Distributed State Estimation Algorithms for Electric Power Systems

Ariana Minot, Blue Waters Graduate Fellow
Professor Na Li, Professor Yue M. Lu
Harvard University, School of Engineering and Applied Sciences

Blue Waters Symposium 2015

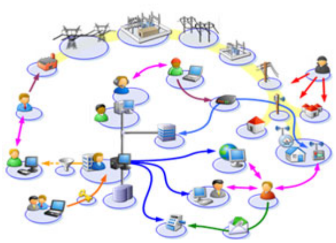
Outline

- Motivation
- Problem Statement
- Method
- Results
- Conclusions

Outline

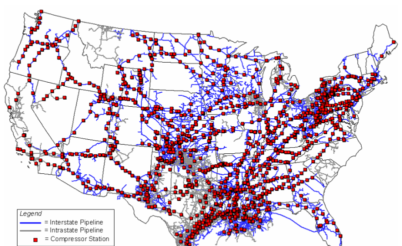
- Motivation
- Problem Statement
- Method
- Results
- Conclusions

Motivation: why we need to improve the electric power grid.



Source: http://www.ornl.gov

Communication Networks



Source: http://www.eia.gov

Fuel supply

Source: mapoftheweek.blogspot.com

Source: ROBERT GIROUX/ Time

Source: http://www.dnvkema.com

- Support for critical infrastructure:
  - Communication networks and fuel and water supply systems
- Prevent blackouts
- Realize a robust efficient grid under increased renewable energy production

Scientific Motivation: better understand large, complex interconnected systems.

Scientific Computing

- Distributed Algorithms
- Scalable with respect to size of :
  - Network
  - Data
- Asynchronous Setting
- Simulation of complex, dynamic systems

Challenge: coordinate control, sensing, and optimization in a decentralized way.

Estimation & Detection

- State Estimation
- Bad Data Detection
- Decision Making Under Uncertainty
- Line Outage Identification

Optimization & Control

- Optimal Power Flow
- Economic Dispatch
- Demand Response
- Wide – Area Control

Improving power grid operations via High Performance Computing

Power Grid Control Room (California)

Source: http://www.caiso.com

Blue Waters Supercomputer (NCSA & UIUC)

Source: http://engineering.illinois.edu

Portion of Power Grids	Midwest	Poland	Europe
# vertices N	118	3375	9241
# edges	186	4161	16049
# generators	54	596	1445

- Efficient matrix inversion for real-time applications
- New sensing technology (PMUs) produces a lot of data
    - 30 samples/sec -> 2 million samples/day
    - For ~100 sensors, ~$2*10^8$ samples/day

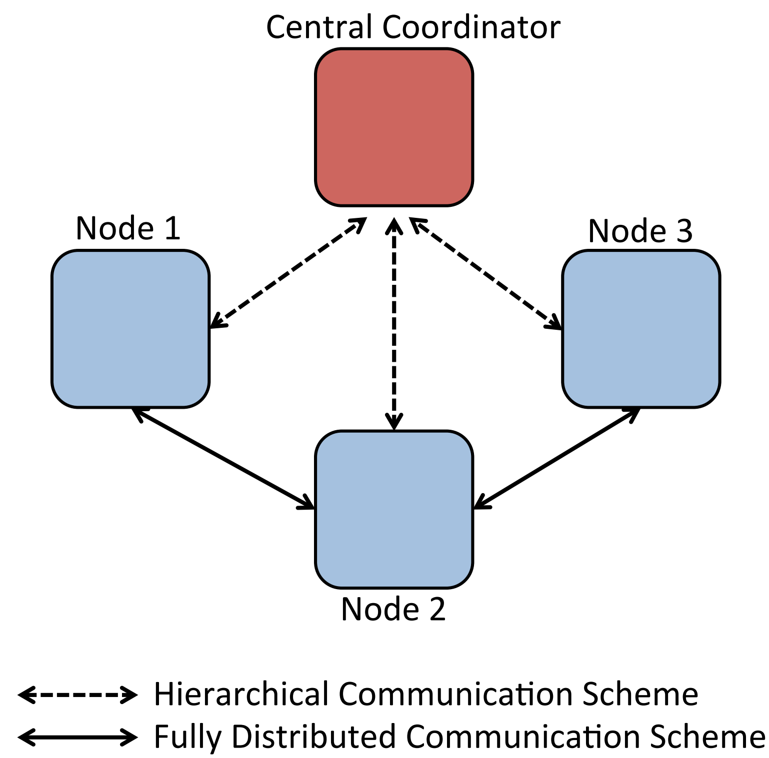
Source: Franz Franchetti, CMU

Outline

- Motivation
- Problem Statement
- Method
- Results
- Conclusions

State Estimation & Distributed Algorithms

- Monitor the electric grid in real-time
- State Estimation task: Given noisy system measurements, infer the state of the system.
    - Reference : [Schweppe 1970]
    - Measurements consist of subset of:
        - Power flows along transmission lines (edges).
        - Power injections and voltage at vertices.
    - State is the voltage (phase angle and magnitude) at each vertex in the network.
    - Non-linear measurement model
- Distributed vs. centralized
- Advantages of distributed approaches:
    - 1) avoid communication bottleneck
    - 2) reduction in computation and memory requirements per area.



Mathematical Problem Statement

- Goal: From noisy system measurements z, determine state, $\boldsymbol{x} = [\boldsymbol{\theta}\ \boldsymbol{V}]$, at every vertex in the power network.

Weighted non-linear least squares optimization

$$\min_{\boldsymbol{x}} f(\boldsymbol{x}) \equiv (\boldsymbol{z} - \boldsymbol{h}(\boldsymbol{x}))^T \boldsymbol{W} (\boldsymbol{z} - \boldsymbol{h}(\boldsymbol{x}))$$

Iterative solution using Newton's method

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} - [\nabla^2 f(\boldsymbol{x}^{(k)})]^{-1} \nabla f(\boldsymbol{x}^{(k)})$$

- Main challenge : Matrix inversion requires full, global knowledge of matrix entries. How to calculate inverse in a fully distributed fashion?
- Idea : use matrix splitting techniques (R. Varga)
- In general, matrix splitting $\boldsymbol{A} = \boldsymbol{M} + \boldsymbol{N}$ provides an iterative approach to solve the linear system, $\mathbf{Ax} = \mathbf{y}$

$$\mathbf{x^{t+1}} = -\mathbf{M^{-1}Nx^{t}} + \mathbf{M^{-1}y}$$

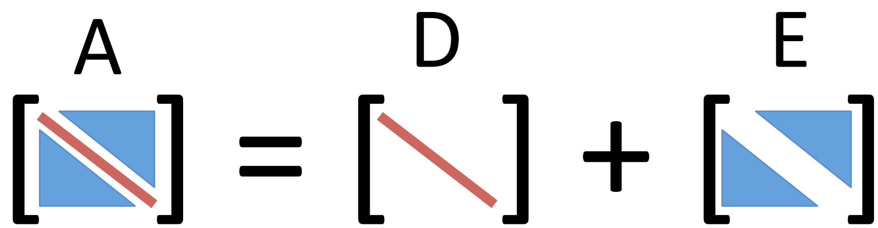
- The sequence $\{\mathbf{x^t}\}$ converges to its limit $\mathbf{x}^*$ as $t \to \infty$ if and only if the spectral radius of the matrix $\mathbf{M^{-1}N}$ is strictly less than 1.

Outline

- Motivation
- Problem Statement
- Method
- Results
- Conclusions

Matrix-Splitting for Distributed State Estimation

- Let $\boldsymbol{A} \equiv \nabla^2 f(\boldsymbol{x}^{(k)}) \succ 0$ and $\bar{E}_{ii} \equiv \alpha \sum_{j=1, j\neq i}^{n} |A_{ij}|$

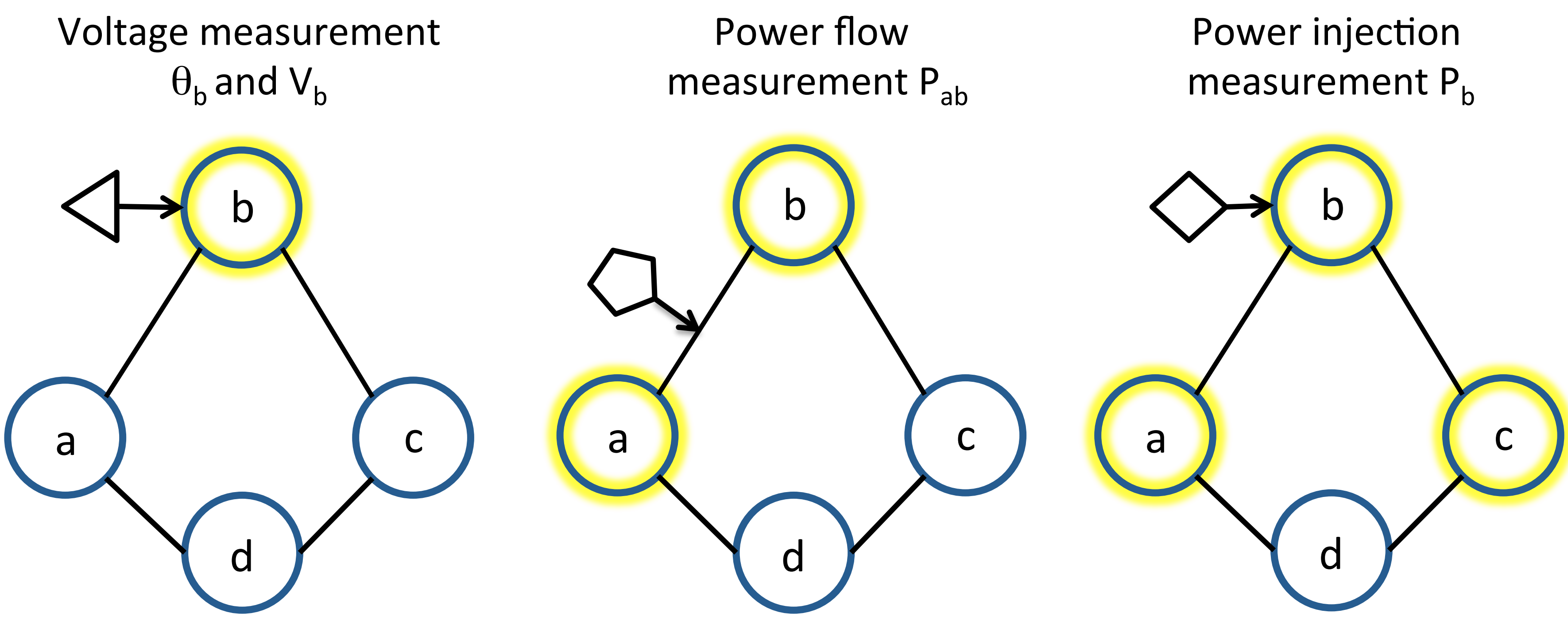


$$= \underbrace{(\mathrm{D} + \bar{\mathrm{E}})}_{\mathrm{M}} + \underbrace{(\mathrm{E} - \bar{\mathrm{E}})}_{\mathrm{N}}$$

Proposition: Let $\mathbf{M} = \mathbf{D} + \bar{\mathbf{E}}$ and $\mathbf{N} = \mathbf{E} - \bar{\mathbf{E}}$.
Then, for $\alpha \geq \frac{1}{2}$, $\rho(\mathbf{M}^{-1}\mathbf{N}) < \mathbf{1}$.

Utilizing Sparsity to Enable Distribution

- **M** is diagonal and easy to invert distributedly. Each vertex needs only its local information.
- Power systems follow Kirchoff's current & voltage laws ➔ **N** has a distinct sparsity pattern related to the power network topology.

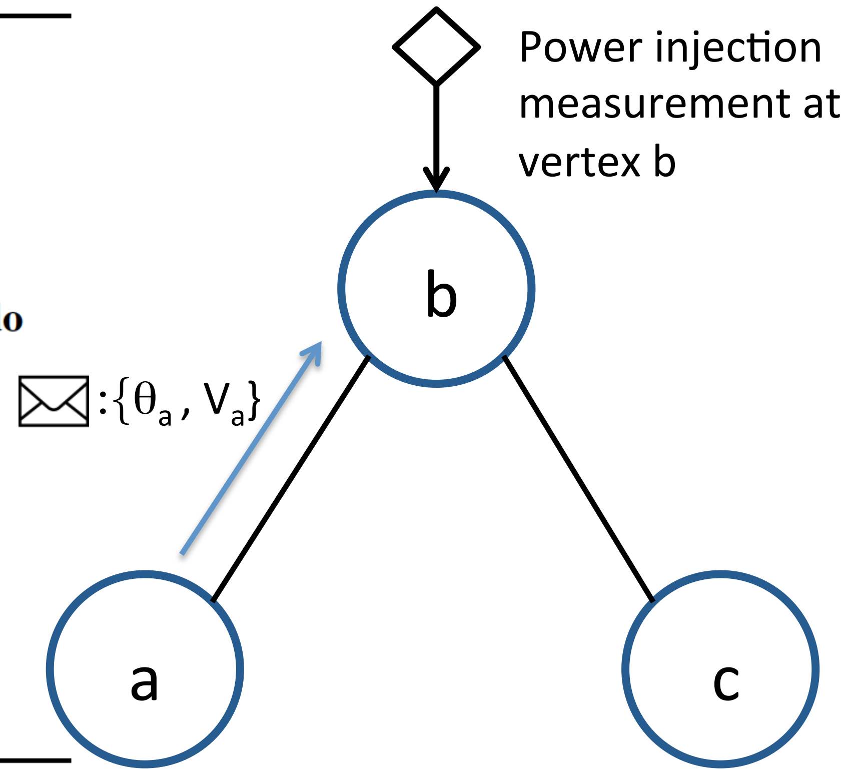


- This sparsity in **N** is what allows for distribution with limited communication.
  - At most need to communicate 2-hop neighbor information.

Distributed Newton Method for DSE

Algorithm 1: Distributed Newton Method for State Estimation

Initialization: $\forall\, 1 \le a \le n$, $\boldsymbol{x}_a^{(0)} = \{\theta_a^{(0)}, V_a^{(0)}\}$ with $\theta_a^{(0)} = 0$ and $V_a^{(0)} = 1$.
for $a := 1$ *to* n **do**
  for $k := 0$ *to* K *(Newton/Outer Loop)* **do**
    $\Delta\boldsymbol{x}_a^{(0,k)} = 0$
    for $t := 0$ *to* T *(Matrix-Splitting/Inner Loop)* **do**
      Calculate $\Delta\boldsymbol{x}_a^{(t+1,k)}$ with matrix-splitting iterative scheme.
      Communication with neighbors.
    end
    $\boldsymbol{x}_a^{(k+1)} = \boldsymbol{x}_a^{k} + \Delta\boldsymbol{x}_a^{(k,T)}$
    Communication with neighbors.
  end
end



- Each vertex is assigned an MPI process.
- MPI Graph Communicator (`MPI_Graph_create`) used to mimic structure of power network.

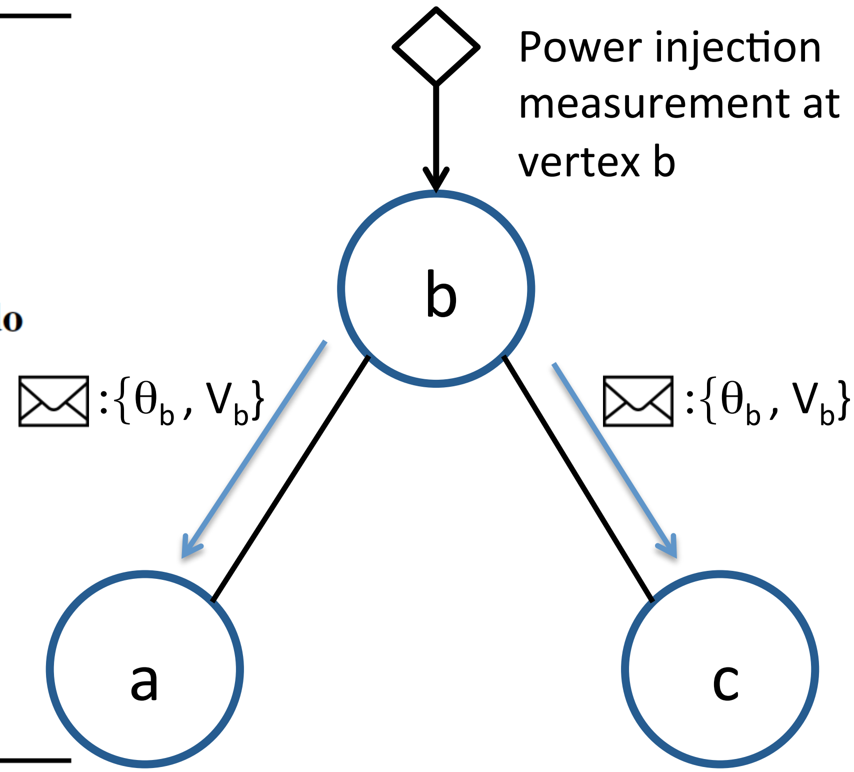
Icon Source: http://icons8.com/license/

Distributed Newton Method for DSE

Algorithm 1: Distributed Newton Method for State Estimation

Initialization: $\forall\, 1 \leq a \leq n,\ \boldsymbol{x}_a^{(0)} = \{\theta_a^{(0)}, V_a^{(0)}\}$ with $\theta_a^{(0)} = 0$ and $V_a^{(0)} = 1$.

for $a := 1$ *to* n **do**
  for $k := 0$ *to* K *(Newton/Outer Loop)* **do**
    $\Delta \boldsymbol{x}_a^{(0,k)} = 0$
    for $t := 0$ *to* T *(Matrix-Splitting/Inner Loop)* **do**
      Calculate $\Delta \boldsymbol{x}_a^{(t+1,k)}$ with matrix-splitting iterative scheme.
      Communication with neighbors.
    end
    $\boldsymbol{x}_a^{(k+1)} = \boldsymbol{x}_a^{k} + \Delta \boldsymbol{x}_a^{(k,T)}$
    Communication with neighbors.
  end
end



- Each vertex is assigned an MPI process.
- MPI Graph Communicator (`MPI_Graph_create`) used to mimic structure of power network.

Icon Source: http://icons8.com/license/

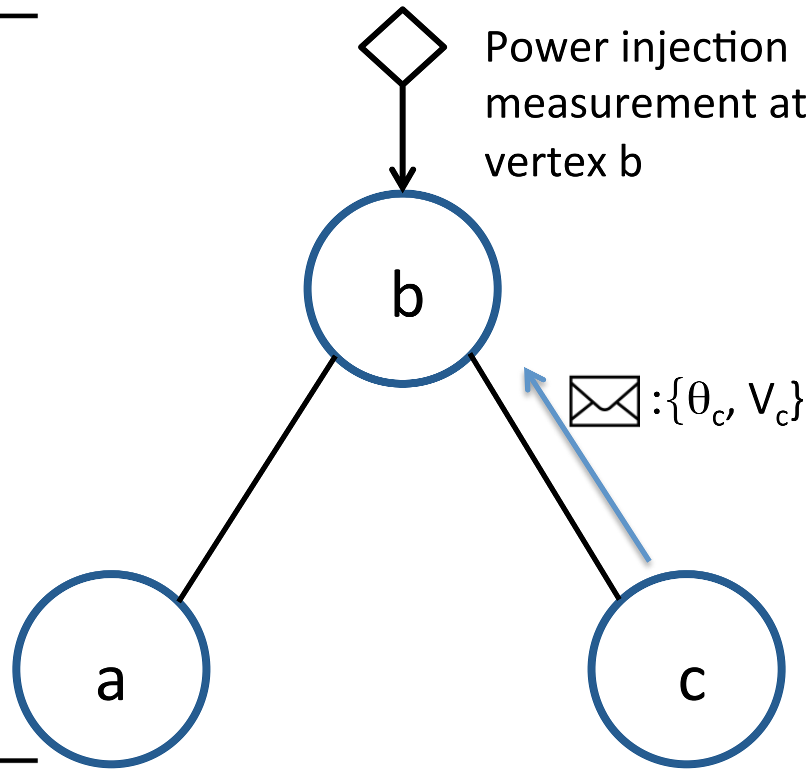
Distributed Newton Method for DSE

Algorithm 1: Distributed Newton Method for State Estimation

Initialization: $\forall\, 1 \le a \le n,\ \boldsymbol{x}_a^{(0)} = \{\theta_a^{(0)}, V_a^{(0)}\}$ with $\theta_a^{(0)} = 0$ and $V_a^{(0)} = 1$.

for $a := 1$ *to* n **do**
- **for** $k := 0$ *to* K *(Newton/Outer Loop)* **do**
  - $\Delta\boldsymbol{x}_a^{(0,k)} = 0$
  - **for** $t := 0$ *to* T *(Matrix-Splitting/Inner Loop)* **do**
    - Calculate $\Delta\boldsymbol{x}_a^{(t+1,k)}$ with matrix-splitting iterative scheme.
    - Communication with neighbors.
  - **end**
  - $\boldsymbol{x}_a^{(k+1)} = \boldsymbol{x}_a^{k} + \Delta\boldsymbol{x}_a^{(k,T)}$
  - Communication with neighbors.
- **end**

end



- Each vertex is assigned an MPI process.
- MPI Graph Communicator (`MPI_Graph_create`) used to mimic structure of power network.

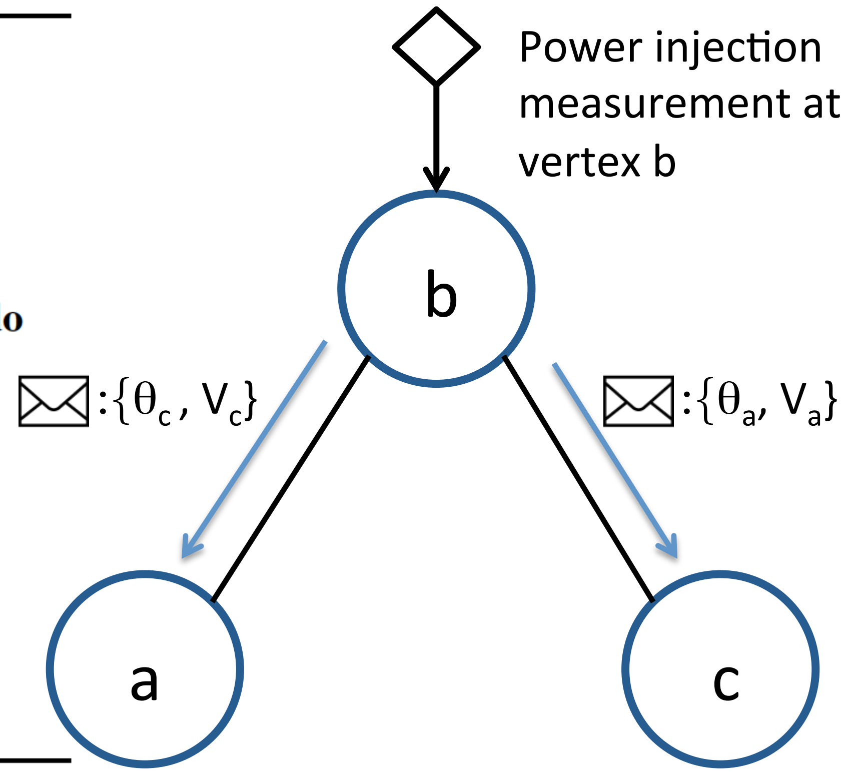
Icon Source: http://icons8.com/license/

Distributed Newton Method for DSE

Algorithm 1: Distributed Newton Method for State Estimation

Initialization: $\forall\, 1 \le a \le n,\ \boldsymbol{x}_a^{(0)} = \{\theta_a^{(0)}, V_a^{(0)}\}$ with $\theta_a^{(0)} = 0$ and $V_a^{(0)} = 1$.

for $a := 1$ *to* n **do**
    for $k := 0$ *to* K *(Newton/Outer Loop)* **do**
        $\Delta\boldsymbol{x}_a^{(0,k)} = 0$
        for $t := 0$ *to* T *(Matrix-Splitting/Inner Loop)* **do**
            Calculate $\Delta\boldsymbol{x}_a^{(t+1,k)}$ with matrix-splitting iterative scheme.
            Communication with neighbors.
        end
        $\boldsymbol{x}_a^{(k+1)} = \boldsymbol{x}_a^{k} + \Delta\boldsymbol{x}_a^{(k,T)}$
        Communication with neighbors.
    end
end



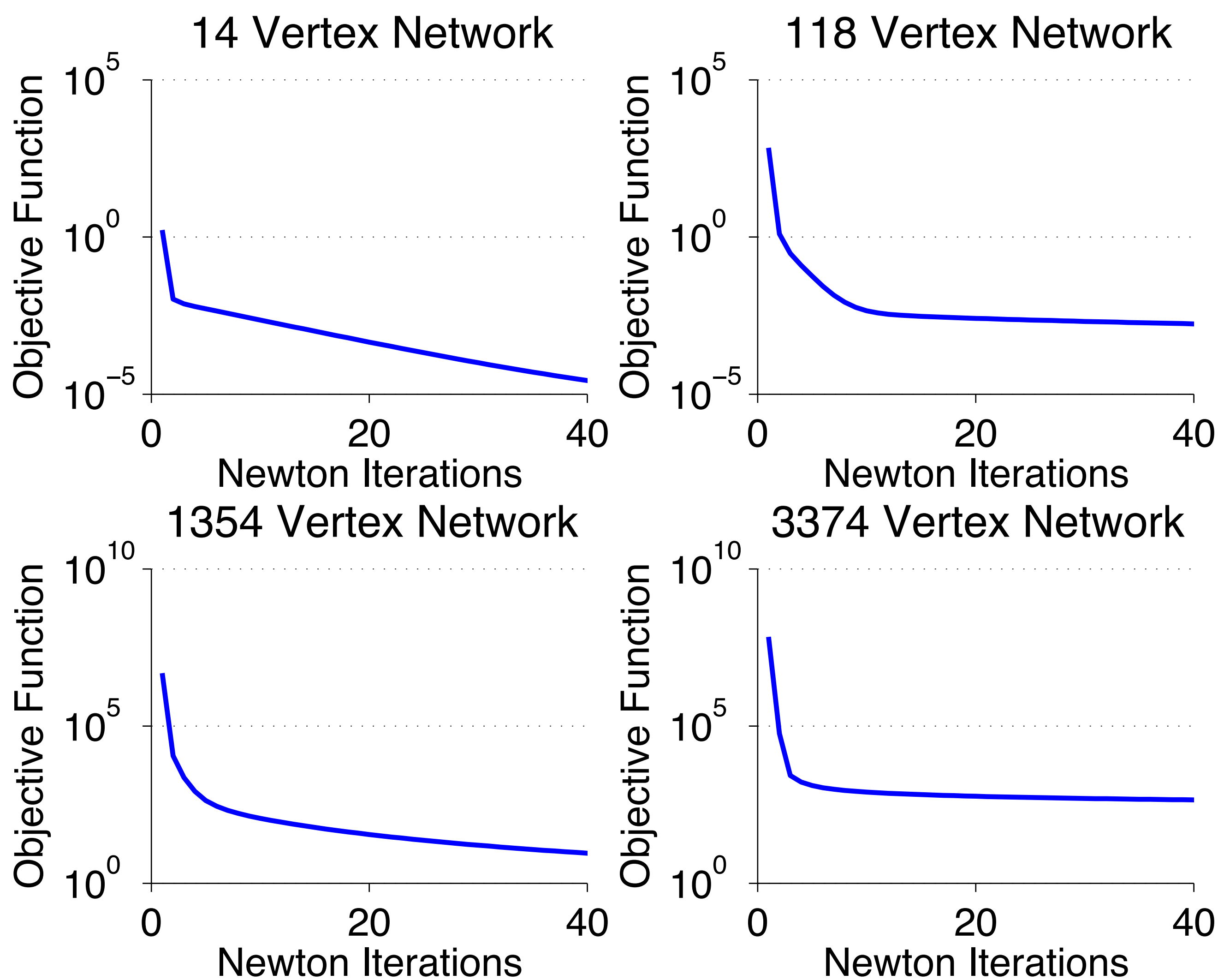
- Each vertex is assigned an MPI process.
- MPI Graph Communicator (`MPI_Graph_create`) used to mimic structure of power network.

Icon Source: http://icons8.com/license/

Outline

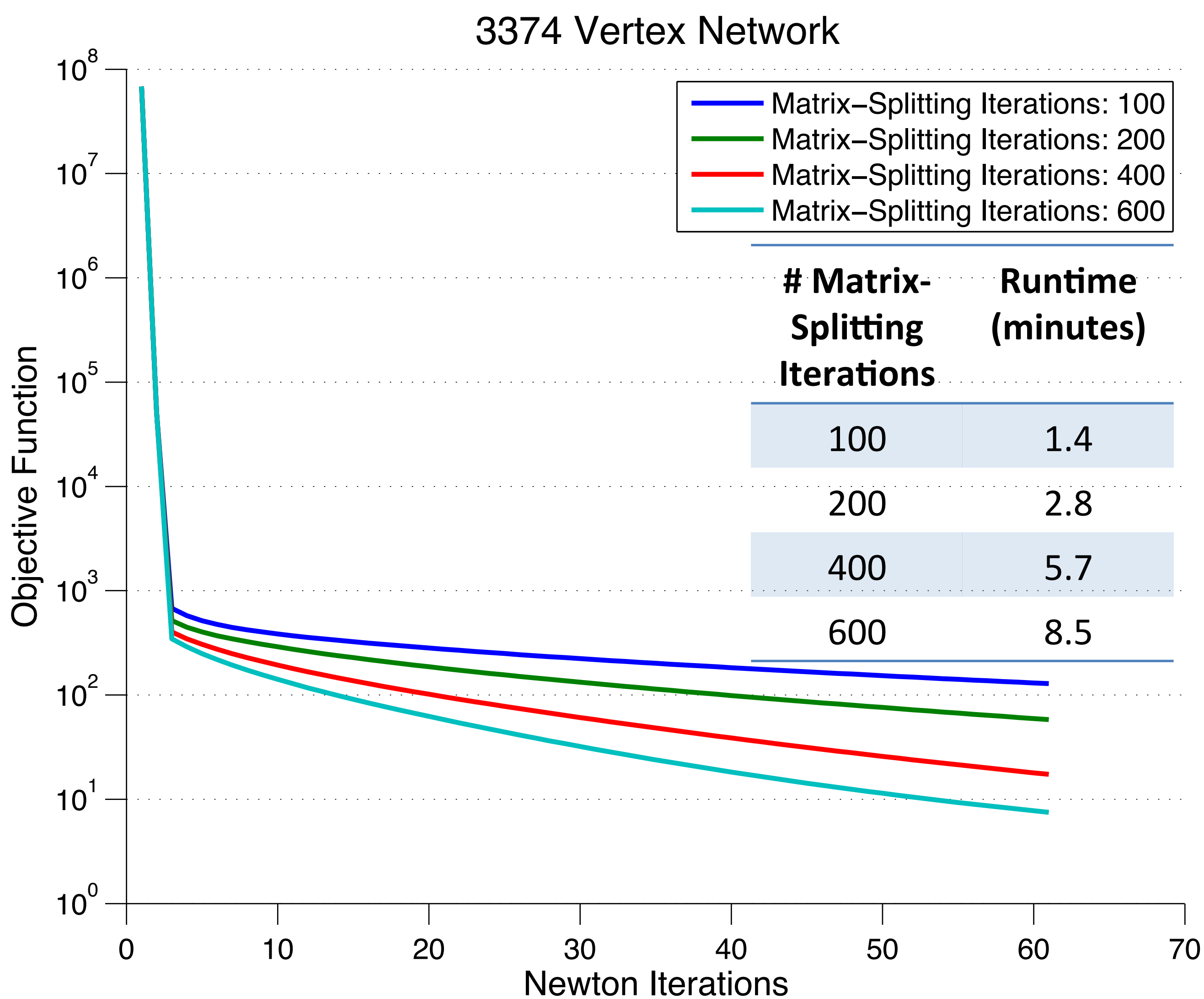
- Motivation
- Problem Statement
- Method
- Results
- Conclusions

Results: Accuracy Performance I



- Each uses 15 matrix-splitting iterations. Number of Newton iterations needed for convergence depends on power network size.

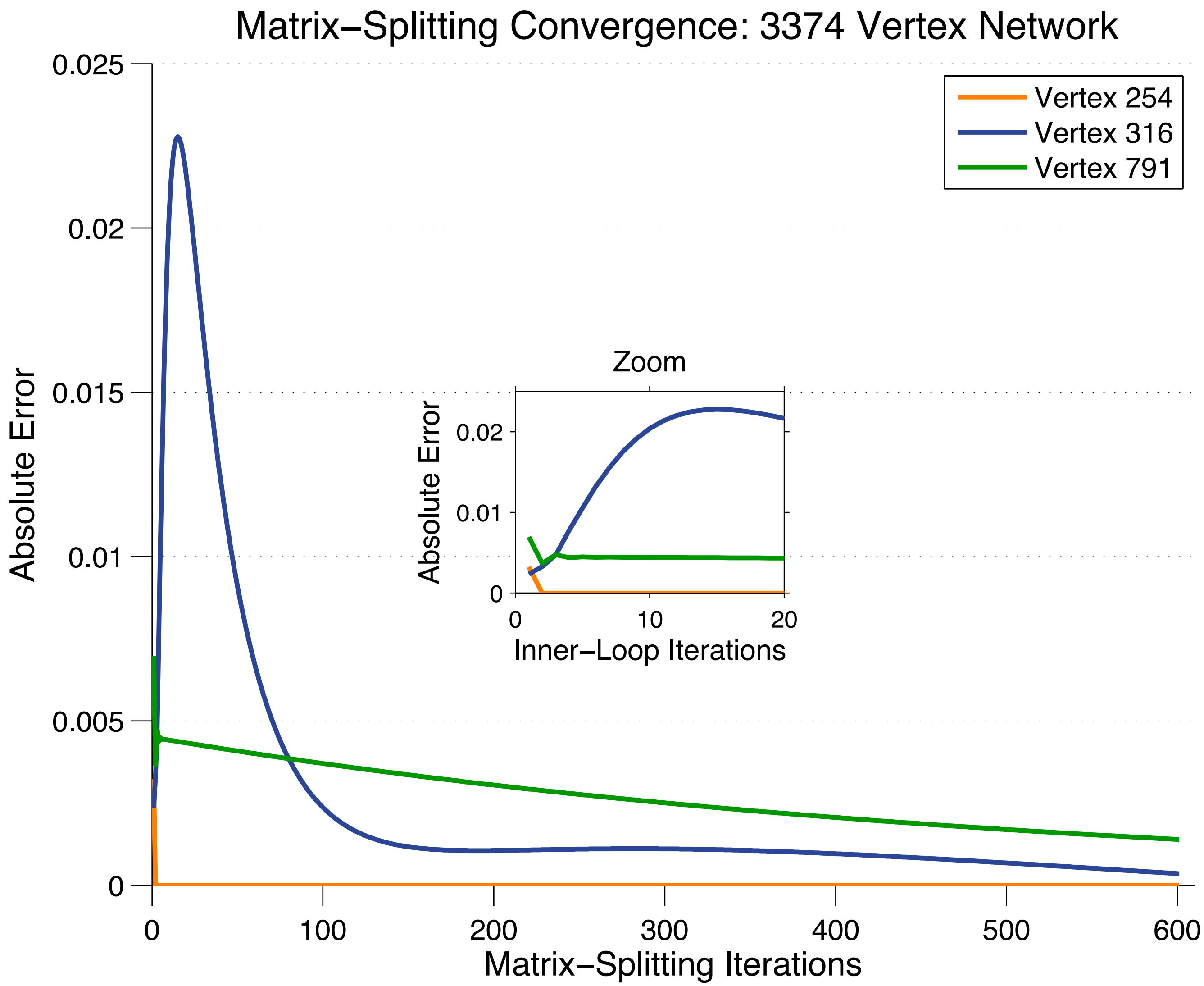
Results: Accuracy Performance II



# Matrix-Splitting Iterations	Runtime (minutes)
100	1.4
200	2.8
400	5.7
600	8.5

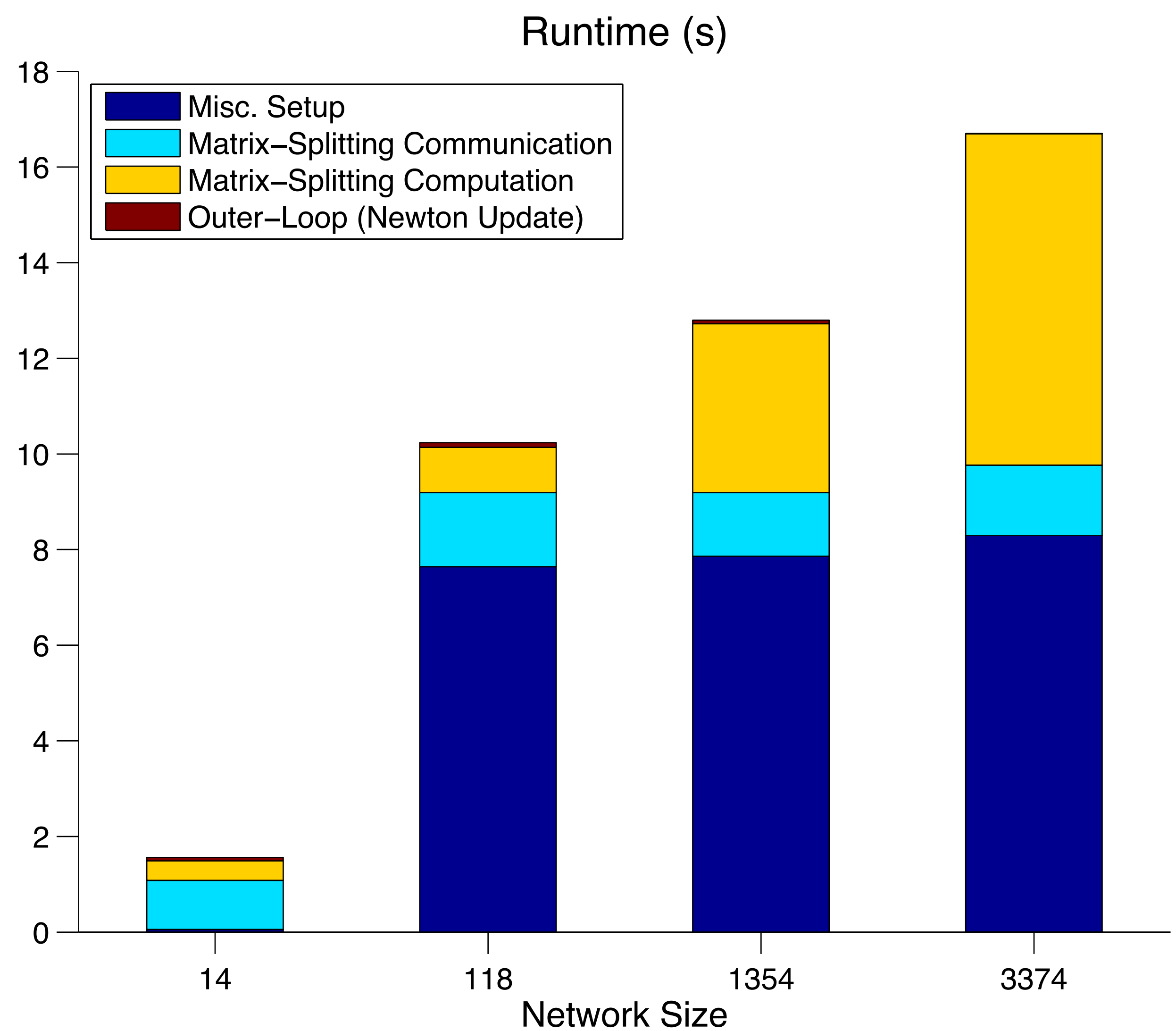
- Tradeoff between convergence and runtime for different number of matrix-splitting iterations.

Results: Accuracy Performance III



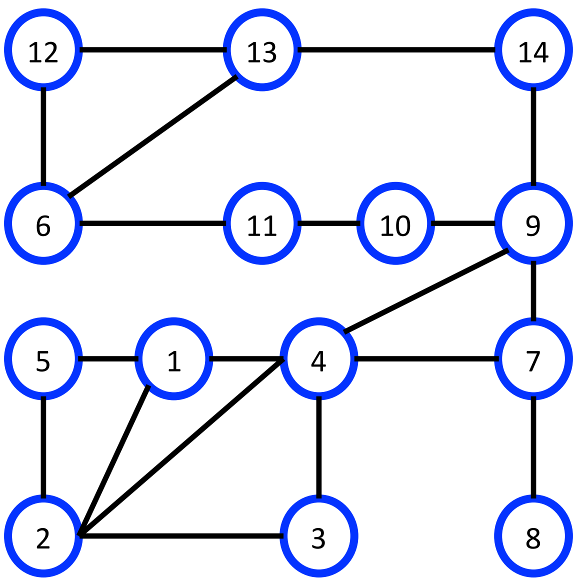
- Convergence of matrix-splitting iterative scheme varies between different vertices in the power network.

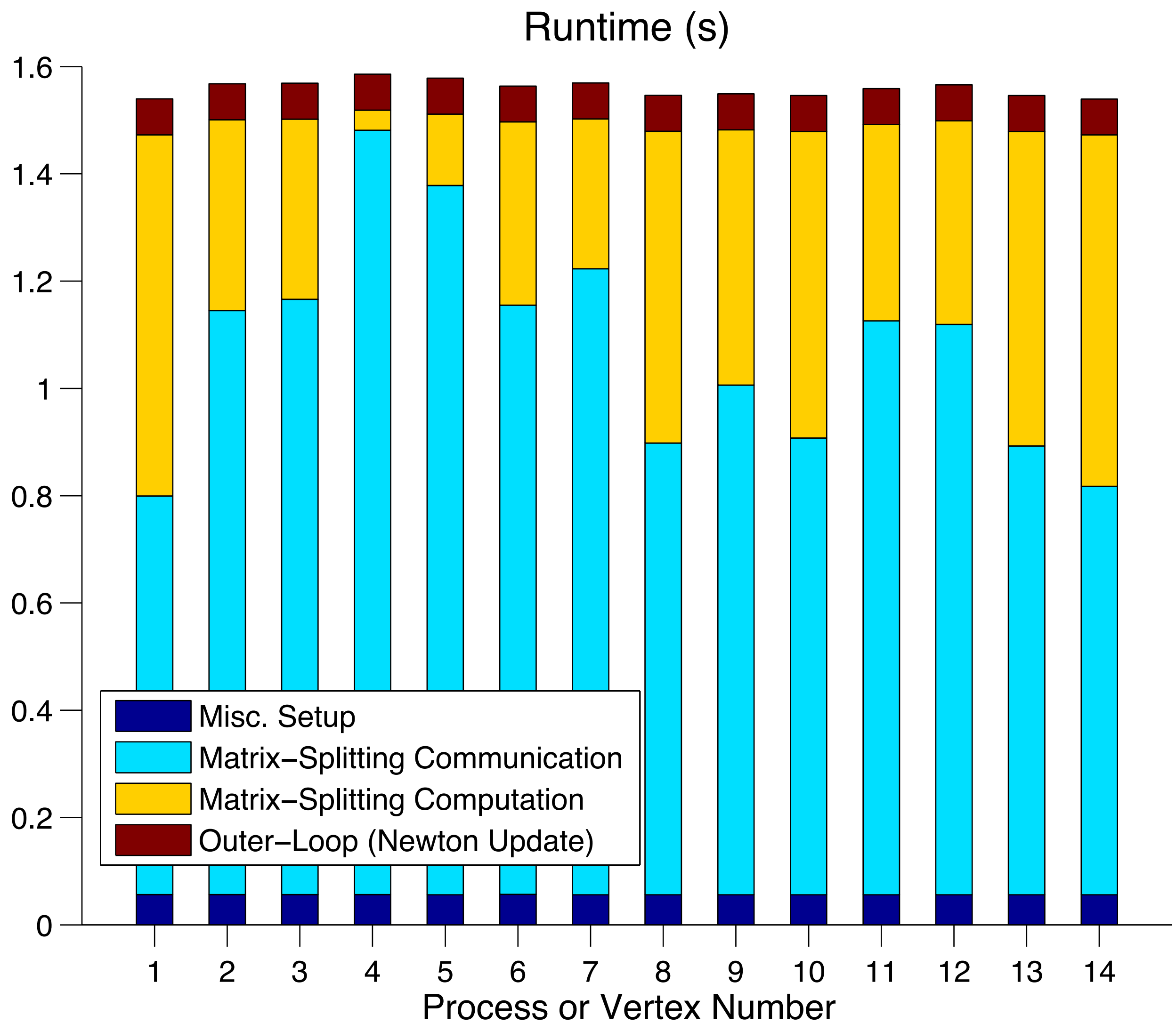
Results: Timing Performance I



- Using same number of iterations (15 matrix-splitting, 40 Newton), compare how computation and communication scale with power network size.

Results: Timing Performance II





- For 14-vertex power network, study how time spent on communication and computation varies among different vertices.

Outline

- Motivation
- Problem Statement
- Method
- Results
- Conclusions

Conclusions

- Use of Blue Waters
  - Allows testing of algorithms on large-scale, realistic systems.
  - Each vertex in the power grid is assigned an MPI process.

Acknowledgement: This research is part of the Blue Waters sustained-petascale computing project, which is supported by the National Science Foundation (awards OCI-0725070 and ACI-1238993) and the state of Illinois. Blue Waters is a joint effort of the University of Illinois at Urbana-Champaign and its National Center for Supercomputing Applications.

Conclusions

- Main result
    - Developed a new fully distributed state estimation algorithm using matrix-splitting techniques.
    - Requires limited sharing of information between neighboring vertices.
    - Saves memory resources.
    - Implementation available in C++ using MPI.
- Future Work
    - Develop new local pre-processing and asynchronous communication schemes.
    - Improve the robustness of state estimation with respect to imperfect communication as well as reduce network traffic.

- Thank you to the Blue Waters Graduate Fellowship Program and NSF for supporting this work, as well as to the NCSA staff and point of contact Craig Steffen, for their help in working with Blue Waters.

References

- F. C. Schweppe and J. Wildes "Power system static-state estimation, pt. I: exact model", *IEEE Trans. Power Apparatus and Systems*, vol. PAS-89, pp.120 -125 1970.
- A. Minot, N. Li, and Y. Lu, "A Distributed Newton Method for Power System State Estimation," Submitted to Trans. on Power Systems , 2015, http://scholar.harvard.edu/files/aminot/files/distr_se__tpsv1.2.pdf
- A. Minot and N. Li, "A Fully Distributed State Estimation Using Matrix Splitting Methods," American Control Conference, 2015.
- A. Abur and A. G. Exposito, Power System State Estimation: Theory and Implementation. CRC Press, 2004.
- L. Xie, D.-H. Choi, S. Kar, and H. V. Poor, "Fully Distributed State Estimation for Wide-Area Monitoring Systems," IEEE Trans. Smart Grid, vol. 3, no. 3, pp. 1154–1169, 2012.
- V. Kekatos and G. Giannakis, "Distributed Robust Power System State Estimation," IEEE Trans. Power Systems, vol. 28, no. 2, pp. 1617–1626, May 2013.
- R. Varga, Matrix Iterative Analysis. Springer Series in Computational Mathematics, 2000.
- E. Wei, A. E. Ozdaglar, and A. Jadbabaie, "A Distributed Newton Method for Network Utility Maximization," Conference on Decision and Control, pp. 1816–1821, 2010.
- D. P. Bertsekas and J. Tsitsiklis *Parallel and Distributed Computation - Numerical Methods*, 1989 :Prentice Hall

Backup Slides

Kirchoff's Voltage and Current Law

Kirchoff Current Law (KCL): conservation of charge

Sum of all currents leaving and entering a node $\Sigma I = 0$

Kirchoff Voltage Law (KVL): conservation of energy

Total voltage around a closed loop $\Sigma V = 0$

Measurement Model

1) If $z_k = \widehat{P}_{ab}$, then

$$\begin{aligned} h_k(\boldsymbol{x}) =& h_{P(a,b)}(\theta_a, V_a, \theta_b, V_b) \\ \stackrel{\text{def}}{=}& - V_a V_b \Big(G_{ab} \cos(\theta_a - \theta_b) + B_{ab} \sin(\theta_a - \theta_b) \Big) \\ & + G_{ab} V_a^2. \end{aligned} \tag{1}$$

2) If $z_k = \widehat{Q}_{ab}$, then

$$\begin{aligned} h_k(\boldsymbol{x}) =& h_{Q(a,b)}(\theta_a, V_a, \theta_b, V_b) \\ \stackrel{\text{def}}{=}& - V_a V_b \Big(G_{ab} \sin(\theta_a - \theta_b) - B_{ab} \cos(\theta_a - \theta_b) \Big) \\ & - (B_{ab}^{\text{sh}} + B_{ab}) V_a^2, \end{aligned} \tag{2}$$

where B_{ab}^{sh} is the shunt susceptance.

3) If $z_k = \widehat{P}_a$, then

$$\begin{aligned} h_k(\boldsymbol{x}) =& h_{a,P}(\theta_a, V_a, \{\theta_b\}_{b \in \mathcal{N}_a}, \{V_b\}_{b \in \mathcal{N}_a}) \\ \stackrel{\text{def}}{=}& \sum_{b \in \mathcal{N}_a} V_a V_b \Big(G_{ab} \cos(\theta_a - \theta_b) + B_{ab} \sin(\theta_a - \theta_b) \Big). \end{aligned} \tag{3}$$

Reference: [Wood & Wollenberg]

B_{ab} : susceptance of transmission line b/t vertices a and b

G_{ab} : conductance of transmission line b/t vertices a and b

Measurement Model

4) If $z_k = \widehat{Q}_a$, then

$$h_k(\boldsymbol{x}) = h_{a,Q}(\theta_a, V_a, \{\theta_b\}_{b\in\mathcal{N}_a}, \{V_b\}_{b\in\mathcal{N}_a}) \stackrel{\text{def}}{=} \sum_{b\in\mathcal{N}_a} V_a V_b \Big(G_{ab}\sin(\theta_a - \theta_b) - B_{ab}\cos(\theta_a - \theta_b)\Big). \quad (4)$$

5) If $z_k = \widehat{\theta}_a$, then

$$h_k(\boldsymbol{x}) = h_{\theta,a}(\theta_a) \stackrel{\text{def}}{=} \theta_a. \quad (5)$$

6) If $z_k = \widehat{V}_a$, then

$$h_k(\boldsymbol{x}) = h_{V,a}(V_a) \stackrel{\text{def}}{=} V_a. \quad (6)$$