UC Dublin, February 23, 2017

The factored approach to solving nonlinear power system problems

Antonio Gómez-Expósito
Dpt. of Electrical Eng. – Endesa Red Chair
University of Sevilla

Contents

Introduction

- John Rice (father of *mathematical software* concept, 1969):

  “solving systems of nonlinear equations is perhaps the most difficult problem in all of numerical computations”

Introduction

- In general, nonlinear equation systems *must* be solved iteratively
  - Some trivial cases allow for closed-form solutions
- **Newton-Raphson** iterative algorithm is the reference method in the general case (since the XVII century):
  - First-order local approximations successively performed
  - Quadratic convergence rate near the solution
  - Same divergence rate outside the basins of attraction
  - Exploitation of Jacobian sparsity for large systems: key for its terrific success in power systems applications

Tinney *et. al.* late in the 60's: sparse LU factorization of Jacobian

Introduction

- Many improvements so far to *vanilla* NR:
  - Quasi-Newton methods: approximate (constant) Jacobian
  - Inexact Newton methods: approximate computation of Newton's step (preconditioners)
  - Higher-order methods: super-quadratic convergence (more costly)
  - Globally convergent Newton methods: line search, continuation/homotopy, trust region
  - Others: polynomial approximation (Chebyshev's), solution of differential equations (Davidenko's), etc.

Motivation

- **Key idea**:

  "Look for a global nonlinear transformation that creates an algebraically equivalent system on which Newton's method does better because the new system is more linear".

  So far, "no general way to apply this idea has been found; its application is problem-specific".

  [Judd K.L., "Numerical methods in economics", MIT Press, New York, 1998, pp. 174-176]

- **Factored approach:** first systematic attempt to achieve this goal on a broad range of nonlinear systems of practical interest

Motivation

- The customary ultra-compact expression:

  $\boxed{h(x)=p}$ (***p*** is given, ***x*** to be found)

  hides the typical structure of ***h(.)***: **sums of nonlinear expressions of different complexity**

- Large-scale nonlinear systems are always **sparse**:
  - Number of additive terms in a set of ***n*** nonlinear eqs. in ***n*** variables is roughly ***O(n)***, **not** ***O(n²)***
  - Some of those linear/nonlinear terms may appear several times:

Let ***m>n*** be the number of distinct terms

Factored solution approach

- Let *y* be an auxiliary vector composed of the ***m*** distinct additive terms in ***h(.)***, each with trivial inverse. Then, the following factored form arises:

$$Ey = p \longrightarrow$$ Under-determined linear system

$$u = f(y) \longrightarrow$$ ***m*** one-to-one mappings with explicit inverse: $y_i = f_i^{-1}(u_i)$

$$Cx = u \longrightarrow$$ Over-determined linear system

Factored solution approach

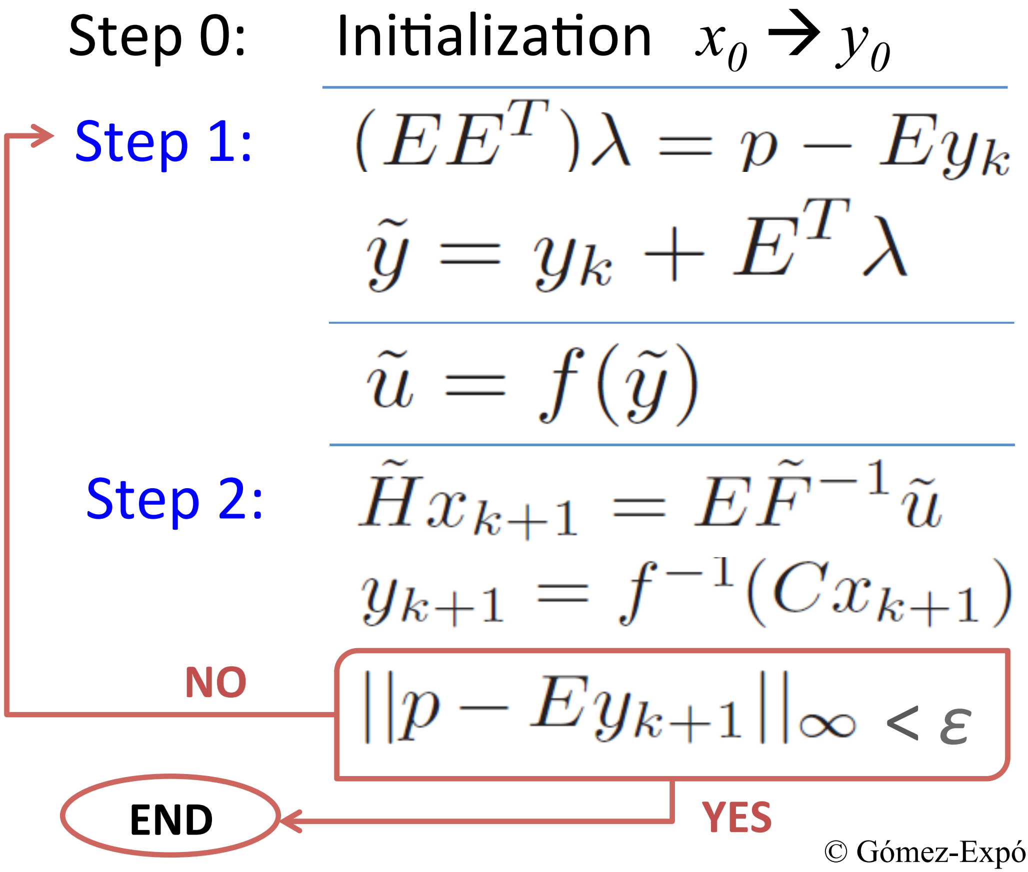
- Conventional NR (for comparison):

$$\boxed{h(x) = p} \qquad H_k \Delta x_k = \Delta p_k = p - h(x_k)$$

- **Two-step factored procedure**:

$$\begin{aligned} Ey &= p \\ u &= f(y) \\ Cx &= u \end{aligned}$$

Factored Jacobian: $\tilde{H} = E\tilde{F}^{-1}C$



Step 0: Initialization $x_0 \rightarrow y_0$

Step 1:

$$(EE^T)\lambda = p - Ey_k$$

$$\tilde{y} = y_k + E^T\lambda$$

$$\tilde{u} = f(\tilde{y})$$

Step 2:

$$\tilde{H}x_{k+1} = E\tilde{F}^{-1}\tilde{u}$$

$$y_{k+1} = f^{-1}(Cx_{k+1})$$

$$||p - Ey_{k+1}||_\infty < \varepsilon$$

NO

YES

END

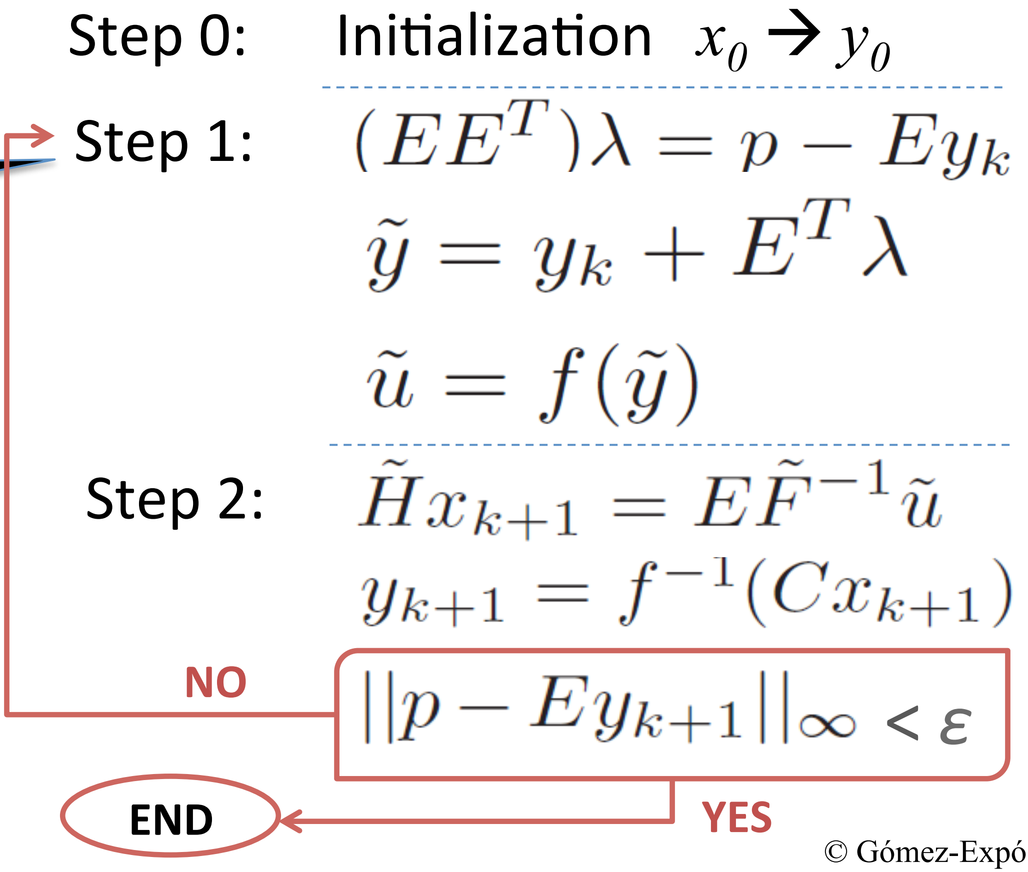
Factored solution approach

- Conventional NR:

$$\boxed{h(x) = p} \qquad H_k \Delta x_k = \Delta p_k = p - h(x_k)$$

Computational issues:

Least-distance problem:
Cholesky factors computed only once



Step 0: Initialization $x_0 \rightarrow y_0$

Step 1:

$$(EE^T)\lambda = p - Ey_k$$

$$\tilde{y} = y_k + E^T\lambda$$

$$\tilde{u} = f(\tilde{y})$$

Step 2:

$$\tilde{H}x_{k+1} = E\tilde{F}^{-1}\tilde{u}$$

$$y_{k+1} = f^{-1}(Cx_{k+1})$$

$$\boxed{||p - Ey_{k+1}||_\infty < \varepsilon}$$

© Gómez-Expósito

Factored solution approach

- Conventional NR:

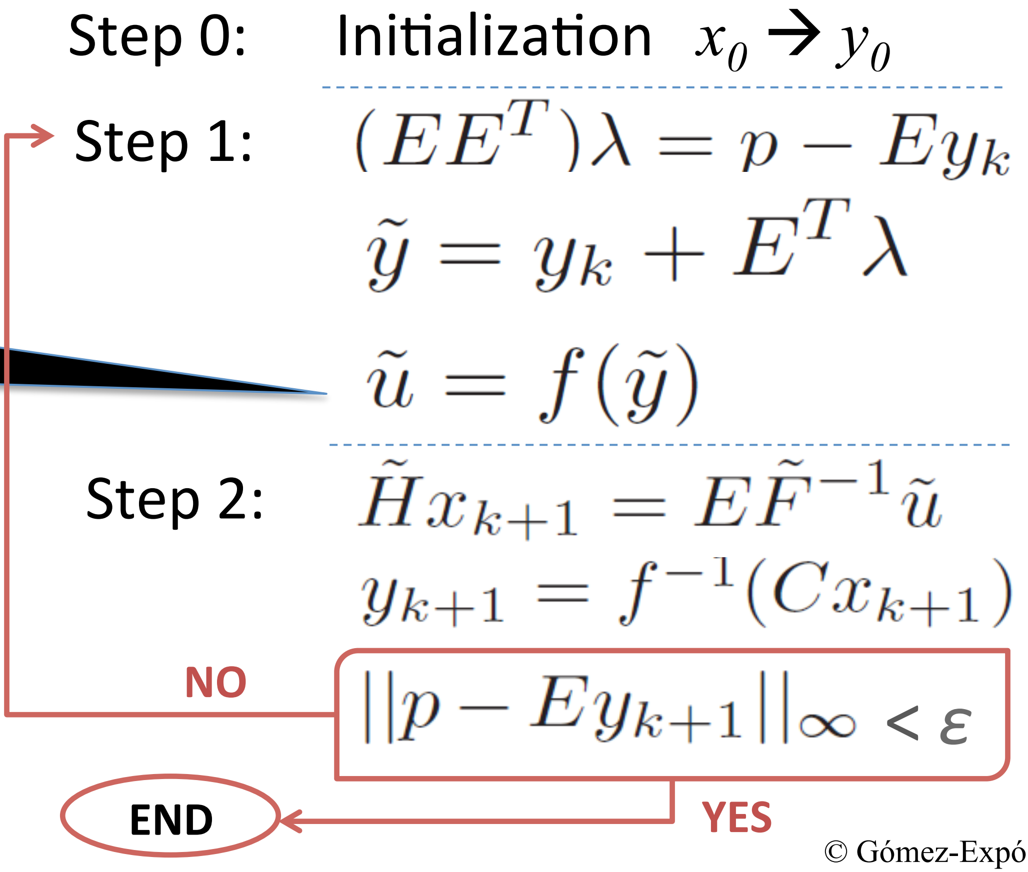
$$\boxed{h(x) = p} \qquad H_k \Delta x_k = \Delta p_k = p - h(x_k)$$

Computational issues:

Trivial one-to-one nonlinear mapping f() with diagonal Jacobian

$$u_i = f_i(y_i), \quad i = 1, \ldots m$$

$$y_i = f_i^{-1}(u_i)$$



Step 0: Initialization $x_0 \rightarrow y_0$

Step 1:

$$(EE^T)\lambda = p - Ey_k$$

$$\tilde{y} = y_k + E^T\lambda$$

$$\tilde{u} = f(\tilde{y})$$

Step 2:

$$\tilde{H} x_{k+1} = E\tilde{F}^{-1}\tilde{u}$$

$$y_{k+1} = f^{-1}(Cx_{k+1})$$

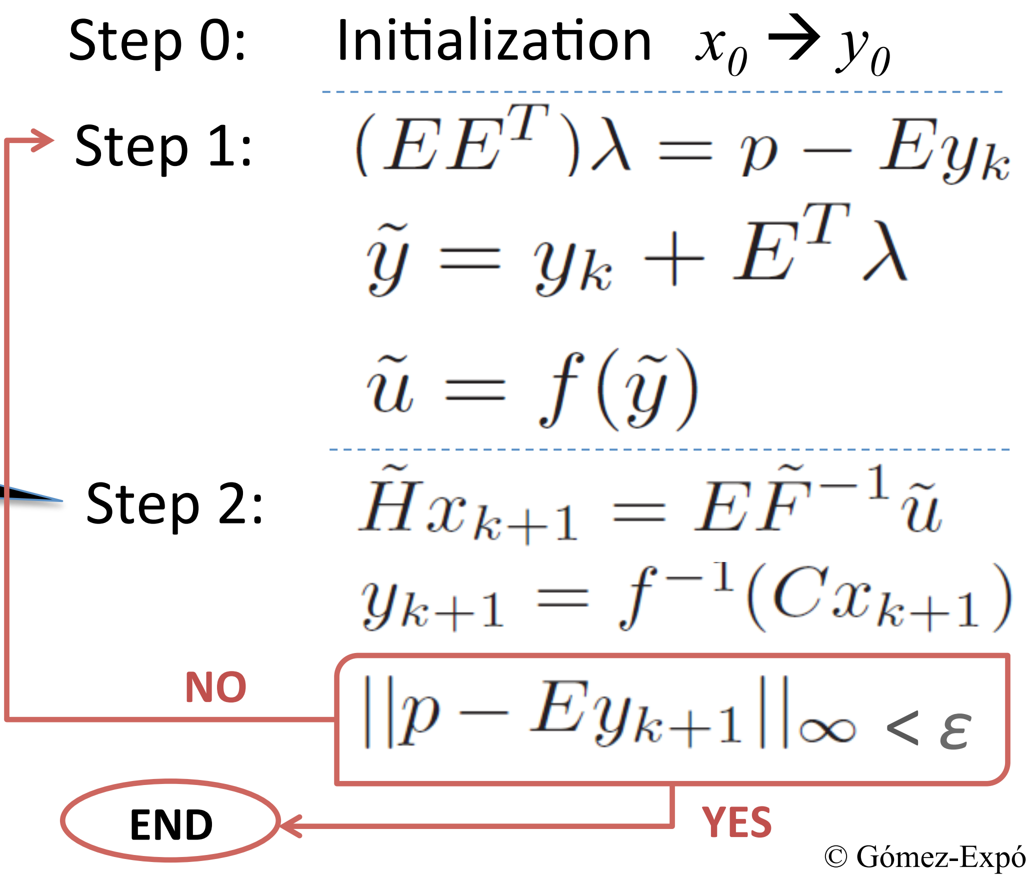
$$||p - Ey_{k+1}||_\infty < \varepsilon$$

Factored solution approach

- Conventional NR:

$$\boxed{h(x) = p} \qquad H_k \Delta x_k = \Delta p_k = p - h(x_k)$$

Computational issues:



Step 0: Initialization $x_0 \rightarrow y_0$

Step 1:

$$(EE^T)\lambda = p - Ey_k$$

$$\tilde{y} = y_k + E^T \lambda$$

$$\tilde{u} = f(\tilde{y})$$

Step 2:

$$\tilde{H} x_{k+1} = E\tilde{F}^{-1}\tilde{u}$$

$$y_{k+1} = f^{-1}(Cx_{k+1})$$

$$\boxed{||p - Ey_{k+1}||_\infty < \varepsilon}$$

Least-squares problem:
Factored Jacobian simpler to compute

$$\tilde{H} = E\tilde{F}^{-1}C$$

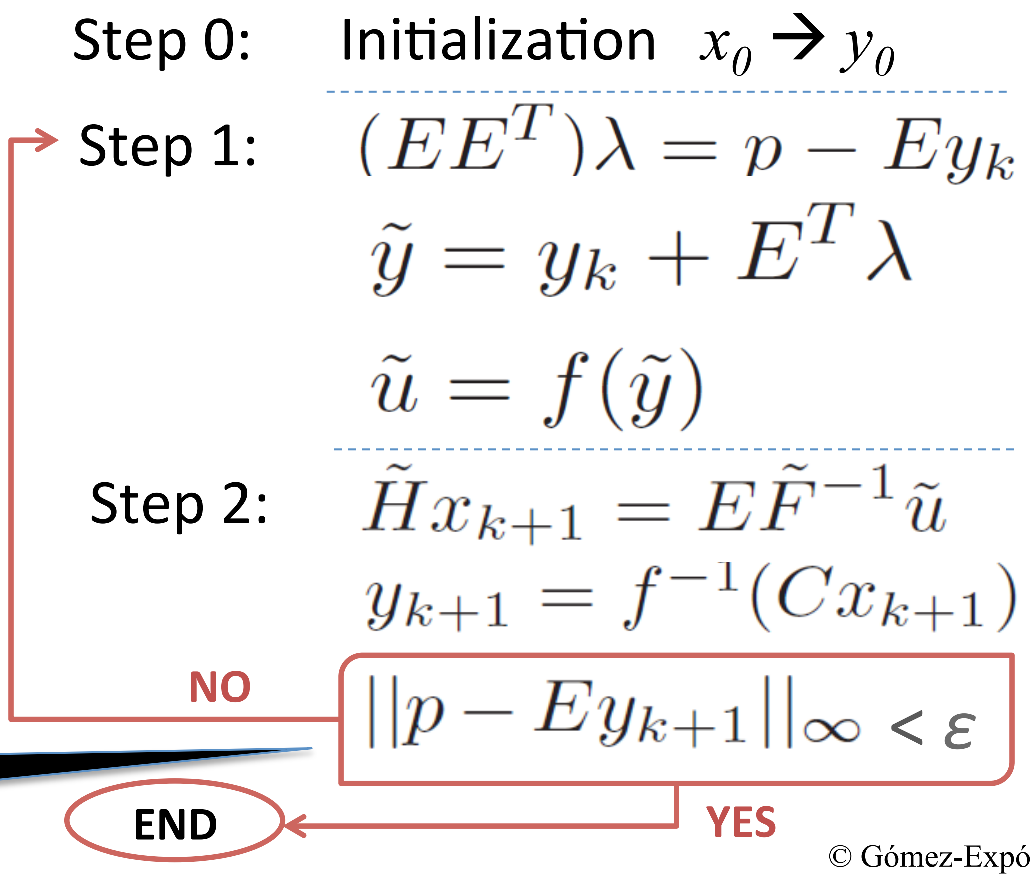
© Gómez-Expósito

Factored solution approach

- Conventional NR:

$$\boxed{h(x) = p} \qquad H_k \Delta x_k = \Delta p_k = p - h(x_k)$$

Computational issues:



Step 0: Initialization $x_0 \rightarrow y_0$

Step 1:

$$(EE^T)\lambda = p - Ey_k$$

$$\tilde{y} = y_k + E^T\lambda$$

$$\tilde{u} = f(\tilde{y})$$

Step 2:

$$\tilde{H}x_{k+1} = E\tilde{F}^{-1}\tilde{u}$$

$$y_{k+1} = f^{-1}(Cx_{k+1})$$

$$\boxed{||p - Ey_{k+1}||_\infty < \varepsilon}$$

NO

YES

END

Linear mismatch vector
used in next iteration

Canonical forms

1. Sums of single-variable nonlinear elementary functions:

$$h_i(x) = \sum_{j=1}^{m_i} c_{ij} h_{ij}(x_k), \qquad y_{ij} = h_{ij}(x_k) \qquad u_{ij} = h_{ij}^{-1}(y_{ij}) = x_k.$$

Example:

$$\left.\begin{array}{c} p = x^4 - x^3 \\ y_1 = x^4 \quad \text{and} \quad y_2 = x^3 \end{array}\right\} \qquad p = y_1 - y_2.$$

$$E = \begin{pmatrix} 1 & -1 \end{pmatrix}, \quad C = \begin{pmatrix} 1 \\ 1 \end{pmatrix},$$

$$u = f(y) = \begin{pmatrix} \sqrt[4]{y_1} \\ \sqrt[3]{y_2} \end{pmatrix}, \quad F^{-1} = \begin{pmatrix} 4u_1^3 & 0 \\ 0 & 3u_2^2 \end{pmatrix}.$$

Canonical forms

2. Sums of products of single-variable power functions:

$$h_i(x) = \sum_{j=1}^{m_i} c_{ij} \prod_{k=1}^{n_j} x_k^{q_k},$$

Preliminary logarithmic change of variables:

$$\alpha_k = \ln x_k, \quad k = 1, \ldots, n.$$

$$y_{ij} = \prod_{k=1}^{n_j} x_k^{q_k} = \prod_{k=1}^{n_j} \exp\left(q_k \alpha_k\right), \quad \left\{ \begin{array}{l} u_{ij} = \ln y_{ij} \quad \Rightarrow \quad y_{ij} = \exp u_{ij}, \\ u_{ij} = \displaystyle\sum_{k=1}^{n_j} q_k \alpha_k. \end{array} \right.$$

$$x_k = \exp \alpha_k, \quad k = 1, \ldots, n.$$

© Gómez-Expósito

Canonical forms

Example:

$$\left.\begin{array}{l}\boxed{\begin{array}{l}p_1 = x_1x_2 + x_1x_2^2 \\ p_2 = 2x_1^2x_2 - x_1^2,\end{array}} \\ y_1 = x_1x_2, \quad y_2 = x_1x_2^2, \\ y_3 = x_1^2x_2, \quad y_4 = x_1^2\end{array}\right\} \quad \begin{pmatrix} p_1 \\ p_2 \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 2 & -1 \end{pmatrix}}_{E} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{pmatrix}.$$

$$u_i = \ln y_i, \quad i = 1,2,3,4 \quad \longrightarrow \quad \underbrace{\begin{pmatrix} 1 & 1 \\ 1 & 2 \\ 2 & 1 \\ 2 & 0 \end{pmatrix}}_{C} \begin{pmatrix} \alpha_1 \\ \alpha_2 \end{pmatrix} = \begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{pmatrix}.$$

$$F^{-1} = \begin{pmatrix} \exp u_1 & 0 & 0 & 0 \\ 0 & \exp u_2 & 0 & 0 \\ 0 & 0 & \exp u_3 & 0 \\ 0 & 0 & 0 & \exp u_4 \end{pmatrix} \qquad x_i = \exp \alpha_i, \quad i = 1,2,3,4.$$

© Gómez-Expósito

Canonical forms

Remark: If ***m=n***, then there is no need to iterate

$$\boxed{\begin{aligned} p_1 &= x_1 x_2 + x_1^2 \\ p_2 &= 2x_1 x_2 - 4x_1^2 \end{aligned}}$$

m=n=2

$$\begin{bmatrix} \boxed{\begin{matrix} p_1 \\ p_2 \end{matrix}} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 2 & -4 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \longrightarrow \begin{aligned} u_1 &= \ln y_1 \\ u_2 &= \ln y_2 \end{aligned}$$

$$\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 2 & 0 \end{bmatrix} \begin{bmatrix} \ln x_1 \\ \ln x_2 \end{bmatrix} \longrightarrow \boxed{\begin{matrix} x_1 \\ x_2 \end{matrix}}$$

Unlike in Newton's method, iterations arise because *m>n*

Canonical forms

3. Conversion to canonical forms: system augmentation

Add variables/equations as needed

Example:

$$\begin{aligned} p_1 &= x_1 \sin\left(x_1^2 + x_2\right) - x_1^2 \\ p_2 &= x_1^2 x_2 - \sqrt{x_2}. \end{aligned}$$

New variable introduced: $x_3 = \sin\left(x_1^2 + x_2\right),$

The equivalent "canonical" system becomes:

$$\begin{aligned} p_1 &= x_1 x_3 - x_1^2, \\ p_2 &= x_1^2 x_2 - \sqrt{x_2} \\ 0 &= x_1^2 + x_2 - \arcsin x_3. \end{aligned}$$

Extending the range of reachable solutions

Multiplicity of solutions: Which solution is iteratively reached?

- **Newton Raphson**: depends on the basin of attraction in which the initial guess x_o lies
- **Factored method**: can be controlled by selecting the computed range for each individual nonlinear function

Examples:

$$\boxed{y = x^q,} \quad q \text{ even} \quad \begin{cases} u = f(y) = \sqrt[q]{y} & \Rightarrow \quad \Re(u) > 0 \\ u = -\sqrt[q]{y}. & \Rightarrow \quad \Re(u) < 0 \end{cases}$$

Periodic functions

$$\boxed{y = \sin x,} \begin{cases} u = \arcsin y & \Rightarrow \quad -\pi/2 < u < \pi/2. \\ u = q\pi + (-1)^q \arcsin y. & \Rightarrow \quad (q - \frac{1}{2})\pi < u < (q + \frac{1}{2})\pi \end{cases}$$

Extending the range of reachable solutions

Example: **2x2 P. Boggs' system**

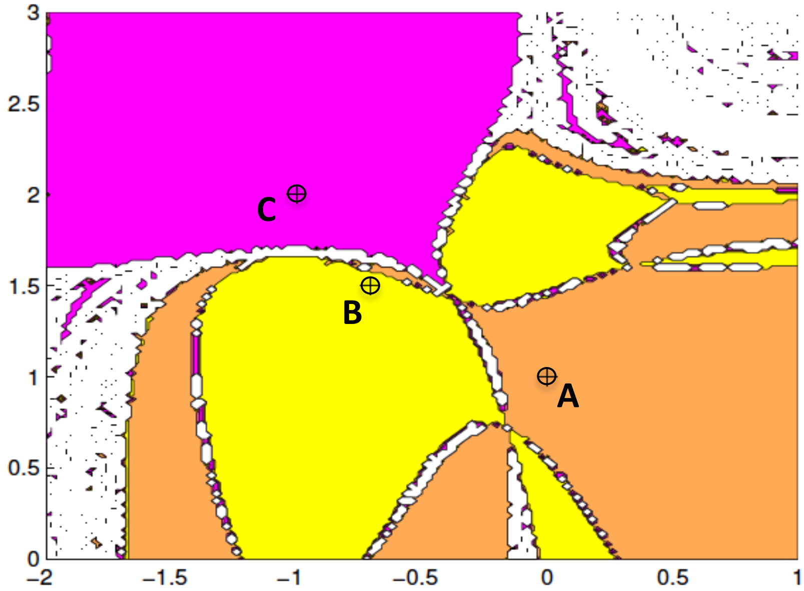
$$\begin{aligned} -1 &= x_1^2 - x_2 \\ 0 &= x_1 - \cos\left(\frac{\pi x_2}{2}\right) \end{aligned}$$

Only three solutions:

$$(0,1),\ (-1/\sqrt{2}, 3/2) \text{ and } (-1,2)$$

A **B** **C**

Newton Raphson's basins of attraction: unpredictable irregular behavior



White areas: divergence

© Gómez-Expósito

Extending the range of reachable solutions

Example: 2x2 P. Boggs' system

$$\boxed{\begin{aligned} -1 &= x_1^2 - x_2 \\ 0 &= x_1 - \cos\left(\frac{\pi x_2}{2}\right) \end{aligned}}$$

For any x_o !!

With the basic definitions converges to A

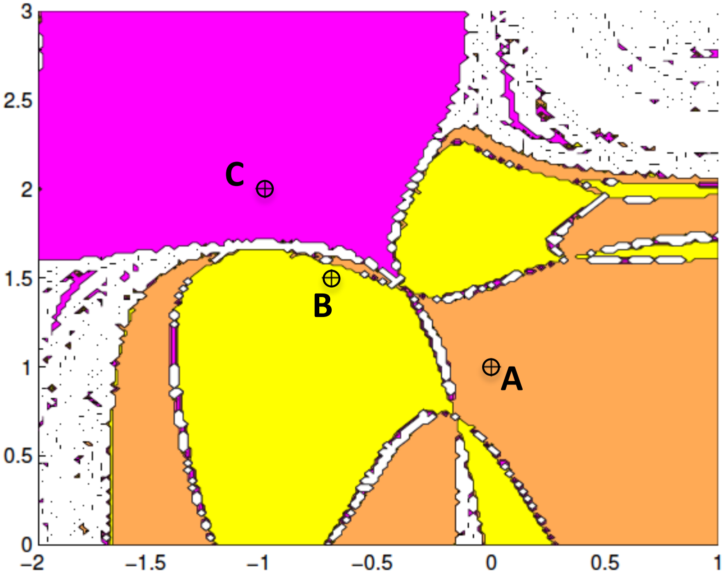
With $u_1 = -\sqrt{y_1},$ converges to B

With $\left\{\begin{aligned} u_1 &= -\sqrt{y_1}, \\ u_4 &= \frac{2(2\pi - \arccos y_4)}{\pi} \end{aligned}\right\}$ converges to C

With $\left\{\begin{aligned} u_1 &= \sqrt{y_1} \\ u_4 &= \frac{2(2\pi - \arccos y_4)}{\pi} \end{aligned}\right\}$ yields complex solution

Factored method: new variables

$$\begin{aligned} y_1 &= x_1^2, & u_1 &= \sqrt{y_1}, \\ y_2 &= x_2, & u_2 &= y_2, \\ y_3 &= x_1, & u_3 &= y_3, \\ y_4 &= \cos\left(\frac{\pi x_2}{2}\right), & u_4 &= \frac{2(\arccos y_4)}{\pi}, \end{aligned}$$



© Gómez-Expósito

Infeasibility and complex solutions

- **Feasible case**: ***p*** is such that there exists at least a **real x** satisfying ***h(x)=p***
- **Infeasible case**: only **complex values** of ***x*** satisfy ***h(x)=p***

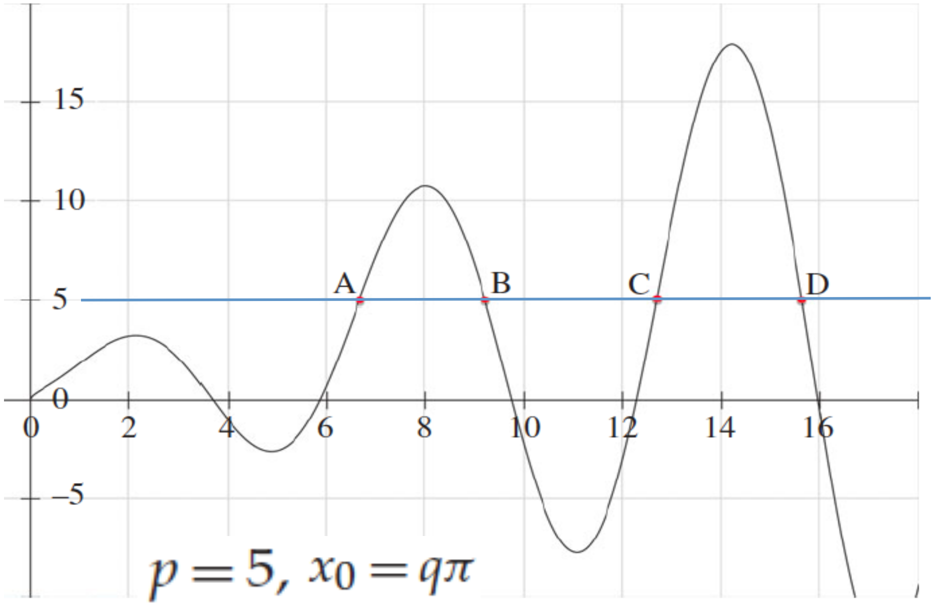
Empirical evidence:

- Newton's method cannot generally converge to complex solutions
- The **factored method** is flexible enough to converge to complex solutions whenever real solutions cannot be found
- Choosing an **initial guess** with a certain **imaginary component** is helpful to achieve convergence to complex solutions
- Quadratic convergence rate also to complex solutions

Infeasibility and complex solutions

Example: $p = x\sin x + \sqrt{x},$

Canonical form: $\left\{\begin{aligned} p &= x_1 x_2 + \sqrt{x_1} \\ 0 &= x_2 - \sin x_1 . \end{aligned}\right.$



$p = 5,\ x_0 = q\pi$

q	iter.	x
0	10	2.1519 + 0.5820i
1	8	2.2158 + 1.0097i
2	5	6.6554 (A)
3	5	9.2097 (B)
4	5	12.6801 (C)
5	4	15.6411 (D)

$$y_1 = x_1 x_2, \qquad u_1 = \ln y_1,$$
$$y_2 = \sqrt{x_1}, \qquad u_2 = \ln y_2,$$
$$y_3 = x_2, \qquad u_3 = \ln y_3,$$
$$y_4 = \sin x_1, \qquad \boxed{u_4 = \ln\left[q\pi + (-1)^q \arcsin y_4\right]}$$

$$\alpha_i = \ln x_i, \quad i = 1, 2.$$

$$\begin{pmatrix} p \\ 0 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 1 \\ \frac{1}{2} & 0 \\ 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \alpha_2 \end{pmatrix} = \begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{pmatrix},$$

Factored Load Flow Solution

$$h(x) = p$$

p: Specified quantities (PQ and PV buses)

x: State variables **(2*N*-1)**

Factored model:

$$\boxed{\begin{array}{rcl} Ey & = & p \\ u & = & f(y) \\ Cx & = & u \end{array}}$$

$$y = \{U_i, K_{ij}, L_{ij}\} \left\{ \begin{array}{l} U_i = V_i^2 \\ K_{ij} \quad = \quad V_i V_j \cos\theta_{ij} \\ L_{ij} \quad = \quad V_i V_j \sin\theta_{ij} \end{array} \right.$$

(*N*+2*b*)

$$u = \{\alpha_i, \alpha_{ij}, \theta_{ij}\} \left\{ \begin{array}{l} \alpha_i = \ln V_i^2 \\ \alpha_{ij} = \alpha_i + \alpha_j \\ \theta_{ij} = \theta_i - \theta_j\,. \end{array} \right.$$

$$x = [\alpha_1, \alpha_2, \ldots, \alpha_N \,|\, \theta_1, \theta_2, \ldots, \theta_{N-1}]^T$$

Factored Load Flow Solution

Linear (underdetermined) power flow model:

Power injections

$$P_i^{sp} = \sum P_{ij}$$

$$Q_i^{sp} = \sum Q_{ij}$$

with

$$P_{ij} = (g_{sh,i} + g_{ij})U_i - g_{ij}K_{ij} - b_{ij}L_{ij}$$

$$Q_{ij} = -(b_{sh,i} + b_{ij})U_i + b_{ij}K_{ij} - g_{ij}L_{ij}$$

- Voltage magnitudes

$$[V_i^2]^{sp} = U_i$$

$$\boxed{Ey = p}$$

Factored Load Flow Solution

The Jacobian and its inverse are trivially obtained

Jacobian elements:

$$\left\{ \begin{array}{l} \dfrac{\partial \alpha_i}{\partial U_i} = \dfrac{1}{U_i} \\ \begin{bmatrix} \frac{\partial \alpha_{ij}}{\partial K_{ij}} & \frac{\partial \alpha_{ij}}{\partial L_{ij}} \\ \frac{\partial \theta_{ij}}{\partial K_{ij}} & \frac{\partial \theta_{ij}}{\partial L_{ij}} \end{bmatrix} = \dfrac{1}{K_{ij}^2 + L_{ij}^2} \begin{bmatrix} 2K_{ij} & 2L_{ij} \\ -L_{ij} & K_{ij} \end{bmatrix} \end{array} \right.$$

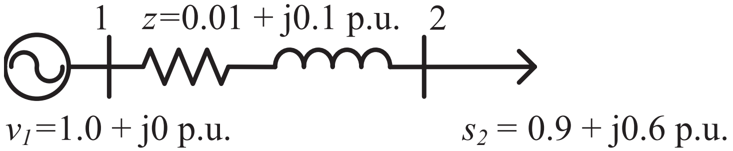
Inverted Jacobian elements:

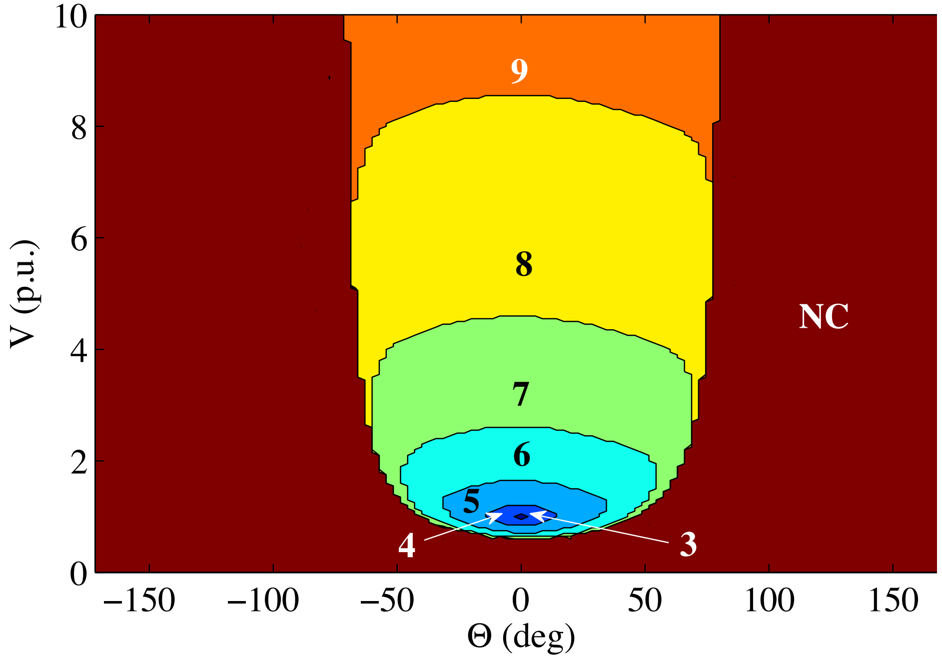
$$\left. \begin{array}{l} \left[\dfrac{\partial \alpha_i}{\partial U_i}\right]^{-1} = U_i \\ \begin{bmatrix} \frac{\partial \alpha_{ij}}{\partial K_{ij}} & \frac{\partial \alpha_{ij}}{\partial L_{ij}} \\ \frac{\partial \theta_{ij}}{\partial K_{ij}} & \frac{\partial \theta_{ij}}{\partial L_{ij}} \end{bmatrix}^{-1} = \begin{bmatrix} \frac{K_{ij}}{2} & -L_{ij} \\ \frac{L_{ij}}{2} & K_{ij} \end{bmatrix}. \end{array} \right\} F^{-1} = \begin{bmatrix} U_1 & & & & & & & & \\ & U_2 & & & & & & & \\ & & U_3 & & & & & & \\ & & & \frac{1}{2}K_{12} & -L_{12} & & & & \\ & & & \frac{1}{2}L_{12} & K_{12} & & & & \\ & & & & & \frac{1}{2}K_{13} & -L_{13} & & \\ & & & & & \frac{1}{2}L_{13} & K_{13} & & \\ & & & & & & & \frac{1}{2}K_{23} & -L_{23} \\ & & & & & & & \frac{1}{2}L_{23} & K_{23} \end{bmatrix}$$

F^{-1} becomes singular iff $U_i = 0$ for any i

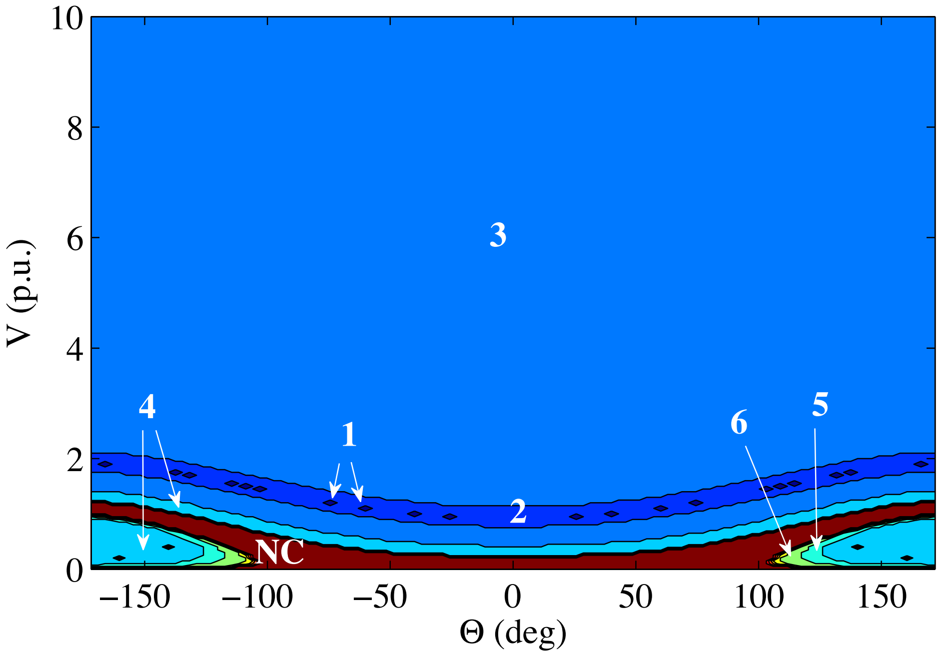
Factored Load Flow Solution

Convergence basins: 2-bus example





Newton-Raphson

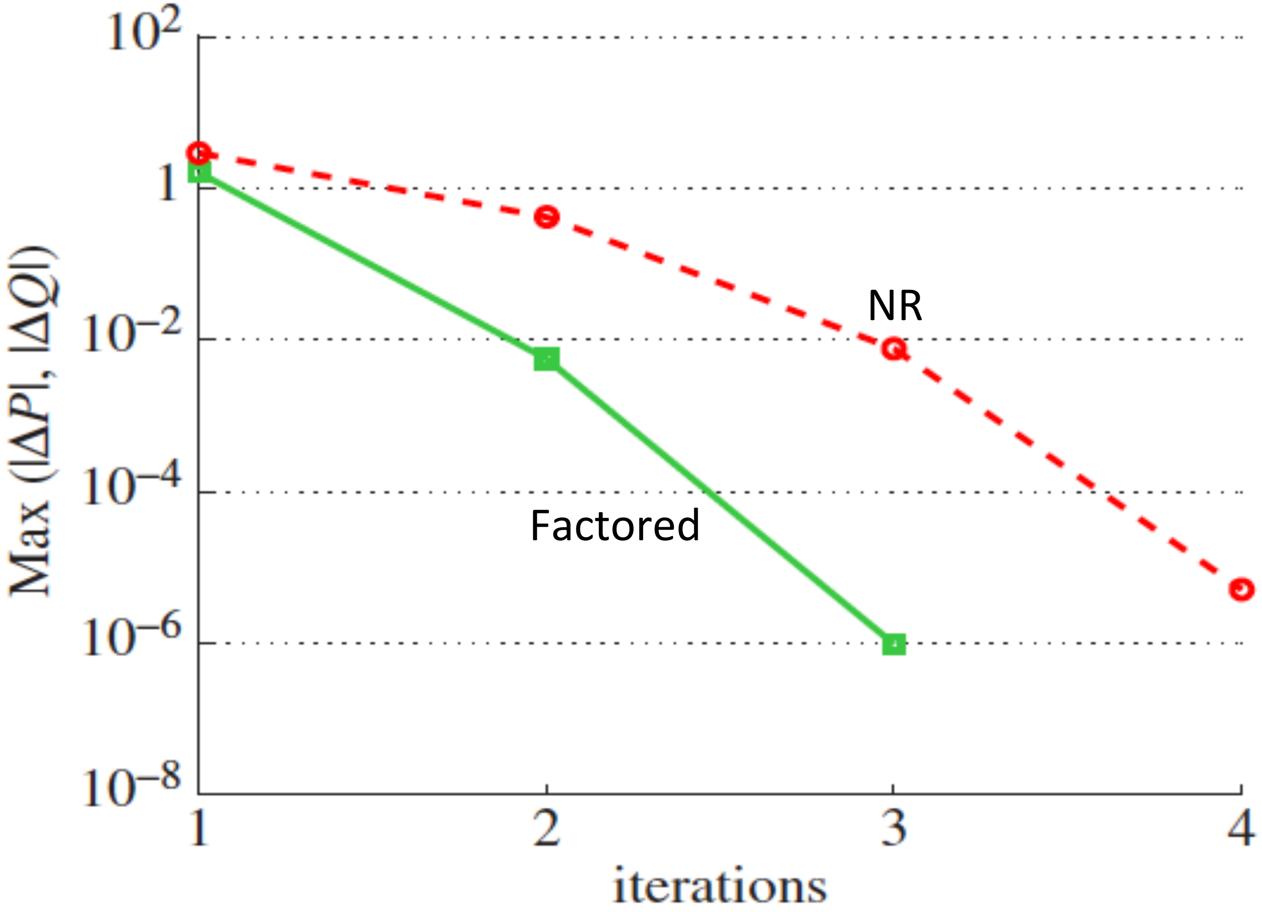


Factored method

© Gómez-Expósito

Factored Load Flow Solution

Convergence rates for the IEEE 300-bus system

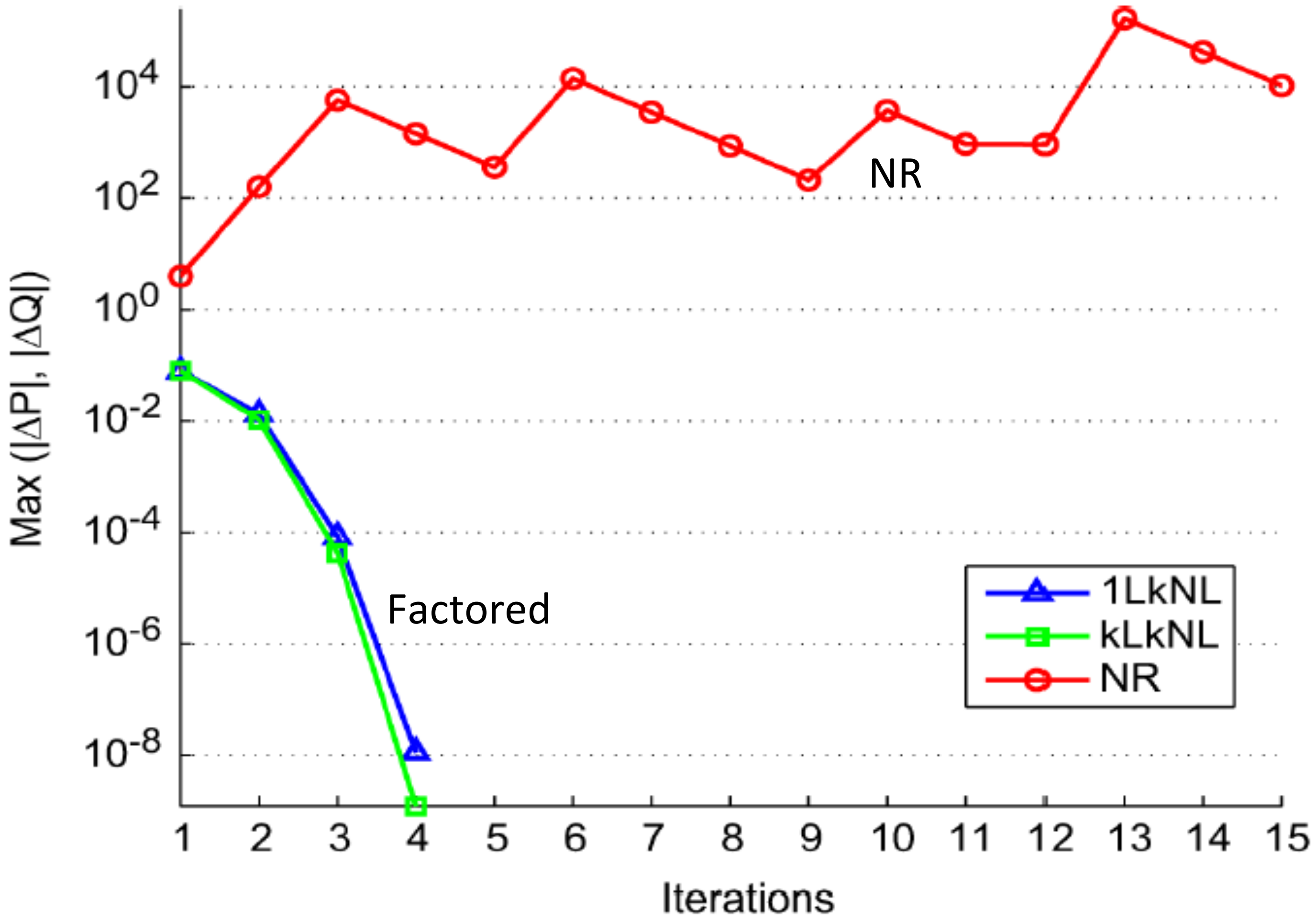


© Gómez-Expósito

Factored Load Flow

Convergence rates for the IEEE 118-bus system

[PV buses converted to PQ buses]

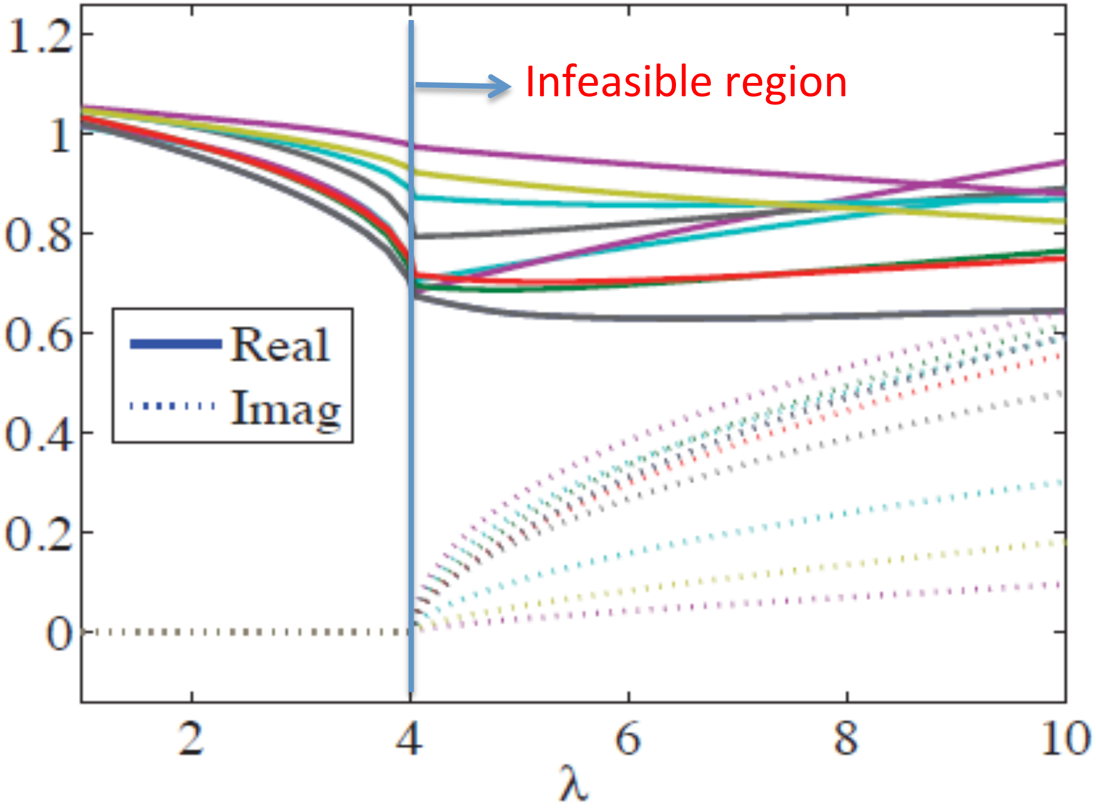


Factored Load Flow Solution

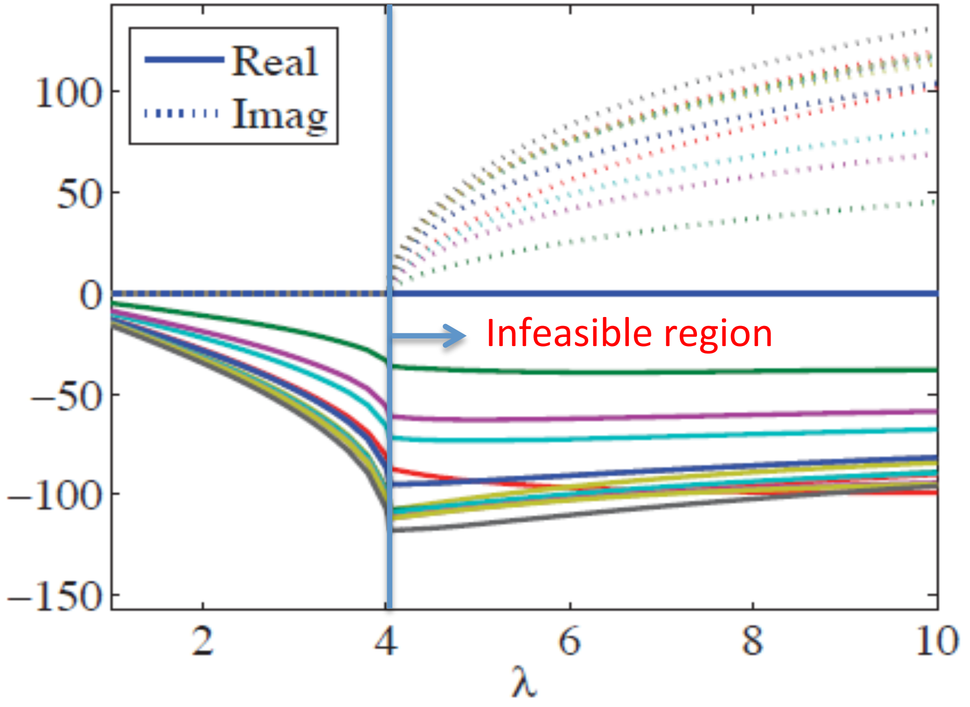
IEEE 14-bus system: evolution of voltages for increasing loadings

$$U_0 = K_0 = 0.5+0.5j$$

Voltage magnitudes (p.u.)

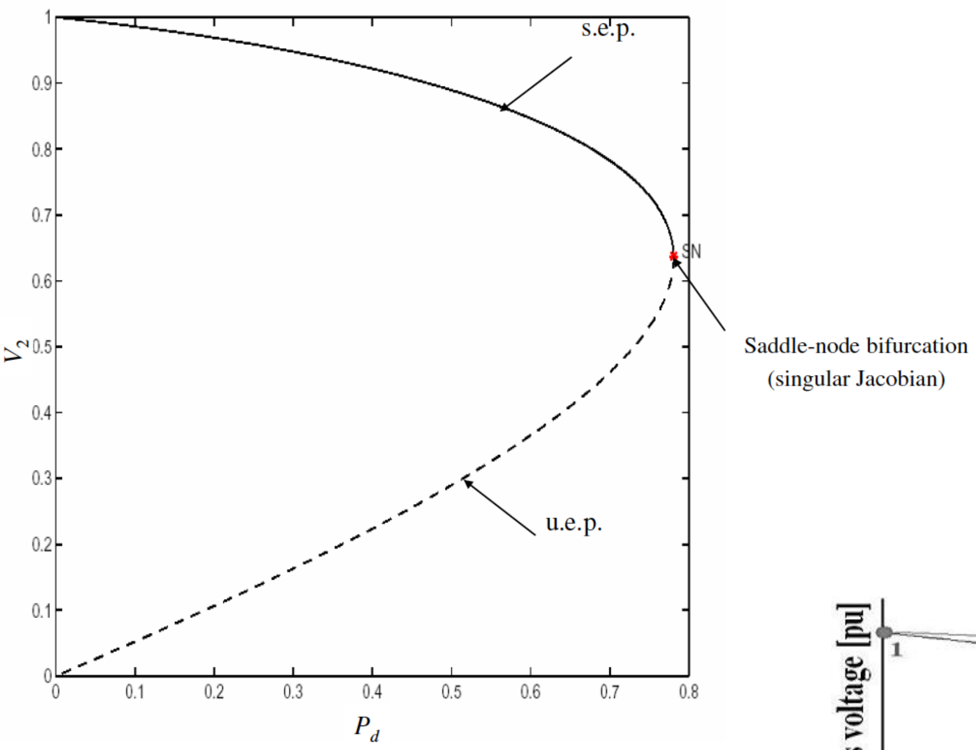


Voltage phase angle (deg.)



Maximum loading point determination

Problem: **Finding the saddle-node or limit-induced bifurcation point**

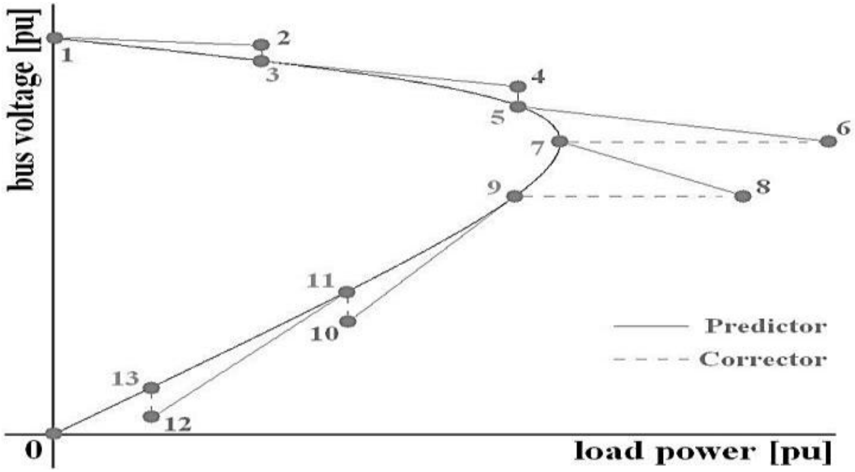


Maximize the value of λ:

$$h(x) = p(\lambda)$$

$$p(\lambda) = \left[\lambda \begin{pmatrix} P_{LG} \\ Q_L \\ U_G \end{pmatrix}_0 \right]$$

- Optimization problem
- Continuation methods

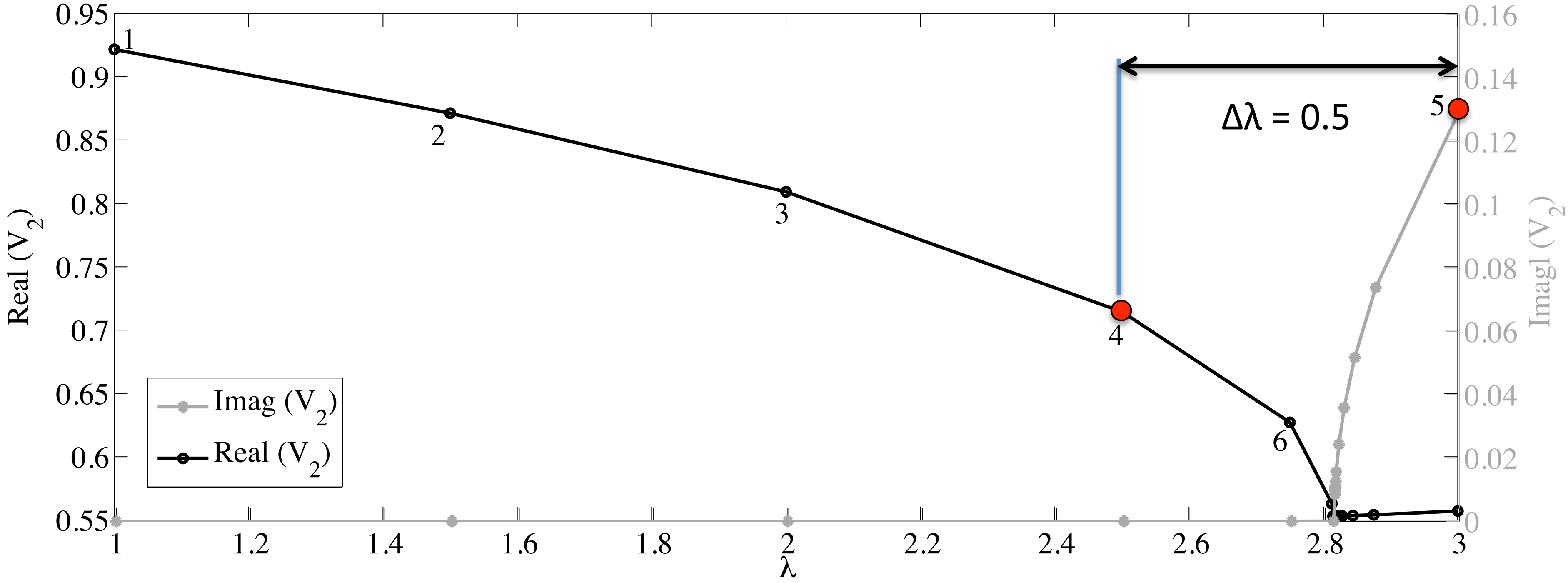


© Gómez-Expósito

Maximum loading point determination

Factored load flow-based procedure:

Phase 1: Increase regularly λ until two consecutive solutions correspond to feasible and infeasible points



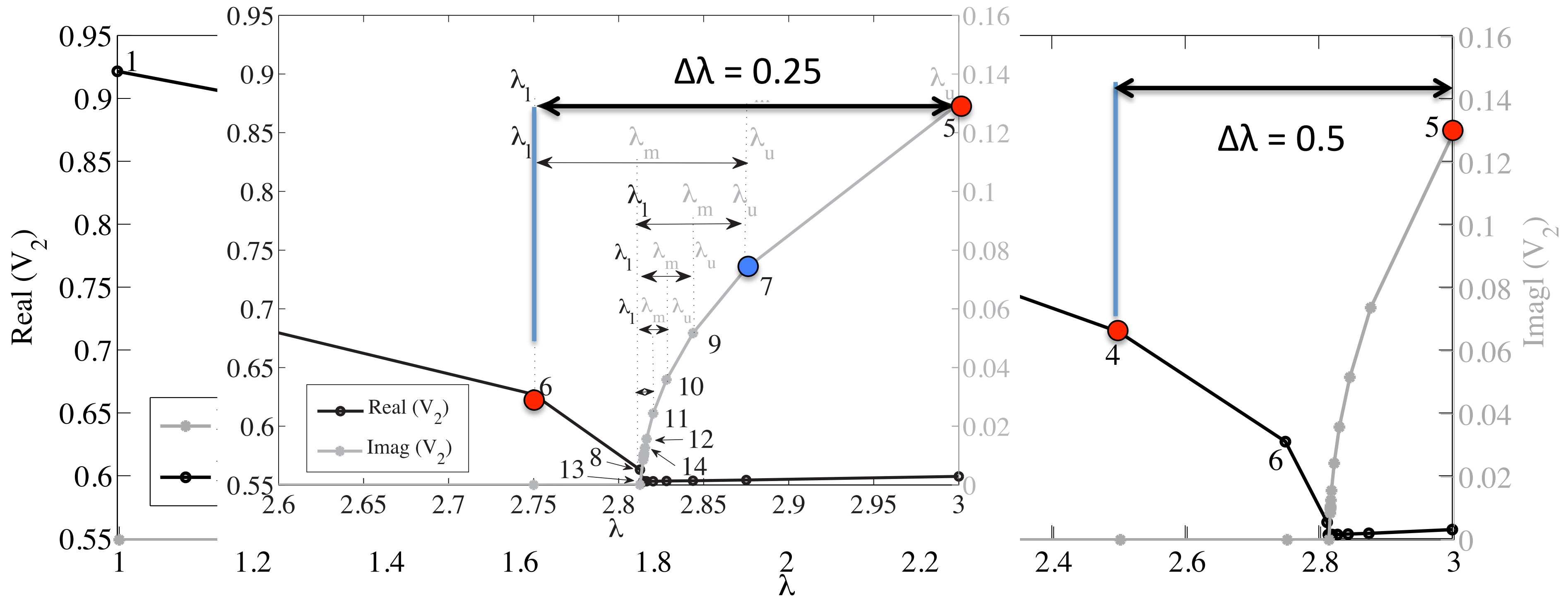
© Gómez-Expósito

Maximum loading point determination

Factored load flow-based procedure:

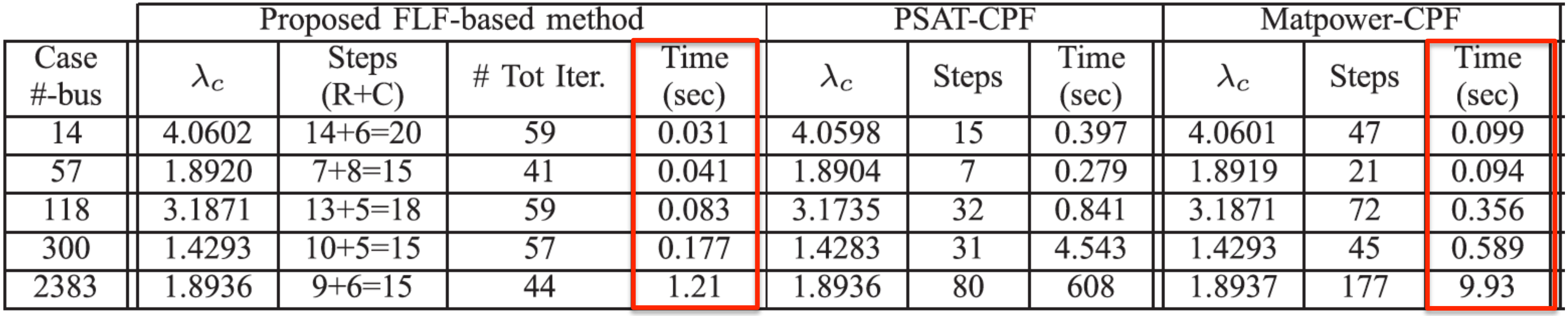
Phase 1: Increase regularly λ until two consecutive solutions correspond to feasible and infeasible points

Phase 2: Sequentially perform bisection searches until Δλ is small



Maximum loading point determination

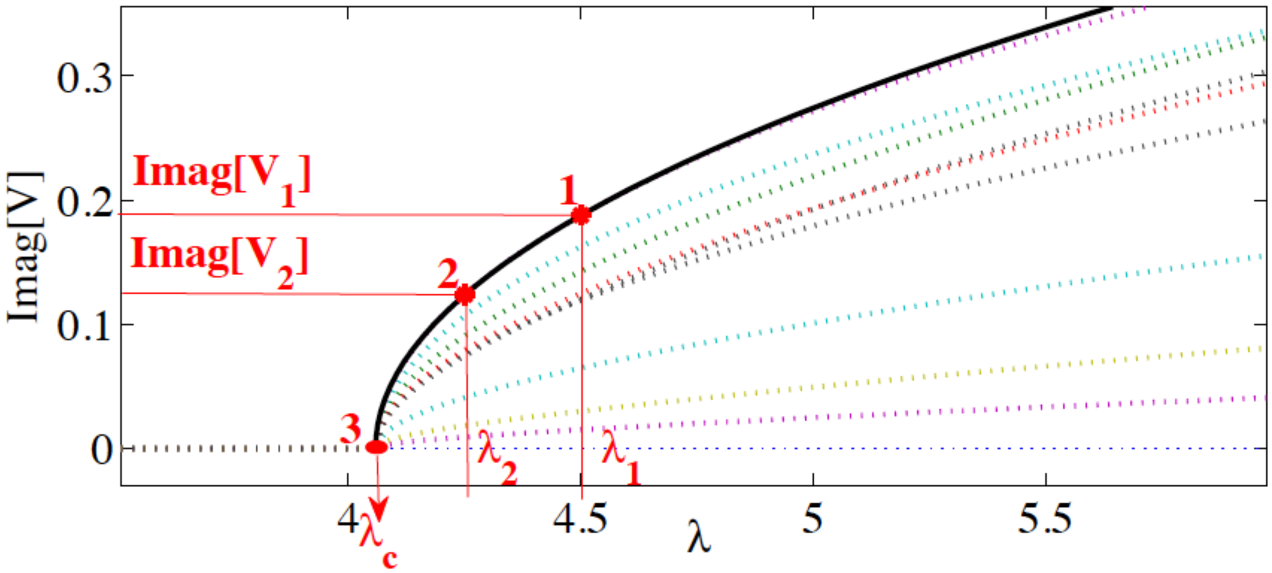
Comparison of simulation results



	Proposed FLF-based method				PSAT-CPF			Matpower-CPF		
Case #-bus	λ_c	Steps (R+C)	# Tot Iter.	Time (sec)	λ_c	Steps	Time (sec)	λ_c	Steps	Time (sec)
14	4.0602	14+6=20	59	0.031	4.0598	15	0.397	4.0601	47	0.099
57	1.8920	7+8=15	41	0.041	1.8904	7	0.279	1.8919	21	0.094
118	3.1871	13+5=18	59	0.083	3.1735	32	0.841	3.1871	72	0.356
300	1.4293	10+5=15	57	0.177	1.4283	31	4.543	1.4293	45	0.589
2383	1.8936	9+6=15	44	1.21	1.8936	80	608	1.8937	177	9.93

Maximum loading point determination

Enhanced solution procedure: parabolic approximations in the infeasible domain



	FLF-bisection		FLF-parabolic	
#-bus	steps	λ_c	steps	λ_c
14	20	4.0604	8	4.0581
30	17	2.9572	8	2.9570
57	15	1.8931	5	1.8932
300	14	1.4299	4	1.4293
2383	15	1.9722	7	1.9720

Factored WLS State Estimation

Conventional solution of nonlinear WLS-SE

Measurement model:

$$\boxed{z = h(x) + e}$$

Maximum likelihood estimation:

Min J

$$\boxed{J = r^T W r = \sum_{i=1}^{m} W_i r_i^2}$$

$$r = z - h(\hat{x})$$
$$W = R^{-1}$$

Normal equations (Gauss-Newton):

$$\boxed{G_k \Delta x_k = H_k^T W [z - h(x_k)]}$$

$$H_k = \partial h / \partial x$$
$$G_k = H_k^T W H_k$$
$$\Delta x_k = x_{k+1} - x_k$$

Covariance (dense) matrices:

$$\boxed{\mathrm{cov}(\hat{x}) = G_k^{-1}}$$

$$\mathbf{cov}(\hat{z}) = H \cdot \mathbf{cov}(\hat{x}) \cdot H^T$$

Factored WLS State Estimation

Factorization of nonlinear WLS problems

Linear WLS SE

Non-linear explicit transformation

Linear WLS SE

Factored WLS State Estimation

Factorization of nonlinear WLS problems

Linear WLS SE

↓

Non-linear explicit transformation

↓

Linear WLS SE

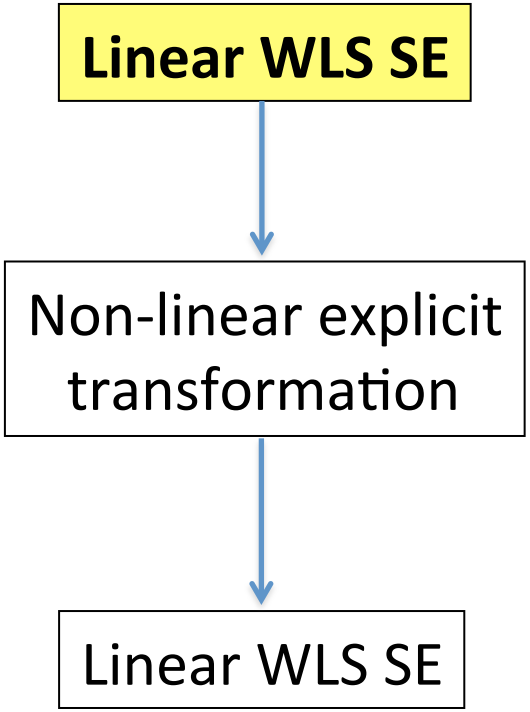
$$z = By + e \quad \longrightarrow \quad \tilde{y}\ ,\ \ \mathrm{cov}^{-1}(\tilde{y}) = G_B = B^T W B$$

$$y = \{U_i, K_{ij}, L_{ij}\} \begin{cases} \text{Branch} \begin{cases} K_{ij} = V_i V_j \cos\theta_{ij} \\ L_{ij} = V_i V_j \sin\theta_{ij} \end{cases} \\ \text{Bus} \quad U_i = V_i^2 \end{cases}$$

(2b+N variables)

Factored WLS State Estimation

Factorization of nonlinear WLS problems



Linear WLS SE

Non-linear explicit transformation

Linear WLS SE

$$z = By + e \quad \longrightarrow \quad \tilde{y}\ ,\ \text{cov}^{-1}(\tilde{y}) = G_B = B^T W B$$

Linear measurement model:

$$
\begin{aligned}
P^m{}_{ij} &= (g_{sh,i} + g_{ij})U_i - g_{ij}K_{ij} - b_{ij}L_{ij} + \varepsilon_p \\
Q^m{}_{ij} &= -(b_{sh,i} + b_{ij})U_i + b_{ij}K_{ij} - g_{ij}L_{ij} + \varepsilon_q \\
P_i^m &= \sum P_{ij} + \varepsilon_{pI} \\
Q_i^m &= \sum Q_{ij} + \varepsilon_{qI} \\
[V_i^2]^m &= U_i + \varepsilon_V
\end{aligned}
$$

- No need to choose initial values
- **"Warm start"** for repeated runs (constant B matrix)

Factored WLS State Estimation

Factorization of nonlinear WLS problems

Linear WLS SE

↓

Non-linear explicit transformation

↓

Linear WLS SE

$$z = By + e \quad \longrightarrow \quad \tilde{y}\ ,\ \mathrm{cov}^{-1}(\tilde{y}) = G_B = B^T W B$$

$$\underbrace{\tilde{u}}_{\left\{\begin{array}{l} \alpha_i = \ln U_i \\ \alpha_{ij} = \ln\left(K_{ij}{}^2 + L_{ij}{}^2\right) \\ \theta_{ij} = \arctan\left(L_{ij} / K_{ij}\right) \end{array}\right.} = f_u(\tilde{y}) \quad \longrightarrow \quad \tilde{u}\ ,\ \mathrm{cov}^{-1}(\tilde{u}) = \tilde{W}_u = \underbrace{\tilde{F}_u^{-T}}_{\boxed{\text{trivial inverse}}} G_B \tilde{F}_u^{-1}$$

Factored WLS State Estimation

Factorization of nonlinear WLS problems

Linear WLS SE

$$z = By + e \quad \rightarrow \quad \tilde{y}\ ,\ \mathrm{cov}^{-1}(\tilde{y}) = G_B = B^T W B$$

↓

Non-linear explicit transformation

$$\tilde{u} = f_u(\tilde{y}) \quad \rightarrow \quad \tilde{u}\ ,\ \mathrm{cov}^{-1}(\tilde{u}) = \tilde{W}_u = \tilde{F}_u^{-T} G_B \tilde{F}_u^{-1}$$

↓

Linear WLS SE

$$\tilde{u} = \underbrace{Cx}_{\left\{\begin{matrix}\alpha_i \\ \theta_i\end{matrix}\right.} + e_u \quad \rightarrow \quad \hat{x}\ ,\ \mathrm{cov}^{-1}(\hat{x}) = G_C = C^T \tilde{W}_u C$$

Factored WLS State Estimation

Factorization of nonlinear WLS problems

Linear WLS SE

↓

Non-linear explicit transformation

↓

Linear WLS SE

Is it a non-iterative procedure?

Not really !!

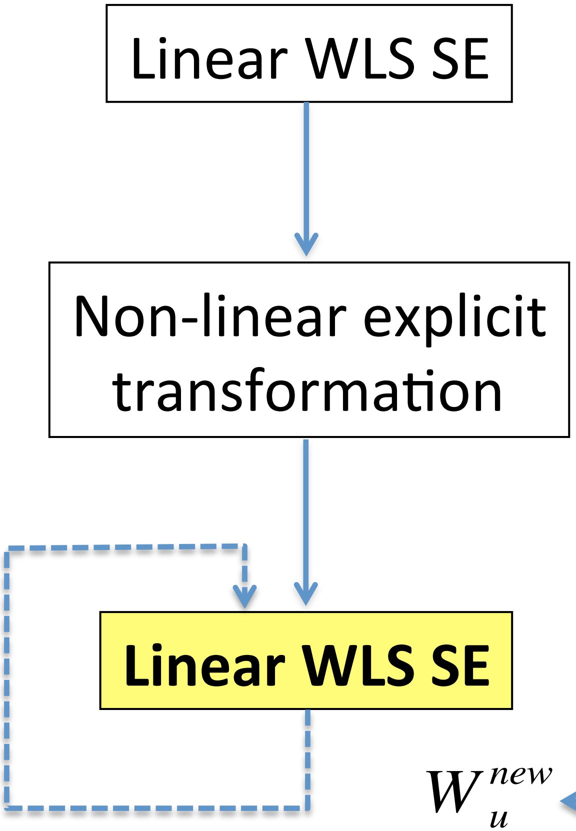
$$\tilde{W}_u = \tilde{F}_u^{-T} G_B \tilde{F}_u^{-1}$$

depends on the operating point: $\tilde{u}$

$$\tilde{u} = Cx + e_u \quad \longrightarrow \quad \hat{x} \ , \ \mathrm{cov}^{-1}(\hat{x}) = G_C = C^T \tilde{W}_u C$$

Factored WLS State Estimation

Factorization of nonlinear WLS problems



Linear WLS SE

$$z = By + e \quad \rightarrow \quad \tilde{y}\ ,\ \mathrm{cov}^{-1}(\tilde{y}) = G_B = B^T W B$$

Non-linear explicit transformation

$$\tilde{u} = f_u(\tilde{y}) \quad \rightarrow \quad \tilde{u}\ ,\ \mathrm{cov}^{-1}(\tilde{u}) = \tilde{W}_u = \tilde{F}_u^{-T} G_B \tilde{F}_u^{-1}$$

Linear WLS SE

$$\tilde{u} = Cx + e_u \quad \rightarrow \quad \hat{x}\ ,\ \mathrm{cov}^{-1}(\hat{x}) = G_C = C^T \tilde{W}_u C$$

$$W_u^{new} \leftarrow u_{new} = C\hat{x}$$

Repeat until $\hat{x}^{k+1} \approx \hat{x}^k$

Factored WLS State Estimation

Stage 1: Linear WLS SE

$$z = By + e \quad \left\{ \begin{array}{l} y = \{ U_i, K_{ij}, L_{ij} \} \\ G_B = B^T W B \end{array} \right.$$

Stage 2: Non-linear one-to-one mapping

$$u = f_u\,(y) \left\{ \begin{array}{l} \alpha_i = \ln U_i \\ \alpha_{ij} = \ln\left(K_{ij}{}^2 + L_{ij}{}^2\right) \\ \theta_{ij} = \arctan\left(L_{ij} / K_{ij}\right) \end{array} \right\} \left\{ \begin{array}{l} u = \{\alpha_i, \alpha_{ij}, \theta_{ij}\} \\ W_u = \mathrm{cov}^{-1}(u) = (Fu)^{-T} G_B \overbrace{(Fu)^{-1}}^{\text{trivial inverse}} \end{array} \right.$$

Stage 3: Linear WLS SE

$$u = Cx + e_u \quad \left\{ \begin{array}{l} x = \{V_i, \theta_i\} \\ G_C = C^T W_u C \end{array} \right.$$

Factored WLS State Estimation

Tests

Benchmark networks: 118-, 298-, 2948- bus
1000 simulations (to obtain pdf’s)
2 redundancy levels

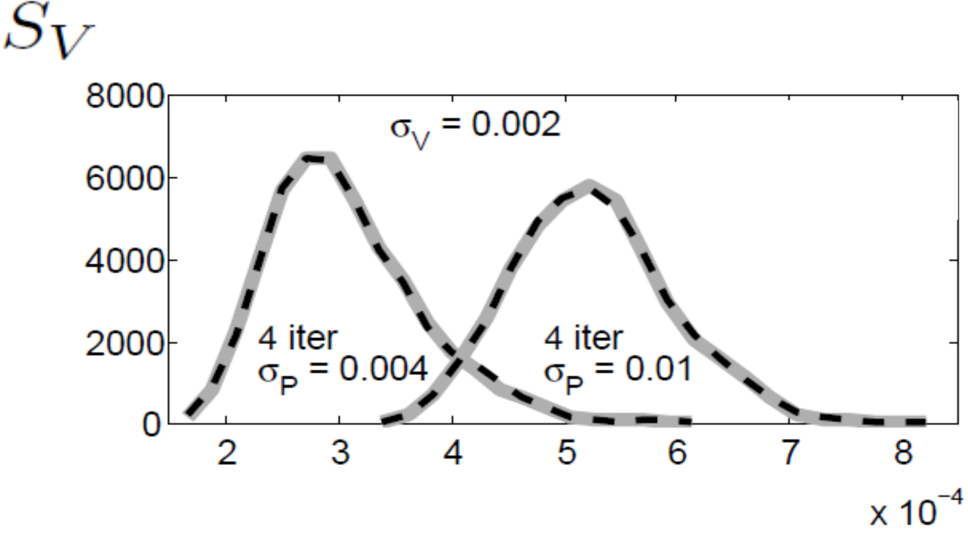
Quality Indices

$$S_V = \frac{1}{N} \sum_{i=1}^{N} |\hat{V}_i - V_i^{\mathrm{ex}}| \qquad S_\theta = \frac{1}{N-1} \sum_{i=1}^{N-1} |\hat{\theta}_i - \theta_i^{\mathrm{ex}}|$$

- Accuracy (with a single run)
- Computation time and speed-ups

Factored WLS State Estimation

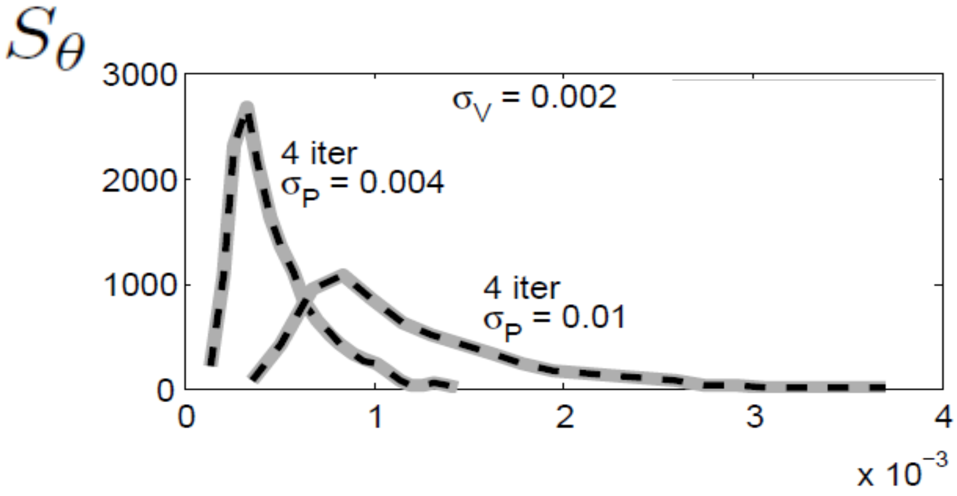
Accuracy (single run)



Voltage errors

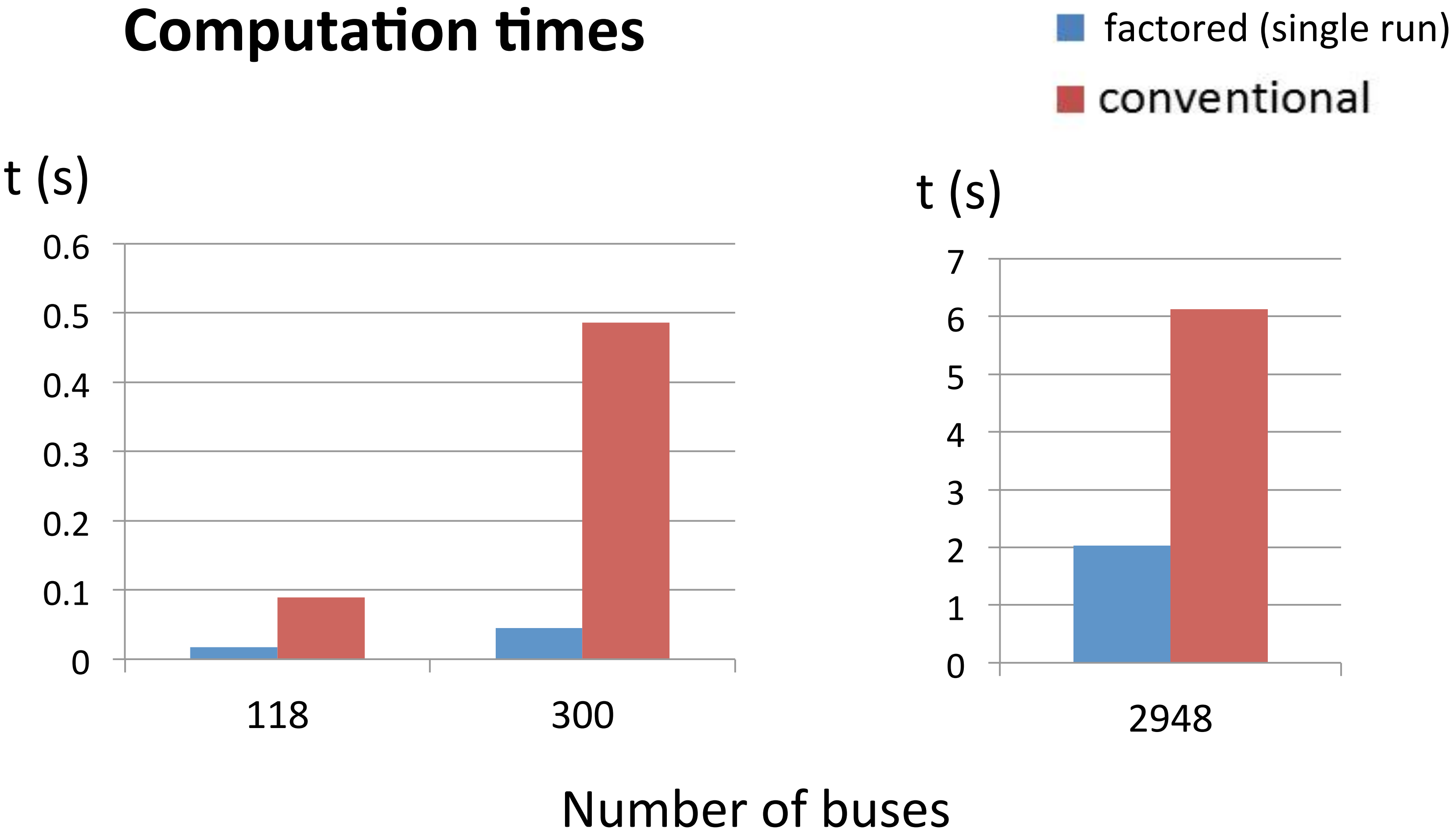
▬ Conv.

- - - Factored (single run)



Phase angle errors

Factored WLS State Estimation



© Gómez-Expósito

Factored WLS State Estimation

Advantages

- Enhanced **convergence rate**

  First (linear) step always gives an estimate

- Early detection of **bad data or topology error**
- Less **computational effort** (constant or simpler Jacobians)

  Approx. 3 times speedup (for accurate enough measurements)

- More advantageous with **accurate measurements**
- More advantageous under **peak loading**

On-going research efforts

Model enhancement & other promising applications

- Incorporation of regulating devices
- Distribution networks (b=N-1)
- Optimal Power Flows
- Nonlinear control
- Eigensystem analysis
- Nonlinear algebraic-differential equations

References

- A. Gómez-Expósito; A. Abur; A. de la Villa Jaén; C. Gómez-Quiles, “A Multilevel State Estimation Paradigm for Smart Grids”, **Proc. of the IEEE**, vol. 99, no. 6, pp. 952-976, June 2011.
- C. Gómez-Quiles, A. de la Villa Jaén, A. Gómez-Expósito, “A Factorized Approach to WLS State Estimation”, **IEEE Transactions on Power Systems**, vol. 26, no. 3, pp. 1724-1732, Aug. 2011.
- A. Gómez-Expósito, C. Gómez-Quiles, A. de la Villa Jaén, “Bilinear Power System State Estimation”, **IEEE Transactions on Power Systems**, vol. 27(1), pp. 493-501, Feb. 2012.
- C. Gómez-Quiles, H. Gil, A. de la Villa Jaén, A. Gómez-Expósito, “Equality-Constrained Bilinear State Estimation”, **IEEE Transactions on Power Systems**, vol. 28 (2), pp. 902-910, May 2013.
- A. Gómez-Expósito, C. Gómez-Quiles, “Factorized Load Flow”, **IEEE Transactions on Power Systems**, vol. 28(4), pp. 4607-4614, November 2013.
- A. Gómez-Expósito, “Factored solution of nonlinear equation systems”, **Proceedings of the Royal Society – A**, 2014 470, 20140236, July 2014.
- A. Gómez-Expósito, C. Gómez-Quiles, W. Vargas, “Factored Solution of Infeasible Load Flow Cases”, *18-th Power Systems Computation Conference*, Wroclaw, August 2014.
- C. Gómez-Quiles, A. Gómez-Expósito, W. Vargas, “Computation of Maximum Loading Points via the Factored Load Flow”, **IEEE Transactions on Power Systems**, vol. 31(5), pp. 4128-4134, September 2016.
- C. Gómez-Quiles, A. Gómez-Expósito “Fast Determination of Saddle-Node Bifurcations via Parabolic Approximations in the Infeasible Region”, IEEE Transactions on Power Systems, in press, 2017.

Thank you !