MATPOWER's Extensible Optimal Power Flow Architecture

Ray Zimmerman, Cornell University
Carlos Murillo-Sánchez, Universidad Autonoma de Manizales
Robert J. Thomas, Cornell University

IEEE PES General Meeting 2009
July 30, 2009
Calgary, Alberta, Canada

CERTS
Consortium for Electric Reliability Technology Solutions

Cornell University

PSERC

Outline

- MATPOWER Overview
- Extensible OPF Formulation
- Standard Extensions
- Software Architecture
- Example: Adding Reserves

Outline

- MATPOWER Overview
  - What does MATPOWER do?
  - MATPOWER History
  - MATPOWER Package
- Extensible OPF Formulation
- Standard Extensions
- Software Architecture
- Example: Adding Reserves

What does MATPOWER do?

- DC power flow
- AC power flow
  - Newton
  - Gauss-Seidel
  - Fast decoupled
- functions to compute ...
  - derivatives of power flow equations
  - generation costs
  - linear shift factors (PTDFs, LODFs)
- DC optimal power flow (OPF)
  - BPMPD (MEX)
  - Primal-Dual Interior Point Method (PDIPM)
- AC optimal power flow (OPF)
  - Primal-Dual Interior Point Method (PDIPM) (pure Matlab & MEX)
  - MINOS (MEX)
  - successive LP's (BPMPD MEX)
  - Optimization Toolbox (fmincon, constr)

MATPOWER History

1st work with Matlab power flow code for PowerWeb
- based on code from Joe Chow & Chris DeMarco

1st PF and OPF code of my own
- based on Opt Tbx, constr()

1st public **MATPOWER** release
- not widely publicized
- PWL costs

MATPOWER 1.0
- in-house successive LP-based OPF

MATPOWER 2.0
- fast decoupled PF
- successive LP-based OPF
- options vector
- User's Manual

MATPOWER 3.0
- MINOS-based OPF (gen. form.)
- fmincon-based OPF
- DC PF & OPF
- multiple gens/bus
- Gauss-Seidel PF
- improved DP de-commitment
- automated tests
- separate disp. load output section
- option for active power line lims
- option to enforce Q_g lims in PF

MATPOWER 3.2
- version 2 case format
- gen capability curves
- branch angle diff lims
- PTDFs
- TSPOPF

MATPOWER 4.0
- refactored OPF (all using gen. form.)
- pure-Matlab PDIPM solver
- OPF with reserves
- userfcn callbacks
- multiple solvers for DC OPF
- LODFs
- support for interior point fmincon

1996 | 1997 | 1998 | 1999 | 2000 | 2001 | 2002 | 2003 | 2004 | 2005 | 2006 | 2007 | 2008 | 2009

MATPOWER Package

- Open source Matlab code available at: http://www.pserc.cornell.edu/matpower/
- No GUI (graphical user interface)
- Set of functions you can run from Matlab command line or include in your own programs
- Example:

```
>> result = runopf('case300');
```

or

```
>> mpc = loadcase('case300');
>> mpc.bus = scale_load(1.1, mpc.bus);
>> result = runopf(mpc);
```

- Primary focus on research and education applications

Outline

- MATPOWER Overview
- Extensible OPF Formulation
  - Standard Formulation
  - Generalized Formulation
  - User-Defined Costs
  - User-Defined Constraints
- Standard Extensions
- Software Architecture
- Example: Adding Reserves

Standard OPF Formulation

$$\min_{x} f(x)$$

subject to

$$g(x) = 0$$

$$h(x) \leq 0$$

$$x_{\min} \leq x \leq x_{\max}$$

Standard OPF Formulation

$$\min_{\Theta,V,P,Q} \sum_{i=1}^{n_g} \left[f_P^i(p_i) + f_Q^i(q_i) \right]$$

subject to

$$g_P(\Theta, V, P) = 0$$

$$g_Q(\Theta, V, Q) = 0$$

$$h_f(\Theta, V) \leq 0$$

$$h_t(\Theta, V) \leq 0$$

$$\theta_{\text{ref}} \leq \theta_i \leq \theta_{\text{ref}}, \qquad i = i_{\text{ref}}$$

$$v_i^{\min} \leq v_i \leq v_i^{\max}, \qquad i = 1 \ldots n_b$$

$$p_i^{\min} \leq p_i \leq p_i^{\max}, \qquad i = 1 \ldots n_g$$

$$q_i^{\min} \leq q_i \leq q_i^{\max}, \qquad i = 1 \ldots n_g$$

Generalized Formulation

$$\min_{x,z} f(x) + f_u(x,z)$$

subject to

$$g(x) = 0$$

$$h(x) \leq 0$$

$$x_{\min} \leq x \leq x_{\max}$$

$$l \leq A \begin{bmatrix} x \\ z \end{bmatrix} \leq u$$

$$z_{\min} \leq z \leq z_{\max}$$

Generalized Formulation

$$\min_{x,z} f(x) + f_u(x,z)$$

subject to

$$g(x) = 0$$

$$h(x) \leq 0$$

$$x_{\min} \leq x \leq x_{\max}$$

$$l \leq A \begin{bmatrix} x \\ z \end{bmatrix} \leq u$$

$$z_{\min} \leq z \leq z_{\max}$$

additional variables

Generalized Formulation

$$\min_{x,z} f(x) + f_u(x,z)$$

additional costs

subject to

$$g(x) = 0$$

$$h(x) \leq 0$$

$$x_{\min} \leq x \leq x_{\max}$$

$$l \leq A \begin{bmatrix} x \\ z \end{bmatrix} \leq u$$

$$z_{\min} \leq z \leq z_{\max}$$

additional variables

Generalized Formulation

$$\min_{x,z} f(x) + f_u(x, z)$$

additional costs

subject to

$$g(x) = 0$$

$$h(x) \leq 0$$

$$x_{\min} \leq x \leq x_{\max}$$

$$l \leq A \left[\begin{array}{c} x \\ z \end{array} \right] \leq u$$

additional constraints

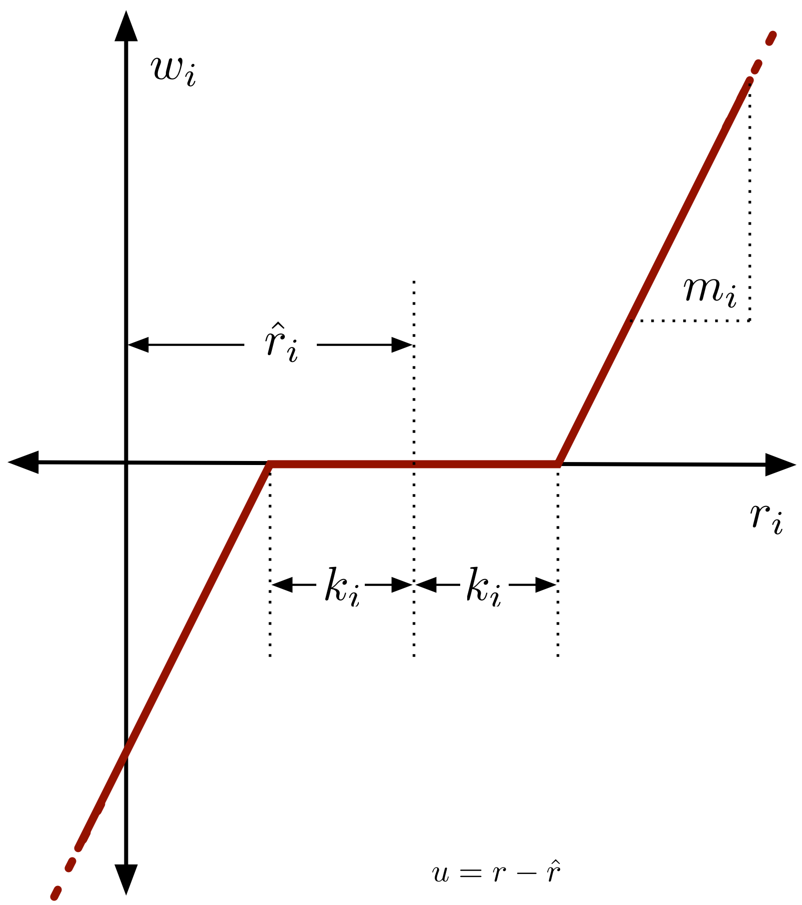
$$z_{\min} \leq z \leq z_{\max}$$

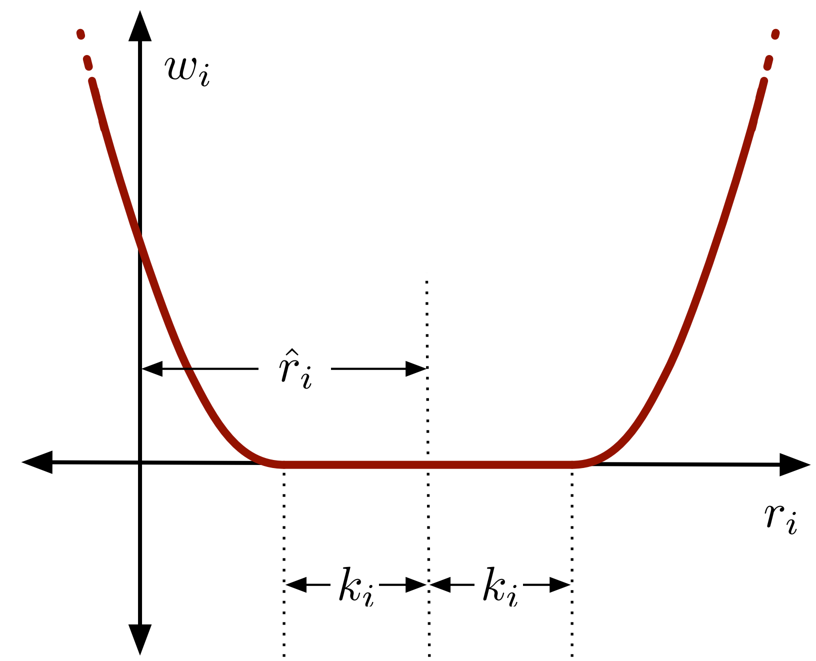
additional variables

User-Defined Costs

$$f_u(x,z) = \frac{1}{2} w^{\mathsf{T}} H w + C^{\mathsf{T}} w$$

$$r = N \begin{bmatrix} x \\ z \end{bmatrix}$$





$$u = r - \hat{r}$$

$$w_i = \begin{cases} m_i f_{d_i}(u_i + k_i), & u_i < -k_i \\ 0, & -k_i \le u_i \le k_i \\ m_i f_{d_i}(u_i - k_i), & u_i > k_i \end{cases}$$

$$f_{d_i}(\alpha) = \begin{cases} \alpha, & \text{if } d_i = 1 \\ \alpha^2, & \text{if } d_i = 2 \end{cases}$$

User-Defined Constraints

- additional linear restrictions on all optimization variables

$$l \leq A \begin{bmatrix} x \\ z \end{bmatrix} \leq u$$

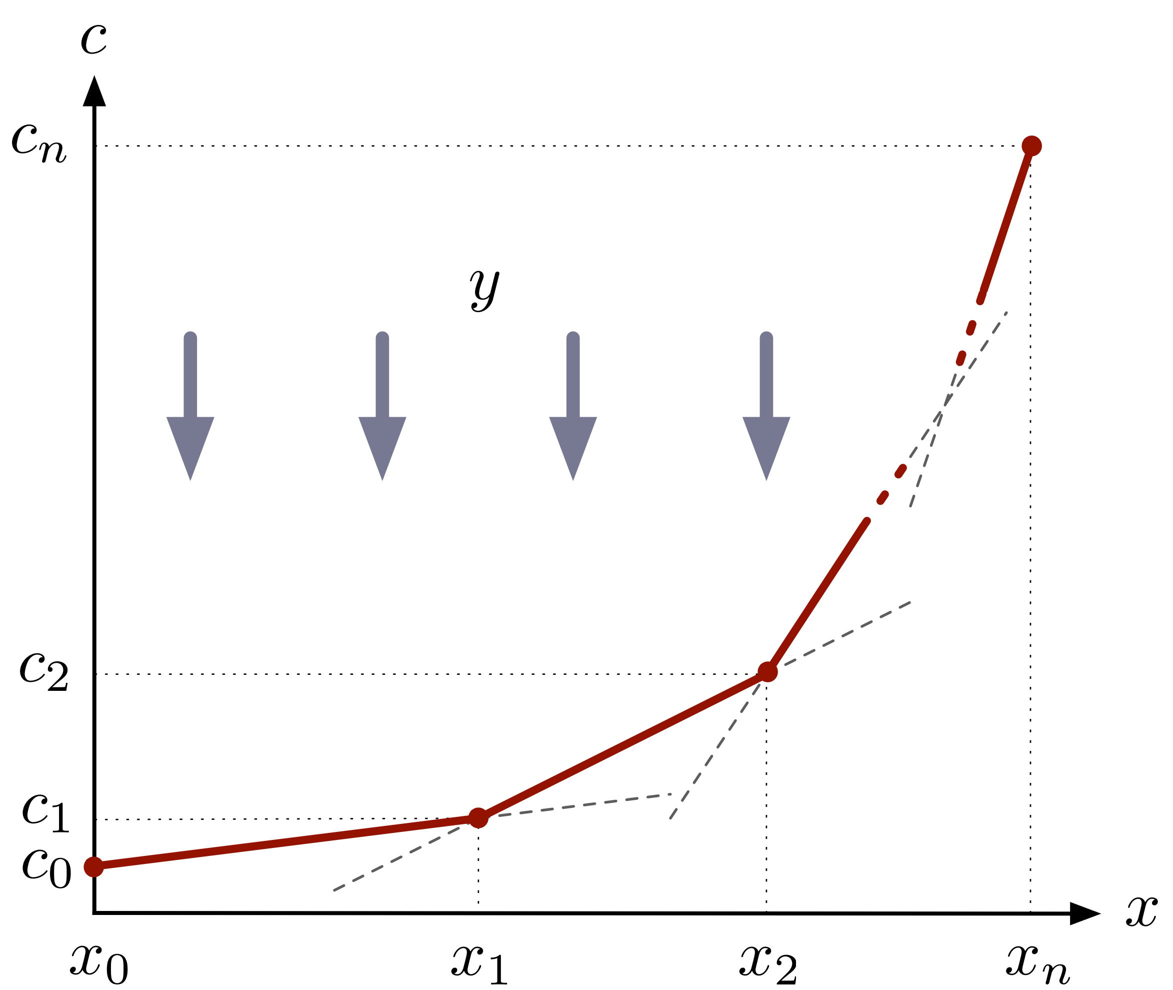
- inequality constraints
- equality constraints if $l = u$

Outline

- MATPOWER Overview
- Extensible OPF Formulation
- Standard Extensions
  - piece-wise linear costs
  - dispatchable (price sensitive) loads
  - generator reactive capability constraints
  - branch angle difference limits
- Software Architecture
- Example: Adding Reserves

Piece-wise Linear Generation Costs

$$c(x) = \begin{cases} m_1(x - x_1) + c_1, & x \le x_1 \\ m_2(x - x_2) + c_2, & x_1 < x \le x_2 \\ \vdots & \vdots \\ m_n(x - x_n) + c_n, & x_{n-1} < x \end{cases}$$



- given the sequence of points

  $$(x_j, c_j), \quad j = 0 \dots n$$

  where m_j is the slope of segment j

  $$m_j = \frac{c_j - c_{j-1}}{x_j - x_{j-1}}, \quad j = 1 \dots n$$

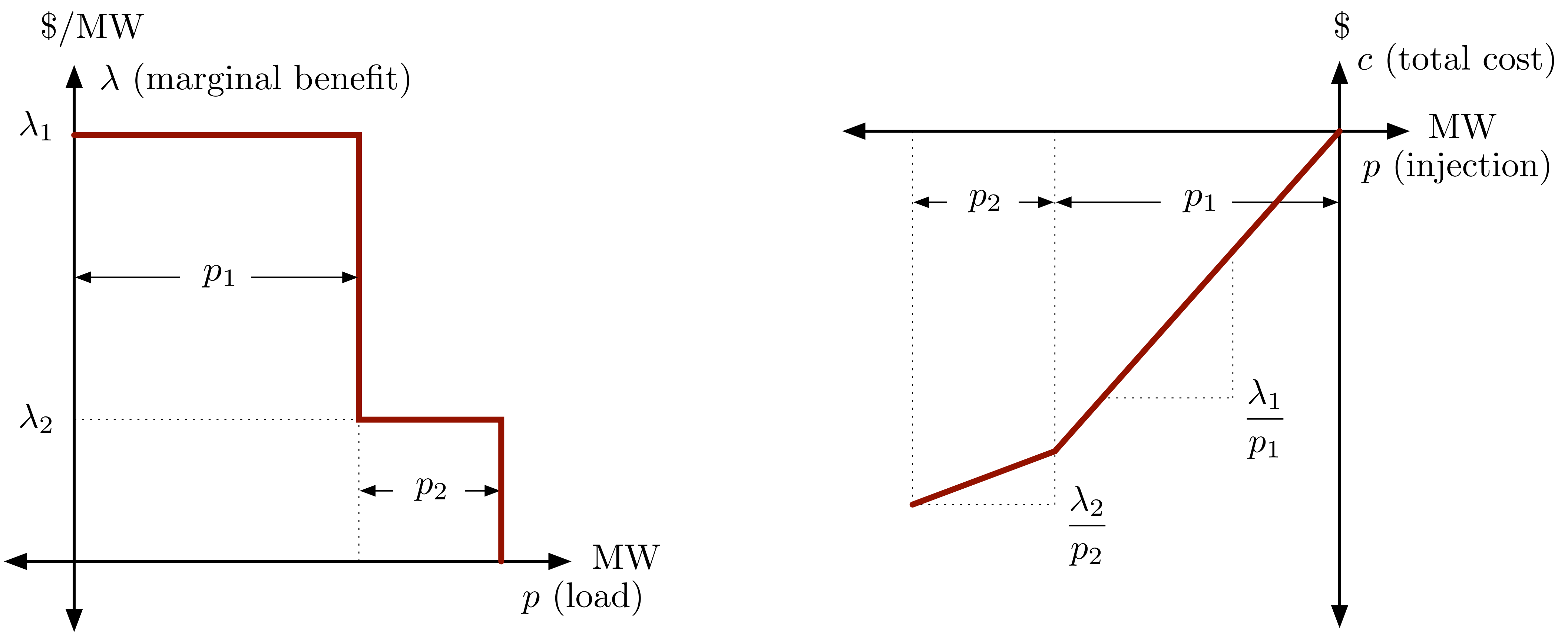
- add a new variable y and, for each segment, a new linear constraint on y

  $$y \ge m_j(x - x_j) + c_j, \quad j = 1 \dots n$$

- use y in place of $c(x)$ in the cost function

Dispatchable (price sensitive) Loads

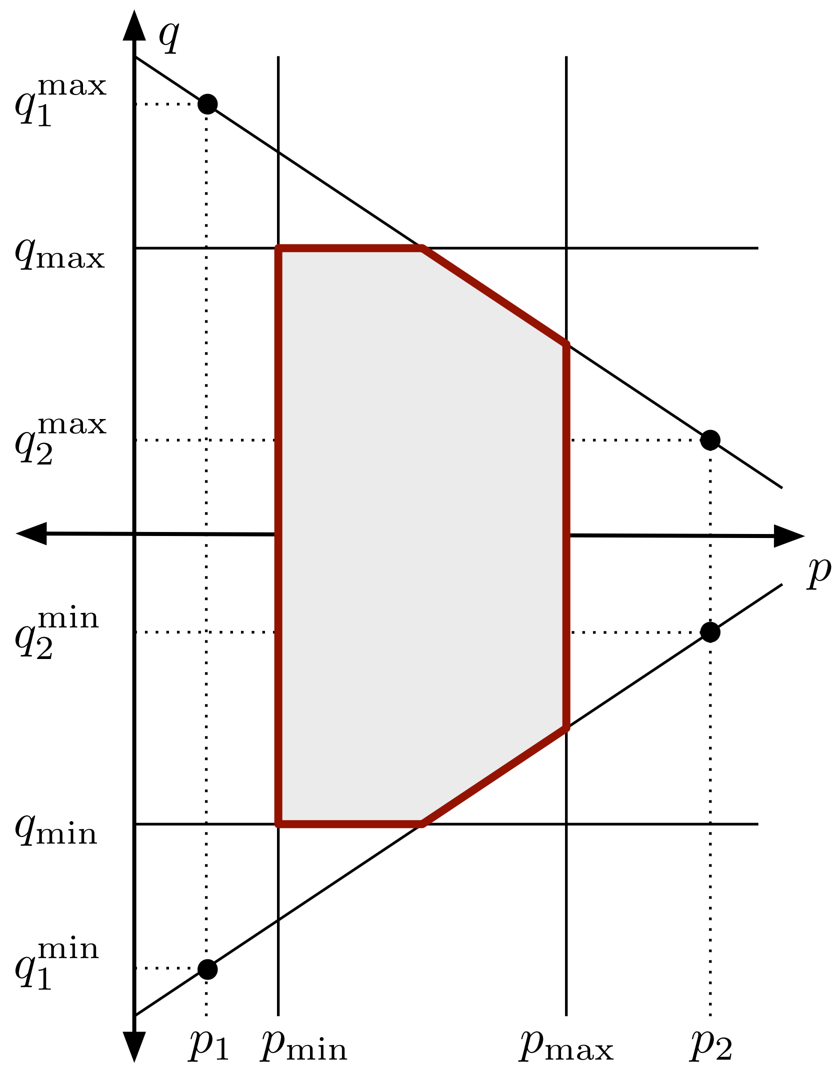
- modeled as "negative generator"



- with an additional constant power factor constraint

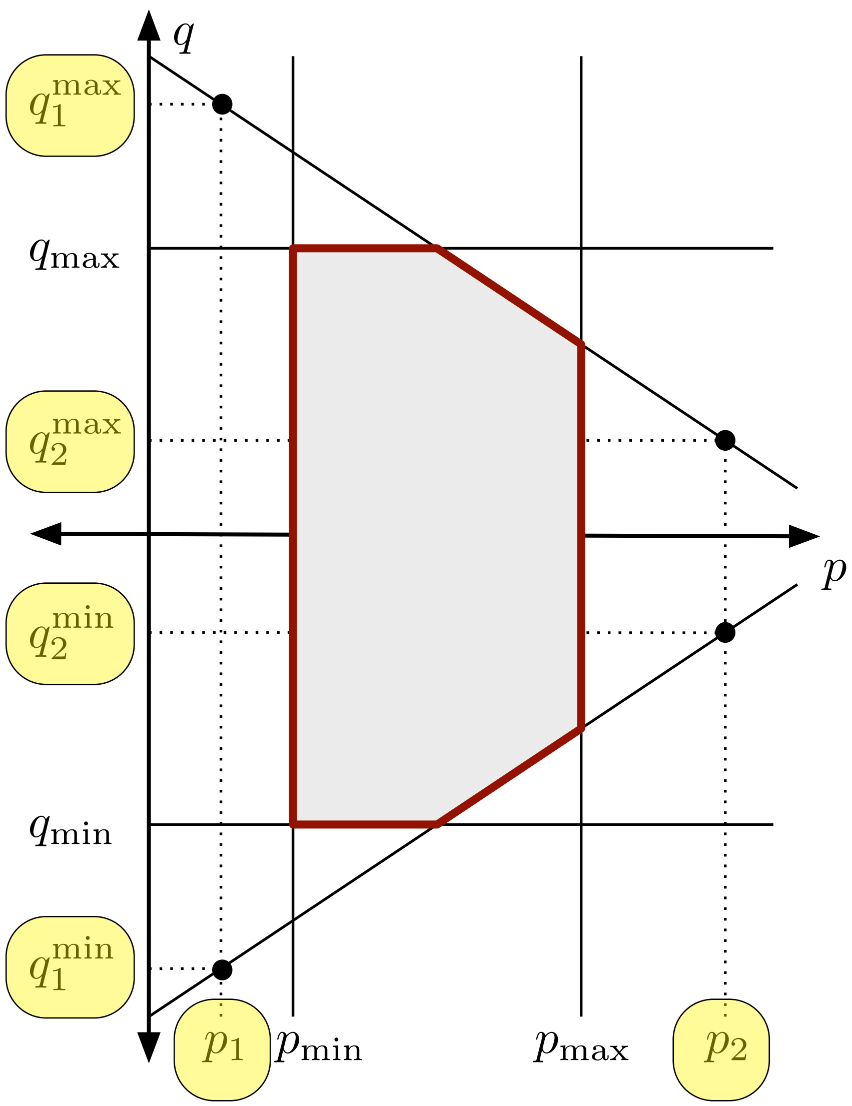
Generator Reactive Capability Constraints

- Instead of simple box constraints ...



Generator Reactive Capability Constraints

- Instead of simple box constraints ...



Outline

- MATPOWER Overview
- Extensible OPF Formulation
- Standard Extensions
- Software Architecture
  - Overview of Execution - Callbacks
  - Adding Variables
  - Adding Constraints
- Example: Adding Reserves

Overview of Execution

- load data
- convert to internal indexing
- set up problem formulation
- run optimization
- convert results back to external indexing
- print results (optional)
- save results (optional)

Overview of Execution – Callbacks

- load data
- convert to internal indexing
- set up problem formulation
- run optimization
- convert results back to external indexing
- print results (optional)
- save results (optional)

Overview of Execution – Callbacks

- load data
- convert to internal indexing
- set up problem formulation
- run optimization
- convert results back to external indexing
- print results (optional)
- save results (optional)

Modifying the Formulation

- Option 1 – externally supply complete constraint matrix *A*, cost coeff matrix *N*, etc. *(taking into account internal conversions)*
- Option 2 – modify formulation directly in a callback function

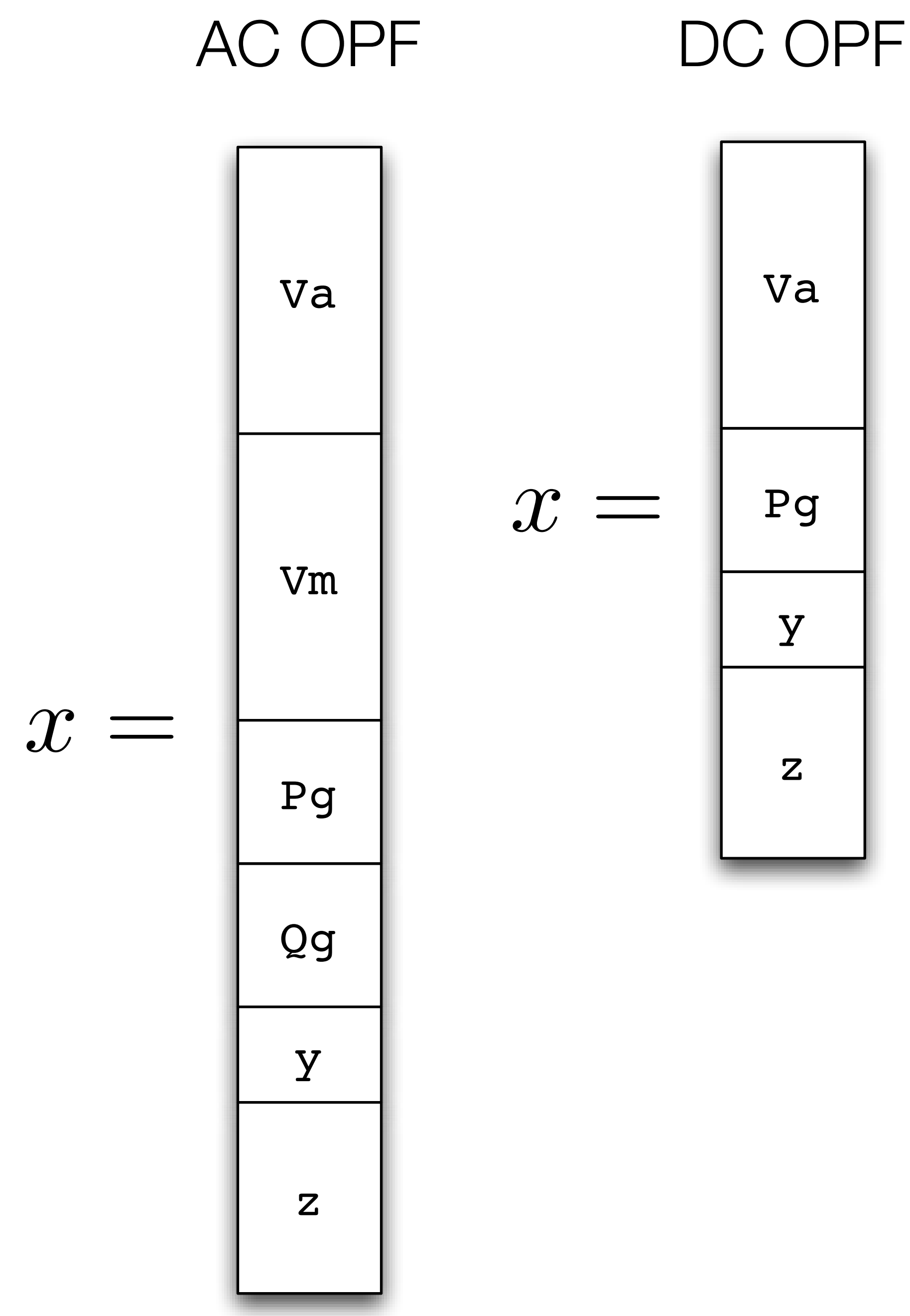
Software Architecture - Variables

- Utilizes an "OPF-Model" object (OM) to manage variable and constraint indexing
- Variables are added in named blocks, with dimension, initial value and bounds, e.g.

```
om = add_vars(om, 'Pg', ng, Pg0, Pmin, Pmax);
```

- Portions of optimization variable *x* or limit shadow prices can be accessed by name, w/o need to keep track of explicit indexing

name	description
`Va`	bus voltage angles
`Vm`	bus voltage magnitudes
`Pg`	generator real power injections
`Qg`	generator reactive power injections
`y`	CCV helper variables for pwl costs
`z`	other user defined variables



Software Architecture - Constraints

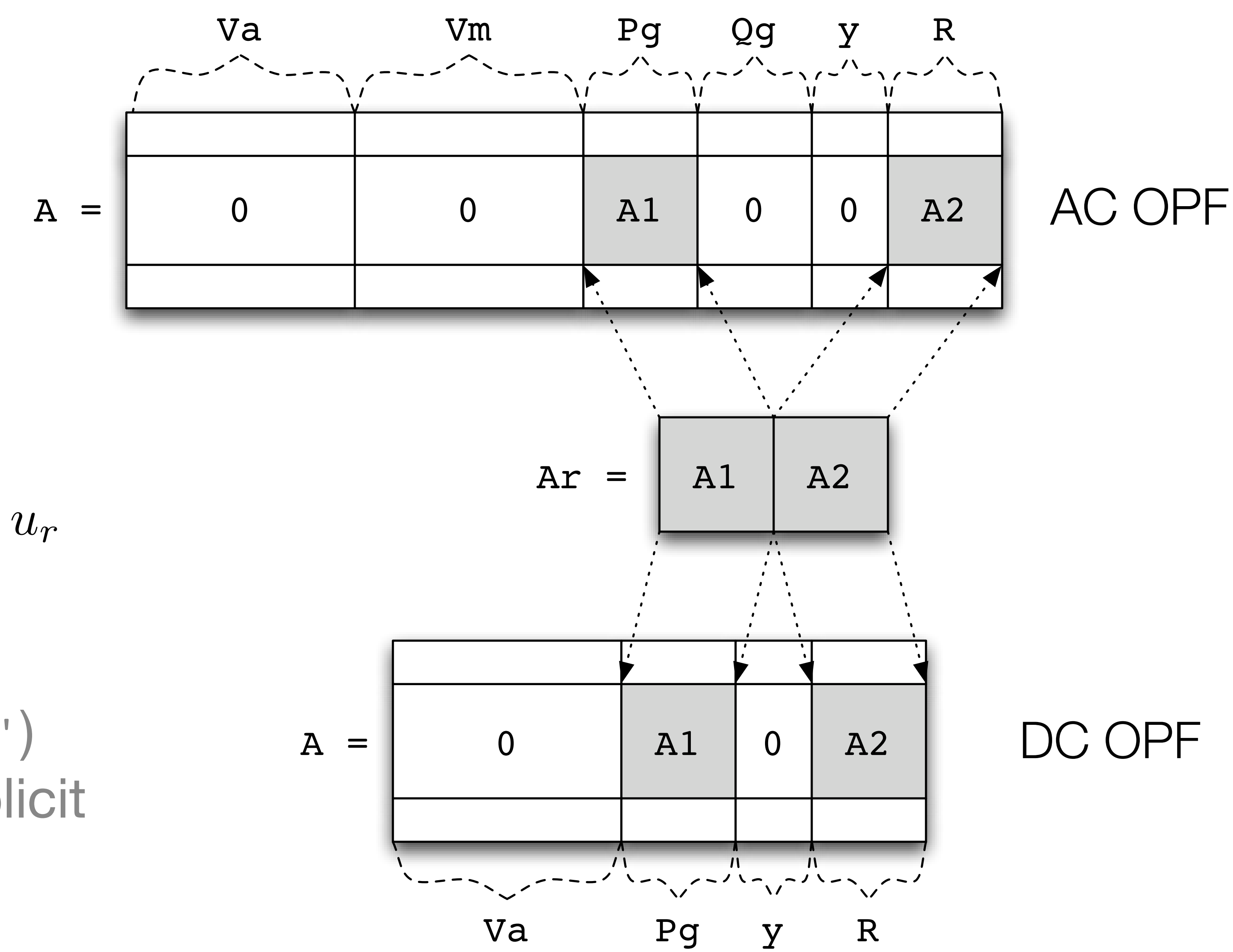
- Constraints added in named blocks, with *A*, *l*, *u* and block column names, e.g.

```
om = add_constraints(om, 'Res', Ar, lr, ur, {'Pg', 'R'});
```

$$l \leq A \begin{bmatrix} x \\ z \end{bmatrix} \leq u$$

$$l_r \leq A_r \begin{bmatrix} P_g \\ R \end{bmatrix} \leq u_r$$

$$l_r \leq \begin{bmatrix} A_1 & A_2 \end{bmatrix} \begin{bmatrix} P_g \\ R \end{bmatrix} \leq u_r$$



- Constraint multipliers can be accessed by name (e.g , `'Res'`) w/o need to keep track of explicit indexing

Outline

- Matpower Overview
- Extensible OPF Formulation
- Standard Extensions
- Software Architecture
- Example: Adding Reserves

Example – Adding Reserves

- Jointly optimize the allocation of both energy and reserves
- Reserve requirements are set of fixed zonal quantities
- New reserve variable: $0 \leq r_i \leq r_i^{\max}$
- Additional reserve cost: $f_u(x, z) = \sum_{i \in U} c_i r_i$
- Reserve constraints: $p_i + r_i \leq p_i^{\max}, \quad \forall i \in U$

$$\sum_{i \in Z_k} r_i \geq R_k, \quad \forall k$$

Adding Reserves – Code

name	description
`om`	OPF model object, already includes standard OPF setup
`ng`	number of generators
`R`	name for new reserve variable vector
`Rmin`	lower bound on `R`, all zeros
`Rmax`	upper bound on `R`, based on ramp rates
`Pmax`	capacity of generators
`I`	identity matrix (`ng` x `ng`)
`Az`	zone definitions, `Az(i,j) = 1`, *iff* gen `j` lies in zone `i`
`Rreq`	vector of reserve requirements for each zone
`Rcost`	cost coefficients for `R`

```
Ar = [I I];
om = add_vars(om, 'R', ng, [], Rmin, Rmax);
om = add_constraints(om, 'Pg_plus_R', Ar, [], Pmax, {'Pg', 'R'});
om = add_constraints(om, 'Rreq', Az, Rreq, [], {'R'});
om = add_costs(om, 'Rcost', struct('N',I,'Cw',Rcost), {'R'});
```

Goals & Applications

- Make it as simple as possible for students and researchers to solve problems that require variations of a power flow or OPF formulation, without having to rewrite the parts that are shared with a standard formulation.

- To be able to easily extend and modify an optimal power flow formulation to include new variables, constraints and/or costs.

- Example applications:
  - co-optimize energy and reserves
  - add environmental costs (e.g. CO2, SOx, NOx) or constraints
  - contingency constrained OPF

➡ MATPOWER 4 available soon at: http://www.pserc.cornell.edu/matpower/