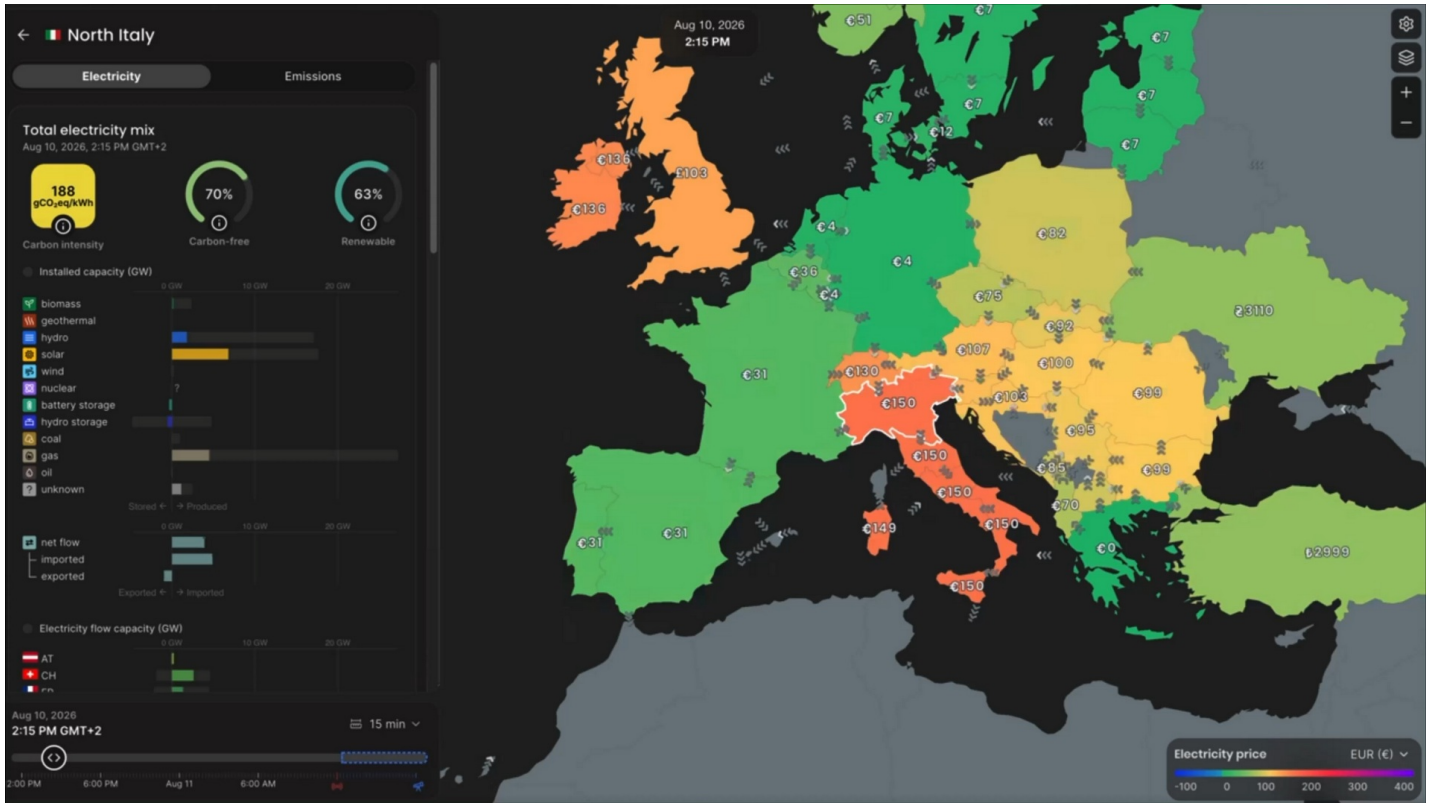
Power Flow Analysis – Part 1

ELEN90060

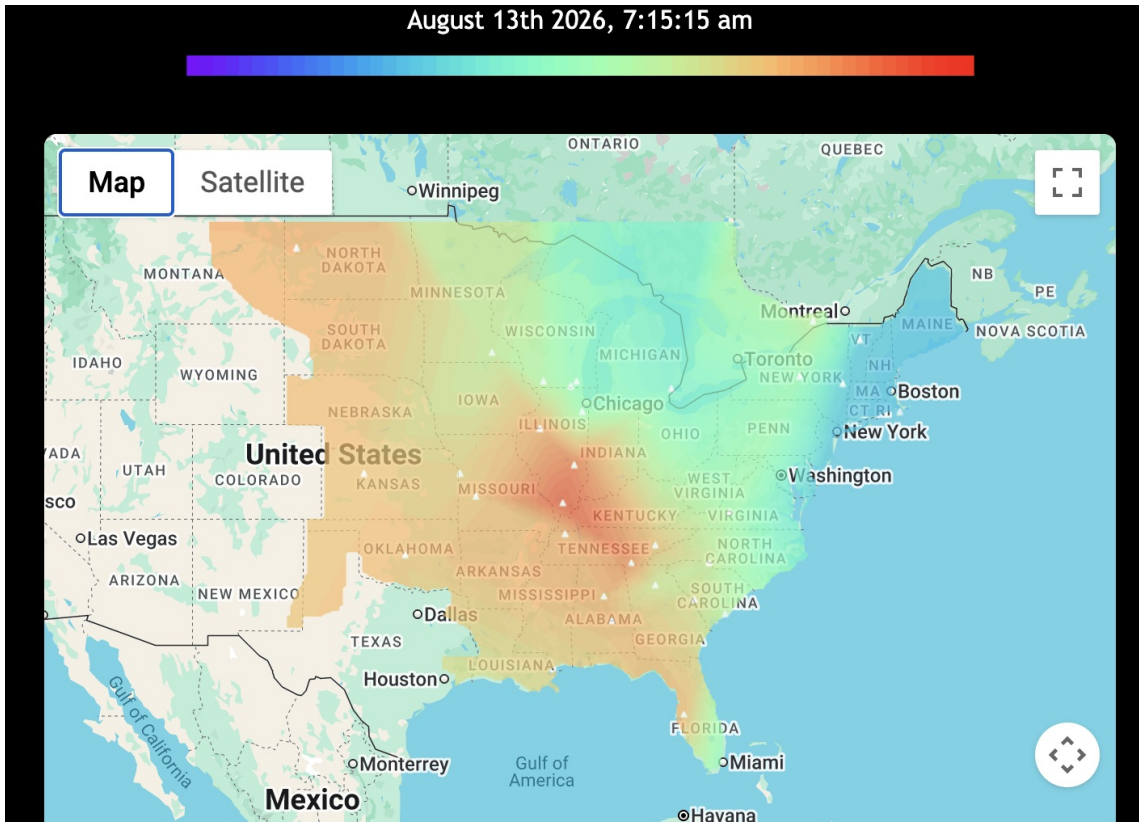
Keith Moffat

This recording demonstrates electricity prices, which are determined by power flows and constraints



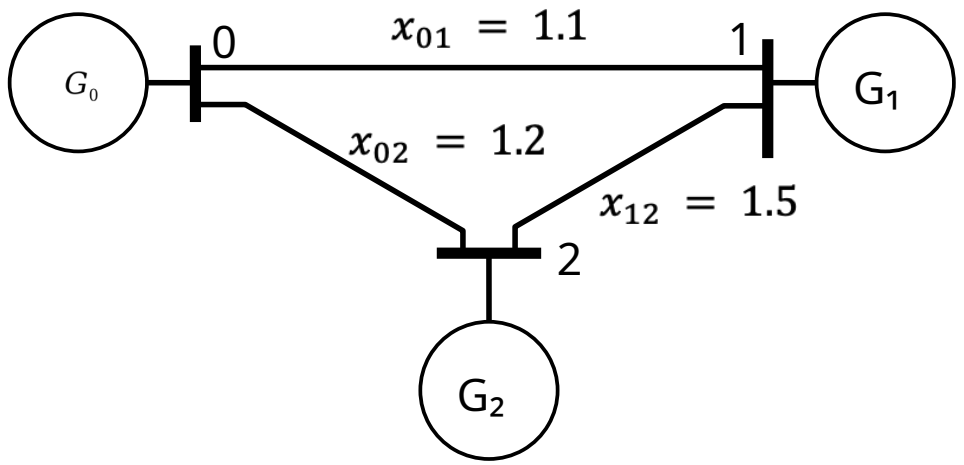
But how do we know how much power is flowing throughout the network?

This map shows the measured electricity angles across the Eastern United States.



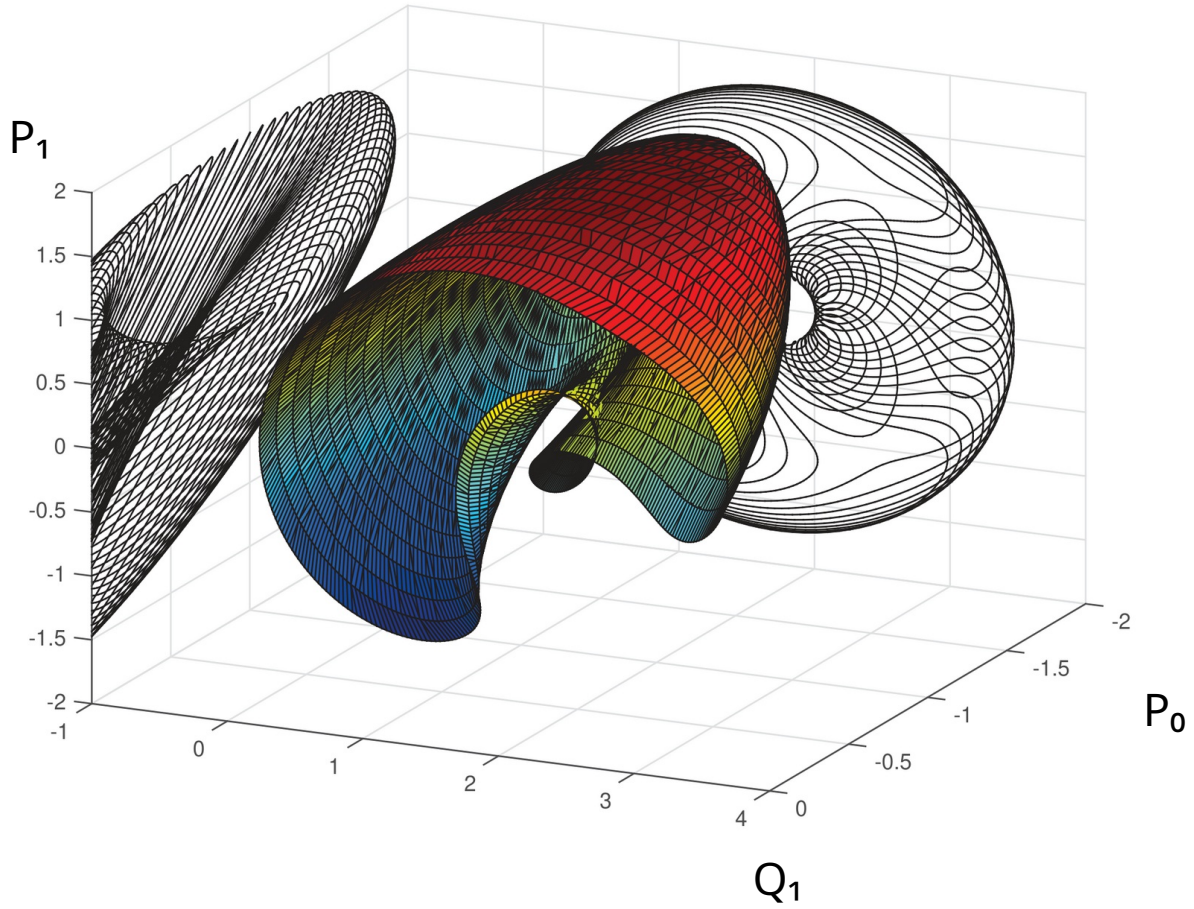
https://fnetpublic.utk.edu/anglecontour.html

Can we predict these electricity angles from the power injections (generation and load)?



$$V_0 = V_1 = V_2 = 1.05$$

Power flow manifold



https://doi.org/10.1109/59.932273
https://molzahn.github.io/pubs/molzahn_hiskens-fnt2019.pdf#page=23

Power Flow Problem Abstraction

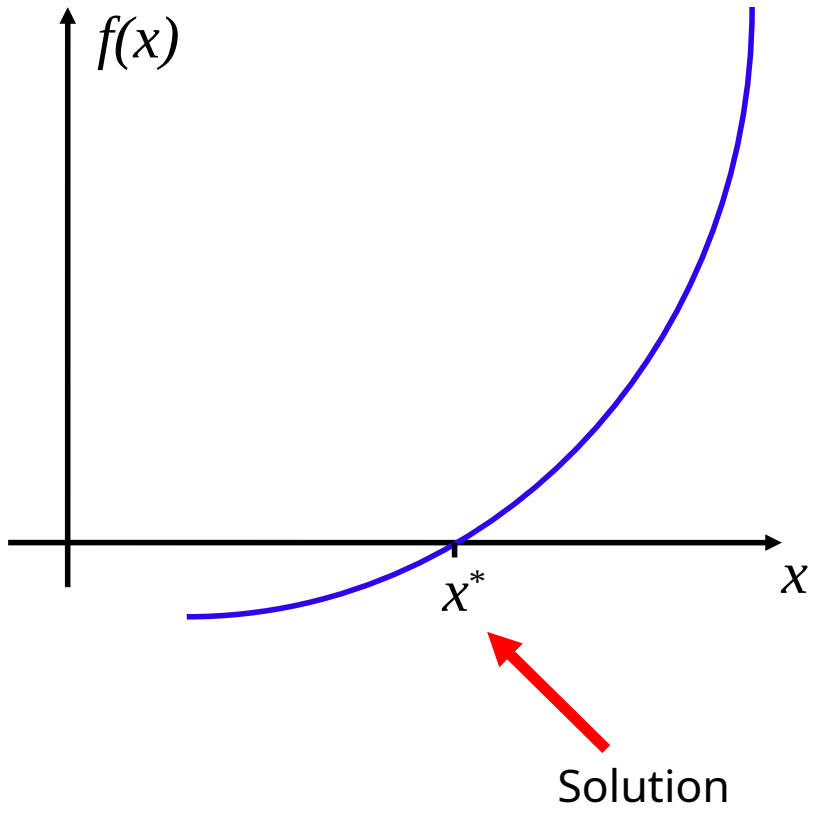
Sets of non-linear equations have no direct solution

→ use an iterative solution method

Newton-Raphson is one such method

Toy example: one function of one variable

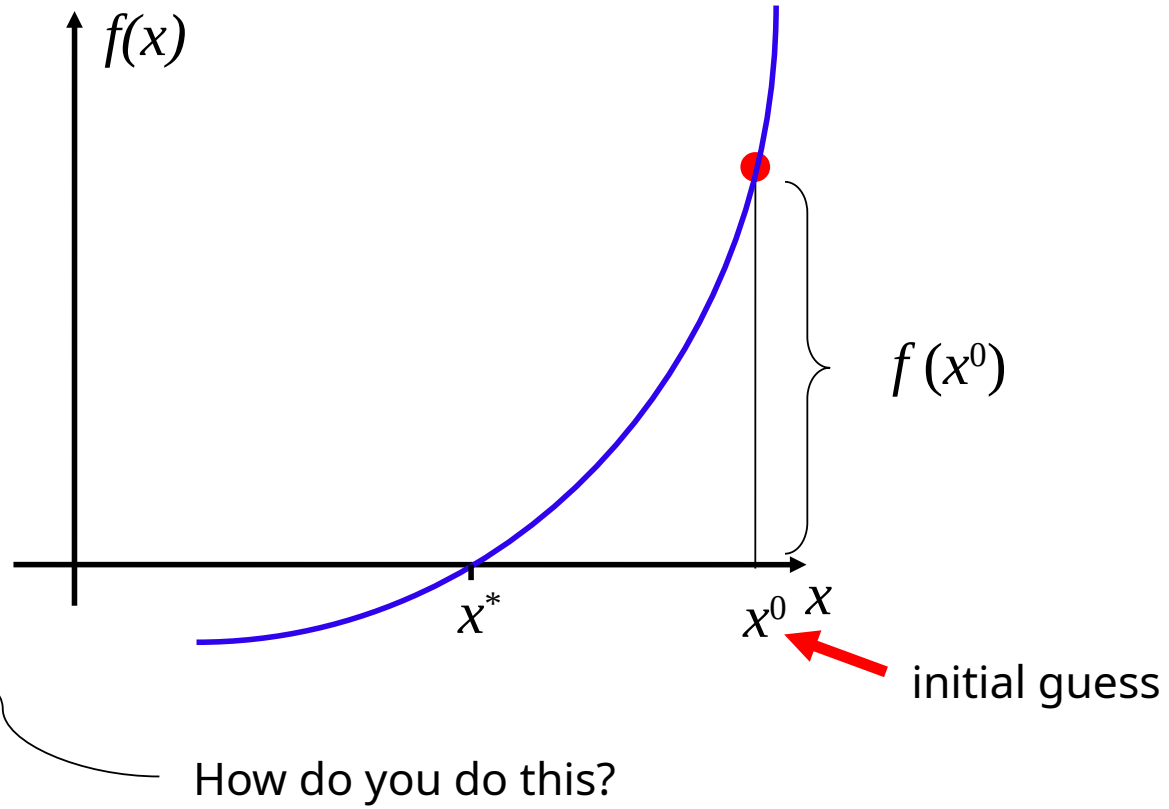
Solve $f(x) = 0$ where $f(x)$ is a non-linear function of scalar variable x



Newton-Raphson Algorithm 1/2

Context: you

- want to find where $f(x) = 0$
- do not have a plot (e.g., on the right)
- but can query $f(x)$ and its derivative $f'(x)$



Approach:

- Start with an initial guess
- Adjust the guess to decrease the error
- Repeat until the error is small enough

How do you do this?

Newton-Raphson Algorithm

Began as a hand-computation rule for finding roots of algebraic equations.

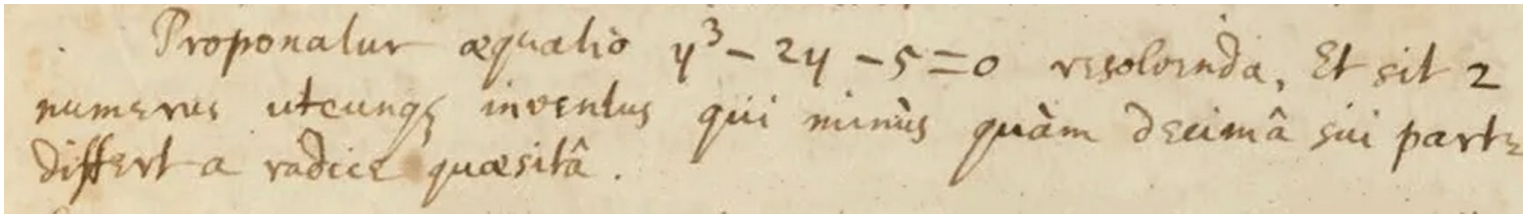
Newton developed a root-finding procedure.

Raphson published a more compact iterative form in 1690.

Simpson later connected the idea to calculus notation and systems of equations.

Today, it’s one of the most important algorithms in scientific computing and optimization.





Insight 1/2

A large fraction of scientific computation reduces to solving linear systems of the form $Ax = b$

Computers are extremely effective at linear algebra,

→ nonlinear problems are often solved by reducing them to a sequence of linear systems

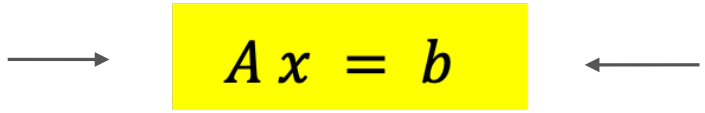
Insight 2/2

Newton's method: reduce hard, nonlinear problem to iterated arithmetic

Humans

- Can solve linear systems (slowly)
- Can only guess-and-check nonlinear systems

→ does not scale



Computers

- Can solve linear systems (fast)
- Can only guess-and-check nonlinear systems

→ does not scale

Idea: use linear system solves to get progressively closer to the solution

This idea allowed

- Newton, later Raphson, to solve scalar nonlinear equations
- Later, Simpson used to solve vector (many functions of many variables) nonlinear equations
- The invention of the computer + sparse matrix theory extended this to thousands and now millions of variables

Taylor Series Expansion

With an initial estimate x^0, our goal is to solve for Δx^0 such that

$$f(x^*) = f(x^0 + \Delta x^0) = 0$$

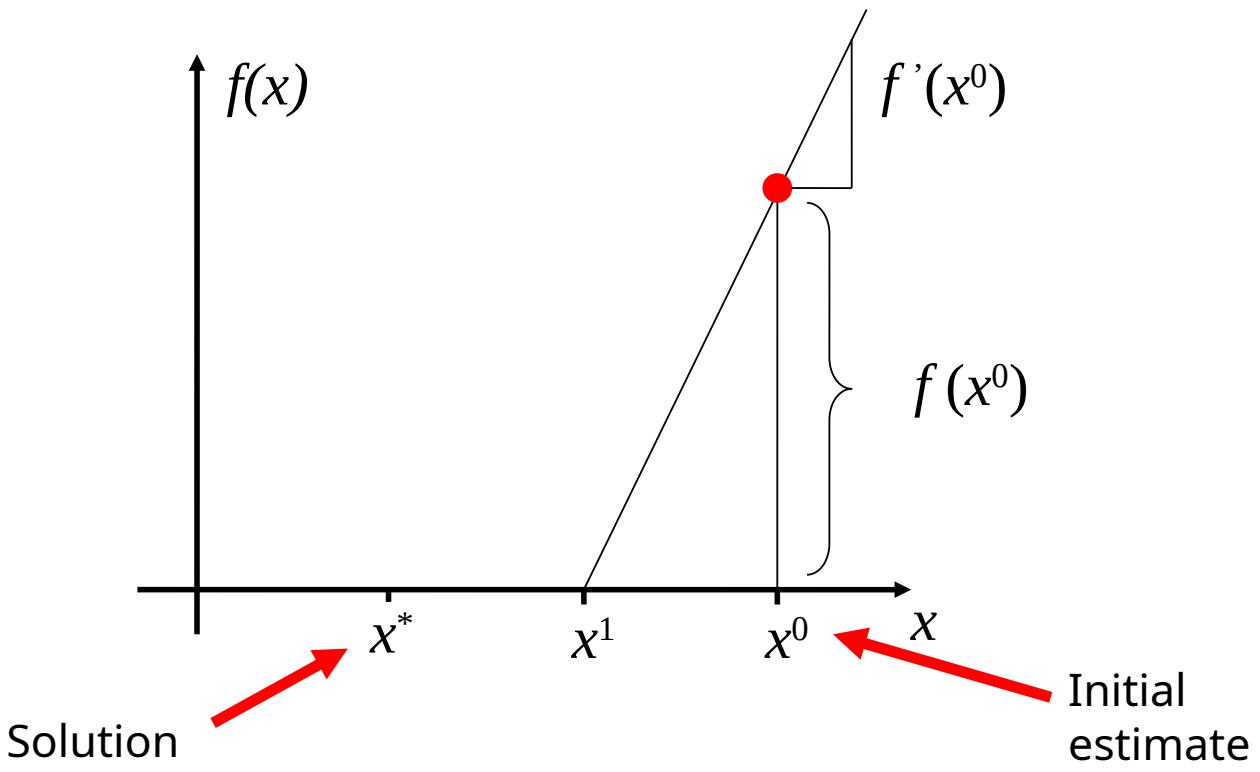
Taylor series:

$$f(x^*) = \underbrace{f(x^0) + f'(x^0) \cdot \Delta x^0}_{\text{linear approximation}} + \underbrace{\frac{1}{2} f''(x^0) \cdot (\Delta x^0)^2 + \ldots}_{\text{higher order terms}}$$

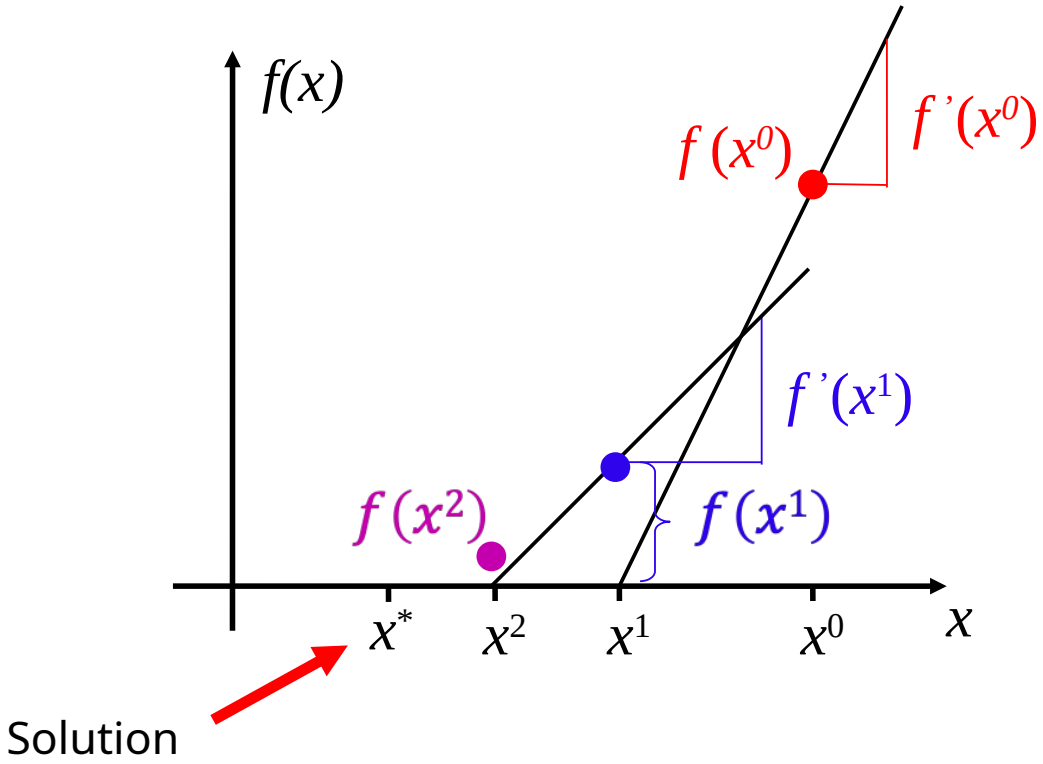
Solving the linear portion $f(x^0) + f'(x^0) \cdot \Delta x^0 = 0$

provides an approximate solution to $f(x^*) = f(x^0 + \Delta x^0) = 0$

→ iterate


f(x)
f ’(x0)
f (x0)
x*
x1
x0
x
Solution
Initial
estimate

Newton-Raphson Toy Demonstration



Newton-Raphson Algorithm 2/2

Initialization: pick an initial guess $x^{(0)}$

Repeat, until convergence:

1. Solve: $-f'(x^{(k)})\Delta x^{(k)} = f(x^{(k)})$
2. Determine the next guess: $x^{(k+1)} = x^{(k)} + \Delta x^{(k)}$
3. Query the nonlinear function to get $f(x^{(k)})$

$$-f'\left(x^{(k)}\right)\Delta x^{(k)} = f\left(x^{(k)}\right)$$

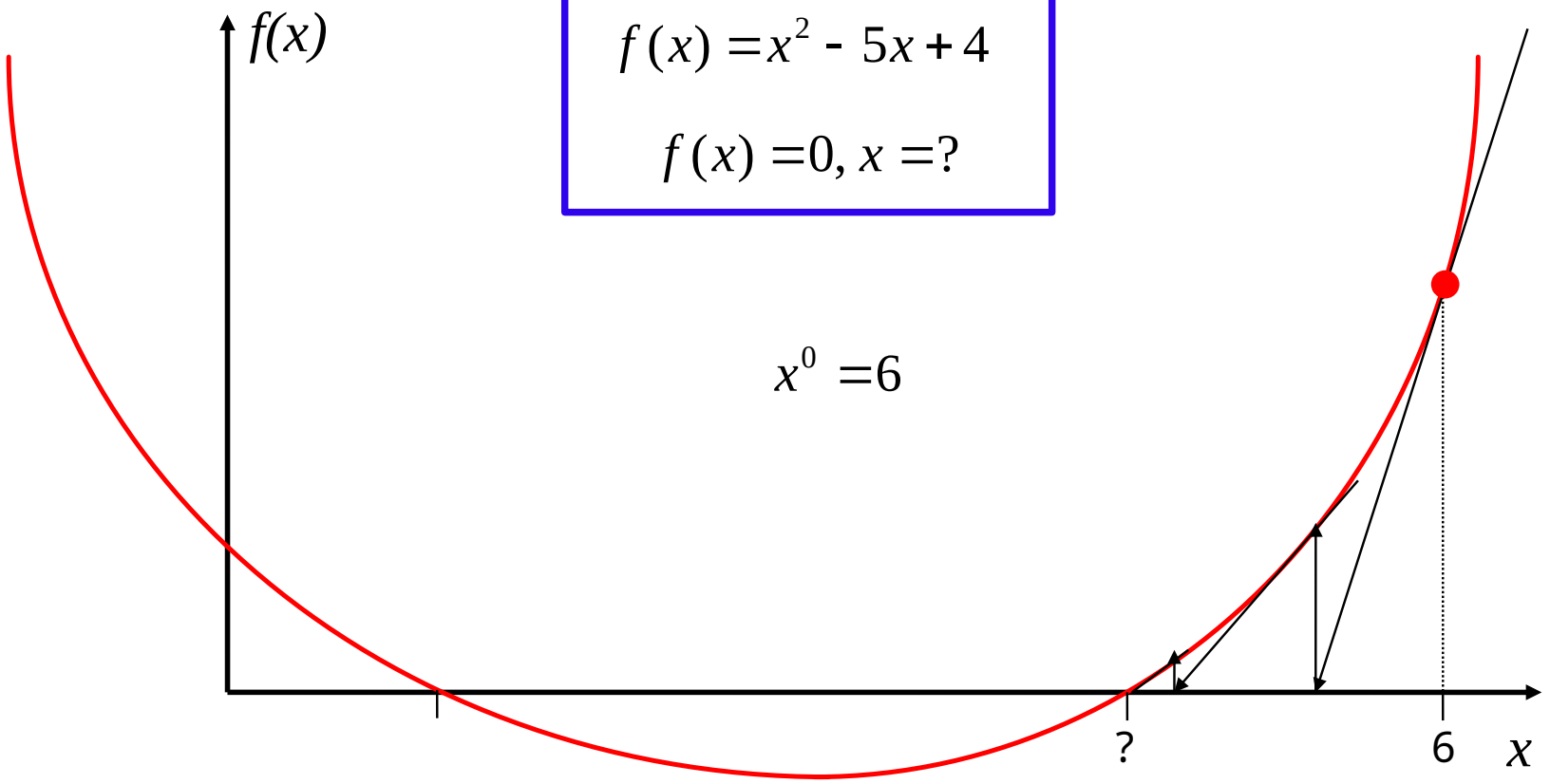
$$x^{(k+1)} = x^{(k)} + \Delta x^{(k)}$$

One function one variable example 1/5

$$f(x) = x^2 - 5x + 4$$

$$f(x) = 0,\ x = ?$$

$$x^0 = 6$$



One function one variable example 2/5

$$f(x) = x^2 - 5x + 4 = 0 \qquad x^{(0)} = 6$$

$$-f'(x^{(k)})\Delta x^{(k)} = f(x^{(k)})$$

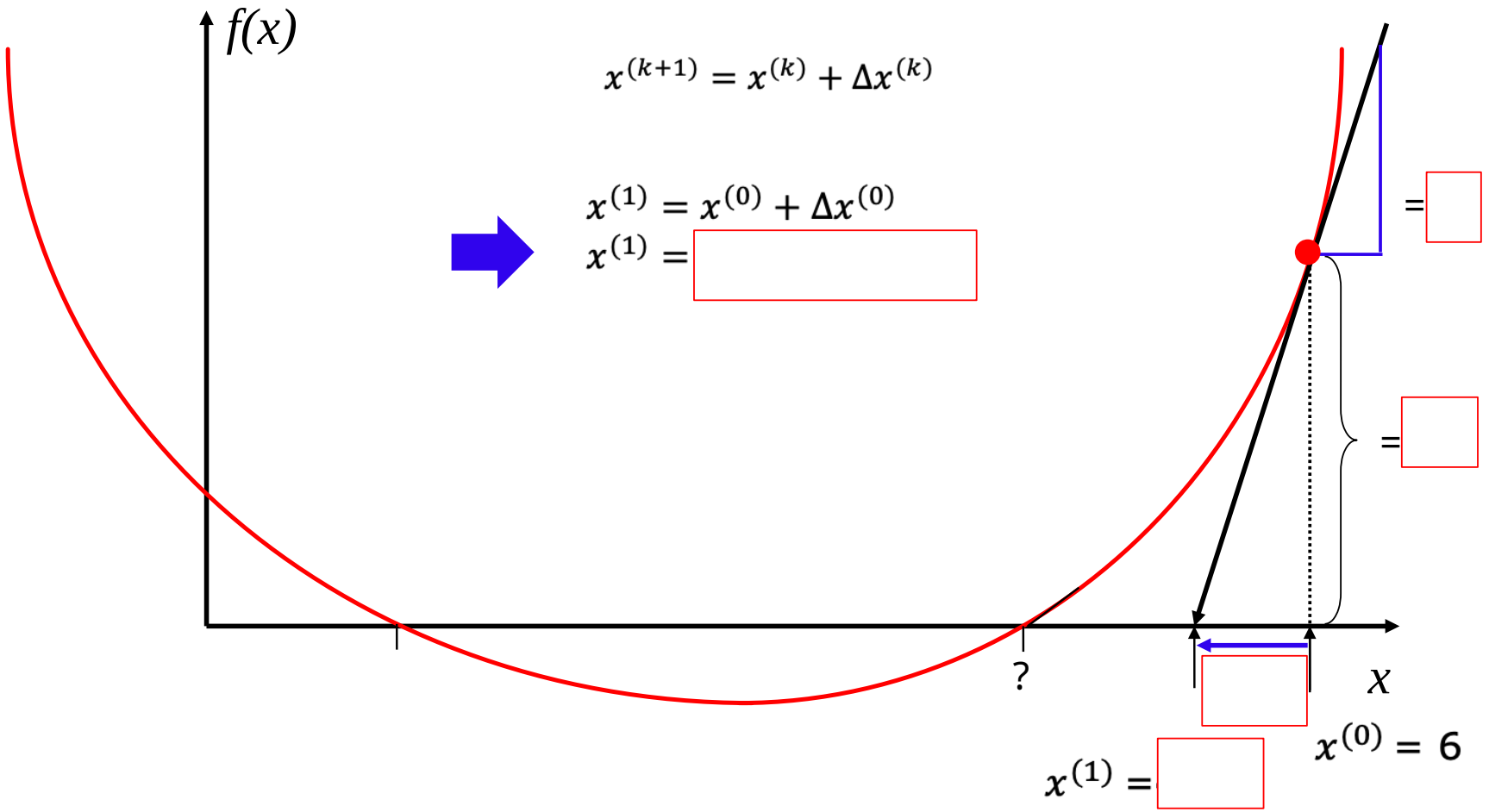
$$x^{(k+1)} = x^{(k)} + \Delta x^{(k)}$$

$$\frac{df(x)}{dx} = 2x - 5 \qquad f'(x^{(0)}) = f'(6) = 2x - 5 = \square$$

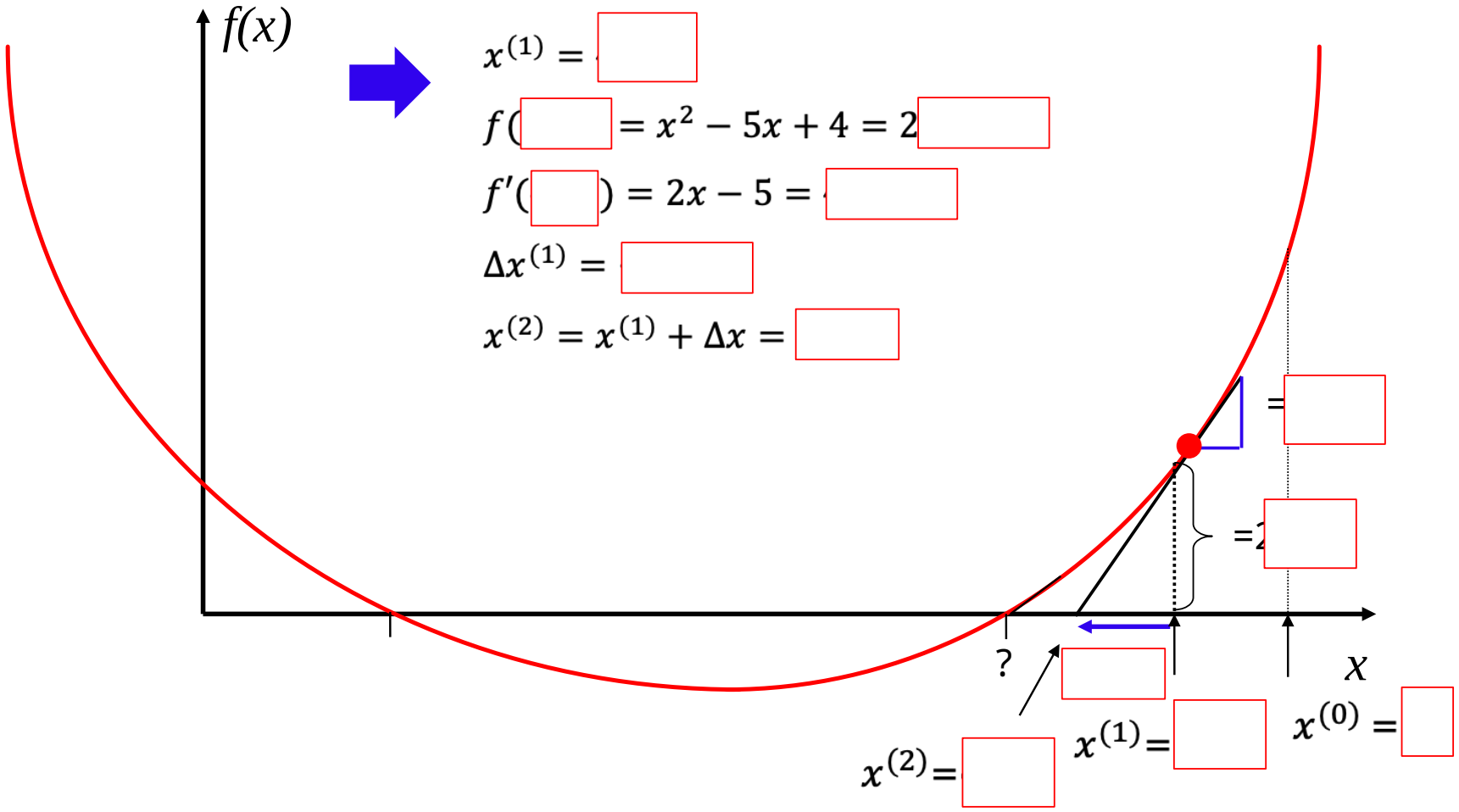
$$f(x^{(0)}) = f(6) = x^2 - 5x + 4 = \square$$

$$\square\, \Delta x^{(0)} = \square \qquad \Delta x^{(0)} = \square = \square$$

One function one variable example 3/5



One function one variable 4/5



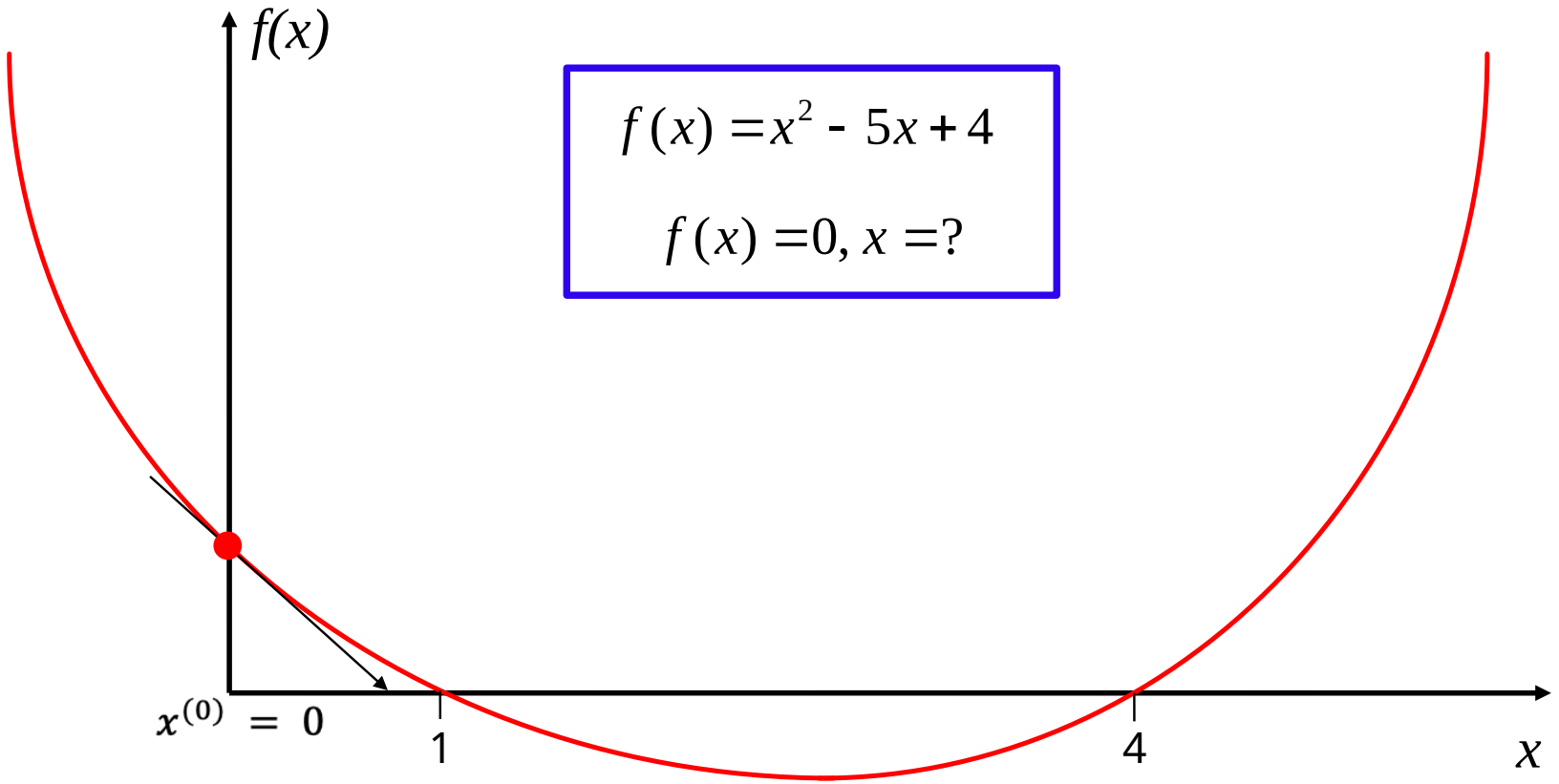
One function one variable example 5/5

$$f(x) = x^2 - 5x + 4$$

$$f(x) = 0,\ x = ?$$

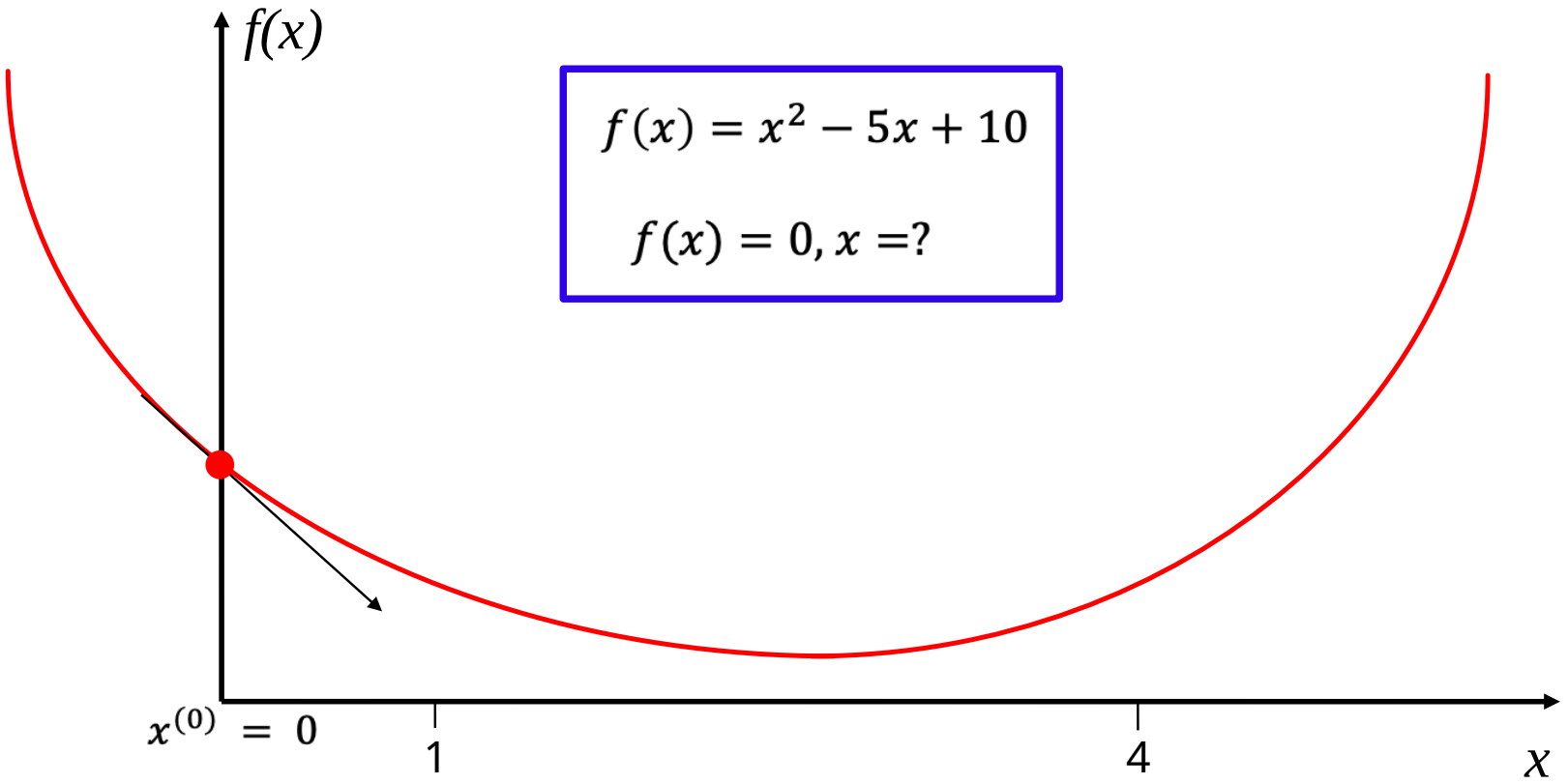
k	$x^{(k)}$	$f(x)$	$f'(x)$	$\Delta x^{(k)}$	$x^{(k+1)}$
0	6.000	10.000	7.000	−1.429	4.571
1	4.571				
2					

One function one variable, Different x^0



The solution x to $f(x) = 0$, where $f(x)$ is a non-linear function, can depend on the initial guess!

One function one variable, solution existence



The solution x to $f(x) = 0$, where $f(x)$ is a non-linear function, can fail to exist!

Newton-Raphson Algorithm Guarantees?

Initialization: pick an initial guess $x^{(0)}$

Repeat, until convergence:

1. Solve: $-f'(x^{(k)})\Delta x^{(k)} = f(x^{(k)})$
2. Determine the next guess: $x^{(k+1)} = x^{(k)} + \Delta x^{(k)}$
3. Query the nonlinear function

Are there general solution guarantees?

No.

But, Newton-Raphson tends to work well in practice.

Two Functions of Two Variables 1/3

Find x and y such that $\begin{cases} f(x,y) = 0 \\ g(x,y) = 0 \end{cases}$

Taylor's series expansion (neglecting higher order terms and with x^k and y^k as estimates):

$$\begin{cases} f(x^*,y^*) \cong f(x^k,y^k) + \dfrac{\partial f(x^k,y^k)}{\partial x} \cdot \Delta x^k + \dfrac{\partial f(x^k,y^k)}{\partial y} \cdot \Delta y^k \\ g(x^*,y^*) \cong g(x^k,y^k) + \dfrac{\partial g(x^k,y^k)}{\partial x} \cdot \Delta x^k + \dfrac{\partial g(x^k,y^k)}{\partial y} \cdot \Delta y^k \end{cases}$$

where $\begin{cases} x^{k+1} = x^k + \Delta x \\ y^{k+1} = y^k + \Delta y \end{cases}$

Two Functions of Two Variables 2/3

If we consider x^k and y^k as the solution, we must have:

$$\begin{cases} f(x^k, y^k) + \dfrac{\partial f(x^k, y^k)}{\partial x} \cdot \Delta x^k + \dfrac{\partial f(x^k, y^k)}{\partial y} \cdot \Delta y^k = 0 \\ g(x^k, y^k) + \dfrac{\partial g(x^k, y^k)}{\partial x} \cdot \Delta x^k + \dfrac{\partial g(x^k, y^k)}{\partial y} \cdot \Delta y^k = 0 \end{cases}$$

or

$$\underbrace{\begin{bmatrix} \dfrac{\partial f(x^k, y^k)}{\partial x} & \dfrac{\partial f(x^k, y^k)}{\partial y} \\ \dfrac{\partial g(x^k, y^k)}{\partial x} & \dfrac{\partial g(x^k, y^k)}{\partial y} \end{bmatrix}}_{\text{Jacobian } J^{(x^k, y^k)}} \cdot \begin{bmatrix} \Delta x^k \\ \Delta y^k \end{bmatrix} = - \begin{bmatrix} f(x^k, y^k) \\ g(x^k, y^k) \end{bmatrix}$$

Two Functions of Two Variables 3/3

$$\mathrm{J}^{(x^k,y^k)} \cdot \begin{bmatrix} \Delta x^k \\ \Delta y^k \end{bmatrix} = - \begin{bmatrix} f(x^k, y^k) \\ g(x^k, y^k) \end{bmatrix}$$

$$\begin{bmatrix} \Delta x^k \\ \Delta y^k \end{bmatrix} = - \left(\mathrm{J}^{(x^k,y^k)} \right)^{-1} \cdot \begin{bmatrix} f(x^k, y^k) \\ g(x^k, y^k) \end{bmatrix}$$



$$\begin{cases} x^{k+1} = x^k + \Delta x^k \\ y^{k+1} = y^k + \Delta y^k \end{cases}$$

Example | Two Functions of Two Variables

Find x and y such that

$$\begin{cases} f(x,y) = x + y - 15 = 0 \\ g(x,y) = x \cdot y - 50 = 0 \end{cases}$$

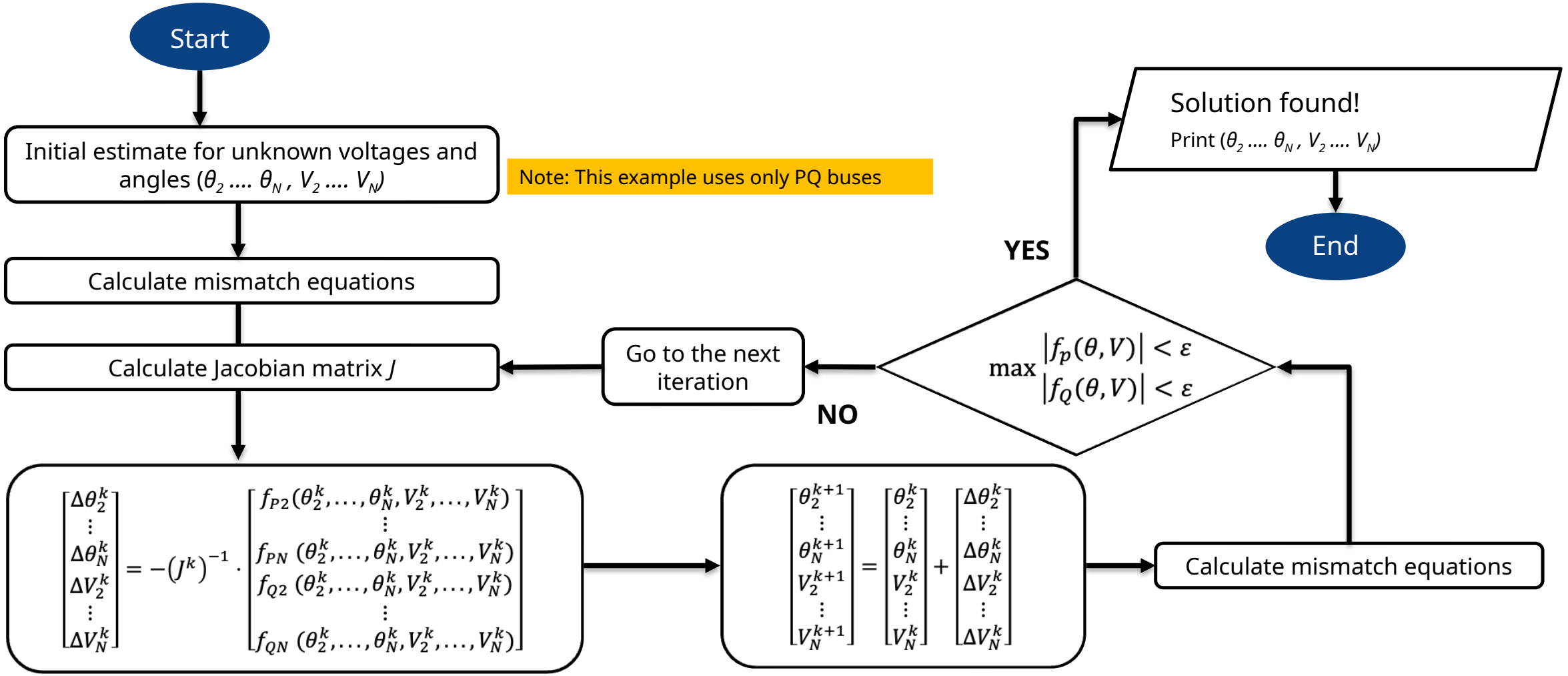
Start from: $x^0 = 4 \; ; \; y^0 = 9$

Iterative expression:

$$\begin{bmatrix} \Delta x^k \\ \Delta y^k \end{bmatrix} = -\left(\mathrm{J}^{(x^k, y^k)}\right)^{-1} \cdot \begin{bmatrix} f(x^k, y^k) \\ g(x^k, y^k) \end{bmatrix}$$

$$\mathrm{J} = \begin{bmatrix} \frac{\partial f}{\partial x} & \frac{\partial f}{\partial y} \\ \frac{\partial g}{\partial x} & \frac{\partial g}{\partial y} \end{bmatrix} = \begin{bmatrix} \quad \end{bmatrix}$$

Newton-Raphson for Power Flow: The Process



Newton-Raphson for Power Flow: The Variables

General notation:

$$x = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$$

Power flow equations:

Voltage angles at non-slack buses

$$\theta_P = \begin{bmatrix} \theta_2 \\ \vdots \\ \theta_N \end{bmatrix}$$

Number of PQ buses: $(N-1)$

Voltage magnitudes at PQ buses

$$V_Q = \begin{bmatrix} V_{n_{PV}+2} \\ \vdots \\ V_N \end{bmatrix}$$

Number of PQ buses: $(N - n_{PV} - 1)$

$$x = \begin{bmatrix} \theta_P \\ V_Q \end{bmatrix}$$

Newton-Raphson for Power Flow: The Equations

General notation:

$$f(x) = \begin{bmatrix} f_1(x) \\ \vdots \\ f_n(x) \end{bmatrix}$$

The Bus Injection Equations:

$$P_k^{injected} = \sum_{i \in N_k} V_k V_i [G_{ki} \cos\theta_{ki} + B_{ki} \sin\theta_{ki}]$$

$$Q_k^{injected} = \sum_{i \in N_k} V_k V_i [G_{ki} \sin\theta_{ki} - B_{ki} \cos\theta_{ki}]$$

... but which of these equations to use for which buses?

Newton-Raphson for Power Flow: The Equations

(special case of no PV-buses)

For the special case of no PV-buses, all buses (except the slack) are PQ buses.

$$\begin{cases} P_2(\theta_2, \dots, \theta_N, V_2, \dots, V_N) \\ \vdots \\ P_N(\theta_2, \dots, \theta_N, V_2, \dots, V_N) \\ Q_2(\theta_2, \dots, \theta_N, V_2, \dots, V_N) \\ \vdots \\ Q_N(\theta_2, \dots, \theta_N, V_2, \dots, V_N) \end{cases}$$

- 2xN power flow equations
- 2xN unknown variables
- 2x(N-1) implicit equations
- 2 explicit equations:

  $P_1(\theta_2, \dots, \theta_N, V_2, \dots, V_N)$

  $Q_1(\theta_2, \dots, \theta_N, V_2, \dots, V_N)$

The Power Flow Jacobian Matrix J
(special case of no PV-buses)

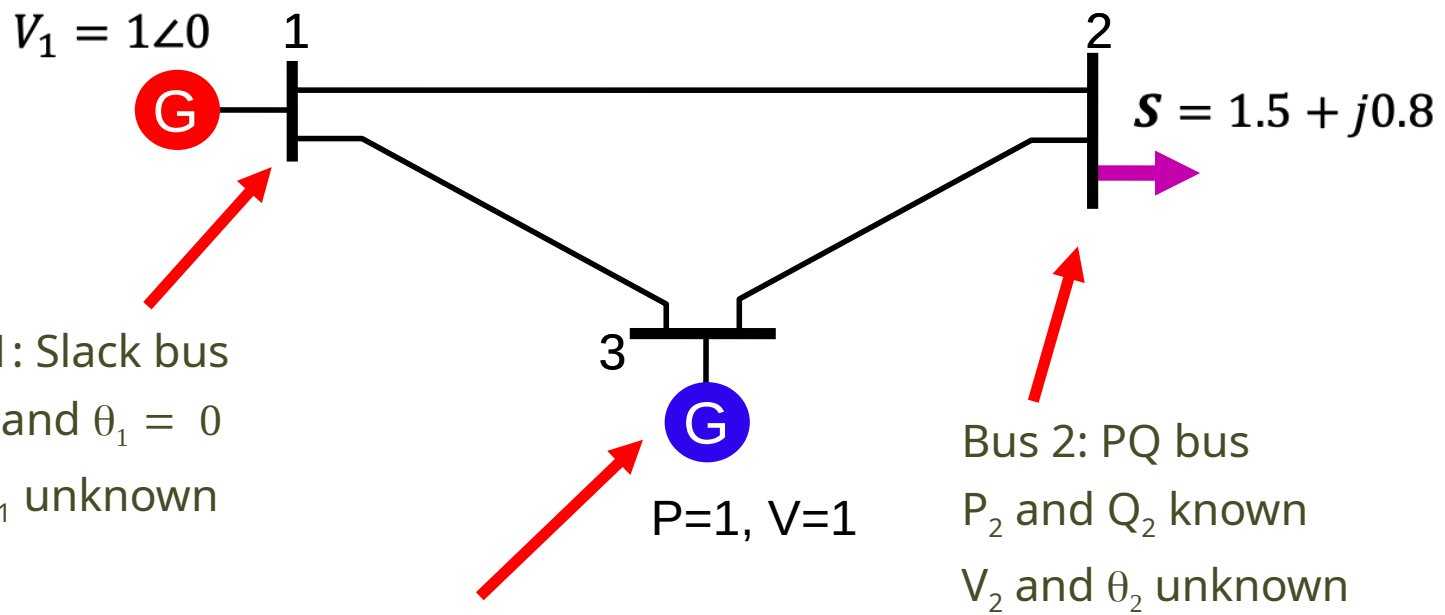
General notation:

$$J(x) = \begin{bmatrix} \frac{\partial f_1(x)}{\partial x_1} & \cdots & \frac{\partial f_1(x)}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n(x)}{\partial x_1} & \cdots & \frac{\partial f_n(x)}{\partial x_n} \end{bmatrix}$$

Power flow equations:

$$J(V,\theta) = \begin{bmatrix} \frac{\partial P_2}{\partial \theta_2} & \cdots & \frac{\partial P_2}{\partial \theta_N} & \frac{\partial P_2}{\partial V_2} & \cdots & \frac{\partial P_2}{\partial V_N} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \frac{\partial P_N}{\partial \theta_2} & \cdots & \frac{\partial P_N}{\partial \theta_N} & \frac{\partial P_N}{\partial V_2} & \cdots & \frac{\partial P_N}{\partial V_N} \\ \frac{\partial Q_2}{\partial \theta_2} & \cdots & \frac{\partial Q_2}{\partial \theta_N} & \frac{\partial Q_2}{\partial V_2} & \cdots & \frac{\partial Q_2}{\partial V_N} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \frac{\partial Q_N}{\partial \theta_2} & \cdots & \frac{\partial Q_N}{\partial \theta_N} & \frac{\partial Q_N}{\partial V_2} & \cdots & \frac{\partial Q_N}{\partial V_N} \end{bmatrix}$$

Three-Bus Example



Bus 1: Slack bus

V_1=1 and $\theta_1 = 0$

P_1, Q_1 unknown

Bus 3: PV bus

P_3 and V_3 known

Q_3 and θ_3 unknown

Bus 2: PQ bus

P_2 and Q_2 known

V_2 and θ_2 unknown

6 unknowns

Three-Bus Example

The Newton-Raphson iterations are applied to the implicit equations

Implicit equations: known power injections

P at non-slack buses → "P_P"

$$P_2 = 10V_2V_1 \sin\theta_{21} + 4V_2V_3 \sin\theta_{23}$$
$$P_3 = 5V_3V_1 \sin\theta_{31} + 4V_3V_2 \sin\theta_{32}$$

Q at PQ buses → "Q_Q"

$$Q_2 = -10V_2V_1 \cos\theta_{21} + 14V_2^2 - 4V_2V_3 \cos\theta_{23}$$

3 equations
3 unknowns

solve these

then solve these

Explicit equations: unknown power injections

$$P_1 = 10V_1V_2 \sin\theta_{12} + 5V_1V_3 \sin\theta_{13}$$
$$Q_1 = 15V_1^2 - 10V_1V_2 \cos\theta_{12} - 5V_1V_3 \cos\theta_{13}$$
$$Q_3 = -5V_3V_1 \cos\theta_{31} - 4V_3V_2 \cos\theta_{32} + 9V_3^2$$

3 equations
3 unknowns

Implicit Equations → Mismatch Equations

$f_P(\theta, V)$ and $f_Q(\theta, V)$ are the active and reactive power mismatch equations

$$f_P(\theta, V) \equiv P_P(\theta, V) - P_P = 0$$
$$f_Q(\theta, V) \equiv Q_Q(\theta, V) - Q_Q = 0$$

non-slack bus injections calculated from Power Flow Equations

given injections for non-slack buses (constant)

given injections for PQ buses (constant)

PQ bus injections calculated from Power Flow Equations

The power flow solution is the values of V and θ that satisfy the mismatch equations

Goal: find the voltages such that the mismatch equations equal zero

The Power Flow Jacobian Matrix J

For an n-bus system with 1 slack bus, n_{PV} PV buses, and n_{PQ} PQ buses,

- variables: $(n-1)$ unknown voltage angles and n_{PQ} unknown voltage magnitudes, and
- implicit equations: $(n-1)$ for active power and n_{PQ} for reactive power.

Thus, the Power Flow Jacobian has dimension $(n-1+n_{PQ})\times(n-1+n_{PQ})$

denotes matrix dimension

non-slack-bus angles

PQ-bus voltages

$$\frac{\partial \begin{bmatrix} P_P \\ Q_Q \end{bmatrix}}{\partial \begin{bmatrix} \theta_P \\ V_Q \end{bmatrix}} = \begin{bmatrix} \frac{\partial P_P}{\partial \theta_P} & \frac{\partial P_P}{\partial V_Q} \\ \frac{\partial Q_Q}{\partial \theta_P} & \frac{\partial Q_Q}{\partial V_Q} \end{bmatrix} = \begin{bmatrix} J_1\left(\begin{bmatrix} \theta_P \\ V_Q \end{bmatrix}\right) & J_2\left(\begin{bmatrix} \theta_P \\ V_Q \end{bmatrix}\right) \\ J_3\left(\begin{bmatrix} \theta_P \\ V_Q \end{bmatrix}\right) & J_4\left(\begin{bmatrix} \theta_P \\ V_Q \end{bmatrix}\right) \end{bmatrix}$$

$\mathbb{R}^{(n-1)\times(n-1)}$ $\mathbb{R}^{(n-1)\times n_{PQ}}$ $\mathbb{R}^{n_{PQ}\times(n-1)}$ $\mathbb{R}^{n_{PQ}\times n_{PQ}}$

Solving the Jacobian linear system

Δx is calculated by solving the linear system:

$$-J(x)\Delta x = f(x)$$

$$-\begin{bmatrix} J_1 & J_2 \\ J_3 & J_4 \end{bmatrix}\begin{bmatrix} \Delta\theta_P \\ \Delta V_Q \end{bmatrix} = \begin{bmatrix} f_P(\theta, V) \\ f_Q(\theta, V) \end{bmatrix}$$

One way to do this is by inverting the Jacobian:

$$\Delta x = -J^{-1}(x)f(x)$$

In practice, there are faster ways of solving linear systems than matrix inversion.

Newton-Raphson power Flow

Plugging into the general Newton-Raphson form,

$$-f'(x^{(k)})\Delta x^{(k)} = f(x^{(k)})$$

$$x^{(k+1)} = x^{(k)} + \Delta x^{(k)}$$

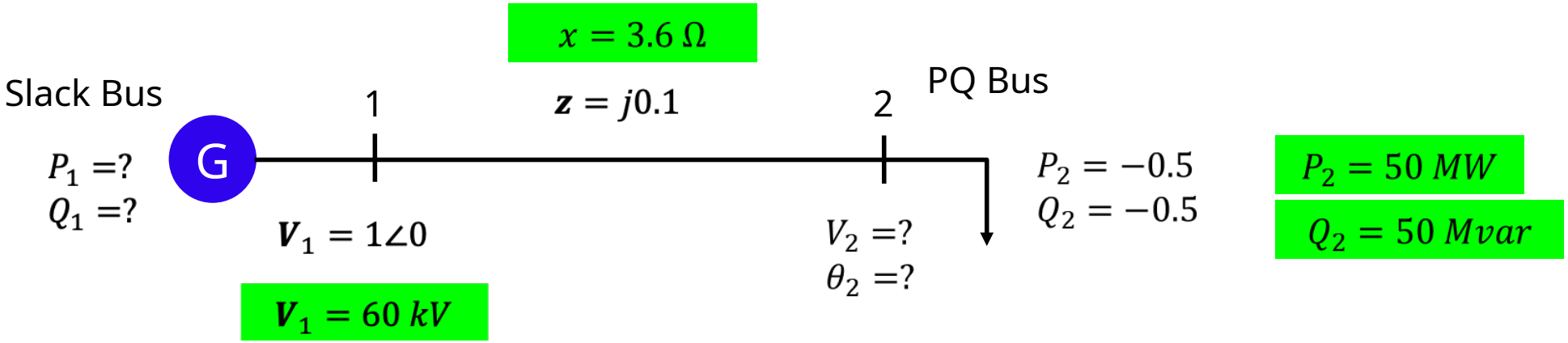
We get the Newton-Raphson Power Flow algorithm:

$$-\begin{bmatrix} J_1\left(\theta_P^{(k)}, V_Q^{(k)}\right) & J_2\left(\theta_P^{(k)}, V_Q^{(k)}\right) \\ J_3\left(\theta_P^{(k)}, V_Q^{(k)}\right) & J_4\left(\theta_P^{(k)}, V_Q^{(k)}\right) \end{bmatrix} \begin{bmatrix} \Delta\theta_P^{(k)} \\ \Delta V_Q^{(k)} \end{bmatrix} = \begin{bmatrix} f_P\left(\theta^{(k)}, V^{(k)}\right) \\ f_Q\left(\theta^{(k)}, V^{(k)}\right) \end{bmatrix}$$

$$\begin{bmatrix} \theta_P^{(k+1)} \\ V_Q^{(k+1)} \end{bmatrix} = \begin{bmatrix} \theta_P^{(k)} \\ V_Q^{(k)} \end{bmatrix} + \begin{bmatrix} \Delta\theta_P^{(k)} \\ \Delta V_Q^{(k)} \end{bmatrix}$$

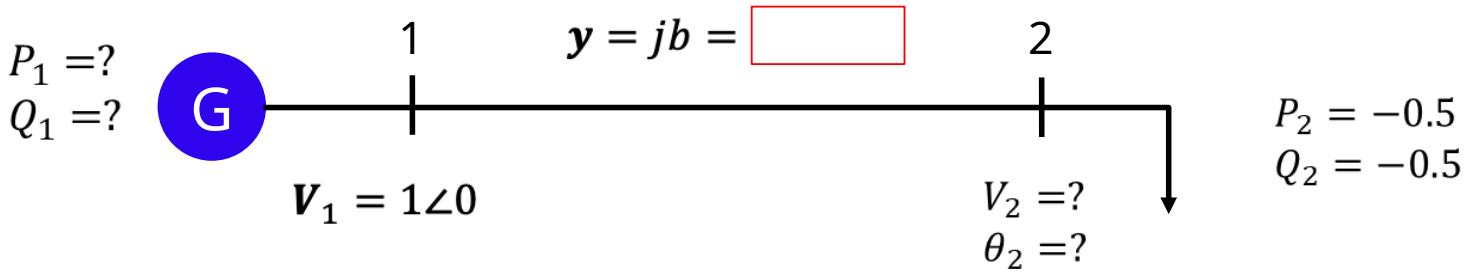
Once the iterations cor
we plug into the explicit equations to get the unknown power injections.

Two Bus Example | Power Flow



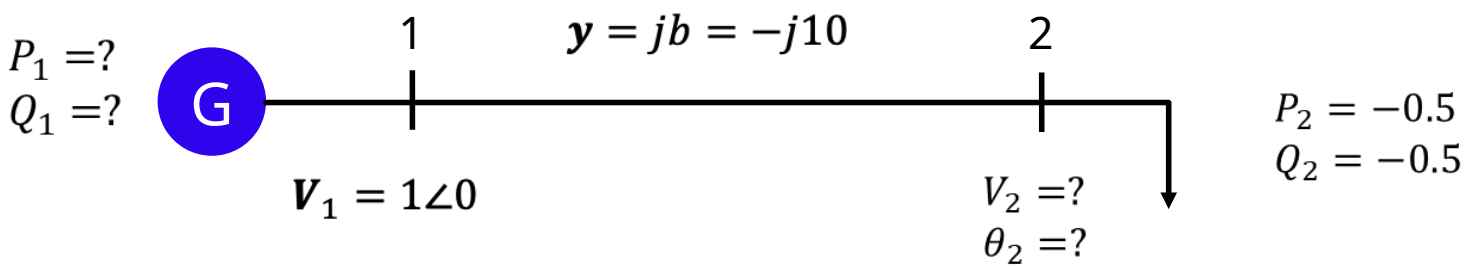
Values in per unit and actual units

Two Bus Example | Admittance Matrix



$$\boldsymbol{Y} = G + jB =$$

Two Bus Example | Power Flow Equations 1/2



$$P_k^{injected} = \sum_{i \in N_k} V_k V_i [G_{ki} \cos\theta_{ki} + B_{ki} \sin\theta_{ki}]$$

$$Q_k^{injected} = \sum_{i \in N_k} V_k V_i [G_{ki} \sin\theta_{ki} - B_{ki} \cos\theta_{ki}]$$

$$\begin{cases} P_1 = \\ Q_1 = \\ P_2 = \\ Q_2 = \end{cases}$$

Two Bus Example | Power Flow Equations 2/2

Implicit equations: compute V_2 and θ_2

$$P_2 = -b \cdot V_1 \cdot V_2 \cdot \sin(\theta_2)$$
$$Q_2 = b \cdot V_1 \cdot V_2 \cdot \cos(\theta_2) - b \cdot {V_2}^2$$

Explicit equations: compute P_1 and Q_1, once we know V_2 and θ_2

$$P_1 = -b \cdot V_1 \cdot V_2 \cdot \sin(-\theta_2)$$
$$Q_1 = -b \cdot {V_1}^2 + b \cdot V_1 \cdot V_2 \cdot \cos(\theta_2)$$

Two Bus Example | Implicit Equations → Mismatch Equations

$f_P(\theta, V)$ and $f_Q(\theta, V)$ are the active and reactive power mismatch equations

$$f_P(\theta, V) \equiv 10 \cdot V \cdot \sin(\theta) + 0.5 = 0$$
$$f_Q(\theta, V) \equiv -10 \cdot V \cdot \cos(\theta) + 10 \cdot V^2 + 0.5 = 0$$

The power flow solution is the values of V and θ that satisfy the mismatch equations

Goal: find the voltages such that the mismatch equations equal zero

Example | Jacobian and Iterative Expression

$$f_P(\theta, V) \equiv 10 \cdot V \cdot \sin(\theta) + 0.5 = 0$$
$$f_Q(\theta, V) \equiv -10 \cdot V \cdot \cos(\theta) + 10 \cdot V^2 + 0.5 = 0$$

$$\begin{bmatrix} \frac{\partial f_P(\theta^k, V^k)}{\partial \theta} & \frac{\partial f_P(\theta^k, V^k)}{\partial V} \\ \frac{\partial f_Q(\theta^k, V^k)}{\partial \theta} & \frac{\partial f_Q(\theta^k, V^k)}{\partial V} \end{bmatrix} \cdot \begin{bmatrix} \Delta\theta^k \\ \Delta V^k \end{bmatrix} = - \begin{bmatrix} f_P(\theta^k, V^k) \\ f_Q(\theta^k, V^k) \end{bmatrix}$$

$$\begin{bmatrix} \Delta\theta^k \\ \Delta V^k \end{bmatrix} = -J(\theta^k, V^k)^{-1} \begin{bmatrix} f_P(\theta^k, V^k) \\ f_Q(\theta^k, V^k) \end{bmatrix}$$

$$\mathrm{J} = \left[\quad \right]$$

Example | 1st iteration

1

$$\theta^0 = 0 \text{ rad} \; ; \; V^0 = 1 \text{ p.u.}$$ ← "Flat" start

$$f_P(\theta^0, V^0) = 10 \cdot V \cdot \sin(\theta) + 0.5 = \square$$

$$f_Q(\theta^0, V^0) = -10 \cdot V \cdot \cos(\theta) + 10 \cdot V^2 + 0.5 = \square$$

2 $$J(\theta^0, V^0) = \begin{bmatrix} \square \end{bmatrix}$$

3 $$J(\theta^0, V^0)^{-1} = \begin{bmatrix} \square \end{bmatrix}$$

4 $$\begin{bmatrix} \Delta\theta \\ \Delta V \end{bmatrix} = -\mathrm{J}^{-1} \begin{bmatrix} f_P \\ f_Q \end{bmatrix} = -\begin{bmatrix} \square \end{bmatrix} \cdot \begin{bmatrix} \square \end{bmatrix} = \begin{bmatrix} \square \end{bmatrix}$$

} For θ^0 and V^0

5 $$\begin{bmatrix} \theta^1 \\ V^1 \end{bmatrix} = \begin{bmatrix} \theta^0 \\ V^0 \end{bmatrix} + \begin{bmatrix} \Delta\theta^0 \\ \Delta V^0 \end{bmatrix} = \begin{bmatrix} \square \end{bmatrix} + \begin{bmatrix} \square \end{bmatrix} = \begin{bmatrix} \square \end{bmatrix}$$ ← θ in rad

Example | Slack bus injections

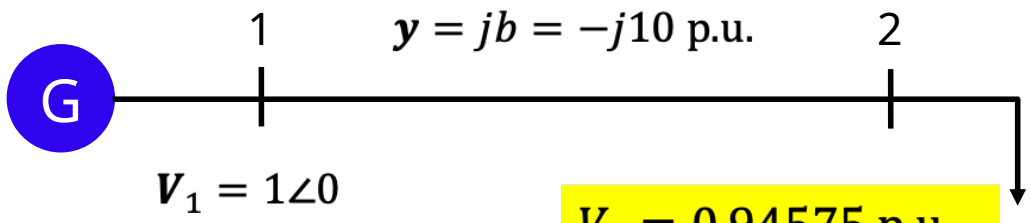
Once we have V_2 and θ_2, solving the Explicit equations gives P_1 and Q_1

$$P_1 = -b \cdot V_1 \cdot V_2 \cdot \sin(-\theta_2)$$
$$Q_1 = -b \cdot {V_1}^2 + b \cdot V_1 \cdot V_2 \cdot \cos(\theta_2)$$

Example | Solution

$P_1 = 0.4999$ p.u. $\cong 0.5$ p.u.
$Q_1 = 0.5557$ p.u.

$P_1 = 50\ MW$
$Q_1 = 55.57\ Mvar$

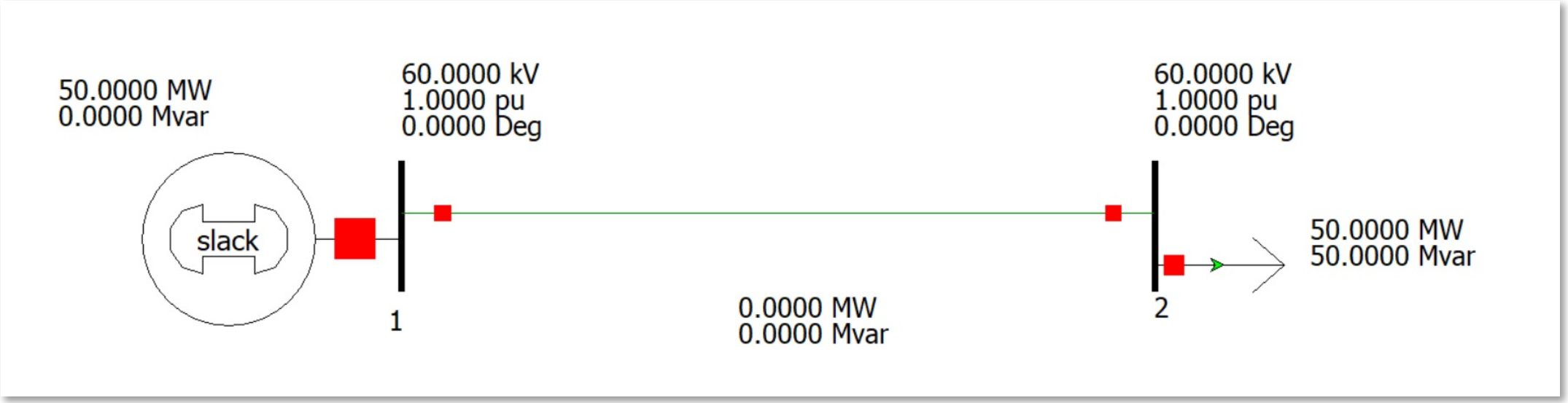


$P_2 = -0.5$ p.u.
$Q_2 = -0.5$ p.u.

$V_2 = 0.94575$ p.u.
$\theta_2 = -3.0298°$

$V_2 = 56.745\ kV$
$\theta_2 = -3.0298°$

PowerWorld is an interface for solving power flow



Voltage-Controlled Buses and Q limits

PV buses (generators) control voltages using reactive power Q, which can be limited.

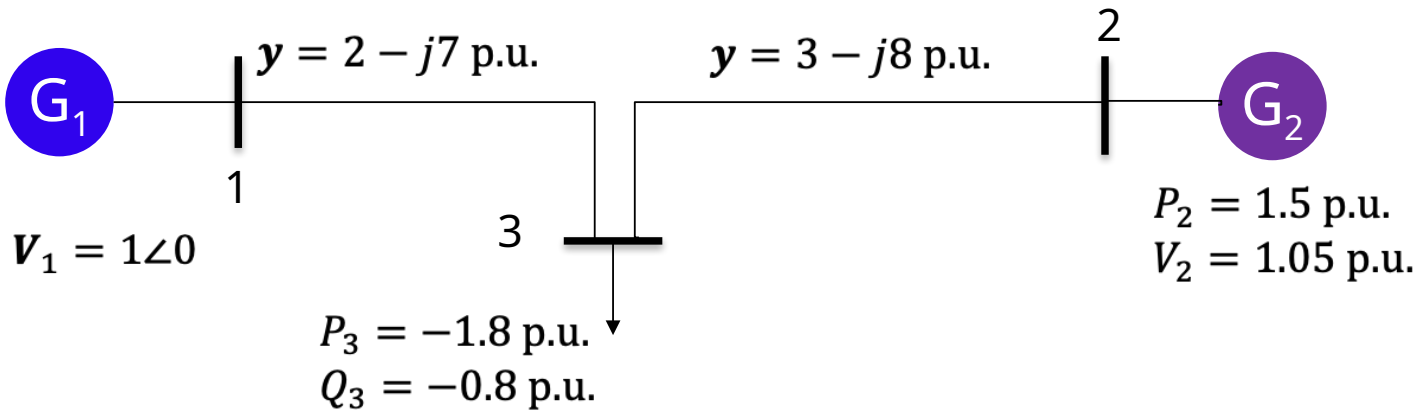
$$Q_{min,i} \leq Q_i \leq Q_{max,i} \quad \text{bus } i \text{ in the system, which is a } PV \text{ bus}$$

If within a NR iteration bus i violates its Q limits,

- the Q at bus i is held at the limit that was violated and it becomes a PQ bus
- the voltage magnitude V_i is introduced as a variable
- an additional mismatch equation is added for the Q_i injection
- the Jacobian increases by 1 row and 1 column

If in a later iteration $Q_{min,i} \leq Q_i \leq Q_{max,i}$, bus i is converted back to a PV bus by reversing the steps above.

Example | PV bus with Q limit



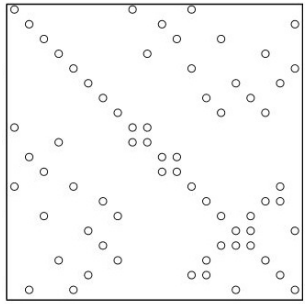
There is a reactive power limit on the generation unit located at Bus 2:

$$-0.5\,p.u. \leq Q_2 \leq 0.95\,p.u.$$

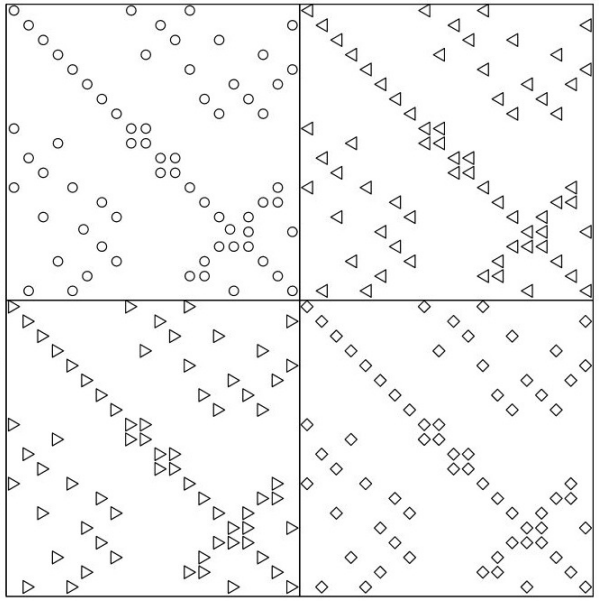
Calculate the first two iterations of Newton-Raphson, considering the Q_2 limit.

Sparsity and Solving the Newton Step

$\boldsymbol{Y}$



$$J = \begin{bmatrix} J_1 & J_2 \\ J_3 & J_4 \end{bmatrix}$$



Like the admittance matrix $\boldsymbol{Y}$, the Jacobian is sparse.

To solve $-\begin{bmatrix} J_1 & J_2 \\ J_3 & J_4 \end{bmatrix}\begin{bmatrix} \Delta\theta_P \\ \Delta V_Q \end{bmatrix} = \begin{bmatrix} f_P(\theta, V) \\ f_Q(\theta, V) \end{bmatrix}$, the Jacobian is factorized (e.g., LU), but not inverted

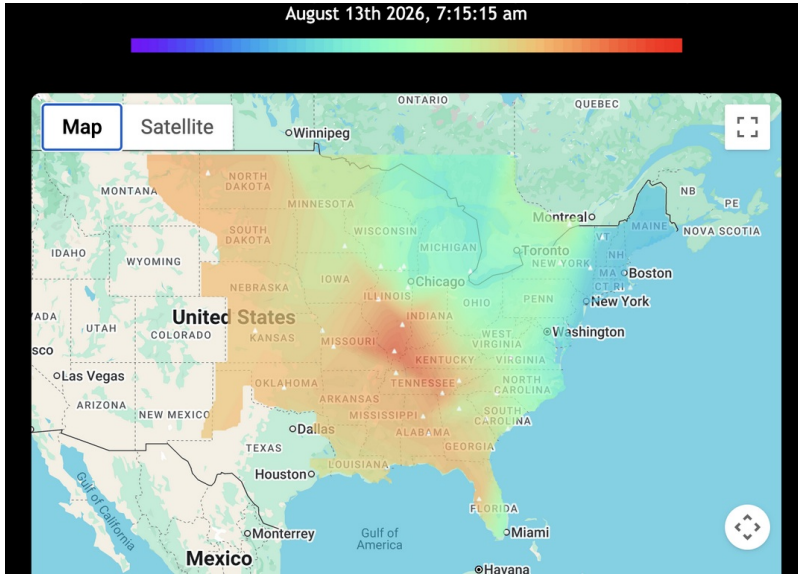
Images from: “Electric energy systems: analysis and operation” A. Gómez-Expósito, A. Conejo, C. Cañizares, 2009

Final Remarks

Newton Raphson Power Flow often has rapid convergence to a very good accuracy

- Thus, 60+ years later it is still the standard for solving power flow





The Jacobian factorization is the time-consuming part of the algorithm.

Thus, methods have been developed to avoid Jacobian factorizations (next week).

Note: Newton-Raphson is used for many applications in addition to power flow

Power Flow Analysis – Part 1

ELEN90060

Keith Moffat

The University of Melbourne – July, 2026