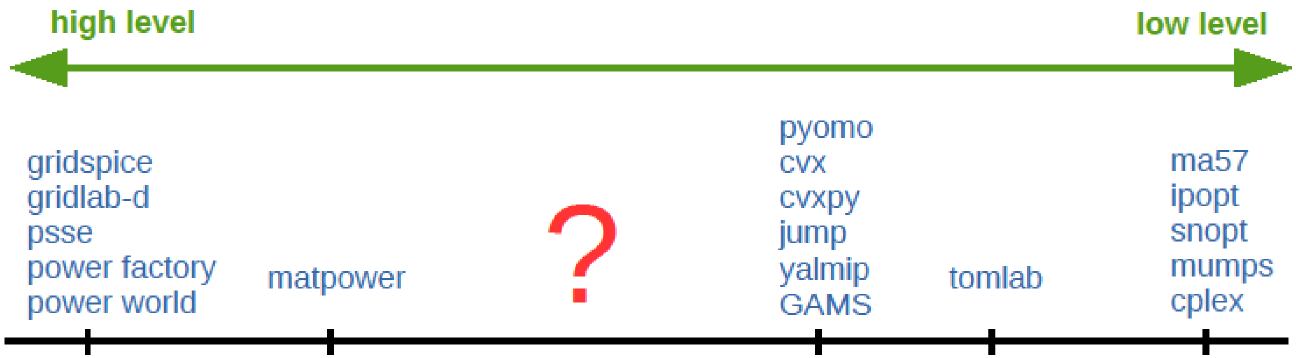
- There are good power system software applications

  → but can be difficult to modify and extend

- There are good optimization software packages

  → but can be difficult to apply for specific fields

Software Spectrum



- **Goals of new software stack**
  - Power system approachable
  - Open platform for sharing and testing new optimization ideas
  - Not just for toy cases (e.g., readily handles 60k+ bus cases)
  - Avoid repeated efforts
  - Fill gap in spectrum
  - Speed up time from research idea to industry applications

ELECTRIC POWER RESEARCH INSTITUTE

History and Contributors

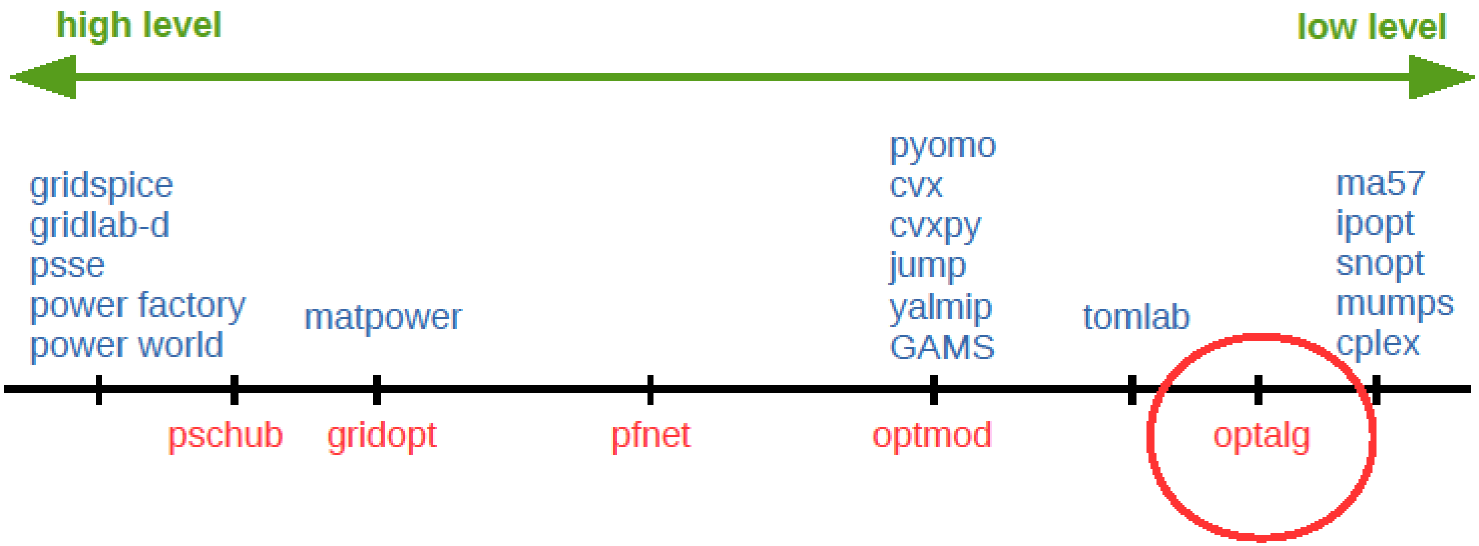
- Tomas Tinoco de Rubira (ETH Zurich)
  - Began software stack as a hobby
    - (the mastermind behind it)
  - Some algorithms based on PhD work at Stanford and EPRI

- Contributors
  - Martin Baltzinger (ETH)
  - Robert Entriken (EPRI)
  - Nick Henderson (EPRI, formerly)
  - Stavros Karagiannopoulos (ETH)
  - Dmitry Shchetinin (ETH)
  - Adam Wigington (EPRI)
  - Martin Zellner (ETH)

Open Source Software Stack

OPTALG Package

- Optimization solvers and interfaces to external solvers



EPRI ELECTRIC POWER RESEARCH INSTITUTE

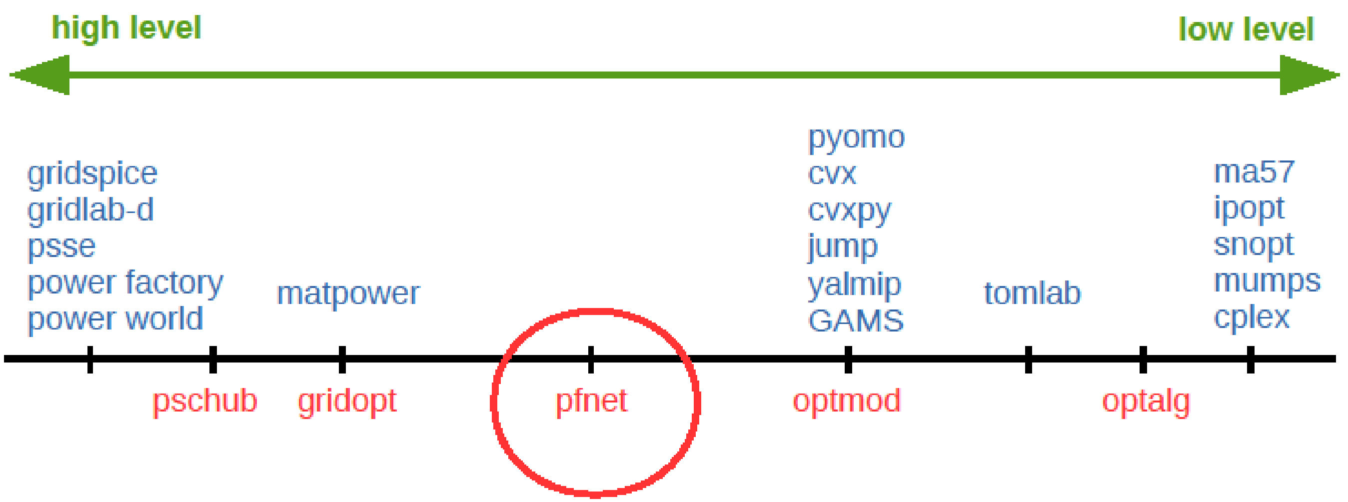
OPTALG Package

- Pure Python
- Optimization Algorithms:
  - Newton-Raphson
  - Interior-Point Quadratic Program
  - Augmented Lagrangian
- Interfaces:
  - IPOPT (interior-point nonlinear)
  - CLP (linear programming)
  - CBC (mixed-integer)
- Linear solver interfaces:
  - SuperLU (scipy)
  - mumps
- General form:

$$
\begin{array}{lll}
\text{minimize} & \varphi(x) & \\
\text{subject to} & Ax = b & : \lambda \\
 & f(x) = 0. & : \nu \\
 & l \leq x \leq u & : \pi, \mu \\
 & Px \in \{0,1\}^m, &
\end{array}
$$

PFNET Package

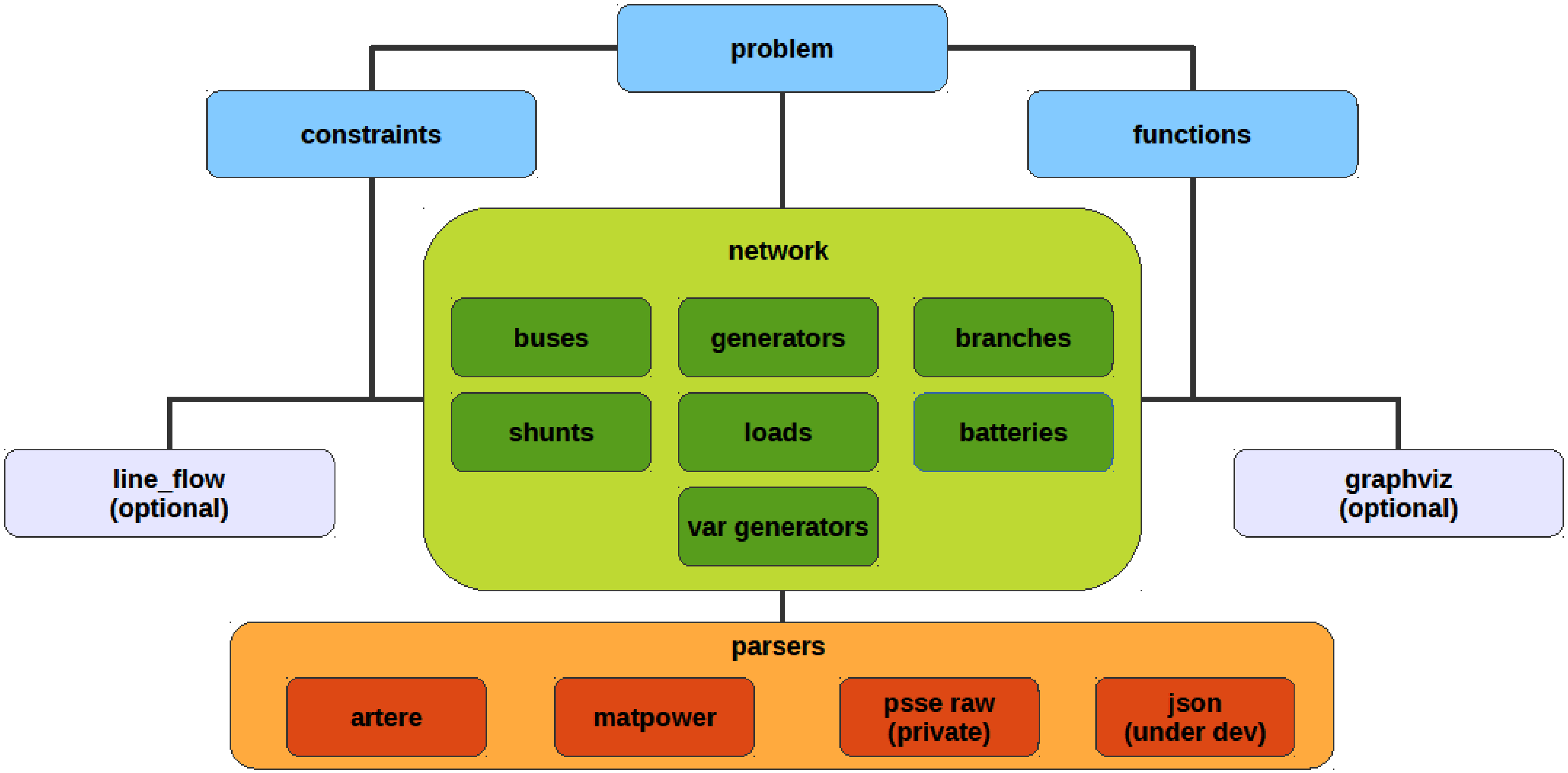
- Link between power system modeling and optimization problem formulation



PFNET Package

- C for numerical efficiency
  - Python wrapper
- Parsers
- Construct problems
  - (Problems get passed to solvers OPTALG)
  - Consists of
    - Variables
      - E.g., bus voltage mag, bus voltage angles, gen mvar powers
    - Objective function components (11 built-in)
      - E.g., generation cost, voltage mag regularization, tap regularization
    - Constraints (16 built-in)
      - E.g., AC power balance, gen mw participation, gen voltage regularization
  - Extensible, define your own!

PFNET Architecture



PFNET Example – Simple Newton-Raphson Solver

```python
def NRsolve(net):

    net.clear_flags()

    # bus voltage angles
    net.set_flags('bus',
                  'variable',
                  'not slack',
                  'voltage angle')

    # bus voltage magnitudes
    net.set_flags('bus',
                  'variable',
                  'not regulated by generator',
                  'voltage magnitude')

    # slack gens active powers
    net.set_flags('generator',
                  'variable',
                  'slack',
                  'active power')

    # regulator gens reactive powers
    net.set_flags('generator',
                  'variable',
                  'regulator',
                  'reactive power')

    p = pfnet.Problem(net)
    p.add_constraint(pfnet.Constraint('AC power balance',net))
    p.add_constraint(pfnet.Constraint('generator active power participation',net))
    p.add_constraint(pfnet.Constraint('generator reactive power participation',net))
    p.analyze()

    x = p.get_init_point()
    p.eval(x)

    residual = lambda x: hstack((p.A*x-p.b,p.f))

    while norm(residual(x)) > 1e-4:
        x = x + spsolve(bmat([[p.A],[p.J]],format='csr'),-residual(x))
        p.eval(x)

    net.set_var_values(x)
    net.update_properties()
```

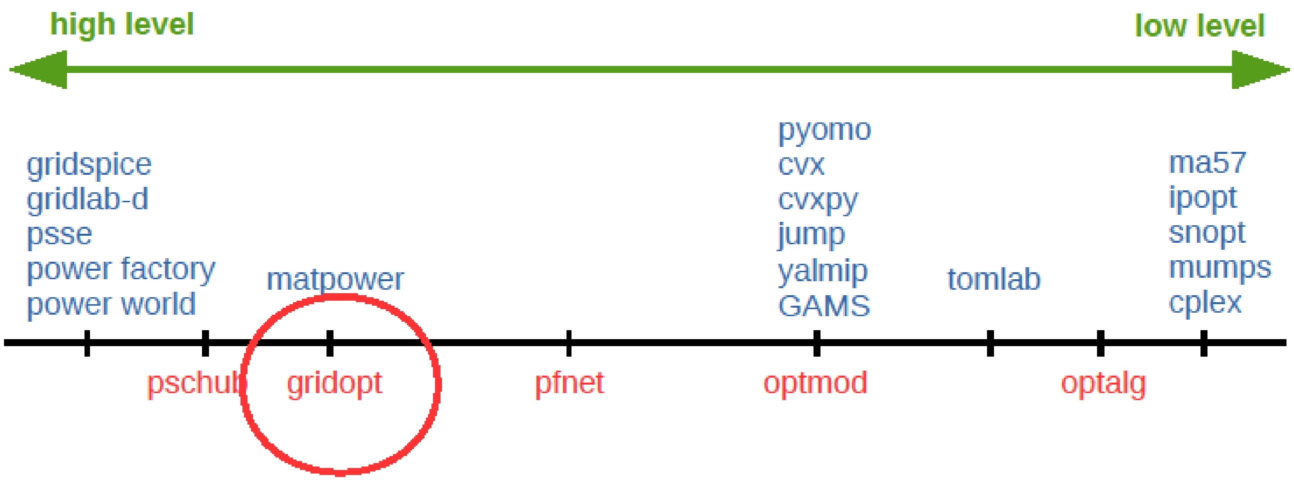
Define variables

Add constraints

Calculations

GRIDOPT Package

- Link between PFNET and OPTALG with ready built power flow and optimal power flow implementations



EPRI | ELECTRIC POWER RESEARCH INSTITUTE

GRIDOPT Package

- Pure Python

- Convenient methods
  - Formulate problems with PFNET
  - Solve with OPTALG

- Power flows
  - DC
  - Newton-Raphson w/ heuristics
  - Augmented Lagrangian

- Optimal power flows
  - DCOPF
  - Augmented Lagrangian
  - IPOPT wrapper

```
>>> import pfnet
>>> import gridopt

>>> net = pfnet.ParserMAT().parse('ieee14.mat')

>>> # max mismatches (MW,MVAr)
>>> print '%.2e %.2e' %(net.bus_P_mis,net.bus_Q_mis)
3.54e-01 4.22e+00

>>> method = gridopt.power_flow.new_method('NRPF')

>>> method.set_parameters({'quiet': True})

>>> method.solve(net)

>>> results = method.get_results()

>>> print results['status']
solved
```

Augmented Lagrangian Method

- Robust to ill-conditioned Jacobians

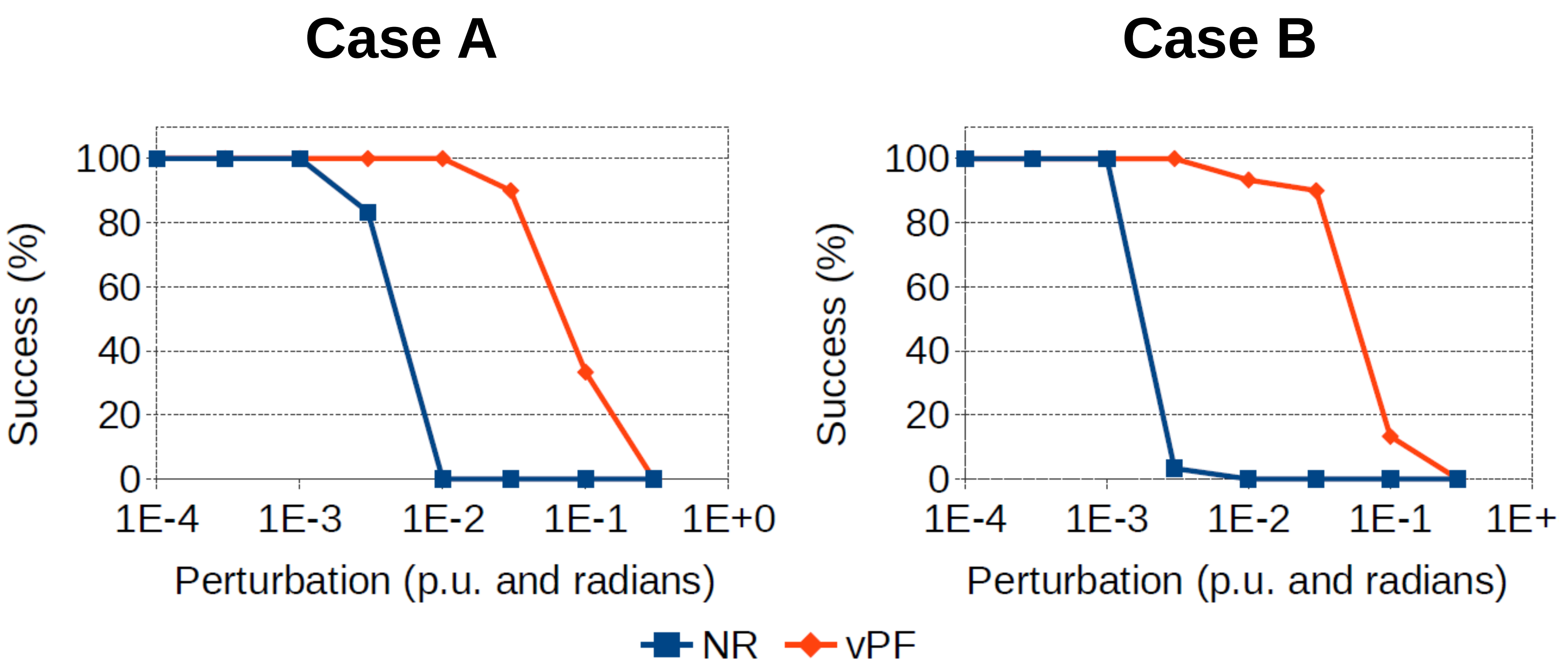
- Complementarity constraints instead of heuristics for PV-PQ switching

- If it does not solve PF equations, it still converges to a minimum and will provide sensitivities that can help

- General form

$$
\begin{array}{lll}
\text{minimize} & \varphi(x) & \\
\text{subject to} & Ax = b & : \lambda \\
 & f(x) = 0 & : \nu \\
 & l \le x \le u. & : \pi, \mu
\end{array}
$$

Augmented Lagrangian Method



Random perturbations of starting point for all variables

Early version of Augmented Lagrangian (vPF) more robust to poor starting points

EPRI Extensions

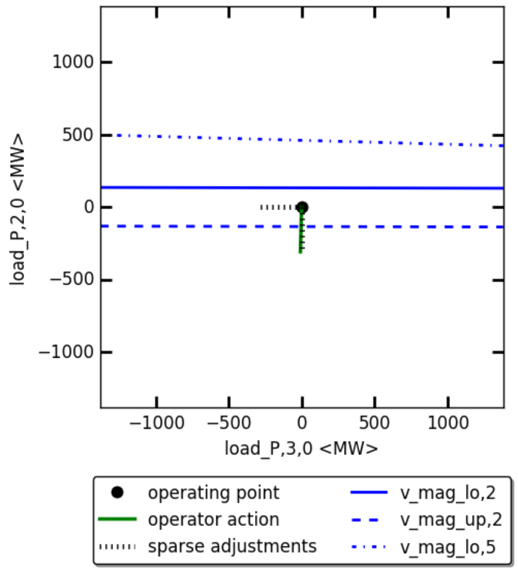
EPRI Extensions to Software Stack

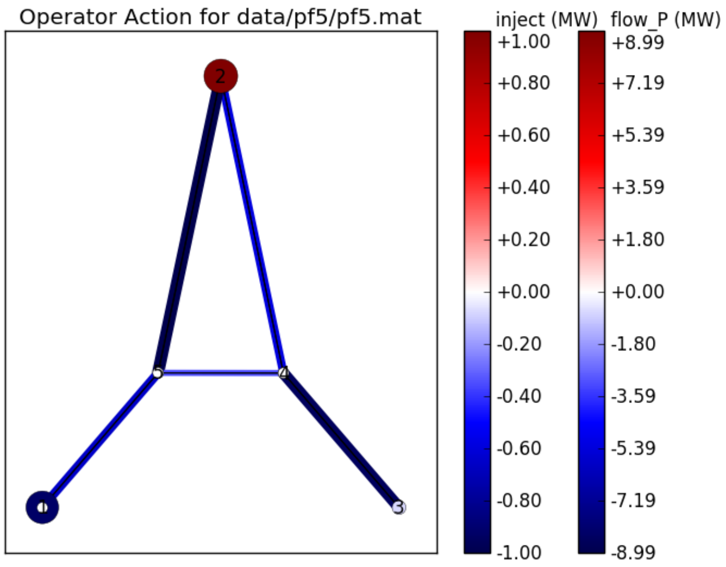
- Practical tools
  - Tracking, naming cases
  - Comparison of networks and results
- Contingency analysis
- Critical operating boundaries
- New Functions and Constraints
  - Interface flows
  - Minimize losses*
  - Voltage control areas*

*TODO

Contribute back to OS tools when appropriate (eg. modeling limitations)

Critical Operating Boundaries





Identify voltage and thermal limits of most concern 1D or 2D Recommended Actions

ELECTRIC POWER RESEARCH INSTITUTE

Getting Started

How to Get Started

- Clone, fork the OS repositories
  - PFNET https://github.com/ttinoco/PFNET
  - OPTALG https://github.com/ttinoco/OPTALG
  - GRIDOPT https://github.com/ttinoco/GRIDOPT
- PFNET building and installing
  - Builds using Autotools for Unix-like system
  - Cmake builds for Windows (tested using mingw) *coming soon*
- Python is easy
  - pypi *coming soon*
- *To come – Unit Commitment, Json file format, ???*

- **or just clone PSCHUB https://github.com/ttinoco/PSCHUB**
  - **JupyterHub docker containe**r

Become a Contributor!

Become a Contributor to the OS Stack

- Find bugs
- Improve documentation
- Add modeling capability
  - E.g., Power flow controllers
- Create new Parsers
- Create new Functions
- Create new Constraints
- Add wrappers to other solvers
- ...

Become a Contributor!

Together…Shaping the Future of Electricity

EPRI ELECTRIC POWER RESEARCH INSTITUTE