Nonlinear optimal power flow

1.0 Some introductory comments

Although the LPOPF does bring in the transmission constraints, it does not handle voltage or reactive power constraints. It is possible to approximate losses via a linear loss function, which improves the LPOPF, but the approximation is not very good.

The nonlinear OPF (NLOPF) addresses both of these issues, but at a "cost" of significantly higher computation. We will also see that the NLOPF admits additional control capabilities that can be very useful.

We utilize [1] significantly in these notes. Please read Section 13.1-13.2 of W&W.

2.0 Formulation

> W&W denote $\underline{y} = \begin{bmatrix} \underline{u} \\ \underline{p} \end{bmatrix}$ where $\underline{u}$ is the same as our $\underline{u}$, but $\underline{p}$ includes "fixed parameters" such as load. I will not use $\underline{p}$, thus, $\underline{y}=\underline{u}$.

Define:

- n_g: number of generators
- N: number of buses
- $\underline{x}$: state vector
- $\underline{u}$: control vector
- E_i: the voltage magnitude at bus i
- θ_i: the angle at bus i
- P_{gi}: the generation level at bus i

We will describe state and control vectors, the objective function, the equality constraints, and the inequality constraints.

State and control vectors:

The state and control vectors are expressed as

$$\underline{\boldsymbol{x}} = \begin{bmatrix} \boldsymbol{\theta}_2 \\ \vdots \\ \boldsymbol{\theta}_{\boldsymbol{N}} \\ \boldsymbol{E}_{\boldsymbol{n}_g+1} \\ \vdots \\ \boldsymbol{E}_{\boldsymbol{N}} \end{bmatrix} \qquad \underline{u} = \begin{bmatrix} P_{g2} \\ \vdots \\ P_{gn_g} \\ E_1 \\ \vdots \\ E_{n_g} \end{bmatrix} \tag{1}$$

Note the numbering scheme: generator buses are numbered first $1,\ldots,n_g$, then load buses $n_g+1,\ldots,N$. Also notice:

- The state vector contains the bus voltage magnitudes and angles that we cannot control (it may also include P_{g1}, as seen later).
- The control vector contains the generation levels and bus voltage magnitudes that we can control (this vector contains what we previously called the *decision variables*). It is also possible to include taps associated with tap-changing transformers in this vector.

Objective function:

The objective function for the most common OPF problem is an economic objective function. In our case, we will assume that it is cost, to be consistent with our LPOPF formulation. It is given by:

$$f(\underline{x},\underline{u}) = \sum_{i=1}^{n_g} F_i(P_{gi}) = F_1(f_{pg1}(\underline{x},\underline{u})) + \sum_{i=2}^{n_g} F_i(P_{gi}) \tag{2}$$

Here, $f_{pg1}(\underline{x},\underline{u})$ represents P_{g1}, and is a dependent variable. So P_{g1} (reference bus), is not a decision variable (and therefore not in the control vector). Yet its dependence on the vectors $\underline{x}$ and $\underline{u}$ requires that we include it in the objective function. Also observe that the dependence of f_{pg1} on $\underline{x}$ comes only through $F_1(P_{g1})$. This means the P_{gi} are all chosen independently, except for P_{g1} which is determined by the solution ($\underline{x}$, $\underline{u}$). As we said before, other objective functions can be used.

Equality constraints:

The equality constraints are given by the power flow equations, expressed below:

$$F_{pk}(\underline{x},\underline{u}) = \sum_{j=1}^{N} |E_k||E_j| \left(G_{kj}\cos(\theta_k - \theta_j) + B_{kj}\sin(\theta_k - \theta_j)\right) - P_k, k = 1,\ldots,N \quad \text{(3a)}$$

$$F_{qk}(\underline{x},\underline{u}) = \sum_{j=1}^{N} |E_k||E_j| \left(G_{kj}\sin(\theta_k - \theta_j) - B_{kj}\cos(\theta_k - \theta_j)\right) - Q_k, k = n_g + 1,\ldots,N \quad \text{(3b)}$$

We see from the above equations that we have $2N\text{-}n_g$ equality constraints. We do not include the reactive power equations for the generator buses because doing so brings in the extra (unknown) variable Q_k. We do not have to include the real power flow equation for the reference bus, but we may, and it is advantageous to do so because we need P_{g1} to evaluate the objective. If we do include the ref bus real power flow equation, then then $\underline{x}$ needs to be changed to include P_{g1} as shown below:

$$\underline{x} = \begin{bmatrix} \theta_2 \\ \vdots \\ \theta_N \\ E_{n_g+1} \\ \vdots \\ E_N \end{bmatrix} \Rightarrow \qquad \underline{x} = \begin{bmatrix} \theta_2 \\ \vdots \\ \theta_N \\ E_{n_g+1} \\ \vdots \\ E_N \\ P_{g1} \end{bmatrix}$$

Assuming we do include the ref bus real power flow equation, we denote the equality constraints as follows:

$$
\begin{aligned}
g_1(\underline{x},\underline{u}) &= F_{p1}(\underline{x},\underline{u}) - P_k = 0\\
&\vdots\\
g_N(\underline{x},\underline{u}) &= F_{pN}(\underline{x},\underline{u}) - P_N = 0\\
g_{N+1}(\underline{x},\underline{u}) &= F_{q1}(\underline{x},\underline{u}) - Q_1 = 0 \qquad \Rightarrow \quad \underline{g}(\underline{x},\underline{u}) = 0 \qquad (3c)\\
&\vdots\\
g_{2N-n_g}(\underline{x},\underline{u}) &= F_{pN}(\underline{x},\underline{u}) - Q_N = 0
\end{aligned}
$$

Inequality constraints:

There are four basic types of inequality constraints.

a. Generator real power limits:

$$P_{gi,\min} \le P_{gi} \le P_{gi,\max} \qquad i = 1,\ldots,n_g \qquad (4a)$$

b. Generator reactive power limits:

$$Q_{gi,\min} \le Q_{gi} \le Q_{gi,\max} \qquad i = 1,\ldots,n_g \qquad (4b)$$

c. Load bus voltage limits:

$$E_{i,\min} \le E_i \le E_{i,\max} \qquad i = n_g + 1, N \qquad (4c)$$

d. Line flow constraints:

$$-\boldsymbol{T}_{\boldsymbol{k},\max} \le \boldsymbol{T}_{\boldsymbol{k}} \le \boldsymbol{T}_{\boldsymbol{k},\max} \qquad \boldsymbol{k} = 1,\ldots,\boldsymbol{N}_{\boldsymbol{Lines}} \qquad (4d)$$

We denote all of (4a)-(4d) as:

$$\underline{\boldsymbol{h}}(\underline{\boldsymbol{x}},\underline{\boldsymbol{u}}) \le \underline{0} \qquad (4)$$

This can be done for any constraint $x_{min} \le x \le x_{max}$ as follows:

Lower Bound: $x \ge x_{min} \Rightarrow -x \le -x_{min} \Rightarrow -x + x_{min} \le 0 \Rightarrow h_1(x) = -x + x_{min} \le 0$

Upper Bound: $x \le x_{max} \Rightarrow x - x_{max} \le 0 \Rightarrow h_2(x) = x - x_{max} \le 0$

We are now in a position to clearly state our problem as:

$$
\begin{aligned}
&\min\ f(\underline{x},\underline{u})\\
&\text{subject to:}\\
&g(\underline{x},\underline{u}) = \underline{0} \qquad (5)\\
&\underline{h}(\underline{x},\underline{u}) \le \underline{0}
\end{aligned}
$$

Lagrangian:

The Lagrangian of our problem (5) is given by

$$\mathcal{L}(\underline{x},\underline{u},\underline{\lambda},\underline{\mu}) = f(\underline{x},\underline{u}) + \underline{\lambda}^T \underline{g}(\underline{x},\underline{u}) + \underline{\mu}^T \underline{h}(\underline{x},\underline{u}) \qquad (6)$$

Notice that we have used a form where the equality and inequality constraint terms are *added*. This is to be consistent with W&W, eq. (13.15). This will result in the corresponding Lagrange multipliers being the negative of what they would be if the equality and inequality constraint terms were *subtracted*.

W&W, eq. (13.18) and (13.19) , do not include the last terms in (7a,7b) because they (p. 519, top) "are only representing equality constraints at this point."

Optimality conditions:

By KKT, the optimality conditions are

$$\frac{\partial \mathcal{L}(\underline{x},\underline{u},\underline{\lambda},\underline{\mu})}{\partial \underline{x}} = \frac{\partial f(\underline{x},\underline{u})}{\partial \underline{x}} + \frac{\partial}{\partial \underline{x}}[\underline{\lambda}^T \underline{g}(\underline{x},\underline{u})] + \frac{\partial}{\partial \underline{x}}[\underline{\mu}^T \underline{h}(\underline{x},\underline{u})] = \underline{0} \quad (7a)$$

$$\frac{\partial \mathcal{L}(\underline{x},\underline{u},\underline{\lambda},\underline{\mu})}{\partial \underline{u}} = \frac{\partial f(\underline{x},\underline{u})}{\partial \underline{u}} + \frac{\partial}{\partial \underline{u}}[\underline{\lambda}^T \underline{g}(\underline{x},\underline{u})] + \frac{\partial}{\partial \underline{u}}[\underline{\mu}^T \underline{h}(\underline{x},\underline{u})] = \underline{0} \quad (7b)$$

$$\frac{\partial \mathcal{L}(\underline{x},\underline{u},\underline{\lambda},\underline{\mu})}{\partial \underline{\lambda}} = \frac{\partial}{\partial \underline{\lambda}}[\underline{\lambda}^T \underline{g}(\underline{x},\underline{u})] = \underline{0} \quad (7c)$$

$$\mu_i h_i(\underline{x},\underline{u}) = 0 \quad (7d)$$

The last condition, 7d, is the complementary condition.

We are now in a position to discuss solution techniques to solving the above set of equations (7a)-(7d).

3.0 Solution by generalized reduced gradient (GRG)

This method is also called the method of steepest descent and was first described for the OPF problem in [2].

It can solve the equality-constrained problem very well.

It solves the inequality-constrained problem only via an iterative approach, as we have done before, where we begin by assuming no inequality constraints are binding, solve the problem, include only those violated inequality constraints in a new solution, then solve again, and repeat until we get a solution with no violated inequality constraints.

Therefore the problem that we need to solve here is the one in the inner loop, where any violated inequality constraint is assumed to be included in the equality constraint vector $\underline{g}(\underline{x},\underline{u})=\underline{0}$. This conforms to what W&W do via "The Gradient Method" of 13.2.1.

So we are trying to solve the following equations:

$$\frac{\partial \mathcal{L}(\underline{x},\underline{u},\underline{\lambda},\underline{\mu})}{\partial \underline{x}} = \frac{\partial f(\underline{x},\underline{u})}{\partial \underline{x}} + \frac{\partial}{\partial \underline{x}}[\underline{\lambda}^T \underline{g}(\underline{x},\underline{u})] = \underline{0} \tag{8a}$$

$$\frac{\partial \mathcal{L}(\underline{x},\underline{u},\underline{\lambda},\underline{\mu})}{\partial \underline{u}} = \frac{\partial f(\underline{x},\underline{u})}{\partial \underline{u}} + \frac{\partial}{\partial \underline{u}}[\underline{\lambda}^T \underline{g}(\underline{x},\underline{u})] = \underline{0} \tag{8b}$$

$$\frac{\partial \mathcal{L}(\underline{x},\underline{u},\underline{\lambda},\underline{\mu})}{\partial \underline{\lambda}} = \frac{\partial}{\partial \underline{\lambda}}[\underline{\lambda}^T \underline{g}(\underline{x},\underline{u})] = \underline{0} \tag{8c}$$

where

$$\mathcal{L}(\underline{x},\underline{u},\underline{\lambda}) = f(\underline{x},\underline{u}) + \underline{\lambda}^T \underline{g}(\underline{x},\underline{u}) \tag{9}$$

The basic idea of the solution procedure is as follows:

1. We desire to minimize f($\underline{x}$,$\underline{u}$).
2. We can only change $\underline{u}$.
3. So we desire to move in the direction of *steepest descent* with respect to the function f($\underline{x}$,$\underline{u}$). This direction is given by
   $$-\nabla_{\underline{u}} f(\underline{x},\underline{u})$$
4. But $\underline{x}$ is a function of $\underline{u}$, that is, $\underline{x}=\underline{Z}(\underline{u})$ where Z is some unknown function that maps the vector $\underline{u}$ to the vector $\underline{x}$.
5. Therefore we can write
   $$f(\underline{x},\underline{u}) = f(Z(\underline{u}),\underline{u})$$
   And the question is, how to obtain $\nabla_{\underline{u}} f$? Notationally, we have
   $$-\nabla_{\underline{u}} f(\underline{x},\underline{u}) \equiv -\frac{df(\underline{x},\underline{u})}{d\underline{u}}$$

where the total derivative represents the change in f per unit change in $\underline{u}$ where $\underline{x}$ also changes (and the partial derivative represents the change in f per unit change in $\underline{u}$ where $\underline{x}$ remains fixed).

To get the desired gradient function, we begin by expressing

$$df(\underline{x},\underline{u}) = \left[\frac{\partial f(\underline{x},\underline{u})}{\partial \underline{x}}\right]^T d\underline{x} + \left[\frac{\partial f(\underline{x},\underline{u})}{\partial \underline{u}}\right]^T d\underline{u} \qquad (10)$$

But d$\underline{x}$ must be a function of d$\underline{u}$ since $\underline{x}$ is a function of $\underline{u}$. Let's express this via the following observation:

If we change $\underline{u}$ by a small amount d$\underline{u}$, and $\underline{x}$ changes by a small amount d$\underline{x}$, then to satisfy $\underline{g}(\underline{x},\underline{u})=\underline{0}$, it must be true that

$$d\underline{g}(\underline{x},\underline{u}) = \frac{\partial \underline{g}(\underline{x},\underline{u})}{\partial \underline{x}} d\underline{x} + \frac{\partial \underline{g}(\underline{x},\underline{u})}{\partial \underline{u}} d\underline{u} = 0 \qquad (11)$$

Solving (11) for d$\underline{x}$, we obtain

$$d\underline{x} = -\left[\frac{\partial \underline{g}(\underline{x},\underline{u})}{\partial \underline{x}}\right]^{-1}\left[\frac{\partial \underline{g}(\underline{x},\underline{u})}{\partial \underline{u}}\right] d\underline{u} \qquad (12)$$

Substitution of (12) into (10) results in

$$df(\underline{x},\underline{u}) = -\left[\frac{\partial f(\underline{x},\underline{u})}{\partial \underline{x}}\right]^T \left[\frac{\partial \underline{g}(\underline{x},\underline{u})}{\partial \underline{x}}\right]^{-1}\left[\frac{\partial \underline{g}(\underline{x},\underline{u})}{\partial \underline{u}}\right] d\underline{u} + \left[\frac{\partial f(\underline{x},\underline{u})}{\partial \underline{u}}\right]^T d\underline{u} \qquad (13)$$

Factoring d$\underline{u}$ to the right,

$$df(\underline{x},\underline{u}) = \left\{-\left[\frac{\partial f(\underline{x},\underline{u})}{\partial \underline{x}}\right]^T \left[\frac{\partial \underline{g}(\underline{x},\underline{u})}{\partial \underline{x}}\right]^{-1}\left[\frac{\partial \underline{g}(\underline{x},\underline{u})}{\partial \underline{u}}\right] + \left[\frac{\partial f(\underline{x},\underline{u})}{\partial \underline{u}}\right]^T\right\} d\underline{u} \qquad (14)$$

$\underline{g}(\underline{x},\underline{u})=\underline{0}$ are the pf eqtns. Observe that we have the same number of equations as we have state variables $\underline{x}$ (2N-N_g with the ref bus real pf equation and P_{g1} in $\underline{x}$, and 2N-N_g-1 without). In the latter case, the matrix being inverted in (12) is exactly the Jacobain.

In the above, d$\underline{u}$ is a column vector and the expression inside the curly brackets is a row vector.

Now we want to bring the d$\underline{u}$ over to the left-hand-side. To do so, we must transpose the left-hand-side so that it will also be a row vector. At the same time, we also rearrange the order of the term on the right.

$$\left[\frac{df(\underline{x},\underline{u})}{d\underline{u}}\right]^T = \left[\frac{\partial f(\underline{x},\underline{u})}{\partial \underline{u}}\right]^T - \left[\frac{\partial f(\underline{x},\underline{u})}{\partial \underline{x}}\right]^T \left[\frac{\partial \underline{g}(\underline{x},\underline{u})}{\partial \underline{x}}\right]^{-1} \left[\frac{\partial \underline{g}(\underline{x},\underline{u})}{\partial \underline{u}}\right] \qquad (15)$$

Recall the linear algebra rule that $[ABC]^T = C^T B^T A^T$. Use this to take the transpose of both sides of (15) to obtain:

$$\nabla_{\underline{u}} f(\underline{x},\underline{u}) = \frac{df(\underline{x},\underline{u})}{d\underline{u}} = \frac{\partial f(\underline{x},\underline{u})}{\partial \underline{u}} - \left[\frac{\partial \underline{g}(\underline{x},\underline{u})}{\partial \underline{u}}\right]^T \left\{\left[\frac{\partial \underline{g}(\underline{x},\underline{u})}{\partial \underline{x}}\right]^{-1}\right\}^T \left[\frac{\partial f(\underline{x},\underline{u})}{\partial \underline{x}}\right] \qquad (16)$$

Now I will make the following claim;

<u>Claim</u>: $\underline{\lambda} = -\left\{\left[\frac{\partial \underline{g}(\underline{x},\underline{u})}{\partial \underline{x}}\right]^T\right\}^{-1} \left[\frac{\partial f(\underline{x},\underline{u})}{\partial \underline{x}}\right]$ (17)

<u>Proof</u>: Repeating (8a), we have

$$\frac{\partial \mathcal{L}(\underline{x},\underline{u},\underline{\lambda},\underline{\mu})}{\partial \underline{x}} = \frac{\partial f(\underline{x},\underline{u})}{\partial \underline{x}} + \frac{\partial}{\partial \underline{x}}[\underline{\lambda}^T \underline{g}(\underline{x},\underline{u})] = \underline{0} \qquad (8a)$$

Use the following <u>fact</u>: If $\underline{\lambda}$ is a constant vector, and $\underline{b}=\underline{b}(\underline{x})$, and both are the same dimension, then

$$\frac{\partial}{\partial \underline{x}}[\underline{\lambda}^T \underline{b}(\underline{x})] = \left[\frac{\partial \underline{b}(\underline{x})}{\partial \underline{x}}\right]^T \underline{\lambda}$$

Use the above in (8a) to obtain

$$\frac{\partial \mathcal{L}(\underline{x},\underline{u},\underline{\lambda},\underline{\mu})}{\partial \underline{x}} = \frac{\partial f(\underline{x},\underline{u})}{\partial \underline{x}} + \left[\frac{\partial \underline{g}(\underline{x},\underline{u})}{\partial \underline{x}}\right]^T \underline{\lambda} = \underline{0} \qquad (18)$$

Solving (18) for $\underline{\lambda}$, we obtain:

$$\underline{\lambda} = -\left\{\left[\frac{\partial \underline{g}(\underline{x},\underline{u})}{\partial \underline{x}}\right]^T\right\}^{-1} \frac{\partial f(\underline{x},\underline{u})}{\partial \underline{x}} \qquad (17)$$

QED

Note that the first term (inside the curly brackets) of (17) is $[\underline{J}^T]^{-1}$, where $\underline{J}$ is the power flow Jacobian with the addition of any binding inequality constraints.

Repeating (16)

$$\nabla_{\underline{u}} f(\underline{x},\underline{u}) = \frac{df(\underline{x},\underline{u})}{d\underline{u}} = \frac{\partial f(\underline{x},\underline{u})}{\partial \underline{u}} - \left[\frac{\partial \underline{g}(\underline{x},\underline{u})}{\partial \underline{u}}\right]^T \left\{\left[\frac{\partial \underline{g}(\underline{x},\underline{u})}{\partial \underline{x}}\right]^{-1}\right\}^T \left[\frac{\partial f(\underline{x},\underline{u})}{\partial \underline{x}}\right] \quad (16)$$

Substitute (17) into (16) results in

$$\nabla_{\underline{u}} f(\underline{x},\underline{u}) = \frac{df(\underline{x},\underline{u})}{d\underline{u}} = \frac{\partial f(\underline{x},\underline{u})}{\partial \underline{u}} + \left[\frac{\partial \underline{g}(\underline{x},\underline{u})}{\partial \underline{u}}\right]^T \underline{\lambda} \quad (19)$$

Although both (16) and (19) are referred to as the *reduced gradient*, when we use this term, we will be referring to (19).

Eq. (16) is the reason for the name, i.e., it is the partial derivative of *f* wrspt $\underline{u}$ "reduced" by the term on the right of (16). The term on the right accounts for the fact that a small change $\Delta\underline{u}$ creates a small induced change $\Delta\underline{x}$ due to the power flow equations (and corresponding changes to power flow equations must be negative, as indicated by (11) and (12)).

4.0 Algorithm

Here is the GRG algorithm for solving the NLOPF problem.

1. Let k = 1. Guess an initial control vector $\underline{u}^{(k)}$. (Use economic dispatch with losses or without losses to make the initial guess).

2. Given $\underline{u}^{(k)}$, solve for $\underline{x}^{(k)}$ from (8c), repeated here for convenience:

$$\frac{\partial \mathcal{L}(\underline{x},\underline{u},\underline{\lambda},\underline{\mu})}{\partial \underline{\lambda}} = \frac{\partial}{\partial \underline{\lambda}} [\underline{\lambda}^T \underline{g}(\underline{x},\underline{u})] = \underline{0} \quad (8c)$$

   This is just a power flow solution!

3. Compute:

$$\underline{\lambda}^{(k)} = -\left\{\left[\frac{\partial \underline{g}(\underline{x},\underline{u})}{\partial \underline{x}}\right]^T\right\}^{-1} \frac{\partial f(\underline{x},\underline{u})}{\partial \underline{x}}\Bigg|_{\underline{x}^{(k)},\underline{u}^{(k)}} \tag{20}$$

4. Compute the "steepest ascent" direction, i.e., the gradient of *f*, according to (19)

$$\nabla_{\underline{u}} f(\underline{x},\underline{u}) = \frac{df(\underline{x},\underline{u})}{d\underline{u}} = \left[\frac{\partial f(\underline{x},\underline{u})}{\partial \underline{u}} + \left[\frac{\partial \underline{g}(\underline{x},\underline{u})}{\partial \underline{u}}\right]^T \underline{\lambda}\right]_{\underline{x}^{(k)},\underline{u}^{(k)}} \tag{19}$$

(the reduced gradient).

5. Update the control vector by moving it in the direction of steepest descent.

$$\underline{u}^{(k+1)} = \underline{u}^{(k)} - \alpha^{(k)} \nabla_{\underline{u}} f(\underline{x},\underline{u}) \tag{21}$$

where $\alpha^{(k)}$ is a step size which is reduced for every iteration.

6. If $\left|\nabla_{\underline{u}} f(\underline{x},\underline{u})\right| < \varepsilon$, stop. Else, k=k+1, and go to (2).

5.0 Examples

Example 1, EDC without losses:

For the system shown below, solve economic dispatch problem without losses. The cost functions are given by

$$C_1(P_{G1}) = 1 + P_{G1} + 3P_{G1}^2$$

$$C_2(P_{G2}) = 0.5 + 0.5P_{G2} + 0.5P_{G2}^2$$

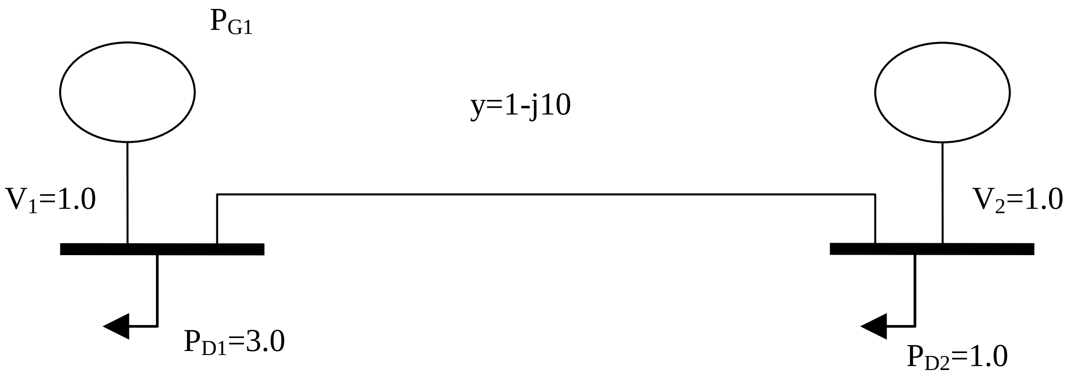

Fig. 1

The optimality conditions are:

$$\lambda = 1 + 6P_{G1}$$

$$\lambda = 0.5 + P_{G2}$$

$$4 = P_{G1} + P_{G2}$$

Writing these in matrix form we obtain

$$\begin{bmatrix} 6 & 0 & -1 \\ 0 & 1 & -1 \\ 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} P_{G1} \\ P_{G2} \\ \lambda \end{bmatrix} = \begin{bmatrix} -1 \\ -0.5 \\ 4 \end{bmatrix}$$

Solving using Matlab, we obtain:

$$\begin{bmatrix} P_{G1} \\ P_{G2} \\ \lambda \end{bmatrix} = \begin{bmatrix} 0.5 \\ 3.5 \\ 4.0 \end{bmatrix}$$

<u>Example 2, EDC with losses</u>:

We will derive the loss function since this is such a simple system.

The real power flowing across a line is expressed as

$$P_{pq} = V_p{}^2 G - V_p V_q G\cos(\theta_p - \theta_q) + V_p V_q B\sin(\theta_p - \theta_q)$$

Applying this to our system, and assuming θ_1=0, we have:

$$\begin{aligned} P_{12} &= (1)(1) - (1)(1)(1)\cos(-\theta_2) + (1)(1)(10)\sin(-\theta_2) \\ &= 1 - \cos\theta_2 - 10\sin\theta_2 \end{aligned} \qquad \text{(E2-1)}$$

$$\begin{aligned} P_{21} &= (1)(1) - (1)(1)(1)\cos(\theta_2) + (1)(1)(10)\sin(\theta_2) \\ &= 1 - \cos\theta_2 + 10\sin\theta_2 \end{aligned} \qquad \text{(E2-2)}$$

Losses may be expressed as the difference between the flow into the line and the flow out of the line. Denoting P'_{12} as the flow out of the line and into bus 2, we have

$$P_L = P_{12} - P'_{12} = P_{12} + P_{21} \qquad \text{(E2-3)}$$

Substituting (E2-1) and (E2-2) into (E2-3), we have

$$\begin{aligned} P_L &= P_{12} - P'_{12} = P_{12} + P_{21} \\ &= 1 - \cos\theta_2 - 10\sin\theta_2 + 1 - \cos\theta_2 + 10\sin\theta_2 \\ &= 2(1 - \cos\theta_2) \end{aligned} \qquad \text{(E2-4)}$$

Recall the Taylor series expansion for cosine:

$$\cos\theta_2 = 1 - \frac{\theta_2^2}{2!} + \frac{\theta_2^4}{4!} - \frac{\theta_2^6}{6!} \pm \ldots \approx 1 - \frac{\theta_2^2}{2} \qquad \text{(E2-5)}$$

Substituting the approximation of (E2-5) into (E2-4), we have

$$P_L = 2(1 - \left(1 - \frac{\theta_2^2}{2}\right)) = \theta_2^2 \qquad \text{(E2-6)}$$

But we want to express the losses as a function of our decision variables P_{G1} and P_{G2}.

We expressed losses as the sum of the flows into either end of the line per (E2-3). Now let’s express the difference of the flows into either end of the line:

$$P_{12} - P_{21} = P_{G1} - 3 - (P_{G2} - 1) = P_{G1} - P_{G2} - 2 \qquad \text{(E2-7)}$$

We may also use (E2-1) and (E2-2) to express the difference of the flows into either end of the line:

$$\begin{aligned} P_{12} - P_{21} &= 1 - \cos\theta_2 - 10\sin\theta_2 - (1 - \cos\theta_2 + 10\sin\theta_2) \\ &= -20\sin\theta_2 \end{aligned} \qquad \text{(E2-8)}$$

If the angle is small, then (E2-8) becomes:

$$P_{12} - P_{21} = -20\theta_2 \qquad \text{(E2-9)}$$

Equating (E2-7) and (E2-8), we get

$$\begin{aligned} &-20\theta_2 = P_{G1} - P_{G2} - 2 \\ &\Rightarrow \theta_2 = \frac{-(P_{G1} - P_{G2} - 2)}{20} \end{aligned} \qquad \text{(E2-10)}$$

By (E2-6), the loss function is the square of (E2-10), i.e.,

$$P_L = \theta_2^2 = \frac{(P_{G1} - P_{G2} - 2)^2}{400} \qquad \text{(E2-11)}$$

From (E2-11), we may compute the penalty factors according to:

$$\begin{aligned} L_1 &= \frac{1}{1 - \frac{\partial P_L}{\partial P_{G1}}} = \frac{1}{1 - P_{G1}/200} \\ L_2 &= \frac{1}{1 - \frac{\partial P_L}{\partial P_{G2}}} = \frac{1}{1 + P_{G2}/200} \end{aligned} \qquad \text{(E2-12)}$$

Setting up the optimality conditions, we have:

$$\begin{aligned} \lambda &= L_1 \frac{dC_1}{dP_{G1}} = \frac{1}{1 - P_{G1}/200}(6P_{G1} + 1) \\ \lambda &= L_2 \frac{dC_2}{dP_{G2}} = \frac{1}{1 + P_{G2}/200}(P_{G2} + 0.5) \end{aligned} \qquad \text{(E2-13a)}$$

Solving (E2-13a) for P_{G1} and P_{G2}, we obtain:

$$P_{G1} = \frac{\lambda - 1}{6 + \lambda / 200}$$
$$P_{G2} = \frac{\lambda - 0.5}{1 - \lambda / 200} \qquad \text{(E2-13b)}$$

We also know that

$$P_D = P_{G1} + P_{G2} - P_L$$

$$\Rightarrow P_D = P_{G1} + P_{G2} - \frac{(P_{G1} - P_{G2} - 2)^2}{400} \qquad \text{(E2-14)}$$

Let's use Lambda iteration to solve (E2-13a) and (E2-13b). Here is matlab code for making the evaluation given λ:

```
pg1=(lam-1)/(6+lam/200)
pg2=(lam-0.5)/(1-lam/200)
ploss=((pg1-pg2-2)^2)/400
pd=pg1+pg2-ploss
```

We initialize with the solution provided by Example 1 and arrive at the solution below, on the left, which is compared to the solution obtained in the previous example on the right:

Solution with losses	Solution without losses
$\lambda = 3.996$	$\lambda = 4.0$
$P_{G1} = 0.4977$	$P_{G1} = 0.5$
$P_{G2} = 3.5673$	$P_{G2} = 3.5$
$P_{Loss} = 0.0643$	$P_{Loss} = 0$
$P_D = 4.0007$	$P_D = 4.0$

Example 3, Optimal power flow:

Solve the problem of Example 2 using the optimal power flow. Ignore all constraints.

Variables:

Since both buses have generators, they are voltage control buses. Therefore the voltage magnitudes are considered to be known. Since the bus 1 angle is the reference, there is only one unknown angle that this will be a state variable, i.e., $x_1=\theta_2$.

We will model both real power flow equality constraints and will therefore need to identify that the bus 1 generation is a state variable, and so $x_2=P_{G1}$. Thus, in the notation of (3), $f_{pg1}(\underline{x},\underline{u})=x_2$.

There is only one control variable and it is $u=P_{G2}$. (We could identify the voltages at each bus as control variables, but we will not here in order to maintain as simple a model as possible here.)

Objective function:
The objective function is given by

$$f(\theta_2, P_{G1}, P_{G2}) = C_1(P_{G1}) + C_2(P_{G2}) =$$

$$1 + P_{G1} + 3P_{G1}^2 + 0.5 + 0.5P_{G2} + 0.5P_{G2}^2$$

Or, in terms of $\underline{x}$ and u, we have that

$$f(\underline{x}, u) = C_1(x_2) + C_2(u) =$$

$$1.5 + x_2 + 3x_2^2 + 0.5u + 0.5u^2$$

Equality constraints:
The equality constraints are the power flow equations. But since there are no PQ buses in this network, there are no reactive power equations. Therefore we need only consider the real power flow equations at the two buses. Recalling (3a) and (3c), we have:

$$g_1(\underline{x},\underline{u}) = F_{p1}(\underline{x},\underline{u}) - P_1 = 0$$

$$g_2(\underline{x},\underline{u}) = F_{p2}(\underline{x},\underline{u}) - P_2 = 0$$

where

$$F_{pk}(\underline{x},\underline{u}) = \sum_{j=1}^{N} |E_k||E_j|\left(G_{kj}\cos(\theta_k - \theta_j) + B_{kj}\sin(\theta_k - \theta_j)\right), k = 1,\ldots, N$$

and

$$P_k = P_{Gk} - P_{dk}$$

This results in:

$$g_1(\underline{x},\underline{u}) = F_{p1}(\underline{x},\underline{u}) - P_{G1} + P_{d1} = 0$$

$$g_2(\underline{x},\underline{u}) = F_{p2}(\underline{x},\underline{u}) - P_{G2} + P_{d2} = 0$$

With y=1-j10, Y_{11}=1-j10, Y_{12}=-1+j10, so the above become:

$$g_1(\underline{x},\underline{u}) = 1 - \cos\theta_2 - 10\sin\theta_2 - P_{G1} + 3 = 0$$

$$g_2(\underline{x},\underline{u}) = 1 - \cos\theta_2 + 10\sin\theta_2 - P_{G2} + 1 = 0$$

Replacing the variables with $x_1=\theta_2$, $x_2=P_{G1}$, and $u=P_{G2}$, the equality constraints become:

$$g_1(\underline{x},\underline{u}) = 1 - \cos x_1 - 10\sin x_1 - x_2 + 3 = 0$$

$$g_2(\underline{x},\underline{u}) = 1 - \cos x_1 + 10\sin x_1 - u + 1 = 0$$

Combining the constants, we get

$$g_1(\underline{x},\underline{u}) = 4 - \cos x_1 - 10\sin x_1 - x_2 = 0$$

$$g_2(\underline{x},\underline{u}) = 2 - \cos x_1 + 10\sin x_1 - u = 0$$

<u>Problem statement</u>:

The problem statement then becomes the following:

$$\min\ f(\underline{x},u) = 1.5 + x_2 + 3x_2^2 + 0.5u + 0.5u^2$$

$$\text{subject to}$$

$$g_1(\underline{x},\underline{u}) = 4 - \cos x_1 - 10\sin x_1 - x_2 = 0$$

$$g_2(\underline{x},\underline{u}) = 2 - \cos x_1 + 10\sin x_1 - u = 0$$

<u>Lagrangian</u>:

The Lagrangian function becomes:

$$\mathcal{L}(\underline{x},u,\underline{\lambda}) = f(\underline{x},u) + \underline{\lambda}^T \underline{g}(\underline{x},u,)$$

$$= f(\underline{x},u) + \begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix}^T \begin{bmatrix} g_1(\underline{x},\underline{u}) \\ g_2(\underline{x},\underline{u}) \end{bmatrix}$$

$$= f(\underline{x},u) + \begin{bmatrix} \lambda_1 & \lambda_2 \end{bmatrix} \begin{bmatrix} g_1(\underline{x},\underline{u}) \\ g_2(\underline{x},\underline{u}) \end{bmatrix}$$

$$= f(\underline{x},u) + \lambda_1 g_1(\underline{x},\underline{u}) + \lambda_2 g_2(\underline{x},\underline{u})$$

$$= 1.5 + x_2 + 3x_2^2 + 0.5u + 0.5u^2$$

$$+ \lambda_1 (1 - \cos x_1 - 10 \sin x_1 - x_2 + 3)$$

$$+ \lambda_2 (g_2(\underline{x},\underline{u}) = 1 - \cos x_1 + 10 \sin x_1 - u + 1 = 0)$$

Optimality conditions:

The appropriate optimality conditions are given by

$$\frac{\partial \mathcal{L}(\underline{x},u,\underline{\lambda})}{\partial \underline{x}} = \frac{\partial f(\underline{x},u)}{\partial \underline{x}} + \frac{\partial}{\partial \underline{x}} [\underline{\lambda}^T \underline{g}(\underline{x},\underline{u})] = 0$$

$$\frac{\partial \mathcal{L}(\underline{x},u,\underline{\lambda})}{\partial u} = \frac{\partial f(\underline{x},u)}{\partial u} + \frac{\partial}{\partial u} [\underline{\lambda}^T \underline{g}(\underline{x},\underline{u})] = 0$$

$$\frac{\partial \mathcal{L}(\underline{x},u,\underline{\lambda})}{\partial \underline{\lambda}} = \frac{\partial}{\partial \underline{\lambda}} [\underline{\lambda}^T \underline{g}(\underline{x},\underline{u})] = 0$$

Homework #6 for next Monday:

Problem 1: Use the Generalized Reduced Gradient procedure to solve the above problem.

The answers you should obtain are:
λ_1=4.2297, λ_2=4.0174, x_1=0.252, x_2=0.5383, u=3.5174

Compare to EDC and EDC+losses

<u>Solution w/ loss</u>	<u>Solution w/o losses</u>	OPF
		$\boldsymbol{\lambda_1 = 4.2297, \lambda_2 = 4.0174}$
$\lambda = 3.996$	$\lambda = 4.0$	$\boldsymbol{P_{G1} = 0.583}$
$P_{G1} = 0.4977$	$P_{G1} = 0.5$	$\boldsymbol{P_{G2} = 3.5174}$
$P_{G2} = 3.5673$	$P_{G2} = 3.5$	$\boldsymbol{P_{Loss} = 0.1004}$
$P_{Loss} = 0.0643$	$P_{Loss} = 0$	$\boldsymbol{P_D = 4.0}$
$P_D = 4.0007$	$P_D = 4.0$	$\boldsymbol{\theta_2 = 0.252 rad}$

Observe that the signs of λ_1 and λ_2 are both positive. Let's think through what this means:

- Our original formulation (see OptimizationIntro.ppt) formulated the Lagrangian by subtracting off the equality constraint terms, according to:

$$\mathcal{L}(\underline{x},\underline{\lambda}) = f(\underline{x}) - \lambda_1\left(h_1(\underline{x}) - c_1\right) - \lambda_2\left(h_2(\underline{x}) - c_2\right) - \ldots - \lambda_m\left(h_m(\underline{x}) - c_m\right)$$

- We interpreted the Lagrange multipliers as the increase in the objective function when the right-hand-side (RHS) of the corresponding constraint is increased by one unit.
- Now we have formulated the Lagrangian by adding the equality constraint terms, according to:

$$\mathcal{L}(\underline{x},u,\underline{\lambda}) = f(\underline{x},u) + \underline{\lambda}^T \underline{g}(\underline{x},u,)$$

- So we interpret the Lagrange multipliers as the decrease in the objective function when the RHS of the corresponding constraint is increased by one unit.
- Observe our equality constraints:

$$g_1(\underline{x},\underline{u}) = 1 - \cos x_1 - 10\sin x_1 - x_2 + 3 = 0$$
$$g_2(\underline{x},\underline{u}) = 1 - \cos x_1 + 10\sin x_1 - u + 1 = 0$$

  When we establish "right-hand-sides" these become:

$$g_1(\underline{x},\underline{u}) = 1 - \cos x_1 - 10\sin x_1 - x_2 = -3$$
$$g_2(\underline{x},\underline{u}) = 1 - \cos x_1 + 10\sin x_1 - u = -1$$

  These RHS are just the negative of the demands. Increasing them makes the demands smaller.
- So the definition of λ_1, λ_2 as "decrease in objective function when RHS of corresponding constraints are increased by one unit" means that positive λ's indicate the cost decreases as demand decreases – makes sense!

6.0 Matrix of second partials approach (Newton method)

The method described in section 5.0 updates the control variables at every iteration step along the direction of steepest descent with respect to the control variables. The problem with this method is that the step size must be small, requiring multiple iterations to identify the solution, and since each iteration requires a full power flow solution, the method can be quite computationally intensive for large systems.

Another way to solve the problem is to view the equations established by the first order conditions as a set of simultaneous nonlinear equations to solve. This means we can use our familiar Newton-Raphson method to solve! Reference [3] provides a good articulation of this method.

A key concept in applying this method is that the nonlinear equations that we must solve are actually the first derivative of the objective and equality constraints. In order to apply the NR approach, we can denote all variables in the equations of the optimality conditions as $\underline{z}$, i.e.,

$$\underline{z} = \begin{bmatrix} \underline{x} \\ \underline{u} \\ \underline{\lambda} \end{bmatrix}$$

Then the Lagrangian function as expressed in (9)

$$\mathcal{L}(\underline{x}, \underline{u}, \underline{\lambda}) = f(\underline{x}, \underline{u}) + \underline{\lambda}^T \underline{g}(\underline{x}, \underline{u}) \qquad (9)$$

becomes

$$\mathcal{L}(\underline{x}, \underline{u}, \underline{\lambda}) = \mathcal{L}(\underline{z}) \qquad (9)$$

To solve the nonlinear equations:

$$\nabla_z \mathcal{L}(\underline{z}) = \underline{0}$$

we need the Jacobian matrix of these equations, which we denote by $\underline{H}(\underline{z})$, where its elements are given by

$$H_{ij} = \frac{\partial^2 \mathcal{L}(\underline{z})}{\partial z_i \partial z_j}$$

The matrix $\underline{H}(\underline{z})$, which is the Jacobian with respect to the first-order conditions, is the *Hessian* with respect to the Lagrangian function.

Once the Hessian is obtained, the NR procedure is performed as usual, based on the update relation:

$$\underline{z}^{(k+1)} = \underline{z}^{(k)} - \left[\underline{H}(\underline{z}^{(k)})\right]^{-1} \nabla_z \mathcal{L}(\underline{z})$$

Homework #6 for next Monday:

Problem 2: For the matrix of the system used in HW6, Problem #1, assume the initial solution $z^{(0)}$ obtained by the economic dispatch solution, obtain the Hessian matrix, and take a single step to obtain a new point $z^{(1)}$.

7.0 Penalty function approach

We motivate this approach by looking at two simple cases. Our general goal is to change a constrained optimization problem into an unconstrained optimization problem. We require the objective function be convex and the feasible space be a convex set. This approach is discussed very briefly in W&W, pp. 530-531 as a method of handling inequality constraints. But as we will see, it can handle both equality and inequality constraints.

Prob. 1.0-a (constrained optimization with equality constraint):

$$\min \ f(\underline{x})$$
$$\text{subject to}$$
$$g(\underline{x}) = 0$$

Observe:

1. $g(\underline{x})$ must be zero at any feasible solution.
2. $[g(\underline{x})]^2=0$ implies $g(\underline{x})=0$, and therefore, by (1), that $[g(\underline{x})]^2=0$ implies that $\underline{x}$ is a feasible solution.
3. $[g(\underline{x})]^2\geq 0$ is always true and therefore $[g(\underline{x})]^2=0$ identifies the minimum value of $[g(\underline{x})]^2$.
4. By (3), minimizing $[g(\underline{x})]^2$ will result in finding the value of x that imposes $[g(\underline{x})]^2=0$. Therefore, by (2), minimizing $[g(\underline{x})]^2$ will result in a feasible value of $\underline{x}$.

With the above in mind, consider the following new problem:

Prob. 1.1-a (unconstrained optimization):

$$\min \phi(\underline{x})$$

where

$$\phi(\underline{x}) = f(\underline{x}) + \alpha\ p_1\big(g(\underline{x})\big); \qquad p_1\big(g(\underline{x})\big) = [g(\underline{x})]^2$$

We cannot guarantee that this new problem will find the solution to problem Prob 1.0-a. To see why, observe in Fig. 2 the functions:

$$y = (x-3)^2 + 3$$

$$y = x^2$$

$$y = (x-3)^2 + 3 + x^2$$

We note that the first one has a minimum at x=3, the second one has a minimum at x=0, and the third one, which is the sum of the first two, has a minimum at about x=1.

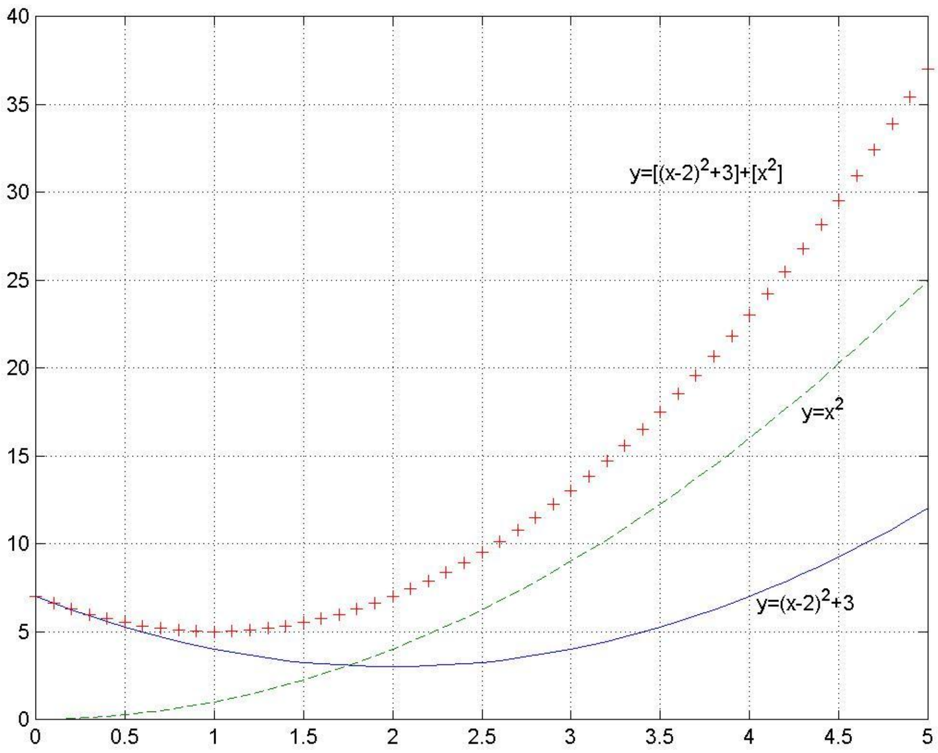

Fig. 2

What we can guarantee is that that Prob. 1.1-a will find a feasible solution to Prob. 1.0-a if we make α large enough. In our example, let's choose α=4. The solution is displayed in Fig. 3 where we observe that the minimum of the sum has moved to the left and now occurs at about 0.3 or 0.4. Clearly, the larger we make α, the more the second function, $y=x^2$, will dominate, and the closer the minimum of the sum will be to the minimum of $y=x^2$.

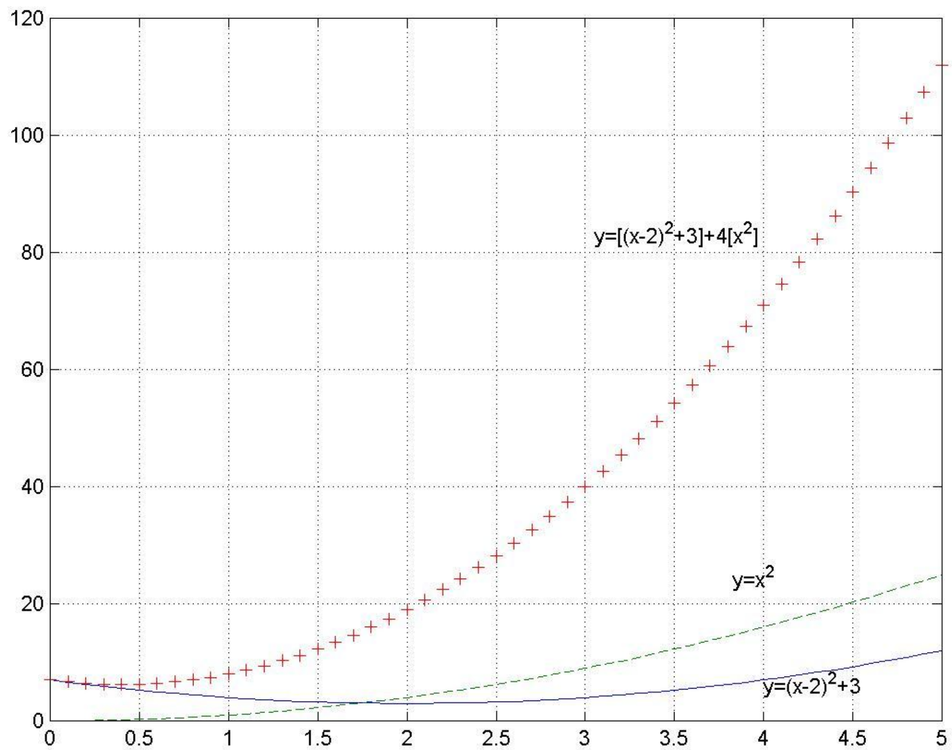

Fig. 3

We can draw the conclusion that Prob. 1.1-a is guaranteed to find a *feasible* solution (one that satisfies the equality constraint) if we make α large enough.

We can generalize this conclusion to the case of multiple equality constraints, as follows.

<u>Prob. 1.0-b (constrained optimization with N equality constraints):</u>

$$\min f(\underline{x})$$
$$\text{subject to}$$
$$\underline{g}(\underline{x}) = 0$$

<u>Prob. 1.1-b (unconstrained optimization):</u>

$$\min \phi(\underline{x})$$

where

$$\phi(\underline{x}) = f(\underline{x}) + \alpha \sum_{i=1}^{N} p_1\big(g_i(\underline{x})\big); \qquad p_1\big(g_i(\underline{x})\big) = [g_i(\underline{x})]^2$$

Now let's consider inequality constraints.

<u>Prob. 2.0-a (constrained optimization with inequality constraint):</u>

$$\min f(\underline{x})$$
$$\text{subject to}$$
$$h(\underline{x}) \le 0$$

Will the same approach work that we used for equality constraints? That is, define

$$\phi(\underline{x}) = f(\underline{x}) + \alpha\, p_2\big(h(\underline{x})\big); \qquad p_2\big(h(\underline{x})\big) = [h(\underline{x})]^2$$

and then solve *min* $\varphi(\underline{x})$ using a large value of α – will this work? That is, will it guarantee to find a feasible solution?

➔ This would only work if we know h($\underline{x}$)≤0 to be *binding* because it would impose h($\underline{x}$)=0, thus, not providing for the possibility that h($\underline{x}$)<0.

So we would like to have a penalty function p_2 which will impose h($\underline{x}$)=0 if h($\underline{x}$)≤0 is binding but allow h($\underline{x}$)<0 if not.

We can write such a penalty function as

$$p_2(h(\underline{x})) = \begin{cases} [h(\underline{x})]^2 & \text{if } h(\underline{x}) > 0 \\ 0 & \text{if } h(\underline{x}) \le 0 \end{cases}$$

The top function corresponds to the case when the constraint is binding, in which case we use the same penalty function that we used for equality constraints. The bottom function corresponds to the case when the constraint is non-binding, in which case we simply add 0 to the objective which has no effect on the solution.

Notationally, we may express the same thing as

$$p_2(h(\underline{x})) = [\max(0, h(\underline{x}))]^2$$

where we see that

- if $h(\underline{x}) \le 0$ (and therefore non-binding), then $p_2 = 0^2 = 0$.
- if $h(\underline{x}) > 0$ (and therefore binding), then $p_2 = [h(\underline{x})]^2$.

The function p_2 is illustrated in Fig. 2.

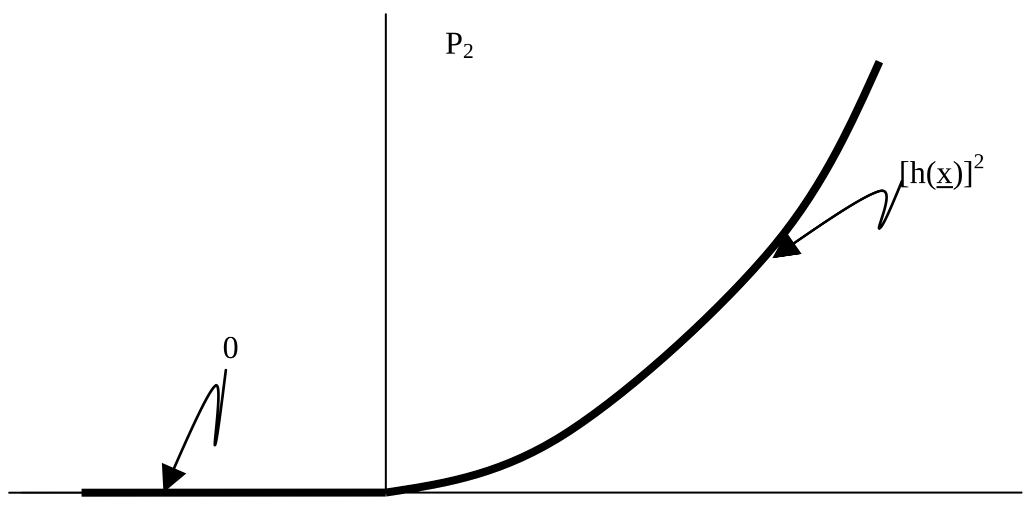

Fig. 2

We can define a continuous function that has a similar characteristic:

$$p_2\left(h(\underline{x})\right) = e^{kh(\underline{x})}; \qquad k > 0$$

which may be appropriately shaped if desired. For example, Fig. 3 illustrates the function

$$p_2\left(h(\underline{x})\right) = e^{5h(\underline{x})}$$

where it is clear that the function is almost 0 where h(x)=0, is 0 for h(x)<0, and gets big for h(x)>0.

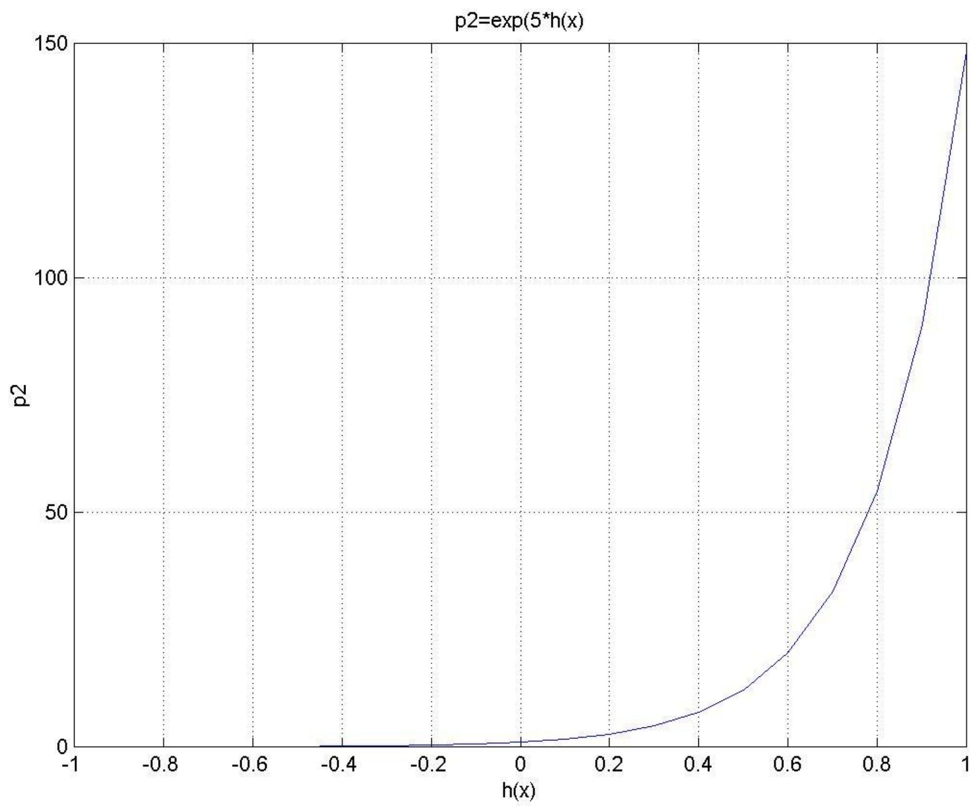

Fig. 3

Use of the continuous function provides that the function is differentiable everywhere, which can be beneficial when solving the equations imposed by the KKT conditions.

And so we see a way to handle inequality constraints with a penalty function as an unconstrained optimization problem.

Prob. 2.1-a (unconstrained optimization):

$$\min \phi(\underline{x})$$

where

$$\phi(\underline{x}) = f(\underline{x}) + \alpha\ p_2\big(h_i(\underline{x})\big); \qquad p_2\big(h(\underline{x})\big) = \begin{cases} [h(\underline{x})]^2 & \text{if } h(\underline{x}) > 0 \\ 0 & \text{if } h(\underline{x}) \le 0 \end{cases}$$

As in the case of equality constraints, we can draw the conclusion that Prob. 2.1-a is guaranteed to find a *feasible* solution (one that satisfies the inequality constraint) if we make α large enough.

We can generalize this conclusion to the case of multiple equality constraints, as follows.

Prob. 2.0-b (constrained optimization with M inequality constraints)

$$\min\ f(\underline{x})$$

$$\text{subject to}$$

$$\underline{h}(\underline{x}) \le \underline{0}$$

Prob. 2.1-b (unconstrained optimization):

$$\min \phi(\underline{x})$$

where

$$\phi(\underline{x}) = f(\underline{x}) + \alpha \sum_{\mathrm{i=1}}^{\mathrm{M}}\ p_2\big(h_i(\underline{x})\big); \qquad p_2\big(h_i(\underline{x})\big) = \begin{cases} [h_i(\underline{x})]^2 & \text{if } h_i(\underline{x}) > 0 \\ 0 & \text{if } h_i(\underline{x}) \le 0 \end{cases}$$

Finally, we may generalize our results to the case of multiple equality constraints and multiple inequality constraints.

Prob. 3.0-a (constrained optimization with 1 equality and 1 inequality constraint)

$$\min\ f(\underline{x})$$
$$\text{subject to}$$
$$g(\underline{x}) = \underline{0}$$
$$h(\underline{x}) \le \underline{0}$$

Prob. 3.1-a (unconstrained optimization):

$$\min \phi(\underline{x})$$

where

$$\phi(\underline{x}) = f(\underline{x}) + \alpha \left[\, p_1\big(g(\underline{x})\big) + p_2\big(h(\underline{x})\big)\right]$$

where

$$p_1\big(g(\underline{x})\big) = [g(\underline{x})]^2$$

$$p_2\big(h(\underline{x})\big) = \begin{cases} [h(\underline{x})]^2 & \text{if } h(\underline{x}) > 0 \\ 0 & \text{if } h(\underline{x}) \le 0 \end{cases}$$

Prob. 3.0-b (constrained optimization with N inequality constraints)

$$\min\ f(\underline{x})$$
$$\text{subject to}$$
$$\underline{g}(\underline{x}) = \underline{0}$$
$$\underline{h}(\underline{x}) \le \underline{0}$$

Prob. 3.1-b (unconstrained optimization):

$$\min \phi(\underline{x})$$

where

$$\phi(\underline{x}) = f(\underline{x}) + \alpha \left[\sum_{i=1}^{N} p_1\big(g_i(\underline{x})\big) + \sum_{i=1}^{M} p_2\big(h_i(\underline{x})\big) \right]$$

where

$$p_1\big(g_i(\underline{x})\big) = [g_i(\underline{x})]^2$$

$$p_2\big(h_i(\underline{x})\big) = \begin{cases} [h_i(\underline{x})]^2 & \text{if } h_i(\underline{x}) > 0 \\ 0 & \text{if } h_i(\underline{x}) \le 0 \end{cases}$$

Now, we ask the following question: If solution to the above unconstrained optimization problems only finds us a feasible solution, what good is it?

To answer this question, we modify our last, most general formulation by replacing α with $\alpha^{(k)}$. Thus, Prob. 3.1-b becomes

<u>Prob. 3.1-b (unconstrained optimization):</u>

$$\min \phi(\underline{x})$$

where

$$\boldsymbol{\phi}(\underline{\boldsymbol{x}}) = \boldsymbol{f}(\underline{\boldsymbol{x}}) + \boldsymbol{\alpha}^{(\boldsymbol{k})} \left[\sum_{i=1}^{N} \boldsymbol{p}_1\big(\boldsymbol{g_i}(\underline{x})\big) + \sum_{i=1}^{M} \boldsymbol{p}_2\big(\boldsymbol{h_i}(\underline{x})\big) \right]$$

where

$$\boldsymbol{p}_1\big(\boldsymbol{g_i}(\underline{x})\big) = [\boldsymbol{g_i}(\underline{x})]^2$$

$$\boldsymbol{p}_2\big(\boldsymbol{h_i}(\underline{x})\big) = \begin{cases} [\boldsymbol{h_i}(\underline{x})]^2 & \text{if } \boldsymbol{h_i}(\underline{x}) > 0 \\ 0 & \text{if } \boldsymbol{h_i}(\underline{x}) \le 0 \end{cases}$$

This suggests that we will develop a *sequence* of solutions corresponding to $\alpha^{(1)}$, $\alpha^{(2)}$, $\alpha^{(3)}$,…

To see how we want to modify $\alpha^{(k)}$, observe that

- If $\alpha^{(k)}$ is very small, then the objective function dominates, and the problem is essentially

$$\min f(\underline{x})$$

- If $\alpha^{(k)}$ is very large (we have already seen this), then the constraints dominate, and the problem is essentially

$$\min \phi(\underline{x}) = \alpha^{(k)} \left[\sum_{i=1}^{N} p_1(g_i(\underline{x})) + \sum_{i=1}^{M} p_2(h_i(\underline{x})) \right]$$

where

$$p_1(g_i(\underline{x})) = [g_i(\underline{x})]^2$$

$$p_2(h_i(\underline{x})) = \begin{cases} [h_i(\underline{x})]^2 & \text{if } h_i(\underline{x}) > 0 \\ 0 & \text{if } h_i(\underline{x}) \le 0 \end{cases}$$

And so we see that we can use $\alpha^{(k)}$ to adjust the relative weight between the objective function and the constraints.

Now consider a sequence of problems as follows:

1. Let k=1 and guess $\{\underline{x}^{(k)}, \alpha^{(k)}\}$ (the starting solution).
2. Solve the unconstrained minimization problem:

$$\min \boldsymbol{\phi}(\underline{\boldsymbol{x}}) = \boldsymbol{f}(\underline{\boldsymbol{x}}) + \boldsymbol{\alpha}^{(\boldsymbol{k})} \left[\sum_{i=1}^{N} \boldsymbol{p}_1(\boldsymbol{g}_{\boldsymbol{i}}(\underline{\boldsymbol{x}})) + \sum_{i=1}^{M} \boldsymbol{p}_2(\boldsymbol{h}_{\boldsymbol{i}}(\underline{\boldsymbol{x}})) \right]$$

where

$$\boldsymbol{p}_1(\boldsymbol{g}_{\boldsymbol{i}}(\underline{\boldsymbol{x}})) = [\boldsymbol{g}_{\boldsymbol{i}}(\underline{\boldsymbol{x}})]^2$$

$$\boldsymbol{p}_2(\boldsymbol{h}_{\boldsymbol{i}}(\underline{\boldsymbol{x}})) = \begin{cases} [\boldsymbol{h}_{\boldsymbol{i}}(\underline{\boldsymbol{x}})]^2 & \text{if } \boldsymbol{h}_{\boldsymbol{i}}(\underline{\boldsymbol{x}}) > 0 \\ 0 & \text{if } \boldsymbol{h}_{\boldsymbol{i}}(\underline{\boldsymbol{x}}) \le 0 \end{cases}$$

   using $\alpha^{(k)}$ and $\underline{x}^{(k)}$ as the starting solution. Denote the new solution as $\underline{x}^{(k+1)}$.
3. If $|\underline{x}^{(k+1)} - \underline{x}^{(k)}| < \varepsilon$, stop. Otherwise,
   a. Let $\alpha^{(k+1)} = \beta \times \alpha^{(k)}$
   b. k=k+1
   c. Go to 2.

Question: Should $\beta<1$ or $\beta>1$?

In other words, as we progress through this sequence of unconstrained optimization problems, do we want to increase emphasis on constraints or decrease emphasis on constraints?

The answer to this question is based on the following information:

- Large values of α create ill-conditioned nonlinear problems (i.e., problems for which nonlinear solvers do not converge quickly or do not converge at all).
- Ill-conditioned problems are very sensitive to the accuracy of the starting solution. If the starting solution is poor, then an ill-conditioned problem may not converge at all.

Since our worst guess is at the beginning of the sequence, we want to make α very small at the beginning of the sequence in order to avoid ill-conditioning.

Then we will change α so that we creep towards the feasible region with each successive solution until the stopping criterion is satisfied.

The implication of the last statement is that $\beta>1$.

Note carefully: We have transformed a constrained optimization problem into a sequence of unconstrained optimization problems whose solutions gradually move from the infeasible region to the feasible region.

This type of penalty function method is referred to as an *Exterior Point* penalty function method.

Two Final Comments:

1. At each stage of the penalty function method, we solve an unconstrained nonlinear optimization problem. There are many methods to do this. Below are three classes of such methods:
    a. Without derivatives: Cyclic coordinate, Hook & Jeeves, Rosenbrock
    b. With derivatives: Steepest descent, Newton's method

c. Conjugate directions (may or may not use derivatives): Davidon-Fletcher-Powell (uses derivatives); Fletcher-Reeves, and Zangwill.

2. Our focus has been on exterior penalty function methods. However, there is another broad class of penalty function solution methods applicable to solution of the Optimal Power Flow. These are called *Interior Point* penalty function methods. This class of solutions has also been referred to as *Barrier Function* methods. The main difference between Exterior Point methods and Interior Point methods is that whereas the former depend on a sequence of infeasible solutions that gradually move towards the feasible region, the latter depend on a sequence of feasible solutions that gradually move towards the boundary of the feasible region.

Example [4]:

Consider the following problem

$$\begin{aligned} &\text{Minimize} && (x_1-2)^4+(x_1-2x_2)^2 \\ &\text{subject to} && x_1^{\,2}-x_2=0 \\ & && X=E_2 \end{aligned}$$

Note that at iteration k, for a given penalty parameter μ_k, the problem to be solved to give $\mathbf{x}_{\mu_k}$ is

$$\text{minimize } (x_1-2)^4+(x_1-2x_2)^2+\mu_k(x_1^{\,2}-x_2)^2$$

Table 9.1 summarizes the computations using the penalty function method. The starting point is taken as $\mathbf{x}_1=(2.0, 1.0)$, where the objective function value is 0.0. The initial value of the penalty parameter is taken as α μ_1 = 0.1, and the

TABLE 9.1 **Summary of the Computations for the Penalty Function Method**

Iteration k	μ_k	$\mathbf{x}_{k+1}=\mathbf{x}_{\mu_k}$	$f(\mathbf{x}_{k+1})$	$\alpha(\mathbf{x}_{\mu_k})=h^2(\mathbf{x}_{\mu_k})$	$\theta(\mu_k)$	$\mu_k\alpha(\mathbf{x}_{\mu_k})$
1	0.1	(1.4539, 0.7608)	0.0935	1.8307	0.2766	0.1831
2	1.0	(1.1687, 0.7407)	0.5753	0.3908	0.9661	0.3908
3	10.0	(0.9906, 0.8425)	1.5203	0.01926	1.7129	0.1926
4	100.0	(0.9507, 0.8875)	1.8917	0.000267	1.9184	0.0267
5	1000.0	(0.9461094, 0.8934414)	1.9405	0.0000028	1.9433	0.0028

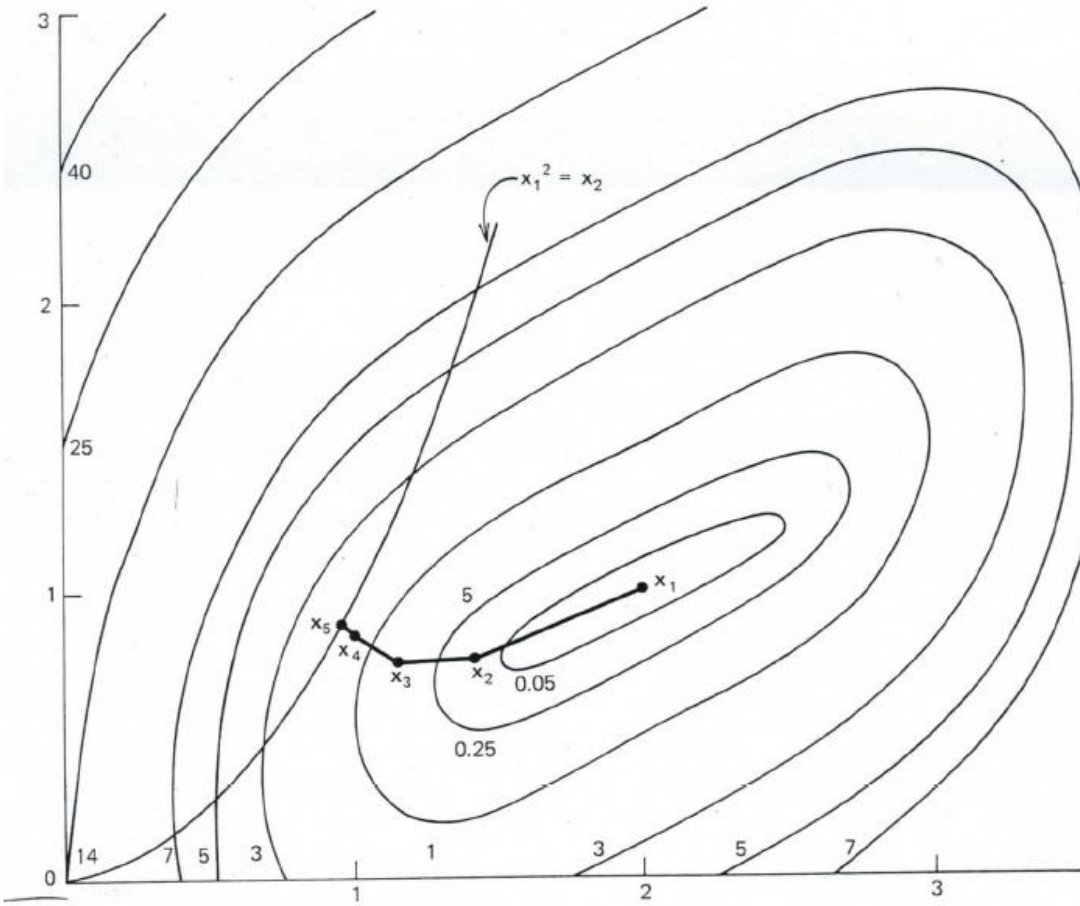

Figure 9.4 Illustration of the penalty function method.

scalar β is taken as 10.0. Note that $f(\mathbf{x}_{\mu_k})$ and $\theta(\mu_k)$ are nondecreasing functions, and $\alpha(\mathbf{x}_{\mu_k})$ is a nonincreasing function. The procedure could have been stopped after the fourth iteration, where $\alpha(\mathbf{x}_{\mu_k}) = 0.000267$. However, to show more clearly that $\mu_k \alpha(\mathbf{x}_{\mu_k})$ does converge to zero according to Theorem 9.2.2, one more iteration was carried out. At the point $\mathbf{x}^t = (0.9461094, 0.8934414)$, the reader can verify that the Kuhn–Tucker conditions are satisfied for $v = 3.3631$. Figure 9.4 shows the progress of the algorithm.

[1] A. Debs, Modern Power Systems Control and Operation," Kluwer, 1988.
[2] H. Dommel and W. Tinney, "Optimal Power Flow Solutions" IEEE Transactions on Power Apparatus and Systems, Vol. PAS-87, Oct. 1968.
[3] D. Sun, B. Ashley, B. Brewer, A. Hughes, and W. Tinney, " Optimal Power Flow by New Approach," IEEE Transactions on Power Systems, 1984.
[4] M. Bazaraa and C. Shetty, "Nonlinear Programming: Theory & Algorithms," John Wiley, 1979.