Power Flow V

Newton–Raphson theory

Buxin She
Mike Wiegers Department of Electrical and Computer Engineering

Lecture outline

1. Scalar Newton: locate a zero using a tangent.
2. Vector Newton: solve coupled correction equations.
3. Power flow: select P/Q equations and construct the Jacobian.

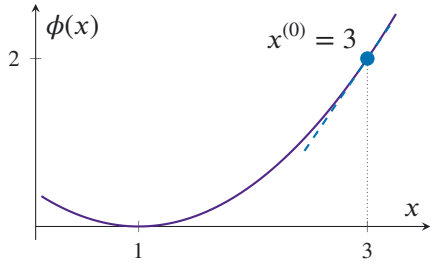
Understand why the correction is needed and where every coefficient comes from.

PART 1

General Newton–Raphson method

Use a tangent to locate a zero. Then apply the same local linear model to coupled equations.

Gradient descent on a quadratic



Consider the marked point:

$$\phi(x) = \tfrac{1}{2}(x-1)^2, \qquad x^{(0)} = 3.$$

The slope is positive: $\phi'(3) = 2$.

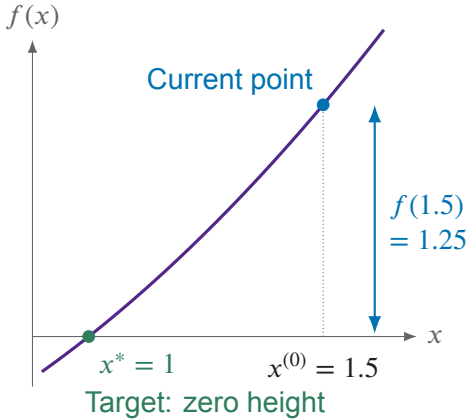
A small move left lowers ϕ.

One gradient descent step:

$$x^{(1)} = x^{(0)} - \alpha\phi'(x^{(0)}), \quad \alpha > 0.$$

The positive step size α controls how far to move.

Solving an equation means finding a zero



Rewrite $A(x) = B(x)$ as

$$f(x) = A(x) - B(x) = 0.$$

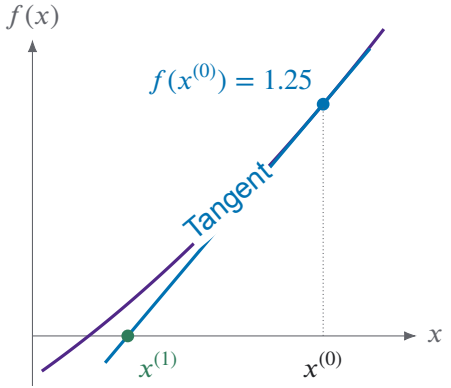
For $x^2 = 1$, define $f(x) = x^2 - 1$.

At $x^{(0)} = 1.5$, the residual is $f(1.5) = 1.25$.

A solution has **zero height** on this curve.

Positive-root region shown.

Use one tangent to predict the zero

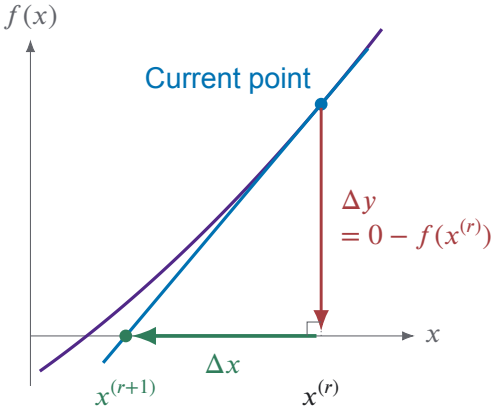


Next guess: tangent intercept

1. **Locate the current point.**
2. **Draw the tangent** using its local slope.
3. **Take the tangent's intercept** as the next guess $x^{(1)}$.

The tangent proposes the next x. We still need to check the original curve.

The correction is a horizontal distance



Define the change in x:

$$\Delta x = x^{(r+1)} - x^{(r)}.$$

On the tangent, slope is signed rise over signed run:

$$f'(x^{(r)}) = \frac{0 - f(x^{(r)})}{\Delta x}.$$

In this drawing, both changes are negative and the slope is positive.

The scalar Newton update

The tangent condition gives a linear equation for the correction:

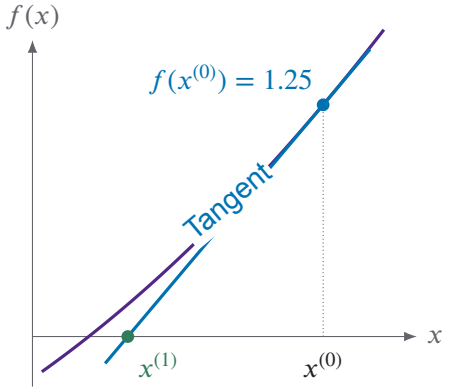
$$f(x^{(r)}) + f'(x^{(r)})\Delta x = 0.$$

$$\boxed{\Delta x = -\frac{f(x^{(r)})}{f'(x^{(r)})}}, \qquad f'(x^{(r)}) \neq 0.$$

Add the correction to the current guess:

$$\boxed{x^{(r+1)} = x^{(r)} + \Delta x = x^{(r)} - \frac{f(x^{(r)})}{f'(x^{(r)})}}.$$

Calculate the first correction



Next guess: tangent intercept

For $f(x) = x^2 - 1$, $f'(x) = 2x$.

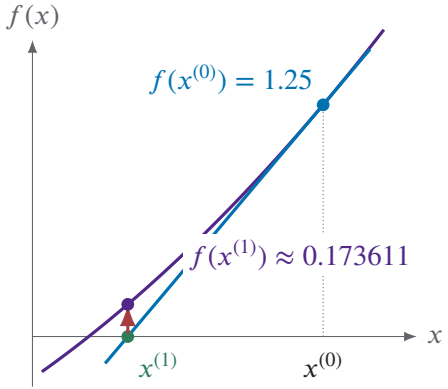
At $x^{(0)} = 1.5$:

$$f = 1.25, \qquad f' = 3.$$

$$\Delta x^{(0)} = -\frac{1.25}{3} \approx -0.416667.$$

$$x^{(1)} = 1.5 + \Delta x^{(0)} \approx 1.083333.$$

Check the original curve



Tangent value here: zero

The tangent reaches zero at $x^{(1)}$.

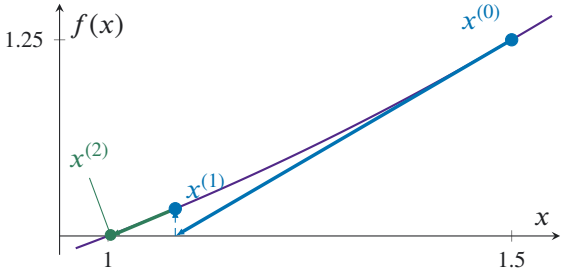
Evaluate the original function there:

$$f(x^{(1)}) = (1.083333\ldots)^2 - 1$$

$$\approx 0.173611 \neq 0.$$

The next tangent must touch the curve at the new x.

Repeat the tangent construction



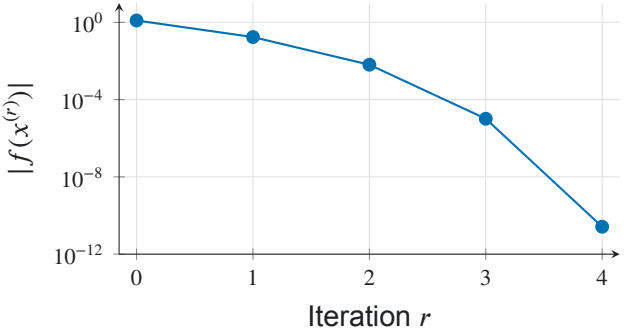
Follow the tangent to its intercept.

Return to the original curve at that x.

Draw a new tangent there.

$x^{(0)} = 1.5 \rightarrow x^{(1)} = 1.083333 \rightarrow x^{(2)} = 1.003205 \rightarrow \cdots$

Track the original residual



Test the original equation after each update:

$$|f(x^{(r)})| \leq \varepsilon.$$

For $\varepsilon = 10^{-8}$, this example passes after four corrections.

Log scale; dots are discrete iterates.

This starting guess converges. Newton convergence is not guaranteed for every initial guess.

Two coupled nonlinear equations

Seek u, v satisfying both $f_1(u, v) = 0$ and $f_2(u, v) = 0$.

Extend the scalar tangent condition $f + f' \Delta x = 0$:

$$f_1 + \frac{\partial f_1}{\partial u} \Delta u + \frac{\partial f_1}{\partial v} \Delta v = 0,$$

$$f_2 + \frac{\partial f_2}{\partial u} \Delta u + \frac{\partial f_2}{\partial v} \Delta v = 0.$$

Each equation includes **both** corrections.

Evaluate all residuals and derivatives at the same current point.

Stack the correction equations

The two scalar equations become one matrix system:

$$\underbrace{\begin{bmatrix} \partial f_1/\partial u & \partial f_1/\partial v \\ \partial f_2/\partial u & \partial f_2/\partial v \end{bmatrix}}_{\mathbf{J}\ \text{(Jacobian)}} \underbrace{\begin{bmatrix} \Delta u \\ \Delta v \end{bmatrix}}_{\Delta \mathbf{x}} = - \underbrace{\begin{bmatrix} f_1 \\ f_2 \end{bmatrix}}_{\mathbf{f}} .$$

Rows identify equations. Columns identify unknowns.

The matrix collects the coefficients already present in the scalar equations.

Newton–Raphson in vector form

For m equations in m unknowns, $J_{ij} = \partial f_i / \partial x_j$.

The local linear model is

$$\mathbf{f}(\mathbf{x} + \Delta\mathbf{x}) \approx \mathbf{f}(\mathbf{x}) + \mathbf{J}(\mathbf{x})\Delta\mathbf{x}.$$

Make its prediction zero, then update:

$$\boxed{\mathbf{J}(\mathbf{x}^{(r)})\Delta\mathbf{x} = -\mathbf{f}(\mathbf{x}^{(r)})}$$

$$\boxed{\mathbf{x}^{(r+1)} = \mathbf{x}^{(r)} + \Delta\mathbf{x}}.$$

Coupled example: Evaluate the current point

$$f_1(u, v) = u^2 + v^2 - 5, \qquad f_2(u, v) = u - v - 1.$$

Start at $(u^{(0)}, v^{(0)}) = (1.5, 0.5)$:

$$\mathbf{f}^{(0)} = \begin{bmatrix} 1.5^2 + 0.5^2 - 5 \\ 1.5 - 0.5 - 1 \end{bmatrix} = \begin{bmatrix} -2.5 \\ 0 \end{bmatrix}.$$

$$\mathbf{J}(u, v) = \begin{bmatrix} 2u & 2v \\ 1 & -1 \end{bmatrix}, \qquad \mathbf{J}^{(0)} = \begin{bmatrix} 3 & 1 \\ 1 & -1 \end{bmatrix}.$$

Coupled example: Solve and update

$$\begin{bmatrix} 3 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} \Delta u \\ \Delta v \end{bmatrix} = \begin{bmatrix} 2.5 \\ 0 \end{bmatrix}.$$

The second row gives $\Delta u = \Delta v$. The first gives $4\Delta u = 2.5$.

$$\Delta u = \Delta v = 0.625.$$

$$\mathbf{x}^{(1)} = \begin{bmatrix} 1.5 \\ 0.5 \end{bmatrix} + \begin{bmatrix} 0.625 \\ 0.625 \end{bmatrix} = \begin{bmatrix} 2.125 \\ 1.125 \end{bmatrix}.$$

Coupled example: Check both equations

At the new point $(u^{(1)}, v^{(1)}) = (2.125, 1.125)$:

$$f_1 = 2.125^2 + 1.125^2 - 5 = 0.78125,$$

$$f_2 = 2.125 - 1.125 - 1 = 0.$$

The second equation remains satisfied.

The first still needs another correction.

Rebuild the Jacobian at the new point and solve the coupled system again.

The general Newton iteration

1 Evaluate

Calculate the original residual $\mathbf{f}(\mathbf{x})$ at the current guess.

2 Test

Accept when $\|\mathbf{f}\|_\infty = \max_i |f_i| \le \varepsilon$.

3 Solve

Build $\mathbf{J}$ at the same point; solve $\mathbf{J}\Delta\mathbf{x} = -\mathbf{f}$.

4 Update

Add $\Delta\mathbf{x}$ and return to the original equations.

A singular Jacobian may prevent a unique step. Stop with failure at the iteration cap.

PART 2

Newton–Raphson power flow

We already know how to solve a zero-residual system. Now construct that system from the network's real and reactive power balance.

Apply the same Newton method

Part 1 gave us a method for any selected system $\mathbf{f}(\mathbf{x}) = \mathbf{0}$:

$$\mathbf{J}(\mathbf{x}^{(r)})\Delta\mathbf{x} = -\mathbf{f}(\mathbf{x}^{(r)}), \qquad \mathbf{x}^{(r+1)} = \mathbf{x}^{(r)} + \Delta\mathbf{x}.$$

To apply it to a power network, identify three ingredients:

1. The unknown voltage angles and magnitudes: $\mathbf{x}$.
2. The power balance equations, with all terms on one side: $\mathbf{f}$.
3. The derivatives of those same residual equations: $\mathbf{J}$.

Start with the physical equations. Their residuals become the input to the existing Newton method.

Voltage determines the network injections

Write $V_i = v_i e^{\mathrm{j}\theta_i}$, $Y_{ik} = G_{ik} + \mathrm{j}B_{ik}$, and $\theta_{ik} = \theta_i - \theta_k$.

$$P_i(\boldsymbol{\theta}, \mathbf{v}) + \mathrm{j}Q_i(\boldsymbol{\theta}, \mathbf{v}) = V_i \left(\sum_{k=1}^{n} Y_{ik} V_k \right)^* .$$

Its real and reactive parts are two voltage-dependent functions:

$$P_i(\boldsymbol{\theta}, \mathbf{v}) = v_i \sum_{k=1}^{n} v_k (G_{ik} \cos\theta_{ik} + B_{ik} \sin\theta_{ik}),$$

$$Q_i(\boldsymbol{\theta}, \mathbf{v}) = v_i \sum_{k=1}^{n} v_k (G_{ik} \sin\theta_{ik} - B_{ik} \cos\theta_{ik}).$$

A trial voltage gives a trial network injection. It need not yet balance the generation and load.

Move the power balance to one side

At bus i, generation minus demand must equal the injection into the network:

$$P_{Gi} - P_{Di} = P_i(\theta, \mathbf{v}), \qquad Q_{Gi} - Q_{Di} = Q_i(\theta, \mathbf{v}).$$

Define one residual for each equation:

$$\boxed{f_{Pi} = P_i(\theta, \mathbf{v}) + P_{Di} - P_{Gi} = 0},$$

$$\boxed{f_{Qi} = Q_i(\theta, \mathbf{v}) + Q_{Di} - Q_{Gi} = 0}.$$

Here G denotes generation and D denotes demand; positive injection is into the network.

For a load-only bus, $f_{Pi} = P_i + P_{Di}$: the network injection must be negative to supply positive demand.

Source: Glover et al., sixth edition, Secs. 6.4, 6.6; Zimmerman et al., IEEE Trans. Power Syst., 2011, Eqs. (30)–(31).

Select equations that match the unknowns

For a fixed bus classification, keep only equations whose net injections are given.

Bus	Given	Voltage unknowns	Residuals in Newton
PQ	$P_G - P_D$, $Q_G - Q_D$	θ_i, v_i	$f_{Pi} = 0$, $f_{Qi} = 0$
PV	$P_G - P_D$, v_i	θ_i	$f_{Pi} = 0$
Slack	θ_i, v_i	None	None

Let A be all non-slack buses and L be the PQ buses.

Unknowns: θ_i $(i \in A)$, v_i $(i \in L)$. Equations: f_{Pi} $(i \in A)$, f_{Qi} $(i \in L)$.

PV reactive generation and slack generation are outputs. Do not prescribe every P and Q at every bus.

The power flow problem is a zero-residual system

Insert the fixed voltage quantities and known injections into the bus equations. Stack the remaining quantities in a fixed bus order:

$$\mathbf{x} = \begin{bmatrix} \theta_A \\ \mathbf{v}_L \end{bmatrix}, \qquad \mathbf{f}(\mathbf{x}) = \begin{bmatrix} \mathbf{f}_P(\mathbf{x}) \\ \mathbf{f}_Q(\mathbf{x}) \end{bmatrix} = \begin{bmatrix} \left(P_i(\mathbf{x}) + P_{Di} - P_{Gi}\right)_{i \in A} \\ \left(Q_i(\mathbf{x}) + Q_{Di} - Q_{Gi}\right)_{i \in L} \end{bmatrix} = \mathbf{0}.$$

For one slack bus, there are $(n-1) + n_{\mathrm{PQ}}$ equations and unknowns.

A three-bus example with bus 2 PV and bus 3 PQ has

$$\mathbf{x} = \begin{bmatrix} \theta_2 \\ \theta_3 \\ v_3 \end{bmatrix}, \qquad \mathbf{f}(\mathbf{x}) = \begin{bmatrix} P_2(\mathbf{x}) - 0.55 \\ P_3(\mathbf{x}) + 0.90 \\ Q_3(\mathbf{x}) + 0.30 \end{bmatrix} = \mathbf{0}.$$

Example injections are in p.u.; the complete numerical iteration follows in the worked example.

Apply the local model to P and Q separately

At the current state, ask each linearized residual to reach zero:

$$0 = f_{Pi} + \sum_{k \in A} \frac{\partial f_{Pi}}{\partial \theta_k} \Delta\theta_k + \sum_{k \in L} \frac{\partial f_{Pi}}{\partial v_k} \Delta v_k, \quad i \in A,$$

$$0 = f_{Qi} + \sum_{k \in A} \frac{\partial f_{Qi}}{\partial \theta_k} \Delta\theta_k + \sum_{k \in L} \frac{\partial f_{Qi}}{\partial v_k} \Delta v_k, \quad i \in L.$$

These are the same local correction equations used in Part 1.

All residuals and derivatives are evaluated at $\mathbf{x}^{(r)}$. The equalities impose zero on the **linear model**; the original nonlinear equations still need to be checked.

> **One voltage correction can affect several residuals. Solve the selected equations together.**

Collect the scalar equations into two vector equations

Stack the P residuals and Q residuals separately:

$$\mathbf{f}_P + \mathbf{H}\Delta\theta_A + \mathbf{N}\Delta\mathbf{v}_L = \mathbf{0},$$

$$\mathbf{f}_Q + \mathbf{M}\Delta\theta_A + \mathbf{L}\Delta\mathbf{v}_L = \mathbf{0}.$$

Each matrix collects one set of derivative coefficients:

$$\mathbf{H} = \frac{\partial \mathbf{f}_P}{\partial \theta_A}, \quad \mathbf{N} = \frac{\partial \mathbf{f}_P}{\partial \mathbf{v}_L}, \quad \mathbf{M} = \frac{\partial \mathbf{f}_Q}{\partial \theta_A}, \quad \mathbf{L} = \frac{\partial \mathbf{f}_Q}{\partial \mathbf{v}_L}.$$

Rows identify residual equations; columns identify voltage unknowns. Stacking changes the notation, not the local model.

Stack them into the Newton matrix equation

Move the current residuals to the right-hand side:

$$\underbrace{\begin{bmatrix} \mathbf{H} & \mathbf{N} \\ \mathbf{M} & \mathbf{L} \end{bmatrix}}_{\mathbf{J}(\mathbf{x}^{(r)})} \underbrace{\begin{bmatrix} \Delta\theta_A \\ \Delta\mathbf{v}_L \end{bmatrix}}_{\Delta\mathbf{x}} = - \underbrace{\begin{bmatrix} \mathbf{f}_P(\mathbf{x}^{(r)}) \\ \mathbf{f}_Q(\mathbf{x}^{(r)}) \end{bmatrix}}_{\mathbf{f}(\mathbf{x}^{(r)})}.$$

For the selected equations, generation and constant-power demand are fixed during this solve. Therefore, for any selected variable x_k,

$$\frac{\partial f_{Pi}}{\partial x_k} = \frac{\partial P_i(\mathbf{x})}{\partial x_k}, \qquad \frac{\partial f_{Qi}}{\partial x_k} = \frac{\partial Q_i(\mathbf{x})}{\partial x_k}.$$

This is exactly $\mathbf{J}\Delta\mathbf{x} = -\mathbf{f}$. The familiar power derivatives are derivatives of our residual equations.

Add the correction and check the original equations

Solve the linear system, then update only the selected voltage unknowns:

$$\theta_i^{(r+1)} = \theta_i^{(r)} + \Delta\theta_i \quad (i \in A), \qquad v_i^{(r+1)} = v_i^{(r)} + \Delta v_i \quad (i \in L).$$

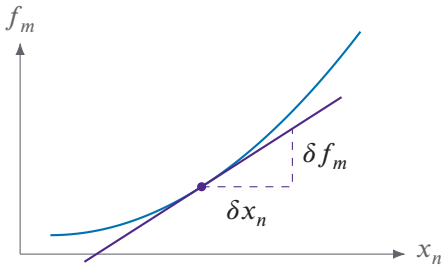
Evaluate the original nonlinear power balances at the new state:

$$f_{Pi}(\mathbf{x}^{(r+1)}) = P_i(\mathbf{x}^{(r+1)}) + P_{Di} - P_{Gi},$$
$$f_{Qi}(\mathbf{x}^{(r+1)}) = Q_i(\mathbf{x}^{(r+1)}) + Q_{Di} - Q_{Gi}.$$

Accept the equation solution when $\|\mathbf{f}(\mathbf{x}^{(r+1)})\|_\infty \le \varepsilon$.

Otherwise, rebuild J at the new state and repeat. A zero linear-model residual does not imply a zero nonlinear residual.

What a Jacobian entry means



schematic; other state entries fixed

With other state entries fixed, one local slope predicts

$$J_{mn} = \frac{\partial f_m}{\partial x_n}, \qquad \delta f_m \approx J_{mn}\,\delta x_n.$$

A complete row combines all selected changes:

$$\delta f_m \approx \sum_n J_{mn}\,\delta x_n.$$

The slope can be positive or negative. δ is a small perturbation, not the Newton correction.

The derivatives depend on the present voltage state. Rebuild J after each correction.

Jacobian blocks

	$\Delta\theta$	$\Delta \mathbf{v}_L$
$\mathbf{f}_P$	**H** $\partial \mathbf{f}_P / \partial \theta$	**N** $\partial \mathbf{f}_P / \partial \mathbf{v}_L$
$\mathbf{f}_Q$	**M** $\partial \mathbf{f}_Q / \partial \theta$	**L** $\partial \mathbf{f}_Q / \partial \mathbf{v}_L$

Rows: f_P at non-slack buses; f_Q at PQ buses.

Columns: non-slack angles and PQ magnitudes.

$$-\begin{bmatrix}\mathbf{f}_P \\ \mathbf{f}_Q\end{bmatrix} = \begin{bmatrix}\mathbf{H} & \mathbf{N} \\ \mathbf{M} & \mathbf{L}\end{bmatrix}\begin{bmatrix}\Delta\theta \\ \Delta\mathbf{v}_L\end{bmatrix}.$$

A PV magnitude is fixed. Its Q is an output checked against generator capability.

The M block predicts changes in Q residuals due to angle corrections. Fixed slack phasors and PV magnitudes have no correction columns.

The power terms to differentiate

Write $Y_{ik} = G_{ik} + \mathrm{j}B_{ik}$, $v_i = |V_i|$, and $\theta_{ik} = \theta_i - \theta_k$. Define

$$A_{ik} = G_{ik}\cos\theta_{ik} + B_{ik}\sin\theta_{ik}, \qquad C_{ik} = G_{ik}\sin\theta_{ik} - B_{ik}\cos\theta_{ik}.$$

Separate the self term, where $\theta_{ii} = 0$:

$$P_i = G_{ii}v_i^2 + v_i \sum_{k\neq i} v_k A_{ik},$$

$$Q_i = -B_{ii}v_i^2 + v_i \sum_{k\neq i} v_k C_{ik}.$$

Here P_i and Q_i abbreviate $P_i(\mathbf{x}^{(r)})$ and $Q_i(\mathbf{x}^{(r)})$. Their derivatives also differentiate the residuals.

Neighbor angle derivatives

For $k \neq i$, only the neighbor-k term depends on θ_k. Since $\partial\theta_{ik}/\partial\theta_k = -1$,

$$\frac{\partial A_{ik}}{\partial \theta_k} = G_{ik}\sin\theta_{ik} - B_{ik}\cos\theta_{ik} = C_{ik},$$

$$\frac{\partial C_{ik}}{\partial \theta_k} = -G_{ik}\cos\theta_{ik} - B_{ik}\sin\theta_{ik} = -A_{ik}.$$

Therefore,

$$\boxed{H_{ik} = v_i v_k C_{ik}}, \qquad \boxed{M_{ik} = -v_i v_k A_{ik}}.$$

The minus sign from the angle difference is essential. Keep the signed Y-bus entries inside A and C.

Neighbor magnitude derivatives

For $k \neq i$, the corresponding terms are $v_i v_k A_{ik}$ in P and $v_i v_k C_{ik}$ in Q.

Hold every angle and other magnitude fixed. A and C do not depend on v_k:

$$N_{ik} = \frac{\partial P_i}{\partial v_k} = v_i A_{ik}, \qquad L_{ik} = \frac{\partial Q_i}{\partial v_k} = v_i C_{ik}.$$

Angle derivative

Differentiates the trigonometric factor and retains $v_i v_k$.

Magnitude derivative

Differentiates the factor v_k and retains v_i.

Use a magnitude column only if that magnitude is an unknown. A PV magnitude remains fixed.

Diagonal angle derivatives

Now $\partial\theta_{ik}/\partial\theta_i = +1$ for every neighbor. The self terms have no angle dependence:

$$H_{ii} = -v_i \sum_{k\neq i} v_k C_{ik} = -Q_i - B_{ii} v_i^2,$$

$$M_{ii} = v_i \sum_{k\neq i} v_k A_{ik} = P_i - G_{ii} v_i^2.$$

These identities follow by removing the self terms from

$$Q_i = -B_{ii} v_i^2 + v_i \sum_{k\neq i} v_k C_{ik}, \qquad P_i = G_{ii} v_i^2 + v_i \sum_{k\neq i} v_k A_{ik}.$$

Use $P_i(\mathbf{x}^{(r)})$ and $Q_i(\mathbf{x}^{(r)})$ in these identities, even before the power balances are satisfied.

Diagonal magnitude derivatives

The self term is quadratic in v_i; each neighbor term is linear in v_i.

$$N_{ii} = 2G_{ii}v_i + \sum_{k \neq i} v_k A_{ik} = \frac{P_i}{v_i} + G_{ii}v_i,$$

$$L_{ii} = -2B_{ii}v_i + \sum_{k \neq i} v_k C_{ik} = \frac{Q_i}{v_i} - B_{ii}v_i.$$

The final forms divide the original calculated power equation by $v_i > 0$ and account for the extra self-term derivative.

Neighbor formulas cannot be reused for a diagonal entry. The bus magnitude appears in both voltage factors of its self term.

Source: Glover et al., sixth edition (2017), Sec. 6.6, Table 6.5.

The complete Newton power flow iteration

1 Evaluate

Form the selected residuals $\mathbf{f}(\mathbf{x})$ from the original P and Q balances.

2 Test

Accept the equation solution if $\|\mathbf{f}\|_\infty \leq \varepsilon$. At the iteration cap, report nonconvergence.

3 Solve

Build $\mathbf{J} = \partial\mathbf{f}/\partial\mathbf{x}$ at the same state. Solve $\mathbf{J}\Delta\mathbf{x} = -\mathbf{f}$.

4 Update

Add the correction to the selected angles and magnitudes, then return to Step 1.

Preserve the fixed slack phasor and PV magnitudes. Use a linear-system solver; do not explicitly form the matrix inverse.

Recover the powers that were not specified

After the selected residuals converge, evaluate the remaining balances:

$$P_{G,s} = P_s(\mathbf{x}) + P_{D,s} \quad \text{at the slack bus } s,$$

$$Q_{Gi} = Q_i(\mathbf{x}) + Q_{Di} \quad \text{at PV and slack buses.}$$

Now check generator reactive limits and other operating requirements.

If a PV generator reaches a Q limit, fix its Q at the limit, release its voltage magnitude, and rebuild the selected equations as a PQ bus.

> **The inner Newton solve uses fixed bus types. A converged equation solution still needs the operating checks.**

Source: Glover et al., sixth edition, Sec. 6.6; Zimmerman et al., IEEE Trans. Power Syst., 2011, Sec. III-A.

Angle units in the Newton system

The derivatives used in Newton assume radians:

$$\frac{\mathrm{d}\sin\theta}{\mathrm{d}\theta} = \cos\theta, \qquad \frac{\mathrm{d}\cos\theta}{\mathrm{d}\theta} = -\sin\theta.$$

If a degree variable d is used instead, $\theta = (\pi/180)d$, so

$$\frac{\partial \boldsymbol{P}}{\partial d} = \frac{\pi}{180}\frac{\partial \boldsymbol{P}}{\partial \theta}.$$

We use radians in the state, Jacobian, and updates. Convert angles to degrees only when reporting them.

For example, a correction of -0.2 rad is -11.4592°. Its numerical value cannot be transferred between units unchanged.

Summary

1. Power balance gives $f_{Pi} = P_i(\mathbf{x}) + P_{Di} - P_{Gi} = 0$ and the corresponding Q equation.
2. Select non-slack angles and PQ magnitudes; stack the matching residuals into $\mathbf{f}(\mathbf{x})$.
3. Differentiate those residuals to build $\mathbf{J}$, then solve $\mathbf{J}\Delta\mathbf{x} = -\mathbf{f}$.
4. Update the voltages and check the original residuals; repeat until the tolerance is met.

Power flow applies the same Newton method developed in Part 1. The worked example will carry out each operation numerically.