

Modelování elektrických sítí KEE/MS

Přednáška na téma:

# **Výpočty chodu sítě**

Ing. Jan Veleba, Ph.D.

doc. Ing. Karel Noháč, Ph.D.

# Výpočet chodu soustavy

(br. e. „load flow“, am. e. „power flow“)


- výpočet napěťových a výkonových poměrů
- výpočet celkových činných a jalových ztrát
- vhodné nastavení převodů u traf. přepínačem odboček
- vhodné nastavení generovaného činného výkonu elektrárenských bloků pro min. výrobní náklady
- plánování budoucího rozvoje ES
- analýza poruchových stavů (výskyt 1f-3f zkratů v ES)

# Výpočet chodu soustavy



software load flow analysis free



- i** [Load Flow Software | Electrical Load Flow Analysis ... - ETAP](#)  
[etap.com/load-flow-analysis/load-flow-analysis.htm](#) ▾ Přeložit tuto stránku  
ETAP Load Flow software performs power flow analysis and voltage drop ... This load flow calculation software is capable of performing analysis on both radial ...
- i** [Power Systems Analysis Software - Open Electrical](#)  
[www.openelectrical.org/.../index.php?...Power...Anal...](#) ▾ Přeložit tuto stránku  
28. 11. 2014 - Power systems analysis and simulation software are ubiquitous in electrical ... Initially, they were used to quickly solve the non-linear load flow problem and ..... AMES is a free open source software package for agent-based ...  
[Commercial Software - Advanced Grounding Concepts ... - ASPEN](#)
- i** [PowerFlow™ | Load Flow Analysis Software | EasyPower ...](#)  
[www.easypower.com/.../EasyPower/.../PowerFlow/P...](#) ▾ Přeložit tuto stránku  
Load Flow Analysis | Instantaneous on the One-Line. With the most ... PowerFlow, our load flow analysis software, offers unrivaled speed, accuracy, and user-friendly interface deliver the ability to: ... Download a free evaluation copy today.
- i** [Free Load Flow Software - InorXL - Excel based - YouTube](#)  
 [www.youtube.com/watch?v=eQz9I1B4aSQ](#) ▾  
20. 8. 2012 - Nahráno uživatelem InorXL  
This is the short introduction to simple, powerful and free load flow analysis software - InorXL. As it is based on ...
- i** [CYMFLOW Power Flow Analysis - Cyme International](#)  
[www.cyme.com](#) ▸ [Software](#) ▾ Přeložit tuto stránku  
The CYMFLOW software is the power flow analysis module of the CYME power engineering software for the analysis of three-phase electric power networks.
- i** [load flow analysis software free download - Softonic](#)  
[en.softonic.com](#) ▸ ... ▸ [Internet](#) ▸ [Traffic Analysers](#) ▾ Přeložit tuto stránku  
load flow analysis software free download - SIP Tester 3.1: SIP tester is a free VoIP load testing tool to check SIP hardware or software, and much more ...

# Výpočet chodu soustavy

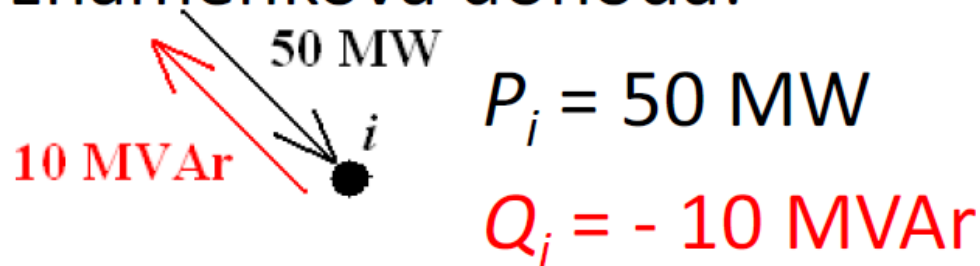
## Výhody použití metody uzlových napětí

- jednoduchá příprava vstupních dat a uzlových rovnic (menší počet uzlů sítě, méně parametrů a rovnic)
- příčné prvky nezvyšují počet neznámých a rovnic popisující procházející větвовé proudy
- snadné změny převodu u regulačních transformátorů
- uzlová napětí jsou dána řešením, větвовé proudy se snadno dopočtou

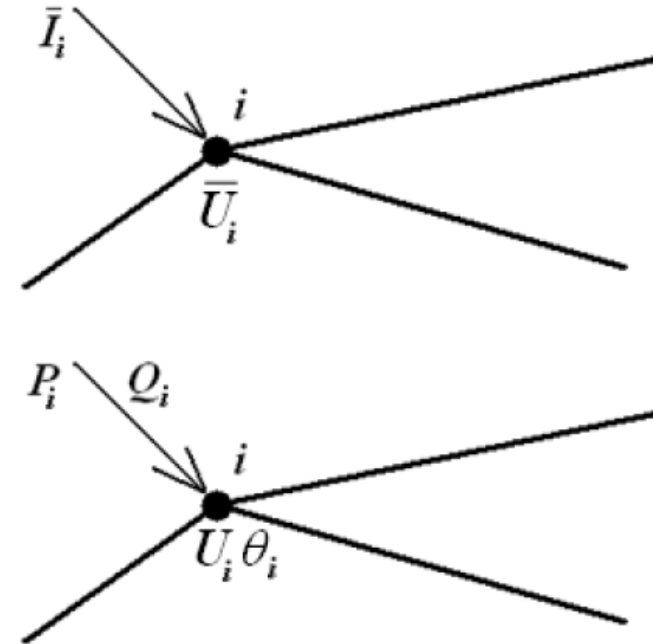
# Výpočet chodu soustavy

## Uzel soustavy

znaménková dohoda:



vzájemný vztah: 
$$\bar{I}_i = \frac{P_i - jQ_i}{\sqrt{3}U_i^*}$$



Typ uzlu:

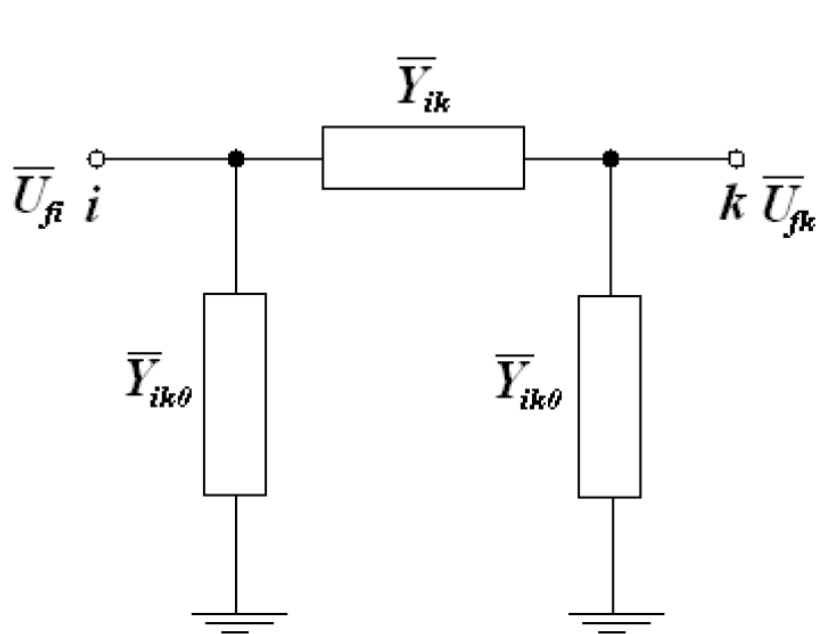
- PQ uzel (tzv. odběrový) –
- PV, PU uzel (tzv. elektrárenský)
- referenční uzel –

# Výpočet chodu soustavy

Definice admitanční/impedanční matice

$$\sqrt{3}\underline{I} = \underline{A} \cdot \underline{U} \quad \text{nebo} \quad \underline{U} = \sqrt{3}\underline{Z} \cdot \underline{I}$$

Síť tvořená vedeními (ve tvaru  $\Pi$ -čláčku)



$$\bar{Z}_{ik} = R_{ik} + jX_{ik} =$$

$$= R_{ik} [\Omega / km] \ell_{ik} [km] +$$

$$+ jX_{ik} [\Omega / km] \ell_{ik} [km] [\Omega]$$

$$\bar{Y}_{ik0} = G_{ik0} + jB_{ik0} =$$

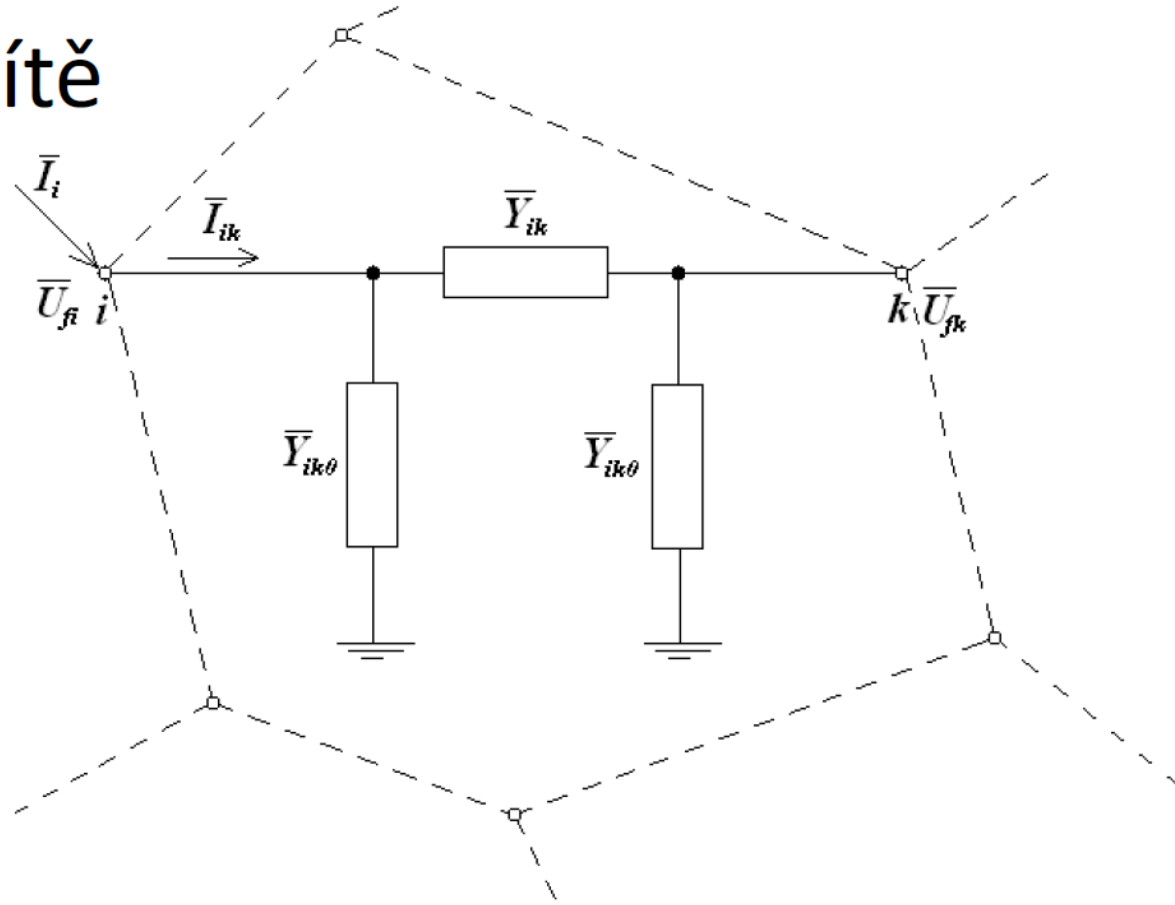
$$= G_{ik0} [S / km] \frac{\ell_{ik} [km]}{2} +$$

$$+ jB_{ik0} [S / km] \frac{\ell_{ik} [km]}{2} [S]$$

$$\bar{Y}_{ik} = \frac{1}{Z_{ik}} = \frac{1}{R_{ik} + jX_{ik}} = \frac{R_{ik}}{R_{ik}^2 + X_{ik}^2} - j \frac{X_{ik}}{R_{ik}^2 + X_{ik}^2} = G_{ik} + jB_{ik} [S]$$

# Výpočet chodu soustavy

Detail el. sítě



$$\begin{aligned}\bar{I}_{ik} &= \bar{U}_{fi} \bar{Y}_{ik0} + (\bar{U}_{fi} - \bar{U}_{fk}) \bar{Y}_{ik} = \bar{U}_{fi} (\bar{Y}_{ik0} + \bar{Y}_{ik}) - \bar{U}_{fk} \bar{Y}_{ik} \\ \rightarrow \bar{I}_i &= \sum_{\substack{k=1 \\ k \neq i}}^n \bar{I}_{ik} = \bar{U}_{fi} \sum_{\substack{k=1 \\ k \neq i}}^n (\bar{Y}_{ik0} + \bar{Y}_{ik}) - \sum_{\substack{k=1 \\ k \neq i}}^n \bar{U}_{fk} \bar{Y}_{ik}\end{aligned}$$

# Výpočet chodu soustavy

Srovnáním s definicí admitanční matice:

$$\bar{I}_i = \sum_{\substack{k=1 \\ k \neq i}}^n \bar{I}_{ik} = \bar{U}_{fi} \sum_{\substack{k=1 \\ k \neq i}}^n (\bar{Y}_{ik0} + \bar{Y}_{ik}) - \sum_{\substack{k=1 \\ k \neq i}}^n \bar{U}_{fk} \bar{Y}_{ik}$$

$$\bar{I}_i = \bar{A}_{i1} \bar{U}_{f1} + \bar{A}_{i2} \bar{U}_{f2} + \dots + \bar{A}_{ii} \bar{U}_{fi} + \dots + \bar{A}_{ik} \bar{U}_{fk} + \dots + \bar{A}_{in} \bar{U}_{fn}$$

$$\bar{A}_{ii} = \sum_{\substack{k=1 \\ k \neq i}}^n (\bar{Y}_{ik0} + \bar{Y}_{ik})$$

$$\bar{A}_{ik} = -\bar{Y}_{ik}$$



# Výpočet chodu soustavy

$$\begin{bmatrix} \frac{P_1 - jQ_1}{\overline{U}_1^*} \\ \vdots \\ \frac{P_i - jQ_i}{\overline{U}_i^*} \\ \vdots \\ \frac{P_n - jQ_n}{\overline{U}_n^*} \end{bmatrix} = \begin{bmatrix} \overline{A}_{11} & \cdots & \overline{A}_{1i} & \cdots & \overline{A}_{1n} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \overline{A}_{i1} & \cdots & \overline{A}_{ii} & \cdots & \overline{A}_{in} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \overline{A}_{n1} & \cdots & \overline{A}_{ni} & \cdots & \overline{A}_{nn} \end{bmatrix} \begin{bmatrix} \overline{U}_1 \\ \vdots \\ \overline{U}_i \\ \vdots \\ \overline{U}_n \end{bmatrix}$$

$$\sqrt{3}\underline{\underline{I}} = \underline{\underline{A}} \times \underline{\underline{U}}$$

- Závěr:
- soustava **n** nelin. rovnic s komplex. koeficienty
  - neznámé - velikosti a úhly uzlových napětí sítě
  - analytický postup lze jen pro velmi malé sítě

# Výpočet chodu soustavy

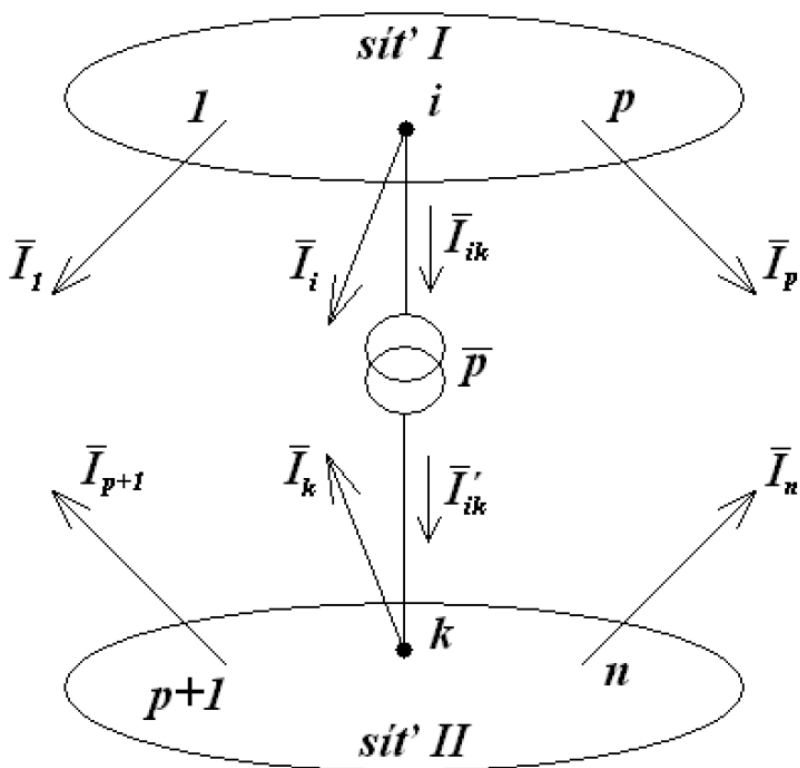
$$\frac{P_i - jQ_i}{\bar{U}_i^*} = \bar{A}_{i1} \cdot \bar{U}_1 + \dots + \bar{A}_{ii} \cdot \bar{U}_i + \dots + \bar{A}_{in} \cdot \bar{U}_n$$

$$P_i - jQ_i = \bar{A}_{i1} \cdot \bar{U}_1 \cdot \bar{U}_i^* + \dots + \bar{A}_{ii} \cdot |\bar{U}_i|^2 + \dots + \bar{A}_{in} \cdot \bar{U}_n \cdot \bar{U}_i^*$$

- Závěr:
- soustava **n** nelin. rovnic s komplex. koeficienty
  - neznámé - velikosti a úhly uzlových napětí sítě
  - analytický postup lze jen pro velmi malé sítě

# Výpočet chodu soustavy

Detail s jedním 2-vinuťovým transformátorem

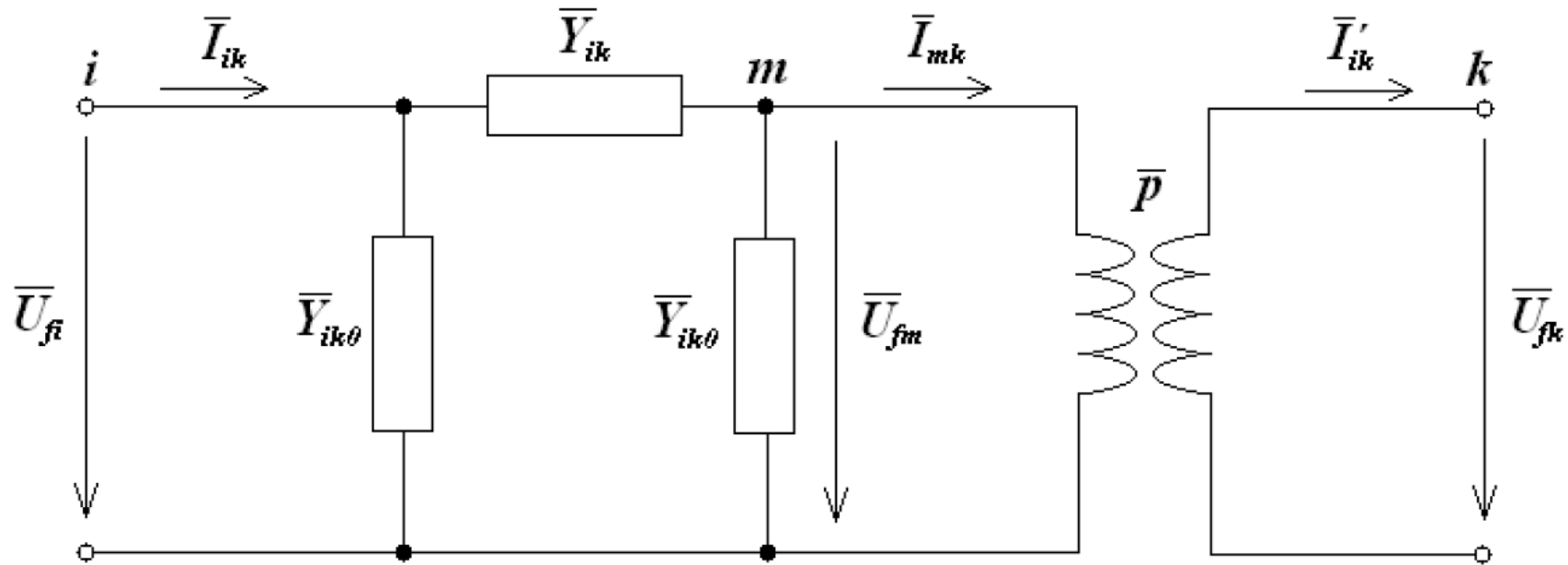


$$\begin{bmatrix} \bar{I}_1 \\ \vdots \\ \bar{I}_i - \bar{I}_{ik} \\ \vdots \\ \bar{I}_p \end{bmatrix} = \begin{bmatrix} \bar{A}_{11} & \cdots & \bar{A}_{1i} & \cdots & \bar{A}_{1p} \\ \vdots & & \vdots & & \vdots \\ \bar{A}_{i1} & \cdots & \bar{A}_{ii} & \cdots & \bar{A}_{ip} \\ \vdots & & \vdots & & \vdots \\ \bar{A}_{p1} & \cdots & \bar{A}_{pi} & \cdots & \bar{A}_{pp} \end{bmatrix} \cdot \begin{bmatrix} \bar{U}_{f1} \\ \vdots \\ \bar{U}_{fi} \\ \vdots \\ \bar{U}_{fp} \end{bmatrix}$$

$$\begin{bmatrix} \bar{I}_{p+1} \\ \vdots \\ \bar{I}_k + \bar{I}'_{ik} \\ \vdots \\ \bar{I}_n \end{bmatrix} = \begin{bmatrix} \bar{A}_{p+1,p+1} & \cdots & \bar{A}_{p+1,k} & \cdots & \bar{A}_{p+1,n} \\ \vdots & & \vdots & & \vdots \\ \bar{A}_{k,p+1} & \cdots & \bar{A}_{kk} & \cdots & \bar{A}_{kn} \\ \vdots & & \vdots & & \vdots \\ \bar{A}_{n,p+1} & \cdots & \bar{A}_{nk} & \cdots & \bar{A}_{nn} \end{bmatrix} \cdot \begin{bmatrix} \bar{U}_{fp+1} \\ \vdots \\ \bar{U}_{fk} \\ \vdots \\ \bar{U}_{fn} \end{bmatrix}$$

# Výpočet chodu soustavy

## Detail 2-vinutového transformátoru



Odvozením dostáváme vyjádření pro  $\bar{I}_{ik}$  a  $\bar{I}'_{ik}$

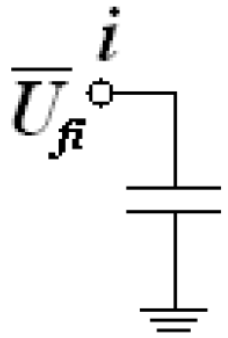
$$\begin{bmatrix} \bar{I}_{ik} \\ \bar{I}'_{ik} \end{bmatrix} = \begin{bmatrix} \bar{Y}_{ik} + \bar{Y}_{ik0} & -\bar{p}\bar{Y}_{ik} \\ \bar{p}^* \bar{Y}_{ik} & -|\bar{p}|^2 (\bar{Y}_{ik} + \bar{Y}_{ik0}) \end{bmatrix} \begin{bmatrix} \bar{U}_{fi} \\ \bar{U}_{fk} \end{bmatrix}$$

# Výpočet chodu soustavy

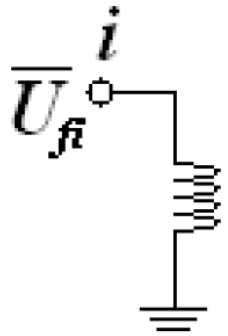
$$\begin{bmatrix} \bar{I}_1 \\ \vdots \\ \bar{I}_i \\ \vdots \\ \bar{I}_p \\ \bar{I}_{p+1} \\ \vdots \\ \bar{I}_k \\ \vdots \\ \bar{I}_n \end{bmatrix} = \begin{bmatrix} \bar{A}_{11} & \dots & \bar{A}_{1i} & \dots & \bar{A}_{1p} & 0 & \dots & 0 & \dots & 0 \\ \vdots & & \vdots & & \vdots & \vdots & & \vdots & & \vdots \\ \bar{A}_{i1} & \dots & \bar{A}_{ii} + \bar{Y}_{ik} + \bar{Y}_{ik0} & \dots & \bar{A}_{ip} & 0 & \dots & -p\bar{Y}_{ik} & \dots & 0 \\ \vdots & & \vdots & & \vdots & \vdots & & \vdots & & \vdots \\ \bar{A}_{p1} & \dots & \bar{A}_{pi} & \dots & \bar{A}_{pp} & 0 & \dots & 0 & \dots & 0 \\ 0 & \dots & 0 & \dots & 0 & \bar{A}_{p+1,p+1} & \dots & \bar{A}_{p+1,k} & \dots & \bar{A}_{p+1,n} \\ \vdots & & \vdots & & \vdots & \vdots & & \vdots & & \vdots \\ 0 & \dots & -p^* \bar{Y}_{ik} & \dots & 0 & \bar{A}_{k,p+1} & \dots & \bar{A}_{kk} + |p|^2 (\bar{Y}_{ik} + \bar{Y}_{ik0}) & \dots & \bar{A}_{kn} \\ \vdots & & \vdots & & \vdots & \vdots & & \vdots & & \vdots \\ 0 & \dots & 0 & \dots & 0 & \bar{A}_{n,p+1} & \dots & \bar{A}_{nk} & \dots & \bar{A}_{nn} \end{bmatrix} \cdot \begin{bmatrix} \bar{U}_{f1} \\ \vdots \\ \bar{U}_{fi} \\ \vdots \\ \bar{U}_{fp} \\ \bar{U}_{fp+1} \\ \vdots \\ \bar{U}_{fk} \\ \vdots \\ \bar{U}_{fn} \end{bmatrix}$$

# Výpočet chodu soustavy

Paralelní kondenzátory a induktory



$$B_{sh\_i} = \frac{Q_c}{U_n^2} > 0 \text{ pro kompenzaci účinníku nn, vn}$$



$$B_{sh\_i} = \frac{Q_c}{U_n^2} < 0 \text{ pro kompenzaci } Q_{cap} \text{ v sítích vvn}$$

$$A_{ii} = A_{ii} + jB_{sh\_i}$$

# Numerické řešení chodu soustavy

## Výstupy výpočtu uzlové:

- velikosti napětí (odhalení přepětí/podpětí)
- rozdíly fázových posunů větví (statická stabilita)
- velikost jalového výkonu v PU uzlech (PU, PU --> PQ)
- velikost injektovaného výkonu v referenčním uzlu

$$\bar{S}_{ref} = \bar{U}_{ref} \sum_{i=1}^n \bar{A}_{ref\ i}^* \bar{U}_i^* \rightarrow P_{ref} = \text{Re}\{\bar{S}_{ref}\}, Q_{ref} = \text{Im}\{\bar{S}_{ref}\}$$

- dodávka výkonu z kompenzačního prvku/nadř. sítě

$$P_{sh\_i} = -G_{sh\_i} U_i^2$$

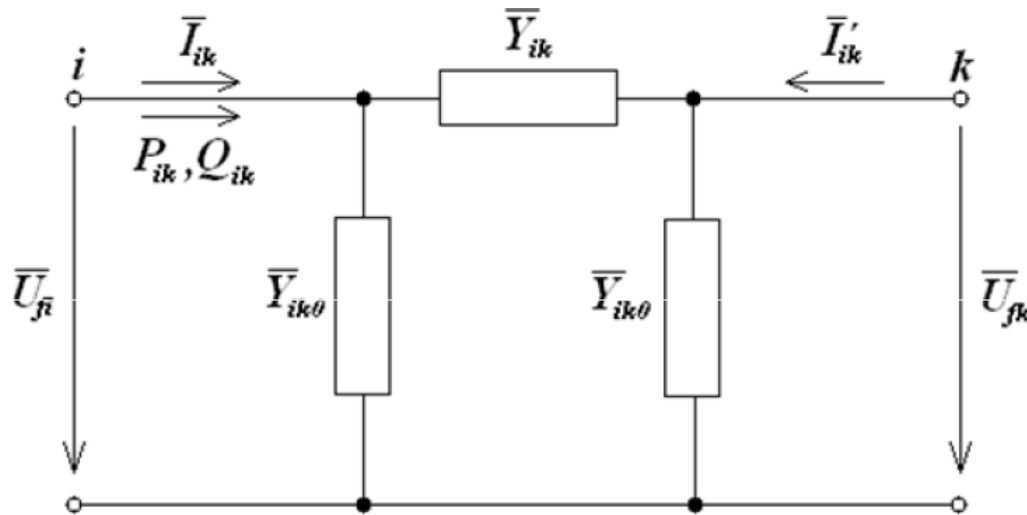
$$Q_{sh\_i} = +B_{sh\_i} U_i^2$$

- injektované proudy do uzlů sítě

$$\bar{I}_i = \frac{P_i - jQ_i}{\sqrt{3}\bar{U}_i^*}$$

# Numerické řešení chodu soustavy

Výstupy výpočtu větvové – činné a jalové toky:



$$P_{ik} = (G_{ik} + G_{ik0})U_i^2 - G_{ik}U_iU_k \cos(\theta_i - \theta_k) - B_{ik}U_iU_k \sin(\theta_i - \theta_k)$$

$$Q_{ik} = -(B_{ik} + B_{ik0})U_i^2 + B_{ik}U_iU_k \cos(\theta_i - \theta_k) - G_{ik}U_iU_k \sin(\theta_i - \theta_k)$$

$$P_{ki} = (G_{ik} + G_{ik0})U_k^2 - G_{ik}U_iU_k \cos(\theta_k - \theta_i) - B_{ik}U_iU_k \sin(\theta_k - \theta_i)$$

$$Q_{ki} = -(B_{ik} + B_{ik0})U_k^2 + B_{ik}U_iU_k \cos(\theta_k - \theta_i) - G_{ik}U_iU_k \sin(\theta_k - \theta_i)$$



# Numerické řešení chodu soustavy

## Výstupy výpočtu větrové – ztráty:

- větrové ztráty větve sítě (pro vedení)

$$\Delta P_{ik} = P_{ik} + P_{ki} = (G_{ik} + G_{ik0})(U_i^2 + U_k^2) - 2G_{ik}U_iU_k \cos(\theta_i - \theta_k)$$

$$\Delta Q_{ik} = Q_{ik} + Q_{ki} = -(B_{ik} + B_{ik0})(U_i^2 + U_k^2) + 2B_{ik}U_iU_k \cos(\theta_i - \theta_k)$$

- ztráty transformátoru naprázdno

$$\Delta P_{ik0} = G_{ik0}(U_i^2 + [p_{ik}U_k]^2)$$

- větrové proudy, zatížení pro vedení/transf., účinník

$$I_{ik} = \frac{\sqrt{P_{ik}^2 + Q_{ik}^2}}{\sqrt{3}U_i} \quad I_{ki} = \frac{\sqrt{P_{ki}^2 + Q_{ki}^2}}{\sqrt{3}U_k} \quad \eta = \frac{\sqrt{P_{ik}^2 + Q_{ik}^2}}{S_{MVA}} \cdot 100 \quad \eta = \sqrt{\frac{\Delta P_{ik}}{\Delta P_{Cu}}} \cdot 100$$

$$\cos \varphi_{ik} = \cos \left( \arctan \left( \frac{Q_{ik}}{P_{ik}} \right) \right) \quad \cos \varphi_{ki} = \cos \left( \arctan \left( \frac{Q_{ki}}{P_{ki}} \right) \right)$$

- celkové činné a jalové ztráty soustavy

$$\Delta P = \sum_{i=1}^n (P_{Gi} + P_{Li} + P_{Ci}) \quad \Delta Q = \sum_{i=1}^n (Q_{Gi} + Q_{Li} + Q_{Ci}) \quad \Delta P = \sum_{i=1}^n \sum_{k=i+1}^n \Delta P_{ik} \quad \Delta Q = \sum_{i=1}^n \sum_{k=i+1}^n \Delta Q_{ik}$$

# Numerické řešení chodu soustavy

Přesné numerické metody:

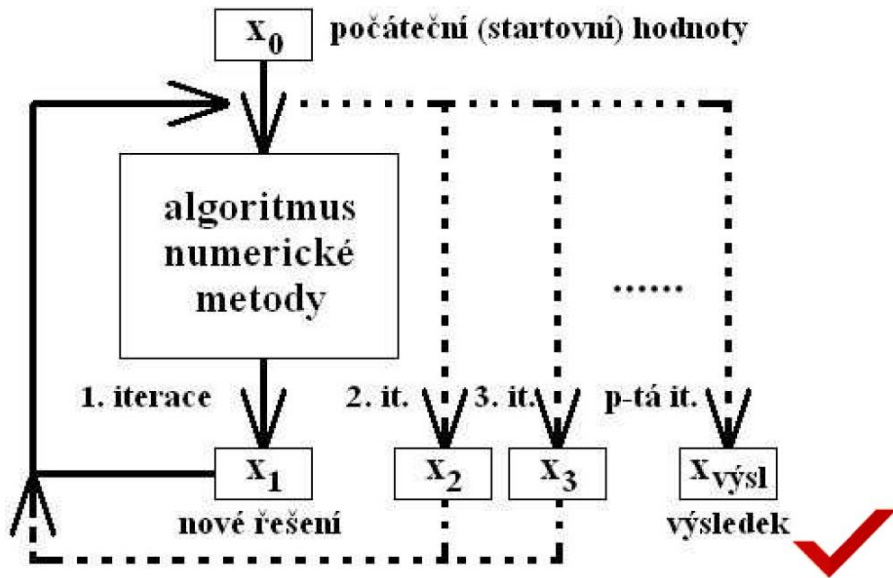
- **Gauss-Seidelova metoda (G-S)**
- **Newton-Raphsonova metoda (N-R)**

Alternativní metody (zjednodušením N-R metody):

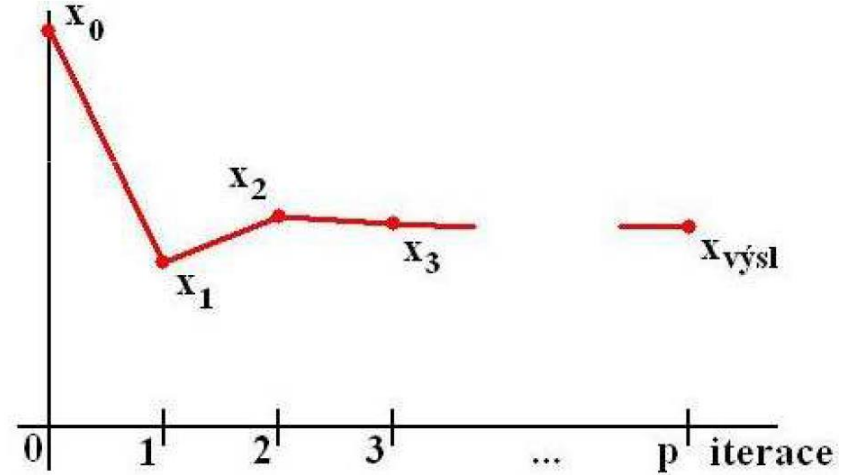
- Decoupled metoda (DLF)
- Fast-Decoupled metoda (F-D)
- DC load flow

# Numerické řešení chodu soustavy

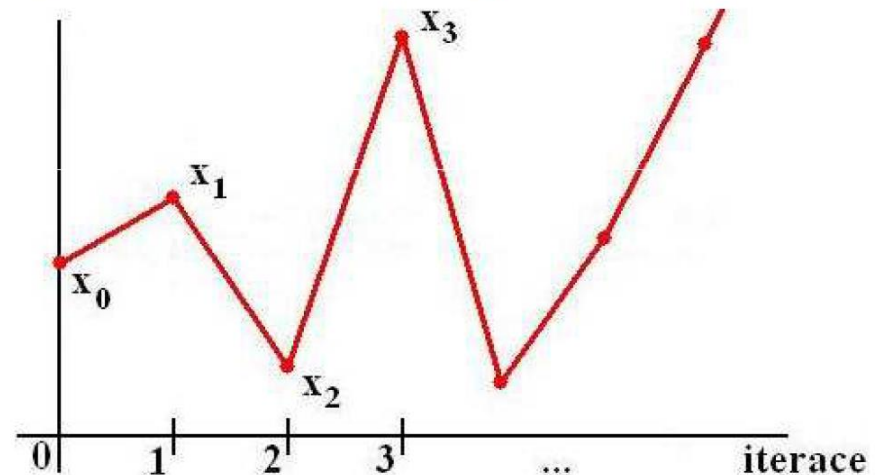
## Princip numerického výpočtu:



Konvergence numerického výpočtu



Divergence numerického výpočtu



# Numerické řešení chodu soustavy

## Gauss – Seidelova metoda

- pro Load Flow aplikována jako první (1950-60)
- výpočtově jednoduchá, výhodná pro první počítače
- pouze pro řešení malých sítí (do 50 uzlů)
- časově náročnější pro modelování velkých sítí (absence požadavku na rychlost výpočtu)
- dnes se používá pouze pro velmi vysokou spolehlivost výsledků (stabilita numerického výpočtu)
- svou jednoduchostí vhodná pro výukové účely

# Numerické řešení chodu soustavy

## Gauss – Seidelova metoda

Řešme soustavu 2 nelineárních rovnic:

$$x + 3\frac{y}{x} = 4$$

$$\underline{-\frac{5}{y} + 8xy = 11} \quad \text{pro } p = 0 \{x_0, y_0\} = \{1; 1\}$$

Postup: vytvoření iteračního algoritmu pro každou neznámou

$$\text{z 1. rovnice: } y^{(p+1)} = \frac{(4 - x^{(p)})x^{(p)}}{3}$$

$$\text{z 2. rovnice: } x^{(p+1)} = \frac{\left(11 + \frac{5}{y^{(p+1)}}\right)}{8y^{(p+1)}}$$

# Numerické řešení chodu soustavy

## Gauss – Seidelova metoda

### Průběh numerického výpočtu

iterace	x	y	odchylka
0	1	1	---
1	2	1	1
2	1.3828	1.3333	0.61719
3	1.5693	1.2064	0.18644
4	1.468	1.2715	0.10125
5	1.5169	1.239	0.048902
...	...	...	...
13	1.5001	1.25	0.00021508
14	1.5	1.25	0.00010898
15	1.5	1.25	5.5215e-005

# Numerické řešení chodu soustavy

The screenshot shows the DYNAST Shell interface with two windows. The left window, titled 'NelinearniRov...', contains the input code for solving a system of nonlinear equations. The right window, titled 'NelinearniRovnice.O', displays the output of the solver, including the number of nodes and equations, the solution values for variables y and x, and performance statistics.

```
File Edit View System Analysis Errors Run Publishing Preferences Window Help
NelinearniRov...
*: Nelinearni rovnice
*SYSTEM;
SYSVAR y;
SYSVAR x;
0 = x+3*y/x-4;
0 = -5/y+8*x*y-11;
*TR;
DC;
PRINT y, x;
RUN;
*END;
```

```
NelinearniRovnice.O
PRINT y, x;
RUN;
*END;

Number of nodes:      2
Number of implicit equations:  2

  Nelinearni rovnice

  1 ...    y
  2 ...    x

           1           2
1.250000e+000  1.500000e+000



Statistics: 0 steps, 0 rejected steps, 7 iterations

Number of errors: 0, Number of warnings: 0
Total seconds used up by DYNAST: 0.002
Program DYNAST exited on November 10, 2016 at 22:07:42
```

Ready

# Numerické řešení chodu soustavy

## Gauss – Seidelova metoda

- Gaussova metoda
  - updatuje všechny neznámé na novou hodnotu až po skončení celé jedné iterace  pomalejší než Gauss-Seidelova metoda (větší počet iterací)
- Gauss-Seidelova metoda
  - po vypočtení jedné neznámé ji okamžitě updatuje a použije pro další výpočty v aktuální iteraci
  -  rychlejší konvergence než Gaussova metoda
  - má **lineární konvergenci** (odchylky se mění lineárně)



# Numerické řešení chodu soustavy

## Gauss – Seidelova metoda

$$\frac{P_i - jQ_i}{\bar{U}_i^*} = \bar{A}_{i1} \cdot \bar{U}_1 + \dots + \boxed{\bar{A}_{ii} \cdot \bar{U}_i} + \dots + \bar{A}_{in} \cdot \bar{U}_n$$

$$\frac{P_i - jQ_i}{\bar{U}_i^*} - \bar{A}_{i1} \cdot \bar{U}_1 - \dots - \bar{A}_{in} \cdot \bar{U}_n = \boxed{\bar{A}_{ii} \cdot \bar{U}_i}$$

$$\frac{1}{\bar{A}_{ii}} \left( \frac{P_i - jQ_i}{\bar{U}_i^*} - \bar{A}_{i1} \cdot \bar{U}_1 - \dots - \bar{A}_{in} \cdot \bar{U}_n \right) = \bar{U}_i$$

$$\bar{U}_i = \frac{1}{\bar{A}_{ii}} \left( \frac{P_i - jQ_i}{\bar{U}_i^*} - \bar{A}_{i1} \cdot \bar{U}_1 - \dots - \bar{A}_{in} \cdot \bar{U}_n \right)$$

## G-S iterační algoritmus

$$\bar{U}_i^{(p+1)} = \frac{1}{\bar{A}_{ii}} \left[ \frac{P_i - jQ_i}{\bar{U}_i^{(p)*}} - \sum_{k=1}^{i-1} \bar{A}_{ik} \bar{U}_k^{(p+1)} - \sum_{k=i+1}^n \bar{A}_{ik} \bar{U}_k^{(p)} \right]$$

# Numerické řešení chodu soustavy

## Gauss – Seidelova metoda

### G-S iterační algoritmus (přímo pro PQ uzly)

$$\bar{U}_i^{(p+1)} = \frac{1}{A_{ii}} \left[ \frac{P_i - jQ_i}{\bar{U}_i^{(p)*}} - \sum_{k=1}^{i-1} \bar{A}_{ik} \bar{U}_k^{(p+1)} - \sum_{k=i+1}^n \bar{A}_{ik} \bar{U}_k^{(p)} \right]$$

pro PU uzly

$$Q_i^{(p+1)} = \mathbf{Im} \left\{ \bar{U}_i^{(p)} \left[ \sum_{k=1}^{i-1} \bar{A}_{ik}^* \bar{U}_k^{(p+1)*} + \sum_{k=i}^n \bar{A}_{ik}^* \bar{U}_k^{(p)*} \right] \right\}$$

$$\bar{U}_i^{(p+1)} = \frac{\bar{U}_i^{(p+1)}}{|\bar{U}_i^{(p+1)}|} U_{i\_set}$$

pro referenční uzel - **žádné výpočty (!)**

# Numerické řešení chodu soustavy

Gauss –

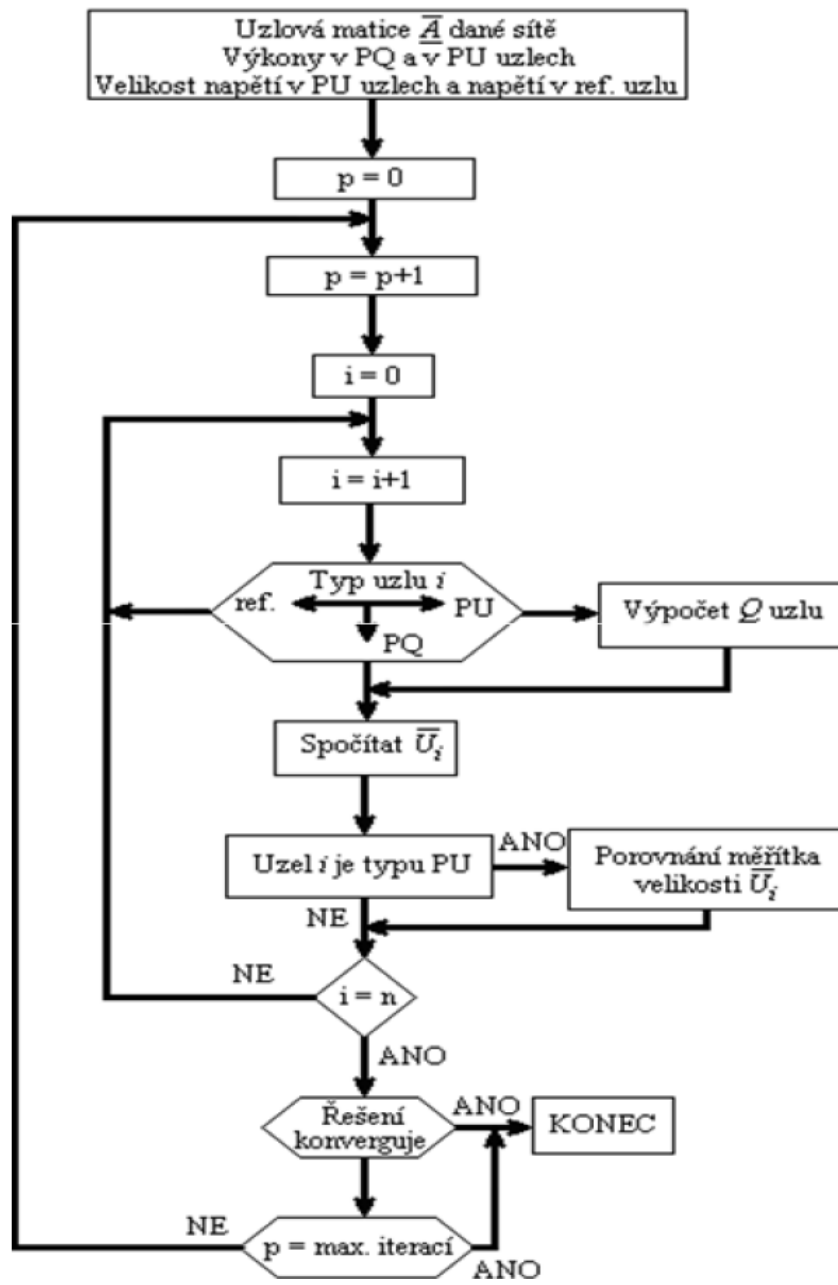
Seidelova metoda

Vývojový diagram  
algoritmu:

Konvergenční kritérium:

$$\max_i \frac{|V_i^{(p+1)} - V_i^{(p)}|}{V_i^{(p)}} \leq \varepsilon$$

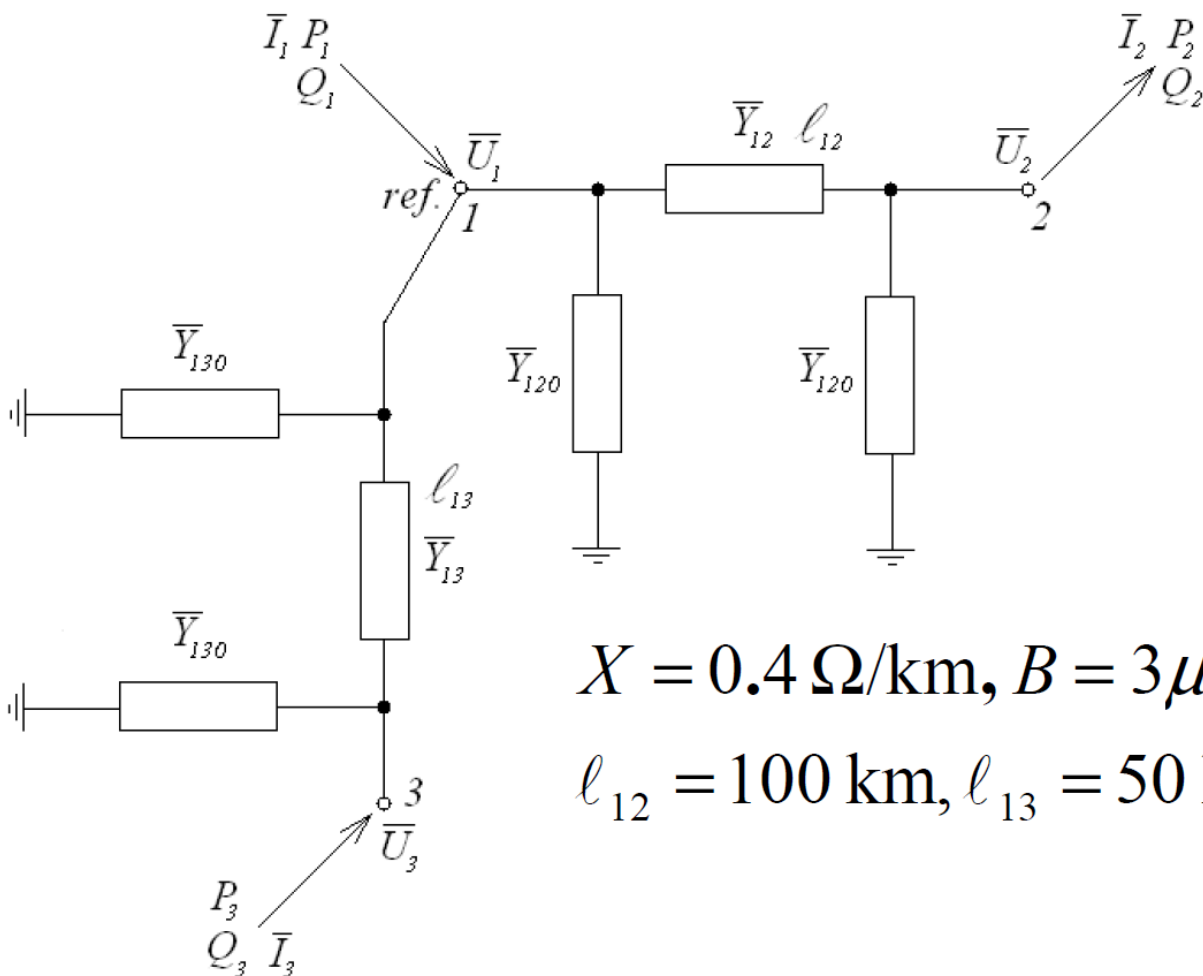
$$\max_i |\theta_i^{(p+1)} - \theta_i^{(p)}| \leq \varepsilon$$



# Numerické řešení chodu soustavy

## Gauss – Seidelova metoda

### Řešte chod 3-uzlové soustavy



$$\bar{U}_1 = 110000 \cdot e^{j0^\circ} \text{ V}$$

$$P_2 = 50 \text{ MW}$$

$$Q_2 = 10 \text{ MVar}$$

$$P_3 = 100 \text{ MW}$$

$$Q_3 = 10 \text{ MVar}$$

$$X = 0.4 \text{ } \Omega/\text{km}, B = 3 \mu\text{S}/\text{km}$$

$$l_{12} = 100 \text{ km}, l_{13} = 50 \text{ km}$$

$$U_2, \theta_2, U_3, \theta_3 = ?$$

# Numerické řešení chodu soustavy

## Gauss – Seidelova metoda

Iterační algoritmus:

$$\bar{U}_2^{(p)} = \frac{1}{A_{22}} \left[ \frac{P_2 - jQ_2}{\bar{U}_2^{*(p-1)}} - \bar{A}_{21}\bar{U}_1 - \sum_3 \bar{A}_{ik}\bar{U}_k^{(p-1)} \right] = \frac{1}{A_{22}} \left[ \frac{P_2 - jQ_2}{\bar{U}_2^{*(p-1)}} - \bar{A}_{21}\bar{U}_1 - \bar{A}_{23}\bar{U}_3^{(p-1)} \right]$$

$$\bar{U}_3^{(p)} = \frac{1}{A_{33}} \left[ \frac{P_3 - jQ_3}{\bar{U}_3^{*(p-1)}} - \bar{A}_{31}\bar{U}_1 - \sum_{k=2}^2 \bar{A}_{ik}\bar{U}_k^{(p)} \right] = \frac{1}{A_{33}} \left[ \frac{P_3 - jQ_3}{\bar{U}_3^{*(p-1)}} - \bar{A}_{31}\bar{U}_1 - \bar{A}_{32}\bar{U}_2^{(p)} \right]$$

- Počáteční hodnoty:

Admitanční matice:

$$\bar{A} = j \begin{bmatrix} -0.074775 & 0.025 & 0.05 \\ 0.025 & -0.02485 & 0 \\ 0.05 & 0 & -0.049925 \end{bmatrix}$$

$$\bar{x}_0 = \begin{bmatrix} 110000 e^{j0} \\ 110000 e^{j0} \\ 110000 e^{j0} \end{bmatrix} [\text{V}]$$

# Numerické řešení chodu soustavy

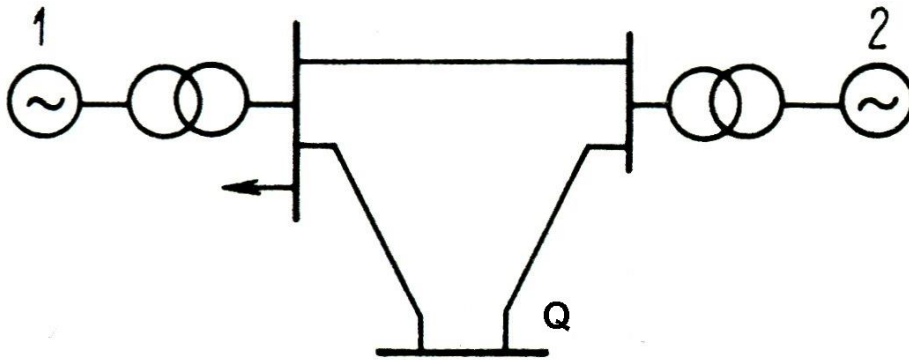
## Gauss – Seidelova metoda

$p$	$U_2$	$U_3$
0	110000,00000	110000,00000
1	108557,79539	113456,92025
2	105374,88998	110502,59407
3	105284,10663	110555,03645
4	105178,50904	110476,87553
5	105175,34781	110478,27315
6	105171,71834	110476,14855
7	105171,60950	110476,18655
8	105171,48461	110476,12876
9	105171,48086	110476,12979
10	105171,47656	110476,12822
11	105171,47643	110476,12824
<b>12</b>	<b>105171,47629</b>	<b>110476,12820</b>

$p$	$\theta_2$	$\theta_3$
0	0	0
1	-9°42'1,41''	9°14'8,01''
2	-9°38'22,42''	9°13'18,51''
3	-9°56'46,96''	9°27'57,71''
4	-9°56'39,67''	9°27'57,02''
5	-9°57'17,58''	9°28'20,93''
6	-9°57'17,33''	9°28'20,90''
7	-9°57'18,63''	9°28'21,55''
8	-9°57'18,62''	9°28'21,54''
9	-9°57'18,67''	9°28'21,56''
10	-9°57'18,67''	9°28'21,56''
11	-9°57'18,67''	9°28'21,56''
<b>12</b>	<b>-9°57'18,67''</b>	<b>9°28'21,56''</b>

# Numerické řešení chodu soustavy

## Gauss – Seidelova metoda



Parametry schématu:

$$\begin{aligned}x_{g1} &= 0.18 & x_{T1} &= 0.1 & x_{g2} &= 0.18 & x_{T2} &= 0.1 \\x_1 &= x_{g1} + x_{T1} = 0.28 & & & x_2 &= x_{g2} + x_{T2} = 0.28 & & \\x_{V1} &= x_3 = 0.052 & & & x_{V2} &= 0.105 = x_4 + x_5 = 0.085 + 0.02 & & \\x_Z &= x_6 = 0.6 & & & & & & \end{aligned}$$

# Numerické řešení chodu soustavy

## Gauss – Seidelova metoda

$$[A_{1,2,3,4,Q}] = \begin{bmatrix} -\frac{i}{x_1} & 0 & \frac{i}{x_1} & & 0 & & 0 \\ 0 & -\frac{i}{x_2} & 0 & & \frac{i}{x_2} & & 0 \\ \frac{i}{x_1} & 0 & -\frac{i}{x_1} & -\frac{i}{x_3} & -\frac{i}{x_4} & -\frac{i}{x_6} & \frac{i}{x_3} & \frac{i}{x_4} \\ 0 & \frac{i}{x_2} & & \frac{i}{x_3} & & & -\frac{i}{x_2} & -\frac{i}{x_3} & -\frac{i}{x_5} & \frac{i}{x_5} \\ 0 & 0 & & \frac{i}{x_4} & & & \frac{i}{x_5} & & -\frac{i}{x_4} & -\frac{i}{x_5} \end{bmatrix}$$

$$[A_{1,2,3,4,Q}] = \begin{bmatrix} -3.5714i & 0 & 3.5714i & 0 & 0 \\ 0 & -3.5714i & 0 & 3.5714i & 0 \\ 3.5714i & 0 & -36.2336i & 19.2308i & 11.7647i \\ 0 & 3.5714i & 19.2308i & -72.8022i & 50.0000i \\ 0 & 0 & 11.7647i & 50.0000i & -61.7647i \end{bmatrix}$$



# Numerické řešení chodu soustavy

## Gauss – Seidelova metoda

$$U_2^{(N+1)/} = \frac{1}{a_{2,2}} \left( \frac{P_2 - i \cdot Q_2}{U_2^{(N)*}} - a_{2,1} U_1^{(0)} - a_{2,3} U_3^{(N)} - a_{2,4} U_4^{(N)} - a_{2,Q} U_Q^{(N)} \right)$$

$$U_2^{(N+1)} = \frac{U_2^{(N+1)/}}{|U_2^{(N+1)/}|} U_2^{(0)} \quad [U^{(0)}] = [1.1 \quad 1.05 \quad 1 \quad 1 \quad 1]^T$$
$$[S] = [0 \quad 0 \quad 0 \quad 0 \quad 0]$$

$$U_3^{(N+1)} = \frac{1}{a_{3,3}} \left( \frac{P_3 - i \cdot Q_3}{U_3^{(N)*}} - a_{3,1} U_1^{(0)} - a_{3,2} U_2^{(N+1)} - a_{3,4} U_4^{(N)} - a_{3,Q} U_Q^{(N)} \right)$$


$$U_4^{(N+1)} = \frac{1}{a_{4,4}} \left( \frac{P_4 - i \cdot Q_4}{U_4^{(N)*}} - a_{4,1} U_1^{(0)} - a_{4,2} U_2^{(N+1)} - a_{4,3} U_3^{(N+1)} - a_{4,Q} U_Q^{(N)} \right)$$

$$U_Q^{(N+1)} = \frac{1}{a_{Q,Q}} \left( \frac{P_Q - i \cdot Q_Q}{U_Q^{(N)*}} - a_{Q,1} U_1^{(0)} - a_{Q,2} U_2^{(N+1)} - a_{Q,3} U_3^{(N+1)} - a_{Q,4} U_4^{(N+1)} \right)$$

$$U^{(141)} = [1.1 \quad 1.05 \quad 0.8633 \quad 0.8839 \quad 0.8800]^T$$

# Numerické řešení chodu soustavy

## Gauss – Seidelova metoda

- Výhody použití:
  - snadný matematický model (bez matic, derivací, atd.)
  - pomalé přibližování k hledanému řešení (malé změny v jednotlivých iteracích)  relativně bezpečné dosažení hledaného výsledku (vysoká spolehlivost)
  - nízké výpočtové a časové nároky vztažené na iteraci
- Nevýhody:
  - pracujeme s komplexními čísly
  - silná závislost počtu iterací na velikosti sítě
  - má pouze lineární konvergenci (velký počet iterací)
  - delší doba výpočtu (zvláště pro rozsáhlé sítě)

# Numerické řešení chodu soustavy

## Newton - Raphsonova metoda

Jednoduchý případ (nelin. rovnice o 1 neznámé)

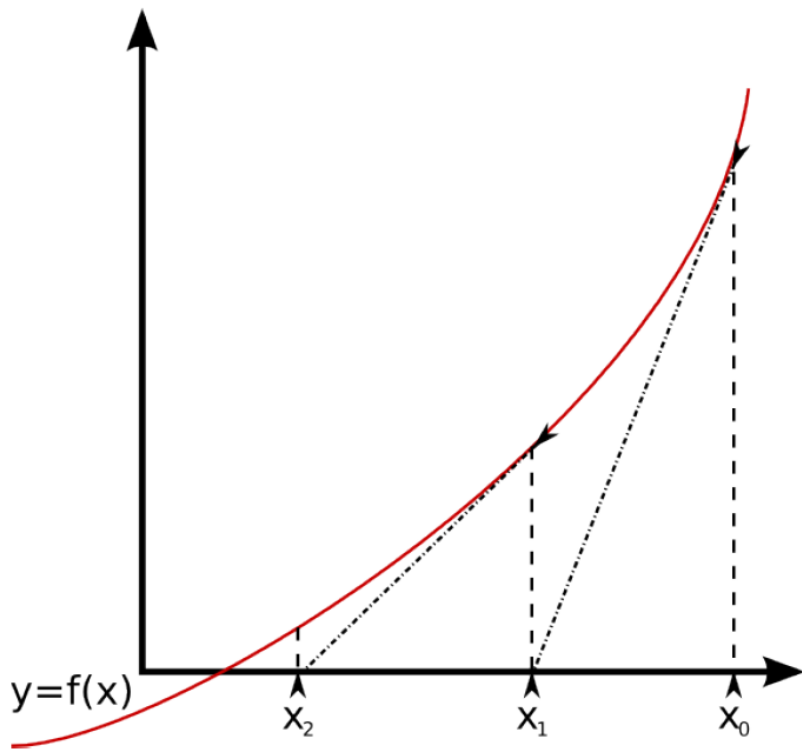
$$f(x) = 0$$

Taylorovým rozvojem dostáváme:

$$f(x^{(0)} + \Delta x) \approx f(x^{(0)}) + \Delta x \left. \frac{df}{dx} \right|_{x=x^{(0)}} + \frac{\Delta x^2}{2!} \left. \frac{d^2 f}{dx^2} \right|_{x=x^{(0)}} + \dots + \frac{\Delta x^N}{N!} \left. \frac{d^N f}{dx^N} \right|_{x=x^{(0)}} = 0$$

Vyšší řády se zanedbávají:

$$f(x^{(0)} + \Delta x) \approx f(x^{(0)}) + \Delta x \left. \frac{df}{dx} \right|_{x=x^{(0)}} = 0$$



# Numerické řešení chodu soustavy

## Newton - Raphsonova metoda

Jednoduchý případ (nelin. rovnice o 1 neznámé)

$$f(x) = 0$$

Taylorovým rozvojem dostáváme:

$$f(x^{(0)} + \Delta x) \approx f(x^{(0)}) + \Delta x \left. \frac{df}{dx} \right|_{x=x^{(0)}} + \frac{\Delta x^2}{2!} \left. \frac{d^2 f}{dx^2} \right|_{x=x^{(0)}} + \dots + \frac{\Delta x^N}{N!} \left. \frac{d^N f}{dx^N} \right|_{x=x^{(0)}} = 0$$

Vyšší řády se zanedbávají:

$$f(x^{(0)} + \Delta x) \approx f(x^{(0)}) + \Delta x \left. \frac{df}{dx} \right|_{x=x^{(0)}} = 0 \quad \longrightarrow \quad \Delta x^{(p+1)} = \frac{0 - f(x^{(p)})}{\left( \frac{df(x^{(p)})}{dx} \right)}$$

Vícerozměrný případ:

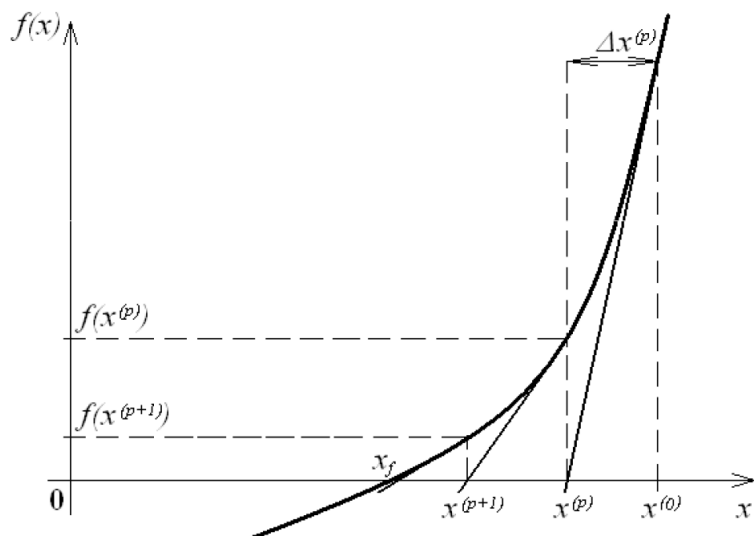
$$\begin{bmatrix} 0 - f_1(x_1^{(0)}, x_2^{(0)}, \dots, x_n^{(0)}) \\ 0 - f_2(x_1^{(0)}, x_2^{(0)}, \dots, x_n^{(0)}) \\ \vdots \\ 0 - f_n(x_1^{(0)}, x_2^{(0)}, \dots, x_n^{(0)}) \end{bmatrix} = \begin{bmatrix} \left. \frac{\partial f_1}{\partial x_1} \right|_{x_1=x_1^{(0)}} & \left. \frac{\partial f_1}{\partial x_2} \right|_{x_2=x_2^{(0)}} & \dots & \left. \frac{\partial f_1}{\partial x_n} \right|_{x_n=x_n^{(0)}} \\ \left. \frac{\partial f_2}{\partial x_1} \right|_{x_1=x_1^{(0)}} & \left. \frac{\partial f_2}{\partial x_2} \right|_{x_2=x_2^{(0)}} & \dots & \left. \frac{\partial f_2}{\partial x_n} \right|_{x_n=x_n^{(0)}} \\ \vdots & \vdots & \ddots & \vdots \\ \left. \frac{\partial f_n}{\partial x_1} \right|_{x_1=x_1^{(0)}} & \left. \frac{\partial f_n}{\partial x_2} \right|_{x_2=x_2^{(0)}} & \dots & \left. \frac{\partial f_n}{\partial x_n} \right|_{x_n=x_n^{(0)}} \end{bmatrix} \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \\ \vdots \\ \Delta x_n \end{bmatrix}$$

# Numerické řešení chodu soustavy

## Newton - Raphsonova metoda

Výpočet přírůstkového vektoru v dané iteraci:

$$\begin{bmatrix} \Delta x_1^{(p+1)} \\ \Delta x_2^{(p+1)} \\ \vdots \\ \Delta x_n^{(p+1)} \end{bmatrix} = \begin{bmatrix} \left. \frac{\partial f_1}{\partial x_1} \right|_{x_1=x_1^{(p)}} & \left. \frac{\partial f_1}{\partial x_2} \right|_{x_2=x_2^{(p)}} & \dots & \left. \frac{\partial f_1}{\partial x_n} \right|_{x_n=x_n^{(p)}} \\ \left. \frac{\partial f_2}{\partial x_1} \right|_{x_1=x_1^{(p)}} & \left. \frac{\partial f_2}{\partial x_2} \right|_{x_2=x_2^{(p)}} & \dots & \left. \frac{\partial f_2}{\partial x_n} \right|_{x_n=x_n^{(p)}} \\ \vdots & \vdots & \ddots & \vdots \\ \left. \frac{\partial f_n}{\partial x_1} \right|_{x_1=x_1^{(p)}} & \left. \frac{\partial f_n}{\partial x_2} \right|_{x_2=x_2^{(p)}} & \dots & \left. \frac{\partial f_n}{\partial x_n} \right|_{x_n=x_n^{(p)}} \end{bmatrix}^{-1} \begin{bmatrix} 0 - f_1(x_1^{(p)}, x_2^{(p)}, \dots, x_n^{(p)}) \\ 0 - f_2(x_1^{(p)}, x_2^{(p)}, \dots, x_n^{(p)}) \\ \vdots \\ 0 - f_n(x_1^{(p)}, x_2^{(p)}, \dots, x_n^{(p)}) \end{bmatrix}$$



# Numerické řešení chodu soustavy

## Newton - Raphsonova metoda

Výpočet přírůstkového vektoru v dané iteraci:

$$\begin{bmatrix} \Delta x_1^{(p+1)} \\ \Delta x_2^{(p+1)} \\ \vdots \\ \Delta x_n^{(p+1)} \end{bmatrix} = \begin{bmatrix} \left. \frac{\partial f_1}{\partial x_1} \right|_{x_1=x_1^{(p)}} & \left. \frac{\partial f_1}{\partial x_2} \right|_{x_2=x_2^{(p)}} & \dots & \left. \frac{\partial f_1}{\partial x_n} \right|_{x_n=x_n^{(p)}} \\ \left. \frac{\partial f_2}{\partial x_1} \right|_{x_1=x_1^{(p)}} & \left. \frac{\partial f_2}{\partial x_2} \right|_{x_2=x_2^{(p)}} & \dots & \left. \frac{\partial f_2}{\partial x_n} \right|_{x_n=x_n^{(p)}} \\ \vdots & \vdots & \ddots & \vdots \\ \left. \frac{\partial f_n}{\partial x_1} \right|_{x_1=x_1^{(p)}} & \left. \frac{\partial f_n}{\partial x_2} \right|_{x_2=x_2^{(p)}} & \dots & \left. \frac{\partial f_n}{\partial x_n} \right|_{x_n=x_n^{(p)}} \end{bmatrix}^{-1} \begin{bmatrix} b_1 - f_1(x_1^{(p)}, x_2^{(p)}, \dots, x_n^{(p)}) \\ b_2 - f_2(x_1^{(p)}, x_2^{(p)}, \dots, x_n^{(p)}) \\ \vdots \\ b_n - f_n(x_1^{(p)}, x_2^{(p)}, \dots, x_n^{(p)}) \end{bmatrix}$$

Update proces:

$$\begin{bmatrix} x_1^{(p+1)} \\ x_2^{(p+1)} \\ \vdots \\ x_n^{(p+1)} \end{bmatrix} = \begin{bmatrix} x_1^{(p)} \\ x_2^{(p)} \\ \vdots \\ x_n^{(p)} \end{bmatrix} + \begin{bmatrix} \Delta x_1^{(p+1)} \\ \Delta x_2^{(p+1)} \\ \vdots \\ \Delta x_n^{(p+1)} \end{bmatrix}$$

Konvergenční kritérium:

$$\max_i \begin{bmatrix} b_1 - f_1(x_1^{(p)}, x_2^{(p)}, \dots, x_n^{(p)}) \\ b_2 - f_2(x_1^{(p)}, x_2^{(p)}, \dots, x_n^{(p)}) \\ \vdots \\ b_n - f_n(x_1^{(p)}, x_2^{(p)}, \dots, x_n^{(p)}) \end{bmatrix} \leq \varepsilon$$

# Numerické řešení chodu soustavy

## Newton - Raphsonova metoda

Řešme soustavu 2 nelineárních rovnic:

$$x + 3\frac{y}{x} = 4$$

$$\underline{-\frac{5}{y} + 8xy = 11} \quad \text{pro } p = 0 \{x_0, y_0\} = \{1; 1\}$$

$$\text{ukončovací podm.: } \boldsymbol{\varepsilon}_{mez} = 1 \cdot 10^{-4}$$

Postup: hlavní numerický předpis pro N-R metodu

$$\begin{bmatrix} \Delta f_1 \\ \Delta f_2 \end{bmatrix} = - \begin{bmatrix} \frac{\partial f_1}{\partial x} & \frac{\partial f_1}{\partial y} \\ \frac{\partial f_2}{\partial x} & \frac{\partial f_2}{\partial y} \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \quad + \text{ update: } \begin{aligned} x^{(p+1)} &= x^{(p)} + \Delta x \\ y^{(p+1)} &= y^{(p)} + \Delta y \end{aligned}$$

# Numerické řešení chodu soustavy

## Newton - Raphsonova metoda

Upravený numerický předpis pro zadaný příklad:

$$\begin{bmatrix} x + 3\frac{y}{x} - 4 \\ -\frac{5}{y} + 8xy - 11 \end{bmatrix} = - \begin{bmatrix} 1 - 3\frac{y}{x^2} & \frac{3}{x} \\ 8y & \frac{5}{y^2} + 8x \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}$$

Pro  $p = 1$  (ze znalosti  $x_0 = \{1; 1\}$ ):

$$\begin{bmatrix} 0 \\ -8 \end{bmatrix} = - \begin{bmatrix} -2 & 3 \\ 8 & 13 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \rightarrow \Delta x = 0.48, \Delta y = 0.32 \rightarrow x^{(1)} = 1.48, y^{(1)} = 1.32$$

update

Průběh výpočtu:

iterace	x	y	odchylka
0	1	1	8
1	1.48	1.32	0.84092
2	1.4976	1.2502	0.020998
3	<b>1.5</b>	<b>1.25</b>	<b>7.1501e-006</b>

$$- \begin{bmatrix} -2 & 3 \\ 8 & 13 \end{bmatrix}^{-1} = - \begin{bmatrix} -0.26 & 0.06 \\ 0.16 & 0.04 \end{bmatrix}$$

Získané řešení:

$$x = 1.5, y = 1.25$$



# Numerické řešení chodu soustavy

## Newton - Raphsonova metoda

Základní iterační algoritmus N-R metody:

$$\begin{bmatrix} \Delta P^{(p-1)} \\ \Delta Q^{(p-1)} \end{bmatrix} = \begin{bmatrix} \underline{H}^{(p-1)} & \underline{N}^{(p-1)} \\ \underline{J}^{(p-1)} & \underline{L}^{(p-1)} \end{bmatrix} \cdot \begin{bmatrix} \Delta \theta^{(p)} \\ \frac{\Delta U^{(p)}}{U^{(p-1)}} \end{bmatrix} \quad \leftarrow \quad \frac{\Delta U^{(p)}}{U^{(p-1)}} = |\Delta u^{(p)}|$$

$A_{ik} = G_{ik} + jB_{ik}$

kde: rozdílový vektor (power mismatch vector)

$$\Delta P_i = P_i - U_i \sum_{k=1}^n U_k (G_{ik} \cos \theta_{ik} + B_{ik} \sin \theta_{ik}) \quad \Delta P_i = P_i - U_i \sum_{k=1}^n A_{ik} U_k \cos(\theta_{ik} - \varphi_{ik})$$

nebo

$$\Delta Q_i = Q_i - U_i \sum_{k=1}^n U_k (G_{ik} \sin \theta_{ik} - B_{ik} \cos \theta_{ik}) \quad \Delta Q_i = Q_i - U_i \sum_{k=1}^n A_{ik} U_k \sin(\theta_{ik} - \varphi_{ik})$$

Pozn.:  $\theta_{ik} = \theta_i - \theta_k$

prvky Jacobi matice

$$H_{ik} = \frac{\partial P_i}{\partial \theta_k} \quad N_{ik} = U_k \frac{\partial P_i}{\partial U_k} \quad J_{ik} = \frac{\partial Q_i}{\partial \theta_k} \quad L_{ik} = U_k \frac{\partial Q_i}{\partial U_k}$$

Iterační algoritmus obsahuje: 2 rovnice pro každý PQ uzel, 1 rovnici pro PV uzel, žádnou pro referenční uzel

# Numerické řešení chodu soustavy

## Newton - Raphsonova metoda

Výpočet prvků Jacobi matice:

$$H_{ii} = -Q_i - B_{ii}U_i^2 \quad N_{ii} = P_i + G_{ii}U_i^2 \quad J_{ii} = P_i - G_{ii}U_i^2 \quad L_{ii} = Q_i - B_{ii}U_i^2$$

$$H_{ik} = U_i U_k (G_{ik} \sin \theta_{ik} - B_{ik} \cos \theta_{ik}) \quad N_{ik} = U_i U_k (G_{ik} \cos \theta_{ik} + B_{ik} \sin \theta_{ik})$$

$$J_{ik} = -U_i U_k (G_{ik} \cos \theta_{ik} + B_{ik} \sin \theta_{ik}) = -N_{ik}$$

$$L_{ik} = U_i U_k (G_{ik} \sin \theta_{ik} - B_{ik} \cos \theta_{ik}) = H_{ik}$$

Pro PV uzly:

$$Q_i^{(p-1)} = U_i^{(p-1)} \sum_{k=1}^n U_k^{(p-1)} (G_{ik} \sin \theta_{ik}^{(p-1)} - B_{ik} \cos \theta_{ik}^{(p-1)})$$

$$\theta_i^{(p)} = \theta_i^{(p-1)} + \Delta \theta_i^{(p)}$$

$$U_i^{(p)} = U_i^{(p-1)} + \left( \frac{\Delta U_i^{(p)}}{U_i^{(p-1)}} \right) U_i^{(p-1)}$$

$$\max_i \left| \frac{\Delta P}{\Delta Q} \right| \leq \varepsilon$$

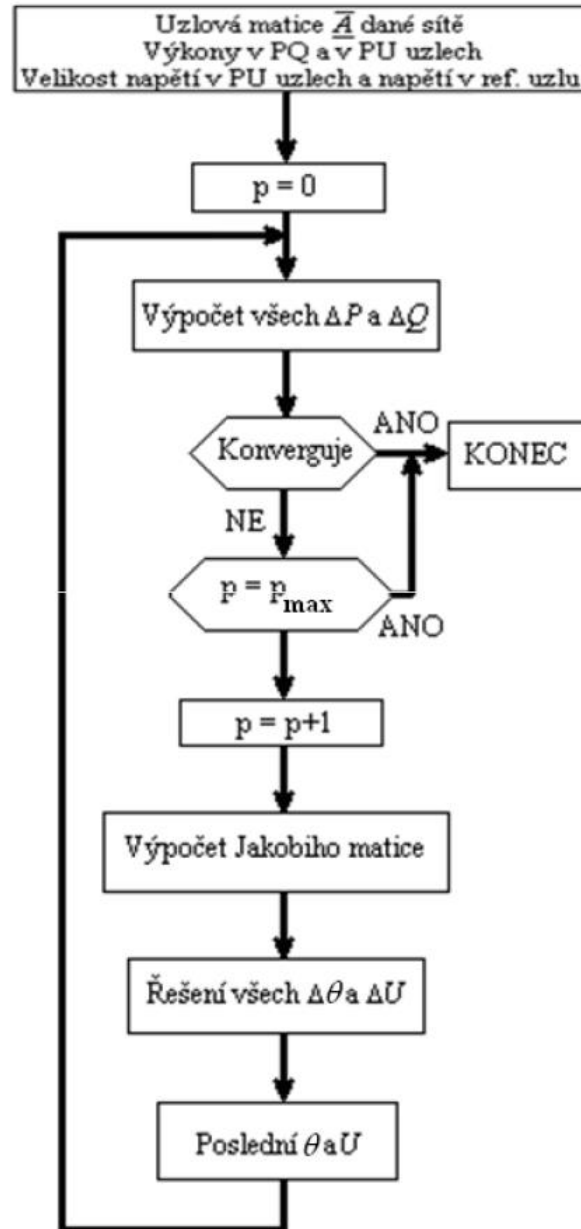
# Numerické řešení chodu soustavy

## Newton - Raphsonova metoda

Vývojový diagram  
algoritmu:

Konvergenční kritérium:

$$\max_i \left| \frac{\Delta P}{\Delta Q} \right| \leq \varepsilon$$



# Numerické řešení chodu soustavy

Newton - Raphsonova metoda

Počáteční (startovní) hodnoty:

$$U^{(0)} = \begin{bmatrix} U_2^{(0)} \\ U_3^{(0)} \end{bmatrix} = \begin{bmatrix} 110000 \\ 110000 \end{bmatrix}; \quad \theta^{(0)} = \begin{bmatrix} \theta_2^{(0)} \\ \theta_3^{(0)} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Základní algoritmus N-R metody pro  $p = 1$ :

$$\begin{bmatrix} -50 \\ 100 \\ -8.185 \\ 10.9075 \end{bmatrix} = \begin{bmatrix} 310.685 & 0 & -50 & 0 \\ 0 & 594.0925 & 0 & 100 \\ -50 & 0 & 290.685 & 0 \\ 0 & 100 & 0 & 614.0925 \end{bmatrix} \begin{bmatrix} \Delta\theta_2^{(p)} \\ \Delta\theta_3^{(p)} \\ \left( \frac{\Delta U_2^{(p)}}{U_2^{(p-1)}} \right) \\ \left( \frac{\Delta U_3^{(p)}}{U_3^{(p-1)}} \right) \end{bmatrix}$$

$$\Delta\theta_2 = -0.170177, \Delta\theta_3 = 0.169993, \frac{\Delta U_2}{U_2^{(0)}} = -0.057429, \frac{\Delta U_3}{U_3^{(0)}} = -0.009920$$

# Numerické řešení chodu soustavy

## Newton - Raphsonova metoda

Update v iteraci  $p = 1$ :

$$\theta_2^{(1)} = \theta_2^{(0)} + \Delta\theta_2 = -0.170177 \text{ rad}, \quad \theta_3^{(1)} = \theta_3^{(0)} + \Delta\theta_3 = 0.169993 \text{ rad}$$

$$U_2^{(1)} = U_2^{(0)} + \frac{\Delta U_2}{U_2^{(0)}} \cdot U_2^{(0)} = 103682.81 \text{ V}$$

$$U_3^{(1)} = U_3^{(0)} + \frac{\Delta U_3}{U_3^{(0)}} \cdot U_3^{(0)} = 108908.80 \text{ V}$$

Celkový průběh:

$p$	$U_2$	$U_3$
0	110000,00000	110000,00000
1	103682,77009	108908,78612
2	105170,26641	110477,69448
3	105171,47627	110476,12763
4	105171,47628	110476,12820
<b>5</b>	<b>105171,47628</b>	<b>110476,12820</b>

$\theta_2$	$\theta_3$
0	0
-9°45'1,54''	9°44'23,73''
-9°57'21,6''	9°27'59,88''
-9°57'18,67''	9°28'21,56''
-9°57'18,67''	9°28'21,56''
<b>-9°57'18,67''</b>	<b>9°28'21,56''</b>

# Numerické řešení chodu soustavy

## Newton - Raphsonova metoda

- Výhody použití:
  - vysoká (kvadratická) rychlost konvergence
  - počet iterací - nezávislý na velikosti sítě (cca 3-7 iterací)
  - časové nároky - nízké (i pro velké sítě), zvyšují se jen lineárně
- Nevýhody použití:
  - komplikovanější matematický model (matice, derivace, atd.)
  - počet iterací může být rapidně navýšen při zahrnutí jalových mezí v PV uzlech a LTC transformátorů
  - silně závislá na počátečních odhadech stavových veličin
  - vysoká nejistota N-R metody v průběhu update procesu
  - problémy u řídicí propojených a ill-conditioned sítí

# Modifikace procesu N-R metody

- Inverze Jacobi matice – výpočtově nejnáročnější
  - výpočet Jacobi matice pouze v prvních 2 iteracích
  - výpočet Jacobi matice pouze 1x za 2-3 iterace (Lazy N-R)
  - aplikace řídkostních algoritmů (sparsity techniques)
- Jalové meze v PV uzlech – začlenění  $Q_i$  řádku do Jacobiánu

Př.: 4-uzlová síť, uzel č. 1 – ref., uzel č. 3 – PV

$$\begin{bmatrix} \Delta P_2 \\ \Delta P_3 \\ \Delta P_4 \\ \Delta Q_2 \\ \Delta Q_4 \end{bmatrix} = \begin{bmatrix} H_{22} & H_{23} & H_{24} & N_{22} & N_{24} \\ H_{32} & H_{33} & H_{34} & N_{32} & N_{34} \\ H_{42} & H_{43} & H_{44} & N_{42} & N_{44} \\ J_{22} & J_{23} & J_{24} & L_{22} & L_{24} \\ J_{42} & J_{43} & J_{44} & L_{42} & L_{44} \end{bmatrix} \begin{bmatrix} \Delta \theta_2 \\ \Delta \theta_3 \\ \Delta \theta_4 \\ \frac{\Delta V_2}{V_2} \\ \frac{\Delta V_4}{V_4} \end{bmatrix} \quad \begin{bmatrix} \Delta P_2 \\ \Delta P_3 \\ \Delta P_4 \\ \Delta Q_2 \\ 0 \\ \Delta Q_4 \end{bmatrix} = \begin{bmatrix} H_{22} & H_{23} & H_{24} & N_{22} & 0 & N_{24} \\ H_{32} & H_{33} & H_{34} & N_{32} & 0 & N_{34} \\ H_{42} & H_{43} & H_{44} & N_{42} & 0 & N_{44} \\ J_{22} & J_{23} & J_{24} & L_{22} & 0 & L_{24} \\ 0 & 0 & 0 & 0 & 1 & 0 \\ J_{42} & J_{43} & J_{44} & L_{42} & 0 & L_{44} \end{bmatrix} \begin{bmatrix} \Delta \theta_2 \\ \Delta \theta_3 \\ \Delta \theta_4 \\ \frac{\Delta V_2}{V_2} \\ \frac{\Delta V_3}{V_3} \\ \frac{\Delta V_4}{V_4} \end{bmatrix}$$

# Fast-Decoupled metoda

- možné rozdělení Jacobi matice (decoupling) díky silné závislosti  $P-\theta$  a  $Q-V$   $\longrightarrow$  významné matice  $H, L$
- výrazně menší hodnoty matic  $N, J$  způsobené:
  - buď malými rozdíly fázových posuvů  $\theta_{ik}$
  - nebo malými poměry  $r/x$ , resp.  $g/b$  (v přenosových sítích)
- základní iterační algoritmus Decoupled metody

$$\begin{bmatrix} \underline{\Delta P}^{(p-1)} \\ \underline{\Delta Q}^{(p-1)} \end{bmatrix} = \begin{bmatrix} \underline{H}^{(p-1)} & 0 \\ 0 & \underline{L}^{(p-1)} \end{bmatrix} \cdot \begin{bmatrix} \Delta \theta^{(p)} \\ \frac{\Delta V^{(p)}}{V^{(p-1)}} \end{bmatrix}$$

- možná další zjednodušení



# Fast-Decoupled metoda

- zjednodušení:

$$\cos \theta_{ik} \approx 1 \quad \sin \theta_{ik} \approx 0 \quad G_{ik} \sin \theta_{ik} \ll B_{ik} \quad Q_i \ll B_{ii} V_i^2$$
$$V_i \approx 1 \text{ pu} \quad V_k \approx 1 \text{ pu}$$

- modifikované prvky matic  $H$  a  $L$ :


$$H_{ii} = L_{ii} = -B_{ii} V_i^2 \quad H_{ik} = L_{ik} = -B_{ik} V_i V_k$$

- základní iterační algoritmus Fast-Decoupled metody

$$\underline{\Delta P}^{(p-1)} / \underline{V}^{(p-1)} = \underline{B}^{(p-1)} \Delta \theta^{(p)}$$

$$\underline{\Delta Q}^{(p-1)} / V^{(p-1)} = \underline{B}'^{(p-1)} \Delta V^{(p)}$$

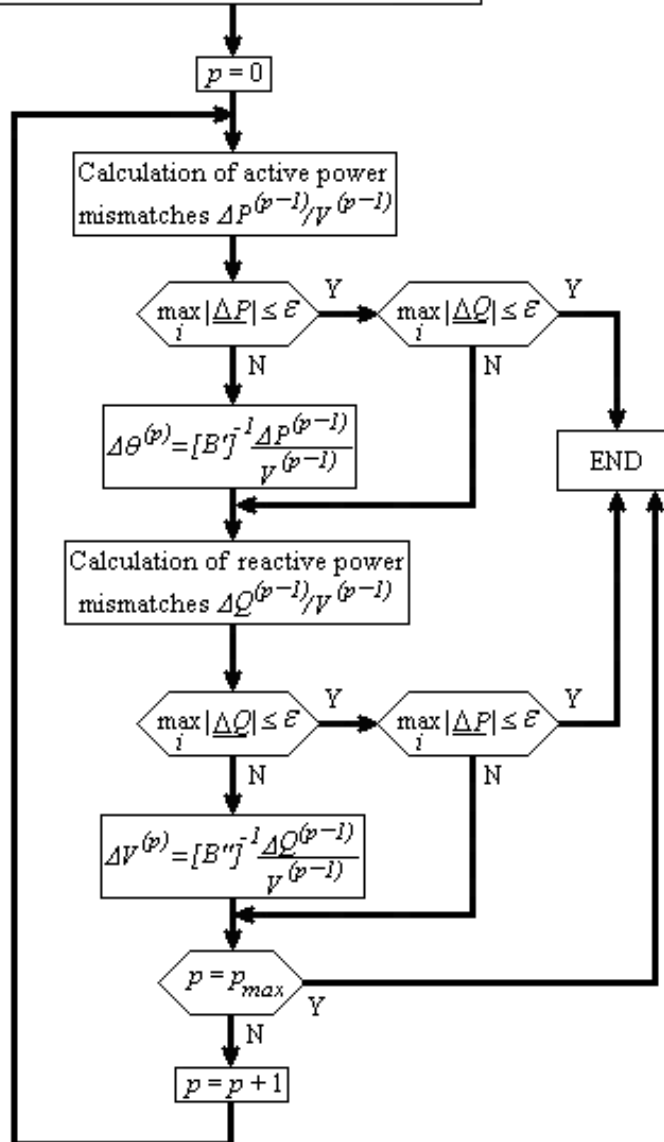
- zanedbání příčných kompenzačních prvků v matici  $H$  (i obsažených v pi-článcích), brány jen jmenovité převody
- zanedbány phase-shiftery v matici  $L$

# Fast-Decoupled metoda

- Další zjednodušení:
  - zanedbání sériových odporů v matici  $B'$  - FDLF typu XB
  - zanedbání sériových odporů v matici  $B''$  - FDLF typu BX
- Vlastnosti F-D metody:
  - obě matice  $B'$ ,  $B''$  se vypočítají pouze jednou
  - rychlost konvergence zhruba stejná jako u N-R, v blízkosti hledaného řešení klesá
  - náklady na iteraci jsou cca 4-5x menší než u N-R, o cca 50 % vyšší než u G-S
  - překvapivá spolehlivost F-D při tolika zanedbání u PS
  - problémy s konvergencí - vysoce zatížené sítě, DS (velké  $r/x$ )

# Vývojový diagram F-D metody

Bus admittance matrix  $\bar{A}$  of the network,  
power injections in PQ and PV buses,  
specified voltage magnitudes in PV buses,  
voltage value in the slack bus,  
initial voltage estimates in PQ buses,  
value of  $p_{max}$ , tolerance factor  $\varepsilon$ .



# DC load flow

- DC load flow = linearizovaná aproximace AC load flow
- Předpoklady/zjednodušení pro každou větev sítě:

- $V_i = V_k = 1$

- $x_{ik} \gg r_{ik} \rightarrow g_{ik} = \frac{r_{ik}}{r_{ik}^2 + x_{ik}^2} \approx 0 \quad b_{ik} = -\frac{x_{ik}}{r_{ik}^2 + x_{ik}^2} \approx -\frac{1}{x_{ik}}$

- $\theta_i - \theta_k \approx 0 \rightarrow \cos(\theta_i - \theta_k) \approx 1 \quad \sin(\theta_i - \theta_k) \approx \theta_i - \theta_k$



$$P_{ik} \approx \frac{1}{x_{ij}} (\theta_i - \theta_k) \rightarrow P_i = \sum_j \frac{1}{x_{ij}} (\theta_i - \theta_k)$$

- V maticovém tvaru:

$$\begin{bmatrix} P_1 \\ P_2 \\ \dots \\ P_n \end{bmatrix} = [B_x] \begin{bmatrix} \theta_1 \\ \theta_2 \\ \dots \\ \theta_n \end{bmatrix}, \text{ kde } B_{x,ii} = \sum_j \frac{1}{x_{ij}}; B_{x,ij} = -\frac{1}{x_{ij}}$$

# Aplikace jalových mezí v PV uzlech

- Generovaný jalový výkon musí být v daných mezích

$$Q_{Gi \min} \leq Q_{Gi} \leq Q_{Gi \max}$$

- Meze mohou nabývat kladných i záporných hodnot
- Napěťový regulátor řídí velikost napětí v uzlu a udržuje hodnotu jalového výkonu v daném rozmezí.
- V krizových situacích (ztráta regulace) přepne daný PV uzel na PQ uzel, ve kterém je jalový výkon konstantní ( $Q_{Gi \min}$  nebo  $Q_{Gi \max}$ ) a napětí se musí vypočítat nové.
- Při load flow analýze se provádí přepnutí typu uzlu z PV na PQ, ale není zřejmé, zda toto přepnutí platí po zbytek numerického výpočtu nebo jen pro danou iteraci

# Aplikace jalových mezí v PV uzlech

- Bus-Type Switching Logics
- A) Standartní PV-PQ logika
  - iterativně porovnává jalový výkon ve všech PV uzlech sítě
  - při překročení meze, přepíná PV uzel na PQ, nastaví jalový výkon na hodnotu překročené meze a uvolní napětí pro

výpočet

$$\begin{aligned} & \text{if } Q_{Gi} > Q_{Gi \max} \\ & \quad Q_{Gi} = Q_{Gi \max}; PV \rightarrow PQ; \\ & \text{else if } Q_{Gi} < Q_{Gi \min} \\ & \quad Q_{Gi} = Q_{Gi \min}; PV \rightarrow PQ; \\ & \text{end} \end{aligned}$$

Výsledek: Nedává spolehlivé výsledky ve všech LF studiích

# Aplikace jalových mezí v PV uzlech

- B) Vylepšená PV-PQ logika

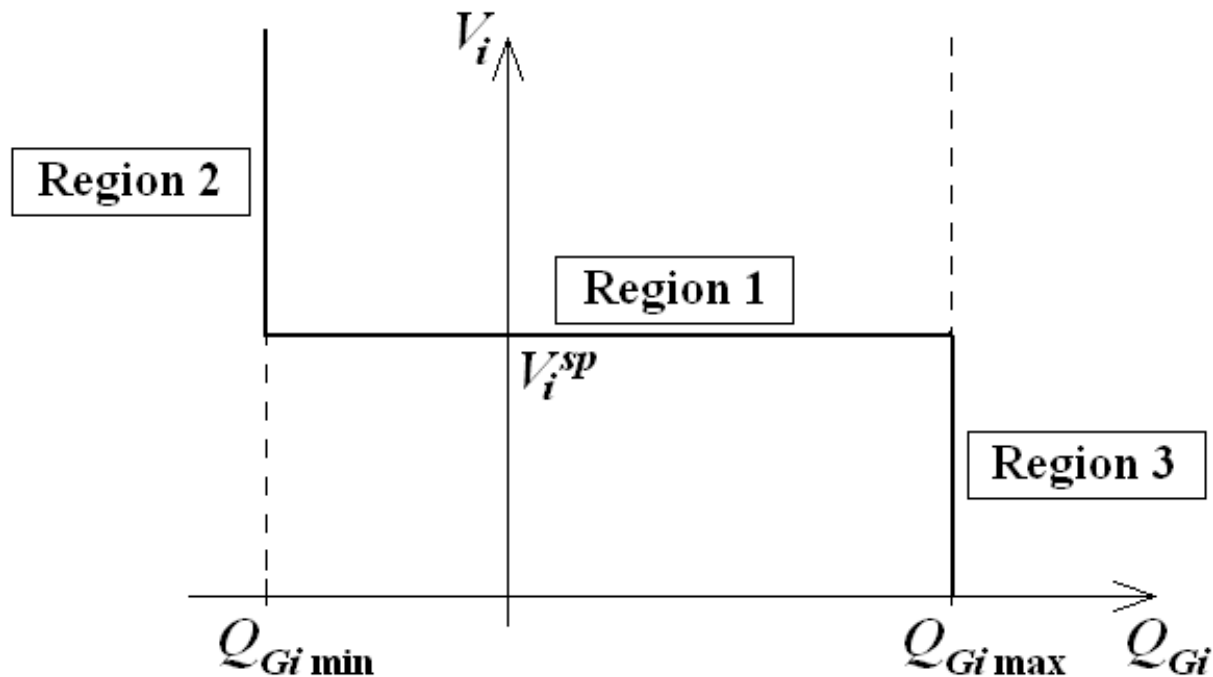
- pracuje pouze blízko konvergence, ne v každé iteraci
- v prvním kroku zjišťuje velikost odchylky jalového výkonu od překročené meze
- v druhém kroku přepne z PV na PQ pouze ten PV uzel s největší odchylkou jalového výkonu od příslušné meze

```
if  $Q_{Gi} > Q_{Gi \max}$                                 if  $M_i == M_{\max}$   
     $M_i = Q_{Gi} - Q_{Gi \max}$  ;                          if  $Q_{Gi} > Q_{Gi \max}$      $Q_{Gi} = Q_{Gi \max}$  ; PV  $\rightarrow$  PQ ;  
else if  $Q_{Gi} < Q_{Gi \min}$                                 else if  $Q_{Gi} < Q_{Gi \min}$      $Q_{Gi} = Q_{Gi \min}$  ; PV  $\rightarrow$  PQ ;  
     $M_i = Q_{Gi \min} - Q_{Gi}$  ;                            end  
end                                                        end
```

- Výsledek: Vykazuje spolehlivost a flexibilitu u G-S metody

# Aplikace jalových mezí v PV uzlech

- C) Robustní PV-PQ a PQ-PV logika
  - pracuje v každé iteraci, 2-kroková - z PV na PQ, z PQ na PV
  - první krok je standardní PV-PQ logika (logika A)
  - v druhém kroku – využití silné Q-V závislosti





# Aplikace jalových mezí v PV uzlech

- C) Robustní PV-PQ a PQ-PV logika

```
if type1 == PQ & type0 == PV
  if (  $Q_{Gi} == Q_{Gi \max} \ \& \ V_i > V_i^{sp}$  ) or (  $Q_{Gi} == Q_{Gi \min} \ \& \ V_i < V_i^{sp}$  )
     $V_i = V_i^{sp}; PQ \rightarrow PV;$ 
  end
end
```

Pozn.: type1 – typ uzlu v aktuální iteraci

type0 – typ uzlu na začátku numerického výpočtu

Výsledek: flexibilní, spolehlivá, rychlá pro N-R metodu

pouze mírný nárůst počtu iterací

minimální zhoršení chodu N-R metody

je aplikována v komerčním programu PowerWorld

možné nebezpečí v četném oscilování mezi PV a PQ

# Zlepšení chování G-S metody

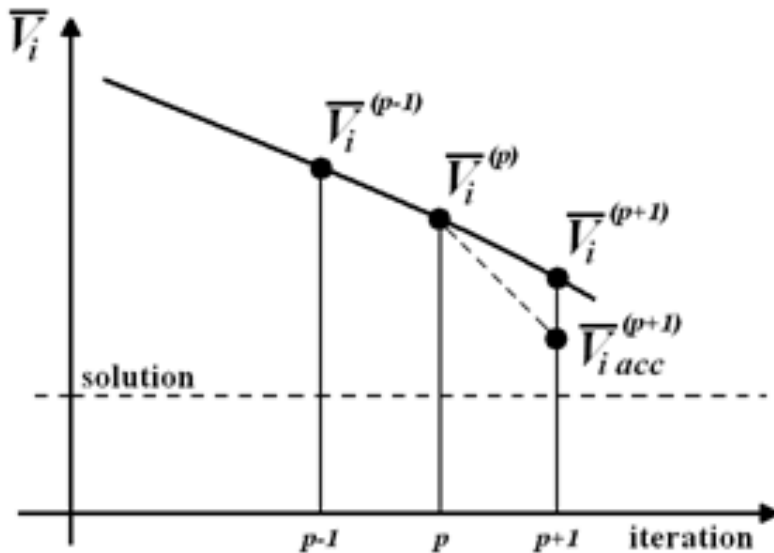
- Akcelerace numerického výpočtu (redukce počtu iterací)
  - 1) akcelerační metoda s výpočtem uzlových napětí 2x za iteraci
    - relativně lehký přístup, snaha zvýšit rychlost konvergence na kvadratickou, pozitivní výsledky
    - výrazně vyšší časové nároky na výpočet (hlavně u větších sítí)
  - 2) akcelerační technika obsahující tzv. succesive overrelaxation process (akcelerační faktor  $\alpha$ )

$$\bar{V}_i^{(p)}{}_{acc} = \bar{V}_i^{(p-1)} + \alpha \left( \bar{V}_i^{(p)} - \bar{V}_i^{(p-1)} \right)$$

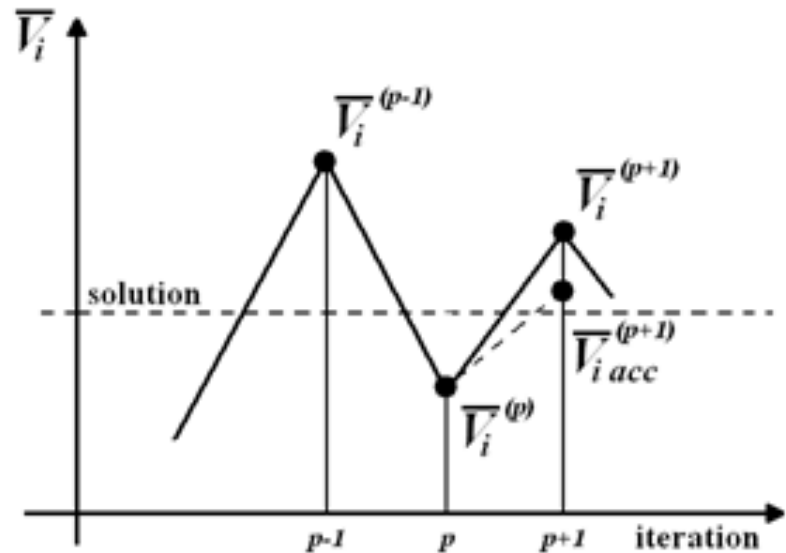
typické hodnoty  $\alpha$  : v rozmezí 1.3 až 1.7, doporučeno 1.6

# Zlepšení chování G-S metody

- Akcelerace numerického výpočtu (redukce počtu iterací)
  - 3) akcelerační metoda s akceleračními a retardačními faktory
    - akcelerační faktor pro zrychlení postupu k výsledku - ad a)
    - retardační faktor pro minimalizaci možných oscilací - ad b)



a)



b)

# Zlepšení chování G-S metody

- Akcelerace numerického výpočtu (redukce počtu iterací)

3) akcelerační metoda s akceleračními a retardačními faktory

$$\text{if } (V_i^{(p+1)} - V_i^{(p)}) \cdot (V_i^{(p)} - V_i^{(p-1)}) \geq 0$$

$$m_{i1} > 1.0;$$

*else if*

$$m_{i1} \leq 1.0;$$

*end*

$$\text{if } (\theta_i^{(p+1)} - \theta_i^{(p)}) \cdot (\theta_i^{(p)} - \theta_i^{(p-1)}) \geq 0$$

$$m_{i2} > 1.0;$$

*else if*

$$m_{i2} \leq 1.0;$$

*end*

$$V_{i \text{ acc}}^{(p+1)} = V_i^{(p)} + m_{i1} (V_i^{(p+1)} - V_i^{(p)}); \theta_{i \text{ acc}}^{(p+1)} = \theta_i^{(p)} + m_{i2} (\theta_i^{(p+1)} - \theta_i^{(p)}); \bar{V}_{i \text{ acc}}^{(p+1)} = V_{i \text{ acc}}^{(p+1)} e^{j\theta_{i \text{ acc}}^{(p+1)}};$$

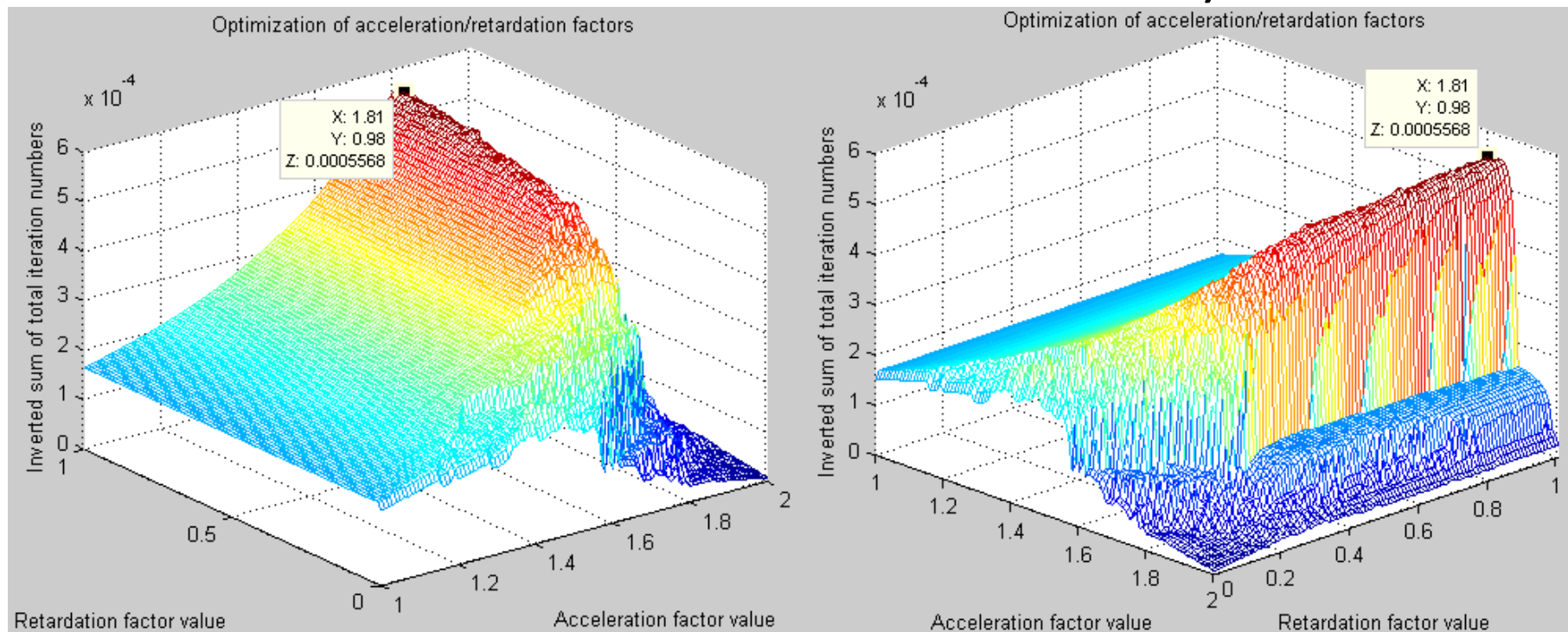
Hodnoty: akcelerace ... 1 - 2      zpomalení ... 0 - 1

- nutno ukládat napěťové hodnoty z posledních 2 iterací

- doporučené hodnoty nejsou v literatuře uvedeny

# Zlepšení chování G-S metody

- Optimalizace hodnot akcelerace a retardace (aplikace na širokou škálu sítí v rozmezí 3 - 300 uzlů)

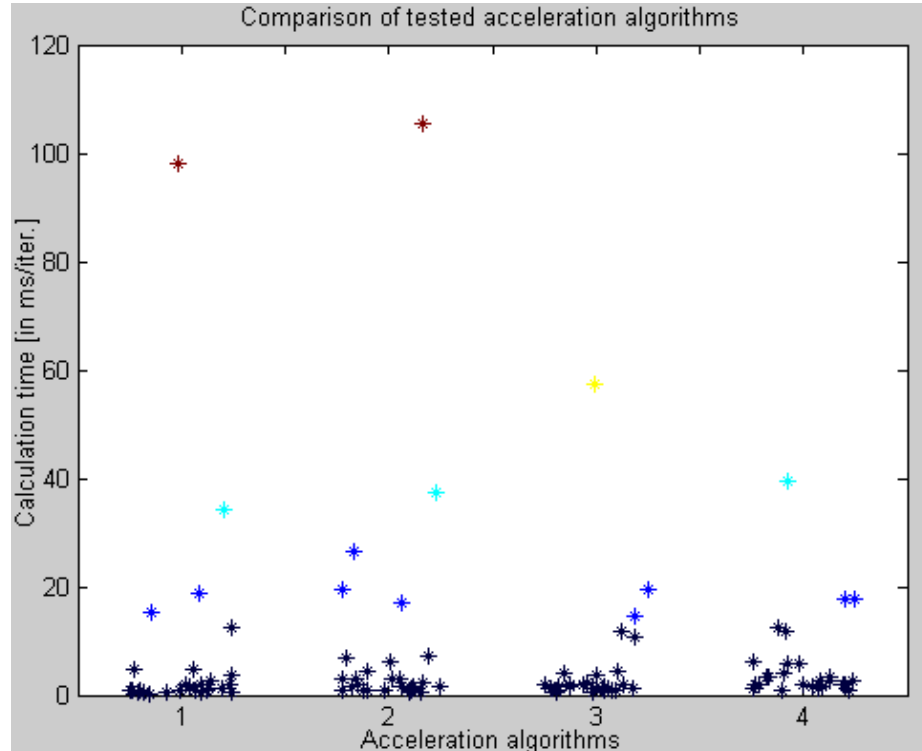
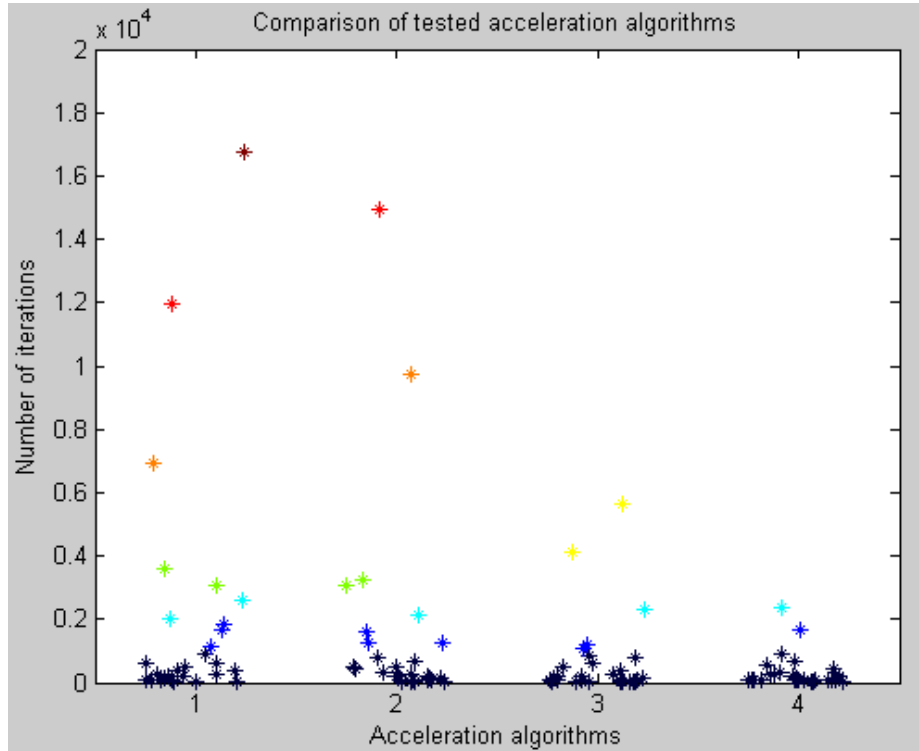


Výsledné hodnoty: akcelerace ... 1.81 zpomalení ... 0.98

V případě divergence: doporučené hodnoty .... 1.5 / 1

# Zlepšení chování G-S metody

- Výsledný testing (28 real sítí v rozmezí 3 - 300 uzlů)





Počet iterací: alg. 2: -20%, alg. 3: -66%, alg. 4: -73% (median)

Doba výpočtu: alg. 2: +52%, alg. 3: +6%, alg. 4: +77% (median)



doporučená spolupráce algoritmů 3 a 4

# Zlepšení chování N-R metody

- N-R metoda může:
    - konvergovat ke správnému řešení
    - divergovat pokud žádné řešení neexistuje
    - divergovat pokud reálné řešení existuje
    - konvergovat k jinému (nereálnému) výsledku
  - Důvody?
    - silná závislost na počátečních startovních hodnotách
    - silná nejistota během update procesu
-  proto je snaha zlepšit chování N-R metody v těchto dvou oblastech
-  tzv. Start Point Estimation and State Update methods

# Zlepšení chování N-R metody

- Start Point Estimation methods
  - tzv. flat start (1.0 pu pro PQ uzly s nulovými  $\theta$ ) nefunguje u všech sítí spolehlivě
  - snaha pro začátek vnutit lepší startovní hodnoty, které vedou ke konvergenci k správnému výsledku
  - možné přístupy:
    - One-Shot Gauss-Seidel
    - nebo
    - One-Shot Fast-Decoupled
  - OSFD použit v load flow programu PowerWorld Simulator
  - očekávané problémy: u sítí s vysokým R/X poměrem (DS)



# Zlepšení chování N-R metody

- State Update methods – Power Mismatch Minimization using relaxation factor  $\beta$

- vhodné  $\beta$  pro minimalizaci rozdílů v každé iteraci

$$\theta_i^{(p+1)} = \theta_i^{(p)} + \beta \Delta\theta_i \quad V_i^{(p+1)} = V_i^{(p)} + \beta \Delta V_i$$

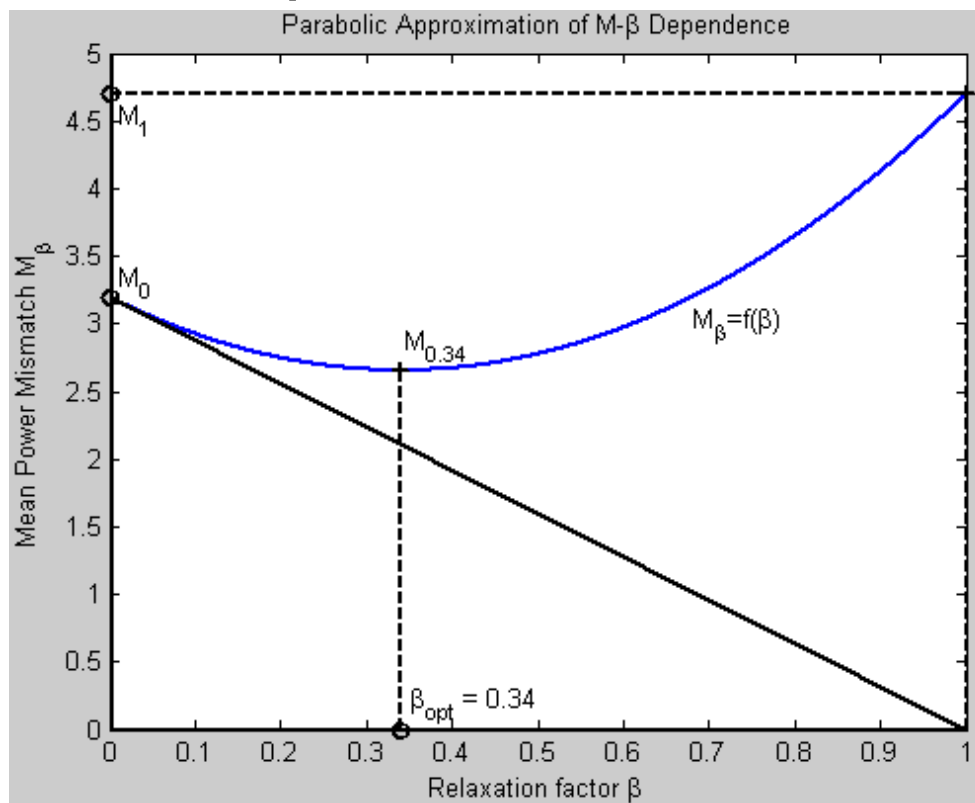
- vyhodnocení tzv. středního výkonového rozdílu

$$M_\beta = \sqrt{\frac{1}{n-1} \sum_{\substack{i=1 \\ i \neq ref}}^n (\Delta P_i^2 + \Delta Q_i^2)}$$

- Postup:
- 1) provedení zkušebního full updatu a spočtení  $M_1$
  - 2) porovnání  $M_1$  s  $M_0$  (z předchozí iterace)
  - 3) pokud  $M_1 < 1/2M_0$ , pak se jde na novou iteraci  
jinak výpočet optimálního  $\beta$ , update a skok na další iteraci

# Zlepšení chování N-R metody

- State Update methods – Power Mismatch Minimization s relaxation factory  $\beta$

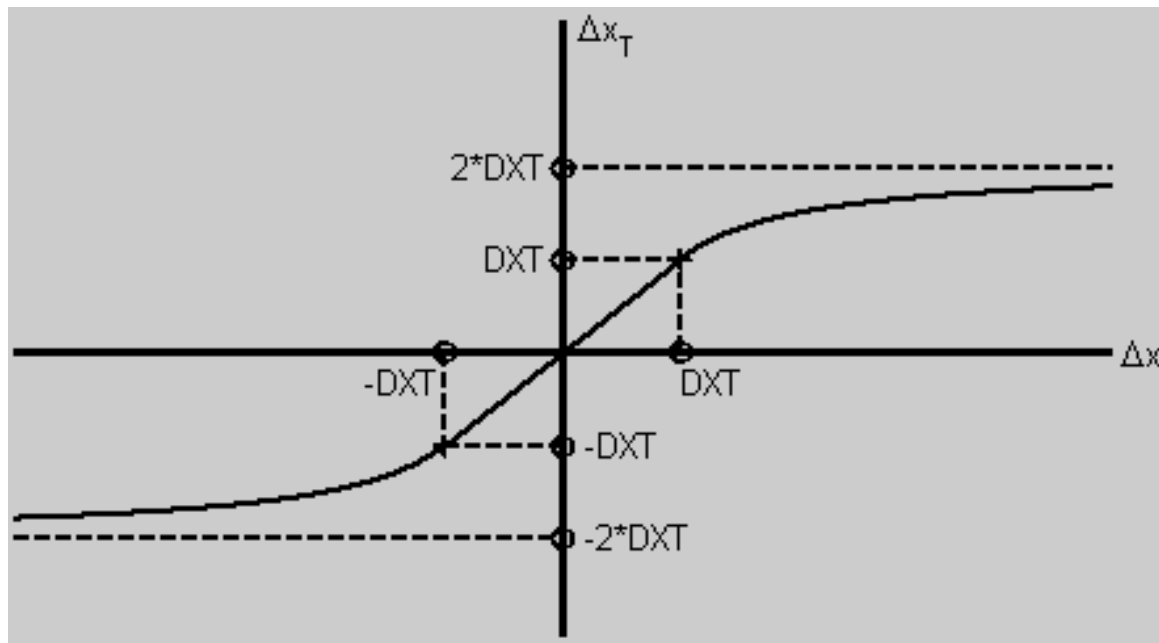


$$\beta_{opt} \cong \frac{M_0}{2M_1}$$

- Problémy: ill-conditioned sítě, vyšší výpočtové nároky

# Zlepšení chování N-R metody

- State Update methods – State Update Truncation
  - použití horních mezí pro přírůstkový vektor N-R metody
  - vhodnější pružnější ořezání než pouze 2-hodnotové omezení



$$\Delta x_T = \begin{cases} \Delta x & \text{if } |\Delta x| < DXT \\ 2 \operatorname{sgn}(\Delta x) DXT - \frac{DXT^2}{\Delta x} & \text{if } |\Delta x| \geq DXT \end{cases}, \text{ kde } DXT_\theta = 0.3 \text{ rad}, DXT_V = 0.2 \text{ pu}$$

# Zlepšení chování N-R metody

- Další možný přístup

- začlenění také 2. řádu rozvoje Taylorovy řady (Hessián)

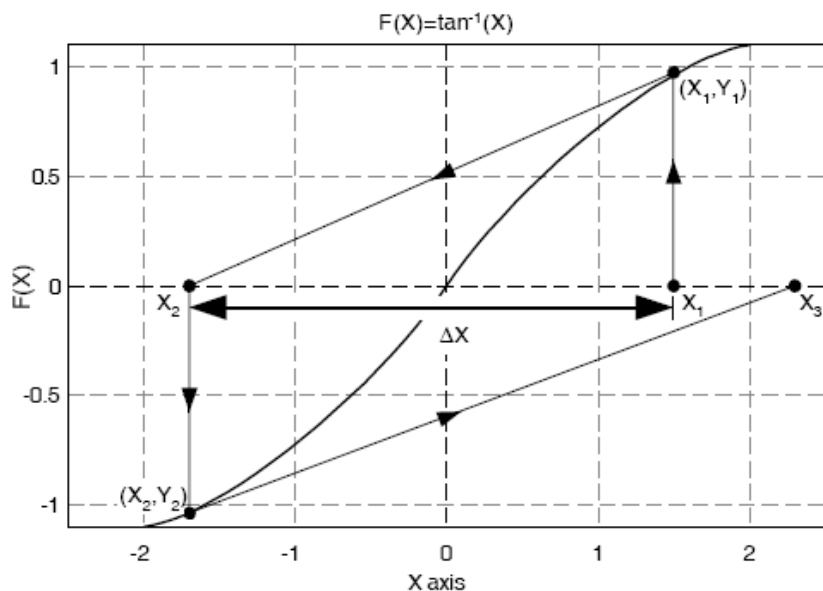


Fig. 1 Pathological case for the Newton method

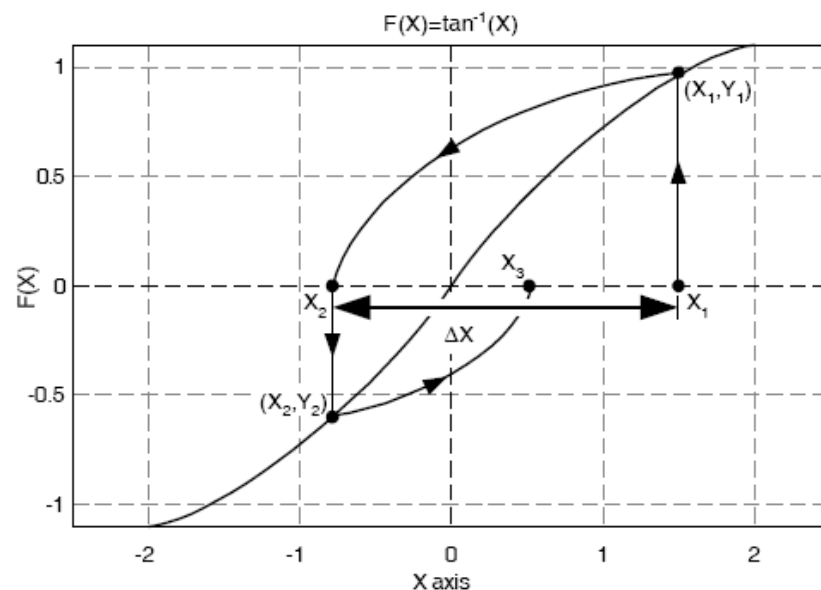


Fig. 2 Convergence using the Hessian matrix

Zdroj: P.J. Lagace, M.H. Vuong and I. Kamwa, "Improving Power Flow Convergence by Newton Raphson with a Levenberg-Marquardt Method", *IEEE Transactions*, Montreal, 2008.

- Problémy: nutné začlenění sparsity techniques

# Zlepšení chování N-R metody

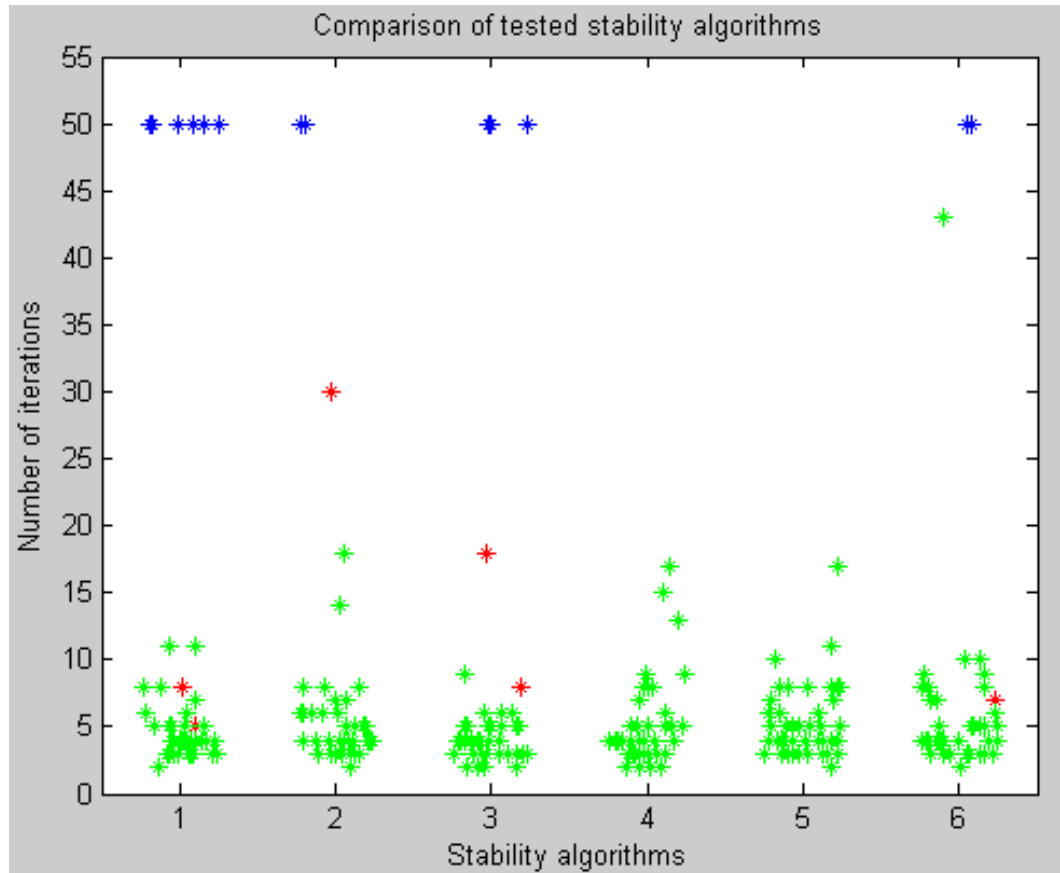
- Výsledný testing (50 real sítí v rozmezí 3 - 2746 uzlů)

Metoda	Popis
1	původní Newton-Raphson
2	State Update Truncation (SUT) pouze
3	One-Shot Fast-Decoupled (OSFD) + $V$ update pouze
4	OSFD algorithm ( $V$ update pouze) + SUT
5	OSGS algorithm ( $V$ update pouze) + SUT
6	Power Mismatch Minimization s $\beta$ -factory

- Důvody:
  - updaty  $\theta$  vykazují rušivé vlivy na numerický výpočet
  - optimalizovaná součinnost více metod k většímu zlepšení chování N-R metody

# Zlepšení chování N-R metody

- Výsledný testing (50 real sítí v rozmezí 3 - 2746 uzlů)



- Nejlepší výsledky:
  - OSFD/OSGS (pouze V updaty) v kombinaci s SUT algoritmem