



U.S. DEPARTMENT OF
ENERGY

PNNL-23736

Prepared for the U.S. Department of Energy
under Contract DE-AC05-76RL01830

Non-iterative Voltage Stability Analysis: Interim Report

YV Makarov
Z Hou
D Meng
ST Elbert
Z Huang

B Vyakaranam
D Wu
S Wang
L Miller

September 2014



Pacific Northwest
NATIONAL LABORATORY

*Proudly Operated by **Battelle** Since 1965*

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor Battelle Memorial Institute, nor any of their employees, makes **any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights.** Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof, or Battelle Memorial Institute. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

PACIFIC NORTHWEST NATIONAL LABORATORY

operated by

BATTELLE

for the

UNITED STATES DEPARTMENT OF ENERGY

under Contract DE-AC05-76RL01830

Printed in the United States of America

Available to DOE and DOE contractors from the
Office of Scientific and Technical Information,
P.O. Box 62, Oak Ridge, TN 37831-0062;
ph: (865) 576-8401
fax: (865) 576-5728
email: reports@adonis.osti.gov

Available to the public from the National Technical Information Service,
U.S. Department of Commerce, 5285 Port Royal Rd., Springfield, VA 22161
ph: (800) 553-6847
fax: (703) 605-6900
email: orders@ntis.fedworld.gov
online ordering: <http://www.ntis.gov/ordering.htm>



This document was printed on recycled paper.

(9/2003)

Non-iterative Voltage Stability Analysis: Interim Report

Y. V. Makarov	B. Vyakaranam
Z. Hou	D. Wu
D. Meng	S. Wang
S. T. Elbert	L. Miller
Z. Huang	

September 2014

Prepared for
the U.S. Department of Energy
under Contract DE AC05 76RL01830

Pacific Northwest National Laboratory
Richland, Washington 99352

Abstract

Voltage stability (VS) is an important problem in power system planning, operations and control. Historically, multiple system blackouts worldwide have been caused by voltage instability and resulted in economical detriments amounting to billions of dollars. Usually, during simulations the VS boundary is explored by gradual change of system generation and system load in certain stress direction. This process requires multiple iterations to solve power flow equations. The stressing process is repeated for multiple contingencies. Consequently, traditional methods to calculate the voltage stability boundary are computationally intensive and are not suitable for real-time applications. To date, faster methods (for instance, near-real-time methods to estimate the voltage stability margin (distance to instability) are usually based on simplifications of the voltage stability problem, not self-evident transformations of power flow equations, or they can explore only few stress directions within the limited time available for real-time applications. This project proposes an efficient combination of computational methods, including a new method proposed in this work, to calculate static voltage stability boundaries (VSBs) using accurate system models and multiple stress directions considered simultaneously. Essentially, the objective is to explore multi-dimensional voltage stability nomograms in coordinates of meaningful power system parameters in near real-time. Such nomograms can be used as security constraints in various real-time applications, for instance in security constrained economic dispatch. The method locates static voltage stability conditions (saddle-node bifurcations), which are often associated with singularities of the power flow Jacobian matrix. The non-iterative method explores the VSB for a large scale power system using the direct (or non-iterative) orbiting method and a new method exploiting the quadric properties of power flow equations in Cartesian coordinates. The singularity problem is formulated as a single nonlinear equation with respect to the scalar stress parameter. The proposed method allows quick exploration of VSBs in coordinates of critical power system parameters such as power flows on critical interfaces, which is essential for real-time system security assessment both locally and in a wide-area. Using the proposed method, several case studies were performed for a Western Interconnection planning model, a California and a Central America model. Significant peculiarities of the boundary configuration were identified, including its non-convexity, discontinuity, branching, and internal singularities (“holes”) that were not known before and cannot be found by traditional methods. This report demonstrates promising performance characteristics of the proposed method using several power systems models. The new method will help to develop a new generation of highly efficient tools suitable for real-time parallel implementation. The ultimate benefit obtained will be early detection of system instability and prevention of system blackouts in real time.

The work described in this report was performed at Pacific Northwest National Laboratory (PNNL) and funded by the Department of Energy Office of Electricity Delivery and Energy Reliability – DOE OE.

Summary

The work described in this report was performed at Pacific Northwest National Laboratory (PNNL) and funded by the Department of Energy Office of Electricity Delivery and Energy Reliability (DOE OE).

A system enters a state of voltage instability when a disturbance, increase in load demand, or change in system conditions causes a progressive and uncontrollable decline in voltage. The main factor causing voltage instability is the inability of the power system to meet the demand for reactive power. Voltage stability is the ability of a power system to maintain acceptable voltages at all buses in the system under normal operating conditions and after being subjected to a disturbance. Voltage stability margin is the distance to instability determined for a selected loading or stress direction in parameter space. It is known that voltage magnitudes alone are poor indicators of voltage stability or security. Voltages can be near normal with generators, synchronous condensers, and static volt-ampere reactive (VAR) compensators near current limiting levels, thus resulting in a possible voltage collapse. Therefore, it is prudent to specify a mega VAR (MVAR) margin or megawatt (MW) margin [Adjarapu 1992].

Usually, during simulations the VS boundary is explored by gradual change of system generation and system load in certain stress direction. This process requires multiple iterations to solve power flow equations. The stressing process is repeated for multiple contingencies. Consequently, traditional methods to calculate voltage stability boundary are computationally intensive and are not suitable for real-time applications. To date, faster methods (for instance, near-real-time methods) to estimate the voltage stability margin (distance to instability) are usually based on simplifications of the voltage stability problem, not self-evident transformations of power flow equations, or they can explore only few stress directions within the limited time available for real-time applications. This project proposes an efficient combination of computational methods, including a new method proposed in this work, to calculate static voltage stability boundaries (VSBs) using accurate system models and multiple stress directions considered simultaneously. Essentially, the objective is to explore multi-dimensional voltage stability nomograms in coordinates of meaningful power system parameters in near real-time. Such nomograms can be used as security constraints in various real-time applications, for instance in security constrained economic dispatch. The method locates static voltage stability conditions (saddle-node bifurcations), which are often associated with singularities of the power flow Jacobian matrix.

The project work reported in this report pursued the following specific objectives:

- Develop a new efficient non-iterative method to determine the voltage stability boundary;
- Contribute to the voltage stability theory associated with the non-iterative method;
- Develop prototype MATLAB® code to implement the non-iterative method;
- Examine and demonstrate the non-iterative method using several power system models.

This work is based on some previous projects funded by National Science Foundation (NSF), California Energy Commission (CEC) Public Interest Energy Research (PIER) program, PNNL Laboratory Directed Research & Development (LDRD) program, DOE OE Office, Consortium for Electricity Technology Reliability Solutions (CERTS), and Bonneville Power Administration (BPA).

This project is based on a solid foundation created by PNNL for the CEC-funded project conducted by CERTS, Electric Power Group (EPG), PNNL, and University of Wisconsin – Madison project “Real-Time Voltage Security Assessment (RTVSA)” [Parashar 2007a], [Parashar 2007b], [Parashar 2007a], [Parashar 2008b]. The connection between various projects in this area is shown in Appendix A.

The selected platform for implementing the procedure includes the Power System Engineering Research Center (PSERC) Continuation Power Flow program implemented using the MATLAB® programming language [Greene 1997], [Dobson 2001], [Alvarado 2001].

The non-iterative method developed in this project explores the VSB for a large scale power system using a direct or non-iterative orbiting method and a new method exploiting the quadric properties of power flow equations in Cartesian coordinates. The singularity problem is formulated as a single nonlinear equation with respect to the scalar stress parameter. The proposed method allows quick exploration of VSBs in coordinates of critical power system parameters such as power flows on critical interfaces, which is essential for real-time system security assessment locally and in a wide-area.

The idea behind our approach is shown in Figure 1.

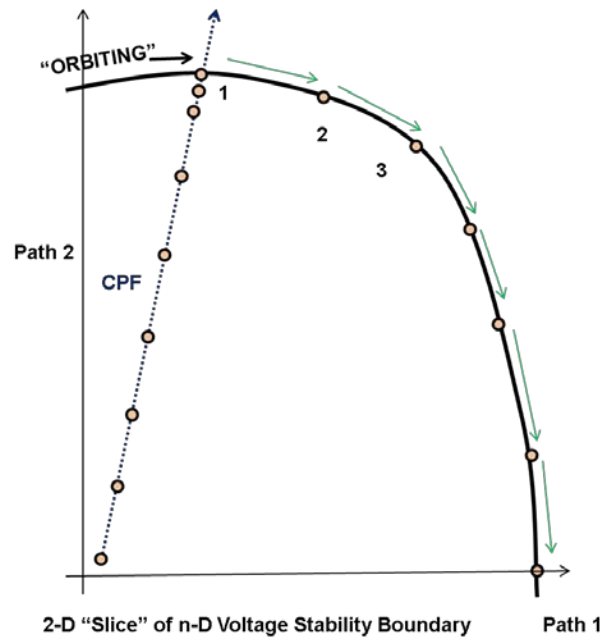


Figure 1. The idea of non-iterative voltage stability analysis.

In Figure 1, a slice of the VSB is shown spanned by power transfers in two transmission paths. The initial VSB point, labeled “1” in Figure 1, can be found using a traditional method such as the continuation power flow (CPF) method. The CPF requires multiple steps to locate the first VSB point. The additional points along the boundary (points 2, 3, etc.) are produced by the direct orbiting method, which produces a new point after each iteration, making it a non-iterative method. Our new method (the modified XR method) is used to restart the process when needed.

The following milestones were achieved in this phase of the research work:

- Upgrade the non-iterative method to the level suitable for testing on industry examples. Develop MATLAB® code capable of reading PowerWorld® formats (simplified models) and run the VSA computation in real system models.
- Explore opportunities of quadratic approximation of the voltage stability boundary. Implement and test critical elements of non-iterative VS analysis algorithms on high-performance computers. Complete non-iterative voltage stability analysis software capable of running on PNNL Institutional Computing (PIC) or equivalent high-performance computing (HPC).
- Technology outreach effort. Through various forums obtain the commitment of at least one organization to participate.

The following main conclusions have been drawn:

- The proposed non-iterative method can find static stability boundary point precisely for real large size power system models;
- The proposed non-iterative method is much faster than the traditional step-by-step stressing methods, e.g. the repeated application of the continuation power flow method to define a single VSB point.
- The piecewise nonlinear quadratic approximation of VSB meets significant difficulties with finding where the approximating quadrics intersect; this makes practically impossible to quantify the actual voltage stability margin. This approach cannot be recommended for a practical implementation.
- At the same time, the non-linear approximation used in the form of high-order numerical method used to solve VSB equations could be useful for tracing the VSB. It can be implemented in the next phase of the project.

Using the proposed method, several case studies were performed for a Western Interconnection planning model, a California model and a Central America model. Significant peculiarities of the boundary configuration were identified, including its non-convexity, discontinuity, branching, and internal singularities (“holes”) that were not known before and cannot be found by traditional methods. This report demonstrates promising outcomes and performance characteristics of the proposed method using several power systems models. This new method will help to develop a new generation of highly efficient tools suitable for real-time parallel implementation. The ultimate benefit obtained will be early detection of system instability and prevention of system blackouts in real time.

This project builds towards the following benefits for system operations:

- Enhanced situational awareness, early detection of potential system instability and improved reliability achieved by providing to power system dispatchers: (1) real-time dynamically adjusted information, and (2) VSB visibility for multiple stress directions and contingencies.

- Actionable information and prevention of system blackouts achieved by providing advisory information using the wide area security region approach [Makarov 2009].
- Better utilization of transmission assets due to near-real-time adjustment of the voltage stability limits and accounting for mutual dependences of power flows on critical transmission paths.
- Developing real-time and HPC applications.
- Potential for predictive/preventive control.
- Potential for closed-loop automatic emergency control systems.

The following recommendations for the next phase of this project are:

(1) Algorithm enhancement: Increase the robustness of the tool for a variety of system models and conditions and demonstrate its trustworthiness for solving real industry cases.

(2) Tool and user interface refinement: Further improve tool performance and improve UI to make it attractive for adoption by vendors.

(3) Vendor engagement: Collaborate with a selected vendor or vendors to transfer the technology. This will include presentations at industry forums and conferences. To date, PNNL has received expression of interest from several vendors.

(4) Non-iterative small signal stability method development: This task will develop methods and prototypes for non-iterative small signal analysis. The method will enable plotting of the small-signal stability boundary in coordinates of critical system parameters in near real-time.

Acknowledgments

The Pacific Northwest National Laboratory (PNNL) is operated by Battelle for the U.S. Department of Energy under contract No. DE-AC05-76RL01830.

This work was supported by the Department of Energy Office of Electricity Delivery and Energy Reliability (OE Office).

The authors would like to acknowledge the following contributors for their inputs to the project.

Mr. Gil Bindewald, U. S. Department of Energy, OE Office

Dr. Dmitry Kosterev, Bonneville Power Administration (BPA)

Ms. Irina Green, California Independent System Operator (CAISO)

Dr. Hassan Ghoudjehbklou, Sempra Energy Utilities

Dr. Mallikarjuna Vallem, Peer Reviewer, PNNL

Dr. Barry Lee, Associate Professor, Department of Mathematics, Southern Methodist University, formerly PNNL employee

Dr. Mark P. Morgan, Project Coordinator, PNNL

Ms. Susan J. Arey, Project Specialist, PNNL

This work is based on some previous projects funded by the National Science Foundation (NSF), California Energy Commission (CEC) Public Interest Energy Research (PIER) program, PNNL Laboratory Directed Research & Development (LDRD) program, DOE OE Office, Consortium for Electricity Technology Reliability Solutions (CERTS), and Bonneville Power Administration (BPA).

In this connection, the authors want to acknowledge contributions from the following projects, colleagues and organizations:

1. Power Systems Engineering Research Center (PSERC) project “Transfer Capability Calculator” funded by National Science Foundation (NSF) Research Centers - Small Firms Collaborative R&D Initiative under NSF Awards 9815325 and 9815678.

PSERC project contributors:

Dr. Scott Greene, Solution Architect: Real-time Dispatch Tool, Portland General Electric

Prof. Ian Dobson, Sandbulte Professor, Iowa State University, formerly University of Wisconsin - Madison

Prof. Fernando L. Alvarado, Emeritus Professor, University of Wisconsin - Madison

Prof. Christopher L. DeMarco, Grainger Professor of Power Engineering, University of Wisconsin - Madison

Dr. Rajesh Rajaraman, Vice President, Christensen Associates

References: [Greene 1997], [Dobson 2001], [Alvarado 2001]

2. “Real-time Voltage Stability Analysis” project funded by the California Energy Commission PIER Program (CEC) and managed by the Consortium for Electricity Technology Reliability

Solutions (CERTS). Participating organizations: PNNL, Electric Power Group (EPG), University of Wisconsin – Madison.

Project contributors:

Dr. Joseph H. Eto, CERTS Project Manager, Grid Integration Group, Lawrence Berkeley National Laboratory

Prof. Ian Dobson, Sandbulte Professor, Iowa State University, formerly with University of Wisconsin – Madison

Dr. Manu Parashar, WAMS Activity Director at Alstom Grid, formerly EPG employee

Dr. Matthew Varghese, Senior Engineer, Reliability Assessment and Performance Analysis, (NERC), formerly EPG employee

Mr. Abhijeet Aggarwal, Director, JSH Power, formerly EPG employee

Dr. Jim Dyer, Principal Consultant with Electric Power Group, LLC.

Dr. Ning Zhou, Assistant Professor at Binghamton University, formerly PNNL employee

References: [Parashar 2007a], [Parashar 2007b], [Parashar 2007a], [Parashar 2008b]

3. “Wide Area Power System Security Region” project cost-shared by DOE OE Office and Bonneville Power Administration, conducted by PNNL.

Project contributors:

Mr. Philip Overholt, Technical Project Officer in the Office of Energy Delivery and Energy Reliability, DOE.

Dr. Jim Gronquist, Lead Operations Engineer at Bonneville Power Administration

Mr. James Burns, Bonneville Power Administration

Dr. Pengwei (David) Du, Senior Engineer-Renewable Integration at Electric Reliability Council of Texas (ERCOT), formerly PNNL employee

Dr. Shuai Lu, Senior Engineer, PNNL

Dr. Tony B. Nguyen, Engineer, PNNL

Mr. Xinxin Guo, Engineer at Qado Energy, formerly PNNL employee

Reference: [Makarov 2009]

Acronyms and Abbreviations

ARC	Australian Research Council
ARPA-E	DOE Advanced Research Projects Agency-Energy
BC	British Columbia
BPA	Bonneville Power Administration
CAISO, California ISO	California Independent System Operator
CEC	California Energy Commission
CERTS	Consortium for Electricity Technology Reliability Solutions
CPF	Continuation power flow
CPU	Central processing unit
DAE	Differential and algebraic equations
DOE	Department of Energy
EERE	DOE Office of Energy Efficiency and Renewable Energy
EPG	Electric Power Group
ERCOT	Electric Reliability Council of Texas
GUI	Graphical user interface
HB	Hopf bifurcation
HELM	Holomorphic embedding load flow method
HO method	High-order numerical method
HPA	Hyperplane approximation
HPC	High-performance computing
IEEE	Institute of Electrical and Electronics Engineers
LDRD	Laboratory Directed Research & Development program
MATLAB	MATrix LABoratory
MVAR	Mega-Volt-Ampere reactive
MW	Megawatt
NERC	North American Electric Reliability Corporation
NR	Newton-Raphson method
NSF	National Science Foundation
MW	Megawatt
OE	DOE Office of Electricity Delivery and Energy Reliability
OM	Direct (non-iterative) orbiting method
PIC	PNNL Institutional Computing
PIER	CEC Public Interest Energy Research
PNNL	Pacific Northwest National Laboratory
PoC	Point of collapse
PQ	Real power (P) and reactive power (Q)

PSERC	Power System Engineering Research Center
p.u.	Per unit
PV	Real power and voltage magnitude
PW	PowerWorld® software package
RTVSA	Real-Time voltage security assessment
SNB	Saddle-node bifurcation
UI	User interface
VAR	Volt-Ampere-reactive
VS	Voltage stability
VSA	Voltage stability analysis
VSB	Voltage stability boundary
VSR	Voltage stability region
WAMS	Wide-area measurement system
WECC	Western Electricity Coordinating Council
XR	X-ray method

Contents

Abstract.....	iii
Summary	v
Acknowledgments	ix
Acronyms and Abbreviations	xi
1.0 Introduction.....	1.1
2.0 Some Characteristics of the Voltage Stability Boundary and Stress Directions	2.1
2.1 Stresses and the VSB in the Parameter and State Space.....	2.1
2.1.1 Mapping of Stresses in the Parameter and State Space.....	2.1
2.2 Search for Indicators to Identify a VSB Point along a Stress Direction in the State Space.....	2.7
2.3 Power Losses in the Vicinity of the VSB.....	2.8
2.4 Voltage Reduction in the Vicinity of VSB.....	2.10
2.5 Mapping Stresses in the Parameter and State Spaces.....	2.12
2.6 Exploring Voltage Stability Boundary Points Using Stresses Defined in Parameter and State Spaces.....	2.16
3.0 New Methodology for Exploring the Voltage Stability Boundary.....	3.1
3.1 XR Methodology.....	3.1
3.1.1 Power Flow Equations in Cartesian Coordinates.....	3.1
3.1.2 Jacobian Matrix in Cartesian Coordinates	3.2
3.1.3 Solving Singularity Equations	3.3
4.0 Exploring Abnormal Singularities within the WECC System Stability Region	4.1
4.1 Details of the Approach to Explore the Voltage Stability Boundary	4.1
5.0 Overall Implementation of Non-iterative Method.....	5.1
5.1 Details of Implementation.....	5.3
5.1.1 Continuation Power Flow (CPF).....	5.3
5.1.2 Direct Method to Refine the Position of VSB Point along the Stress Direction ..	5.7
5.1.3 Inverse Iterations.....	5.7
5.1.4 Direct Orbiting Method to Follow the VSB.....	5.8
5.1.5 Implementation of XR Method.....	5.9
5.1.6 Hyperplane Approximation.....	5.11
5.1.7 High-order Numerical Methods (HO).....	5.13
5.2 Algorithm Implementation and Software Code.....	5.16
5.3 Connectivity with PowerWorld®.....	5.17
6.0 Experimental Results and Improvements	6.1
6.1 Test Cases.....	6.1
6.1.1 Central America System.....	6.1
6.1.2 California ISO Model.....	6.1

6.1.3	Western Electricity Coordinating Council (WECC) Model	6.1
6.2	Transformation from Polar Coordinates to Rectangular Coordinates	6.1
6.3	Improvements and Tests: Discrete Events	6.1
6.3.1	Comparing Bus Types at Boundary Points Obtained From CPF and OM	6.2
6.3.2	Sensitivity of Bus Types to Changes in Stress Trajectory	6.2
6.3.3	Proposed Approach to Directly Detect Discrete Events and Generation Bus Types in the XR Method	6.4
6.3.4	Incorporating Constraints and Discrete Events into the XR Method	6.6
6.4	Hyperplane Method Test	6.9
6.5	Examples of VSB Plots for Test Systems	6.10
6.5.1	Central America System	6.10
6.5.2	California System	6.11
6.5.3	WECC System	6.12
6.6	Computational Time	6.12
6.6.1	Central America System	6.12
6.6.2	California System	6.13
6.6.3	WECC System	6.13
6.6.4	Comparison of Results and Conclusion	6.14
6.7	Accuracy	6.14
7.0	Project Benefits and Expected Impacts	7.1
8.0	Conclusion	8.1
9.0	Recommendations for Next Phase	9.1
10.0	References	10.1
	Appendix A : Voltage Stability Related Blackouts Worldwide	A.1
	Appendix B : Power Flow in Cartesian Coordinates	B.1
	Appendix C : Formulation of the Quadratic Approximation	C.1
	Appendix D : Development of Input File Using SimAuto	D.1
	Appendix E : Jacobian Matrix Transformation from Polar to Rectangular Coordinates	E.1
	Appendix F : SimAuto and MATLAB® Scripts	F.1
	Appendix G : Additional Procedures and Results	G.1

Figures

Figure 1. The idea of non-iterative voltage stability analysis.	vi
Figure 2. Stresses in parameter space.	2.2
Figure 3. Stresses in the state space.	2.2
Figure 4. IEEE 9-bus test system.	2.3
Figure 5. Stress trajectories of active powers at bus 5, 6 and 8 (view 1).	2.4
Figure 6. Stress trajectories of active powers at bus 5, 6 and 8 (view 2).	2.4
Figure 7. Stress trajectories of active powers at bus 5, 6 and 8 (view 3).	2.5
Figure 8. Stress trajectories of nodal voltages at bus 5, 6 and 8 (view 4).	2.5
Figure 9. Stress trajectories of nodal voltages at bus 5, 6 and 8 (view 5).	2.6
Figure 10. Stress trajectories of nodal voltages at bus 5, 6 and 8 (view 6).	2.6
Figure 11. A 3-generator 3-bus system.	2.8
Figure 12. Static voltage stability boundary of 3-bus system in parameter space (p.u.).	2.9
Figure 13. Losses at different operating points within the boundary (p.u.).	2.9
Figure 14. Contour plot of system losses with $Q_2 = 0.5$	2.10
Figure 15. Norm of state variables at different operating points within the boundary.	2.11
Figure 16. Contour of norm of state variables.	2.11
Figure 17. A 2-generator 2-bus system.	2.12
Figure 18. Part of the boundary and some paths in parameter space. (The mapping of Figure 19 to parameter space.)	2.13
Figure 19. Part of the boundary and some paths in state space. (The mapping of Figure 18 to state space.)	2.13
Figure 20. $\ \Delta g\ $ vs. α along path 4.	2.14
Figure 21. $\ \Delta g\ $ vs. α along path 3.	2.15
Figure 22. Contour of $\ x\ $ in parameter space.	2.16
Figure 23. Stability boundary and stress trajectory in parameter space.	2.17
Figure 24. Stability boundary and stress trajectory in state space.	2.17
Figure 25. Exploration of voltage stability boundary on a two-dimensional plane.	4.2
Figure 26. Stability boundary for Western Interconnection system.	4.4
Figure 27. Illustration of the combined method of exploring the VSB.	5.2
Figure 28. Basic flowchart for the combined method of exploring the VSB.	5.2
Figure 29. Continuation power flow step: predictor and corrector.	5.3
Figure 30. Solution of parametric differential equation for a scalar case.	5.4
Figure 31. Parametric motion in the space of mismatches.	5.5
Figure 32. Exploration of distance function (100% denoted a VSB point).	5.10
Figure 33. Illustration of an approximating hyperplane in the parameter space.	5.12
Figure 34: Tangent and approximating hyperplanes.	5.13

Figure 35. Comparing bus types obtained by different methods in California system model.....	6.2
Figure 36. Boundary points from different stressing direction (California system model).....	6.3
Figure 37. The difference between trajectories $Y(\alpha)$ and $Y(\beta)$	6.5
Figure 38. Exploration of $\ R(\alpha)\ $ and distance function ($\ J(\alpha)R(\alpha)\ $) without reactive power constraints.	6.7
Figure 39. Exploration of $\ R(\alpha)\ $ and distance function ($\ J(\alpha)\tilde{R}(\alpha)\ $) without fixed reactive power constraints.	6.8
Figure 40. Approximating lines for the California test case (pink lines).	6.10
Figure 41. Central America System VSB.....	6.11
Figure 42. California system VSB.....	6.11
Figure 43. WECC system VSB.....	6.12
Figure 44. Computational time to get one slice of VSB depending on the system size.	6.14
Figure 45: Illustration for quadratic approximation.	C.2
Figure 46: Shortest distance between current operating point and the surface.....	C.5
Figure 47: Original precision with seven significant digits.....	G.4
Figure 48: Original precision with six significant digits.....	G.5
Figure 49: Original Precision with five significant digits.	G.6
Figure 50: Original precision with seven significant digits.....	G.7
Figure 51: Original precision with six significant digits.....	G.8
Figure 52: Original precision with five significant digits.	G.9

Tables

Table 1. Number of PV buses in each direction.....	6.3
Table 2. Computational time for the Central America system	6.13
Table 3. Computational time for the California system.....	6.13
Table 4. Computational time for the WECC system.....	6.14
Table 5. Comparison of the VSB points obtained by two methods.....	6.15
Table 6. Comparison of simulation time (in second) of different methods.....	G.2

1.0 Introduction

Voltage stability (VS) is an important problem in power system planning, operations and control. Historically, multiple system blackouts worldwide have been caused by voltage instability and resulted in economical detriments amounting to billions of dollars – see Appendix A.

As electrical demand increases, modern power systems are being operated under increasingly stressed conditions, leading power systems closer to their stability limits. Voltage stability is an important problem in power system planning, operation and control. Voltage instability occurs when a power system is unable to maintain stable voltage levels at some buses. Under that condition, voltages can decline progressively throughout the system and may cause a large-scale blackout. Voltage instability has been identified to be responsible for the collapse of several major power worldwide [Makarov 2007]. Identification of the voltage stability boundary (VSB) plays a vital role in the reliable and economic operation of a power system. Although the voltage stability margin depends on numerous possible system stresses in practical real-time applications, only several selected stress directions are actually computed and checked in practical applications, especially, real-time applications. However, with the increasing variability and uncertainty in today's power systems, it is becoming increasingly clear that the stability margin definitions should be based on multiple stresses covering various sources and ranges of uncertainty and variability. Therefore, an accurate and fast estimation of the available voltage stability margin is of paramount importance for the secure and economic operation and control of modern electric power systems.

The voltage stability boundary surrounds the region of feasible and stable operating points in power system parameter space. These operating points cannot cross the VSB without losing their stability [Venkatasubramanian 1995a]. The voltage stability region (VSR) is a safe region for guaranteeing local stability at the equilibrium under slow parametric variations [Venkatasubramanian 1995b]. Voltage stability conditions are usually considered as power flow feasibility conditions; and the VSB is associated with singularity conditions of the power flow Jacobian matrix and saddle-node bifurcation (SNB). References [Venkatasubramanian 1995a] and [Venkatasubramanian 1995b] provide a comprehensive discussion on feasibility boundaries and regions in state and parameter space in the power system domain, and summarize some recent development on the stability analysis of large-scale systems. More recent publications that address the voltage stability problems include [Zhang 2011], [Dong 2012], [Nguyen 2012], [Perninge 2013].

The traditional methods for calculating SNBs traditionally employ iterative procedures, unlike the direct method in this project. An extensive review of these methods is provided in [Parashar 2007a]. The continuation power flow (CPF) and direct methods are two commonly used iterative methods [Eto 2008]. The purpose of CPF is to find a series of load-flow solutions for a given load/generation change scenario [Ibsais 1996]. The CPF method provides reliable convergence due to its predictor-corrector approach, but is computationally intensive. Direct methods were proposed for assessing the VSB, in which augmented power flow equations are solved [Dobson 1992], [Dobson 1993], [Kwatny 1995], [Makarov 2000a]. These methods provide the left or right eigenvectors corresponding to the zero Jacobian matrix eigenvalue at the point of voltage collapse. Direct methods are sensitive to the initial guess, i.e., the initial guess affects the speed of convergence and may even cause divergence of the iterative process [Makarov 2000a]. This iterative process requires a considerable number of calculations

to find a single VSB point. To obtain the full VSB, the computational effort becomes prohibitively significant for large-scale power systems. Computational time becomes critically important for real-time analyses, massive contingency screenings, simulations of blackouts and cascading processes, probabilistic methods, and synchrophasor-based applications.

One of the most discussed recent works in the area of power flow calculations is the holomorphic embedding load flow method HELMTM [Trias 2012]. This method can be potentially applied for voltage stability analysis. HELM is based on the following three ideas:

- Holomorphic embedding and analytical continuation (the idea is close to a parameter continuation approach, but with a complex continuation parameter);
- Hypercomplex reformulation (the idea is similar to a relaxation approach), and
- Nonlinear approximation of the solution function (unlike its linear approximation in the Newton-Raphson method).

Our analysis of reference [Trias 2012] shows, that despite the claim that HELM is a non-iterative algorithm, this method is actually an iterative procedure. Because HELM doubles the number of unknowns, it should be compared with iterative methods in term of its speed and accuracy.

One of the methods used in this project, the orbiting method, is motivated by [Hiskens 2001], and is further developed and implemented in this project for large-size power system models. In [Hiskens 2001], a combination of direct VSB equations and a predictor-corrector method was first proposed to explore the voltage stability boundary.

The main objective of this project is to develop non-iterative methods for accurate multi-dimensional voltage stability assessment in real time using full system models without simplifications. The analysis is limited to the static voltage stability problem. The efficiency of this method is demonstrated by revealing thin structures of the WECC system planning model in a multi-dimensional space, as well as performing the same analysis for some other real system models.

2.0 Some Characteristics of the Voltage Stability Boundary and Stress Directions

In this section we explore some important characteristics of the voltage stability boundary (VSB) and system stress trajectories used to locate a VSB point. We also analyze several criteria to quickly locate a VSB point on the stress trajectory without a need for time-consuming eigenvalue calculations and search procedures. This analysis helped to develop a new methodology reported in this document.

2.1 Stresses and the VSB in the Parameter and State Space

The study team conducted experiments to learn more about the relationship between the stress trajectories in coordinates of state (nodal voltages represented in Cartesian coordinates) and trajectories in coordinates of nodal injections. Stress trajectories are mapped into the state space of the system, which is defined in coordinates of nodal voltages. The same stress trajectories are also mapped into a parameter space, which is defined in coordinates of nodal active and reactive power injections. The transformation between these two spaces results in a mathematical approach for simplifying calculation of points on the VSB boundary.

2.1.1 Mapping of Stresses in the Parameter and State Space

The IEEE standard nine-bus test system model as provided in [Anderson 2003] was used in these experiments.

For a current operating point in parameter space (the space of nodal injections) Y_0 , we can find its corresponding boundary point Y^* , as shown in Figure 2. We divide the interval between Y_0 and Y^* into certain incremental steps. At each step, α increases to obtain a new operating point, i.e., $Y_i = Y_0 + \alpha_i \Delta Y$, ($i=1,2,\dots,n$). For each point Y_i , we can find the corresponding point X_i in the state space, as shown in Figure 3. The curve formed by points $X_1, \dots, X_5, \dots, X^*$ in Figure 3 represents a mapping of the straight stress trajectory $Y_i = Y_0 + \alpha_i \Delta Y$ into the state space.

Next, we divide the straight stress line between X_0 and X^* into certain incremental steps – see Figure 3. For each incremental step on this line, we calculate the corresponding points in the parameter space. By connecting these points with the base point Y_0 , we can form a family of nonlinear curves in parameter space (Figure 2). One expectation is that the straight lines in state space will be located on a plane, and the curves in parameter space will be located on a plane. These stress traversals in state space and parameter space demonstrate the quadratic properties of the power flow in rectangular coordinates.

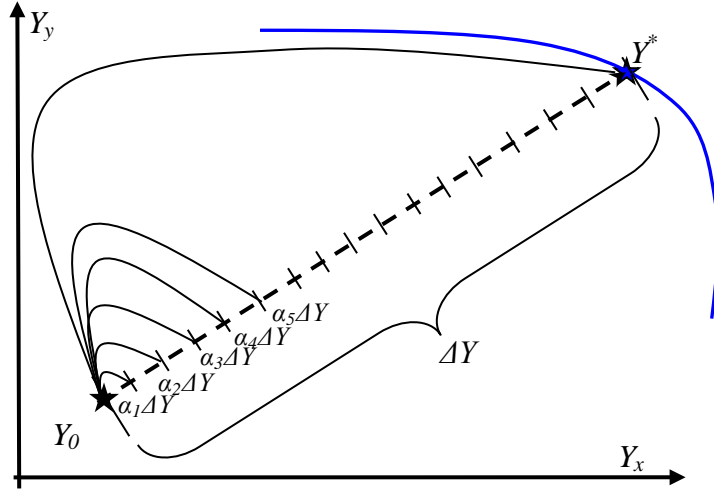


Figure 2. Stresses in parameter space, defined in coordinates of nodal active and reactive power injections.

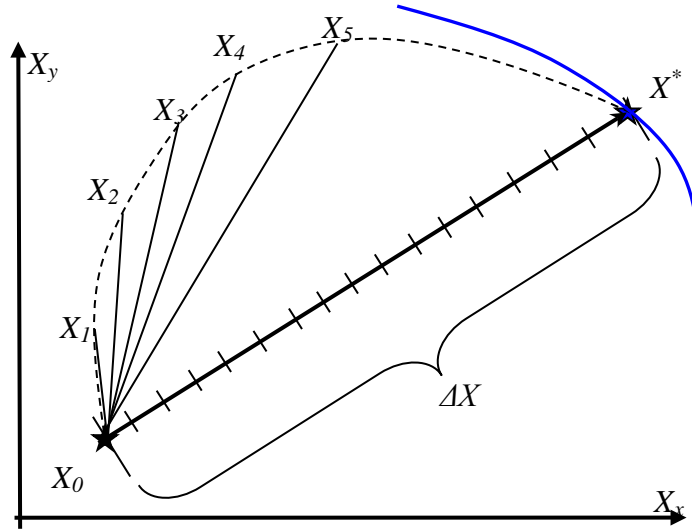


Figure 3. Stresses in the state space, defined in coordinates of nodal voltages..

A three-machine nine-bus power system test system¹ is used to plot the boundary point mapping in state and parameter space – see Figure 4. To determine the boundary point, the initial specified power flow values are changed linearly (here powers at buses 5, 6 and 8 respectively) and the power flow is run until the boundary point is reached in that direction and the corresponding voltage and specified power values are noted. Now, with this boundary point and the initial point, the straight-line distance between the two points is divided into various intermediate points (i.e., intermediate specified powers), and at each point the power flow equations are solved to get the corresponding voltages at those points.

¹ Parameters of the nine-bus standard test system model of the Institute of Electrical and Electronics Engineers (IEEE) can be found in [Anderson 2003].

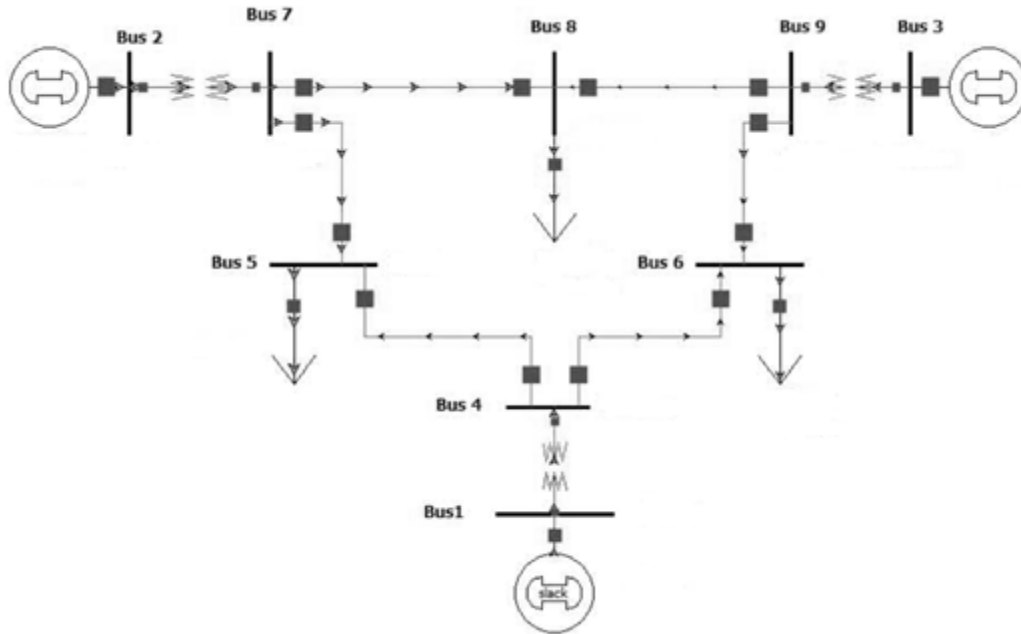


Figure 4. IEEE 9-bus test system.

Again, a straight line distance between the initial voltage point and each of the newly obtained intermediate voltage points is found, and is divided into various intermediate voltage points. Power flow equations are solved to get the corresponding calculated power at each of these intermediate voltage points and the paths traced by the calculated power in the parameter space and voltages in state-space are obtained.

The following figures (Figure 5 through Figure 10) show various views of stress trajectories in the parameter space. It is to be noted that in the simulations, the specified power at buses 5, 6 and 8 are changed and the power at other buses is kept constant.

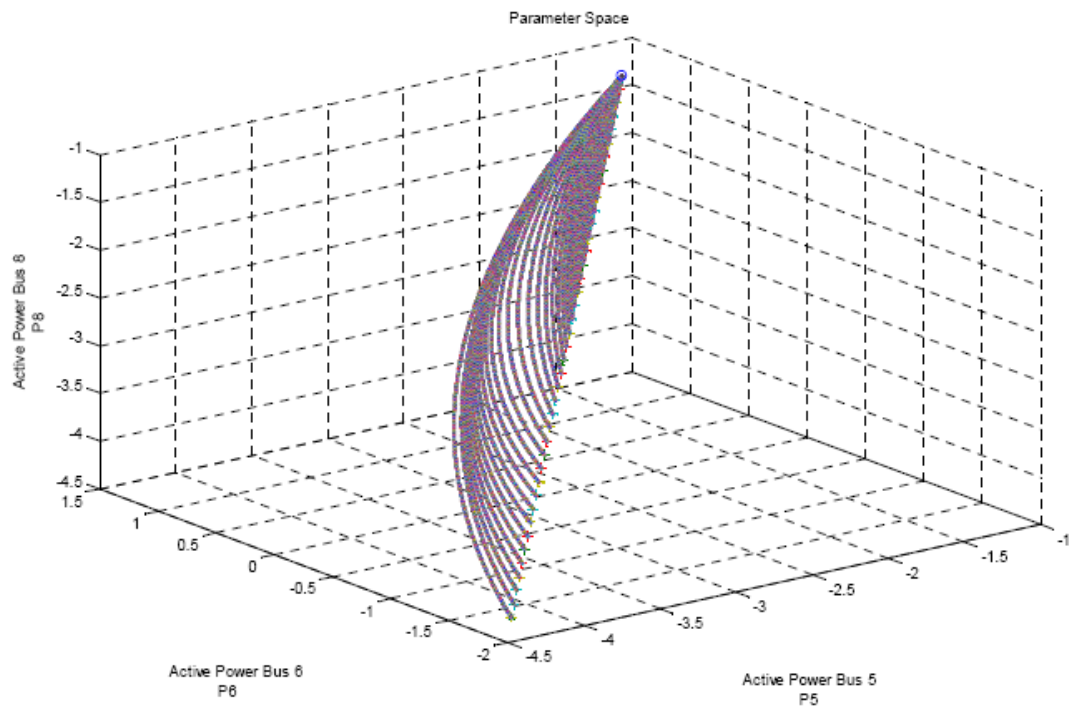


Figure 5. Stress trajectories of active powers at bus 5, 6 and 8 (view 1).

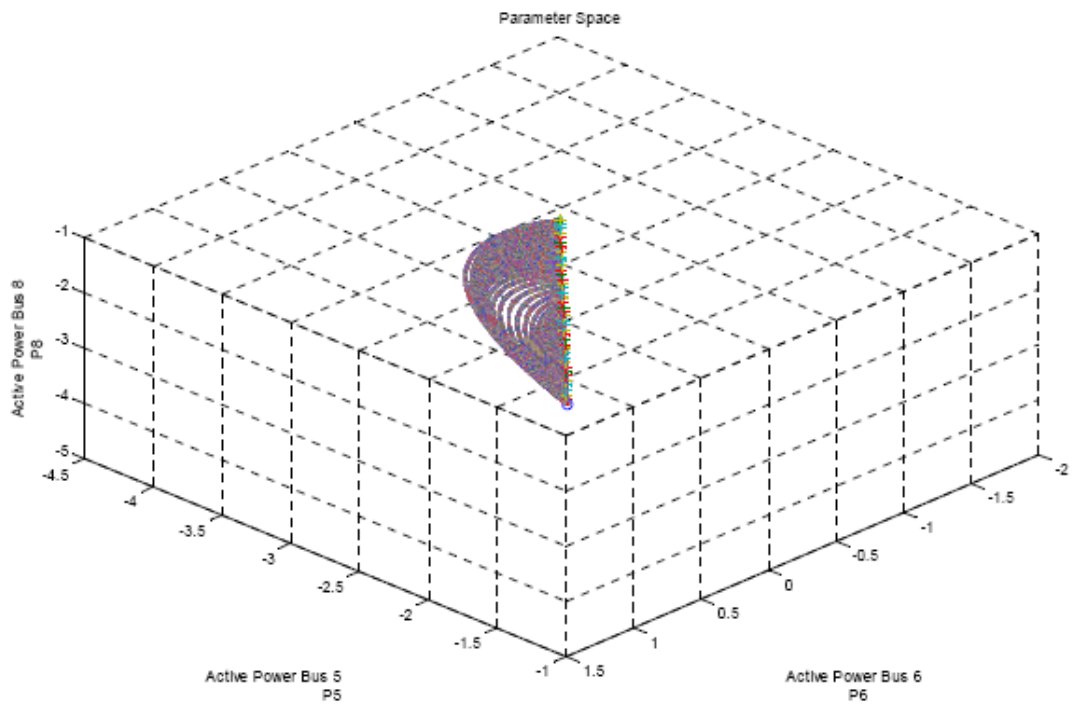


Figure 6. Stress trajectories of active powers at bus 5, 6 and 8 (view 2).

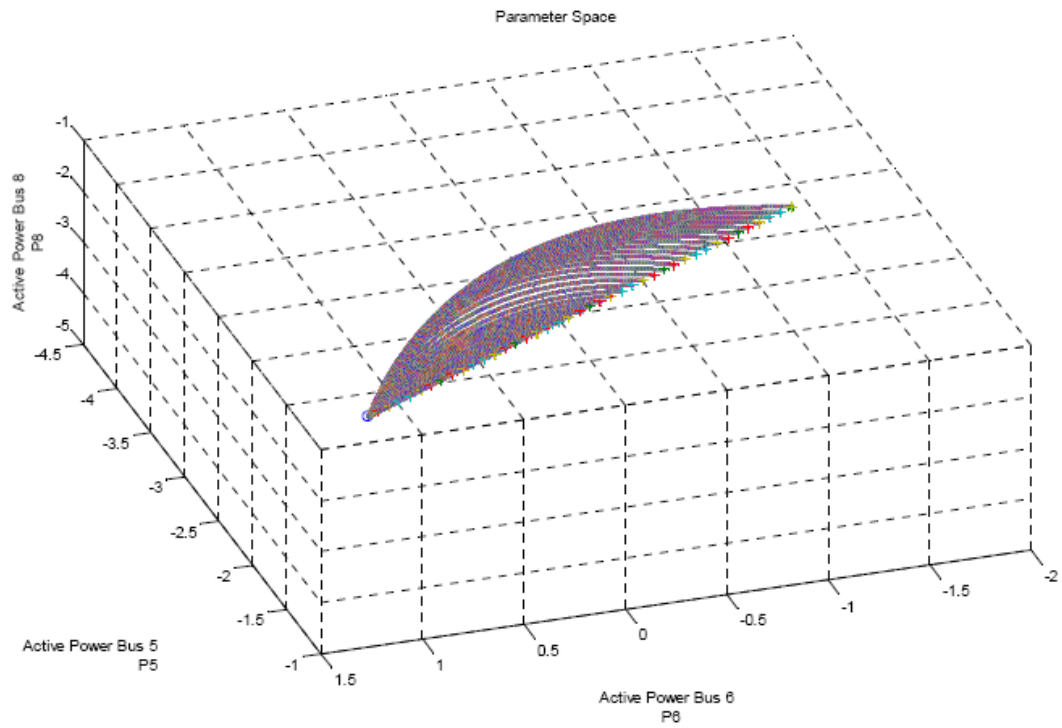


Figure 7. Stress trajectories of active powers at bus 5, 6 and 8 (view 3).

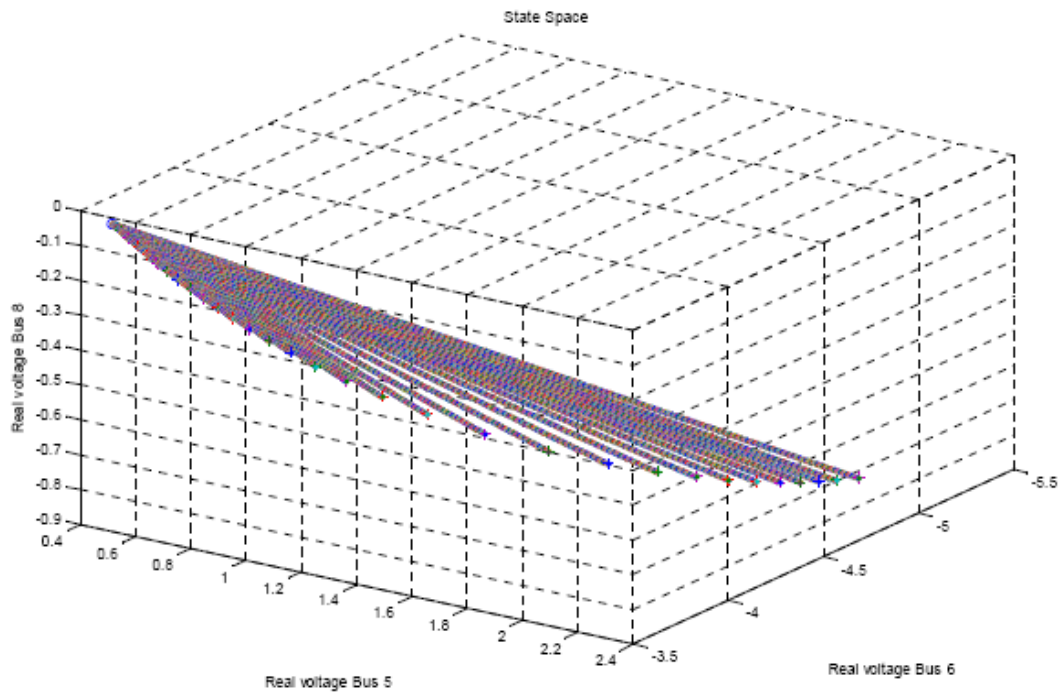


Figure 8. Stress trajectories of nodal voltages at bus 5, 6 and 8 (view 4).

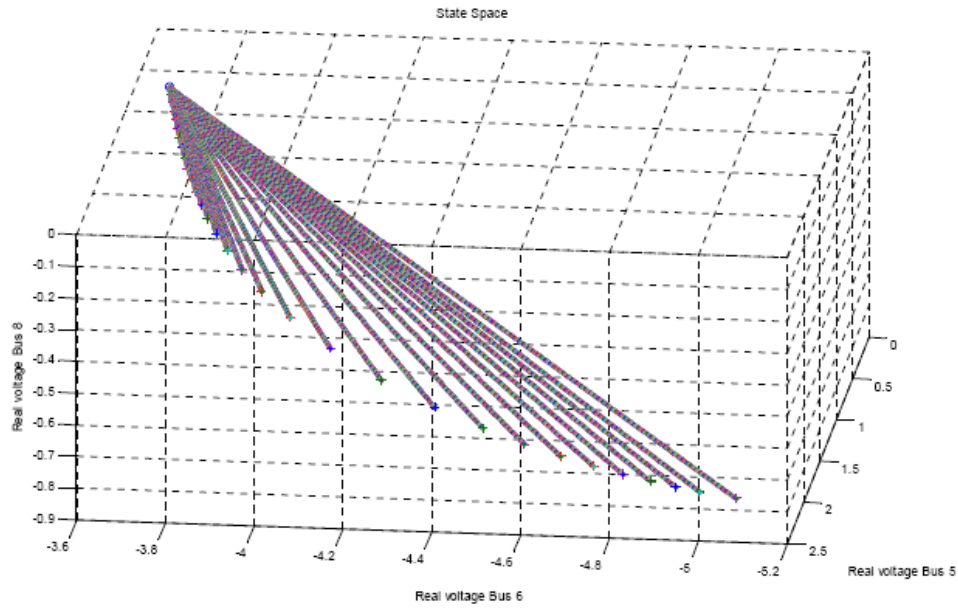


Figure 9. Stress trajectories of nodal voltages at bus 5, 6 and 8 (view 5).

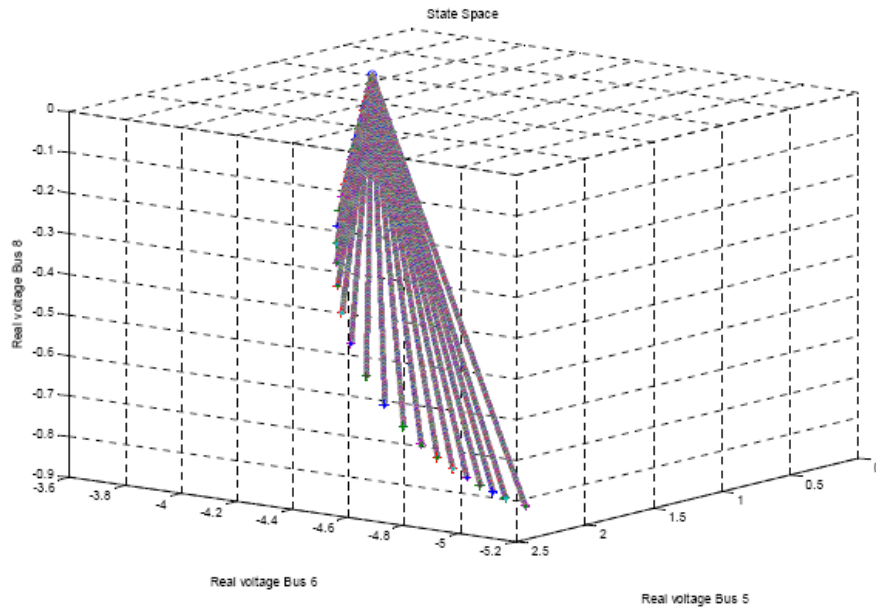


Figure 10. Stress trajectories of nodal voltages at bus 5, 6 and 8 (view 6).

It can be seen that starting from the initial specified power flow point, the path traced to any intermediate point in the parameter space is curved, in accordance with the quadratic nature of the equations. It is also worth noting that all the curves obtained in this parameter-space are co-planar i.e., the

surface of all the curves lies in a plane. Also, the path traced by the voltage points in the state-space is as expected.

2.2 Search for Indicators to Identify a VSB Point along a Stress Direction in the State Space

This section reports results of the effort to find a simple easy-to-compute criterion to detect VSB points along a straight line in coordinates of real and imaginary components of nodal voltages (a.k.a. as rectangular or Cartesian coordinates). Because the purpose is to dramatically speed up VSB calculations, more time-consuming criteria such as eigenvalues or singular values of the power flow Jacobian matrix are not considered.

The rationale behind this search is the following. In Cartesian coordinates, with some simple adjustments, the power flow mismatch functions become quadratic functions of state variables. Any straight line in the state space can be described parametrically as follows:

$$x(\alpha) = x_1 + \alpha(x_2 - x_1) \quad (2.1)$$

Where x_1 and x_2 are any two fixed distinct points in the state space and α is a scalar parameter. Any scalar quadratic function of x , $g(x)$, along the line in Equation (2.1) becomes a quadratic function of α , of the form

$$g(x(\alpha)) = g(\alpha) = a\alpha^2 + b\alpha + c \quad (2.2)$$

If we find a function g that becomes zero at VSB points, we can easily locate VSB points by solving this quadratic equation:

$$a\alpha^2 + b\alpha + c = 0 \rightarrow \alpha_{1,2} \quad (2.3)$$

Another question that we attempt to answer in this section is the correspondence between the stress in the state space given by Equation (2.1) and a realistic stress direction defined in coordinates of nodal active and reactive power injections $y = [P, Q]$, that is the correspondence between

$$y(\alpha) = y(x(\alpha)) \quad (2.4)$$

and

$$y(\beta) = y_0 + \beta\Delta y = F(x) \quad (2.5)$$

where y_0 is the base case power injection vector, Δy is the stress direction in coordinates of nodal powers, β is the stress parameter, and $F(x)$ is the power flow mismatch function.

The purpose of this analysis is that Equation (2.4) can deviate from Equation (2.5) and correspond to unfeasible power system states. We need to know to what extent and in which cases we can rely on the non-iterative VSB search using Equation (2.3).

During our effort to answer those two questions (i.e. finding function Equation (2.2) and finding the correspondence between Equation (2.4) and Equation (2.5)), we have found several interesting facts about the characteristics of VSB and stress processes in coordinates of state variables and power injections. We also tested several ideas including the ones that were previously proposed in the literature for locating VSB points.

2.3 Power Losses in the Vicinity of the VSB

This section examines the power losses in the vicinity of the VSB voltage stability boundary. It has been noticed that power losses rapidly increase when the system approaches voltage collapse. This fact was used to detect VSB points. In this section, we attempt to use power losses as function g in Equation (2.2).

Numerical experiments were performed for the 3-bus system shown in Figure 11. In this example, bus 1 is a PV bus with $|V_1| = 1.05$, bus 2 is a PQ bus, and bus 3 is the slack bus. The line impedance in per unit is $Z_{12} = 0.01 + j0.1$, $Z_{13} = 0.02 + j0.2$, and $Z_{23} = 0.02 + j0.2$. Losses were calculated and plotted for different feasible operating points in parameter space.

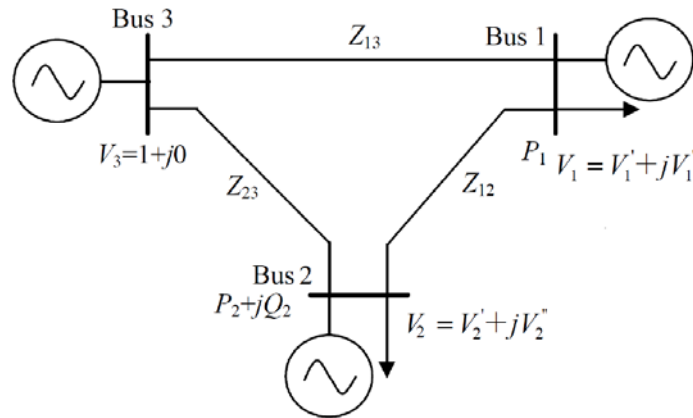


Figure 11. A 3-generator 3-bus system.

To select different feasible operating points, the voltage stability boundary must first be found. In this study, a direct method [Lu 1995] was used to obtain the static stability boundary as shown in Figure 12 and Figure 13.

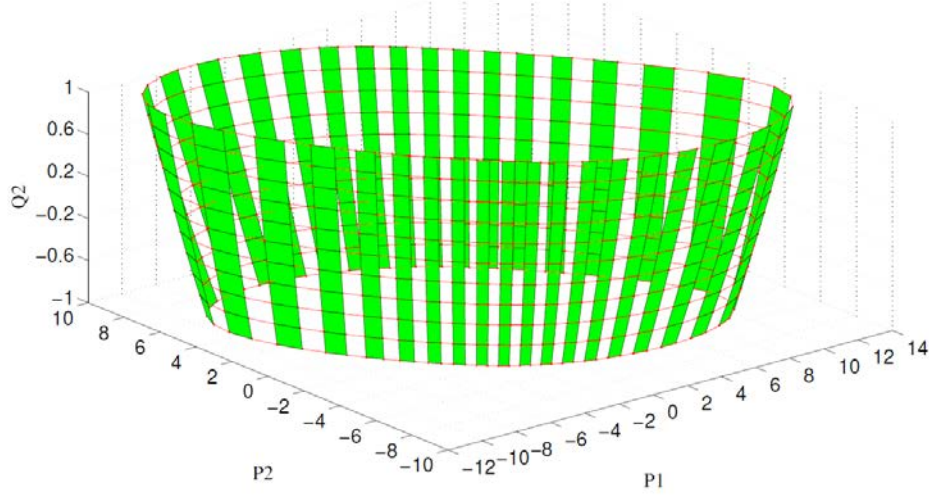


Figure 12. Static voltage stability boundary of 3-bus system in parameter space (p.u.).

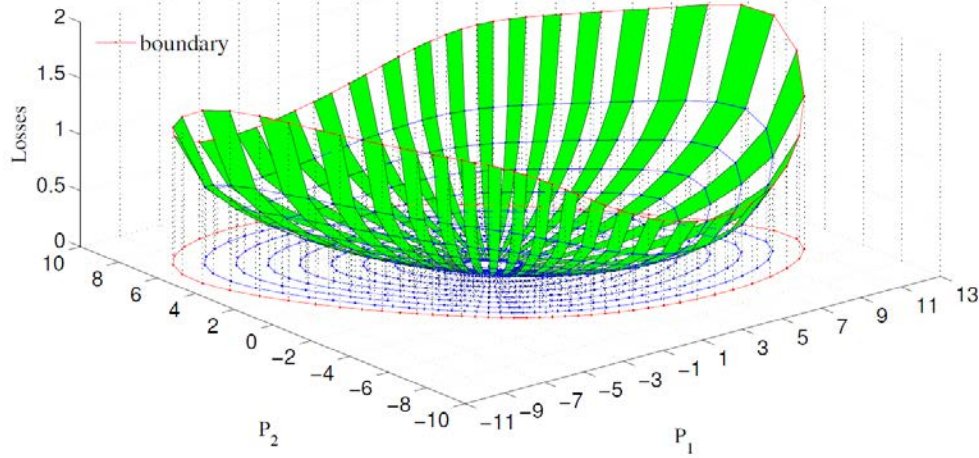


Figure 13. Losses at different operating points within the boundary (p.u.).

By fixing $Q_2 = 0.5$, the system losses are plotted for different P_1 and P_2 within the voltage stability boundary, as shown in Figure 14. The blue points are obtained by evenly interpolating nine points between the “center” operating point ($P_1 = 0$, $P_2 = 0$, and $Q_2 = 0.5$) and each boundary point. As an operating point moves a straight line from the “center” point to a boundary point, the following observation can be made:

- The system losses increase, and the boundary point corresponds to the maximum losses.
- The rate of change of losses also increase (as indicated by the density of contour lines), which becomes more clear in Figure 14.

It should be noted that the losses or the rate of change of losses is not necessarily larger at a boundary point than at any internal points. For example, the losses and rate of change of losses at point B is smaller than point A, although B is a boundary point and A is an internal point. In addition, when an operating point moves from A to B in a straight, the total system losses first decrease and then increase.

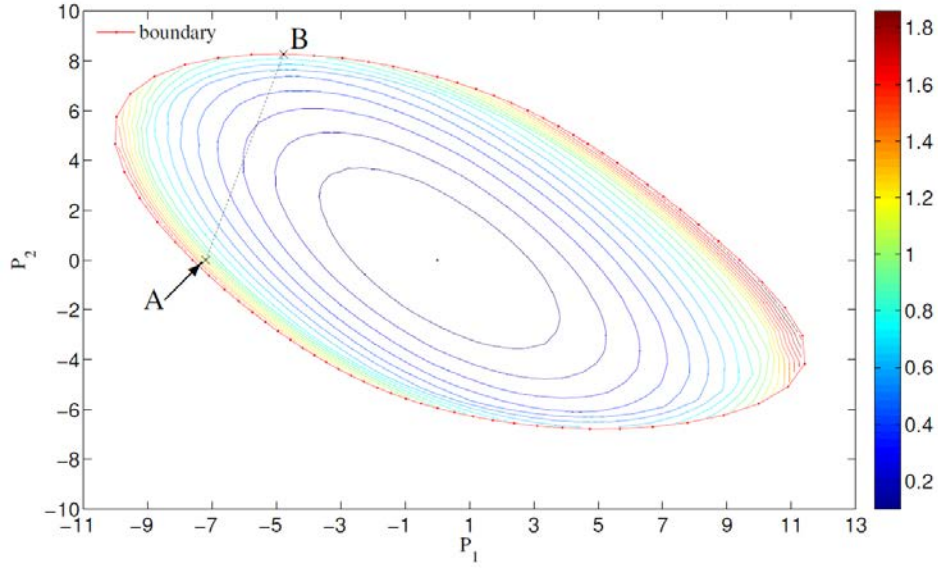


Figure 14. Contour plot of system losses with $Q_2 = 0.5$.

Conclusion:

The increase in power losses and the rate of this increase cannot be considered as a robust and accurate indicator of the VSB proximity. Based on the losses, we cannot confidently and accurately evaluate the current “distance to voltage collapse”.

2.4 Voltage Reduction in the Vicinity of VSB

Another hypothesis tested in this research was the use of nodal voltage magnitudes V_i as indicators of voltage collapse proximity. This idea is similar to P-V curve approach widely use in practical voltage stability studies. The g-function to be used in this case can be constructed as follows:

$$g(\alpha) = \sqrt{\sum_i V_i^2(\alpha)} \quad (2.6)$$

In the 3-bus system, the l^2 -norm of state variables is $(V_1'^2 + V_1''^2 + V_2'^2 + V_2''^2)^{1/2}$. Similarly as the calculation of losses in previous subsection, l^2 -norm of state variables is obtained, as plotted in Figure 15 and Figure 16. As an operating point moves in a straight line from the “center” point to a boundary point, one can observe:

- The norm of state variables decreases, and the boundary point is corresponding to the minimum norm of state variables. At the same time, it is not clear a priori what level of reduction of (2.9) signifies the voltage collapse point.
- The rate of change of norm (2.6) increases.

- Nevertheless, the rate of $g(\alpha)$ change in the vicinity of VSB depends upon the angle between the stress trajectory and VSB. If the trajectory is almost tangent to VSB, this influences the rate of change of (2.9).

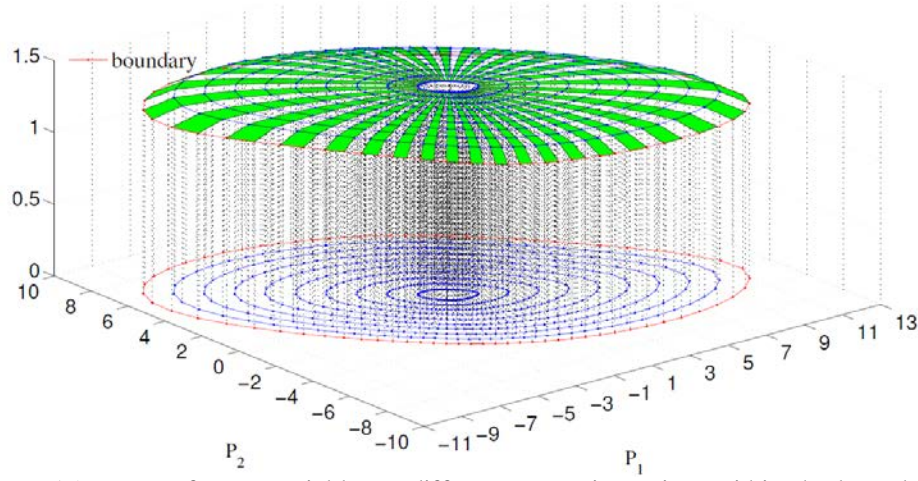


Figure 15. Norm of state variables at different operating points within the boundary.

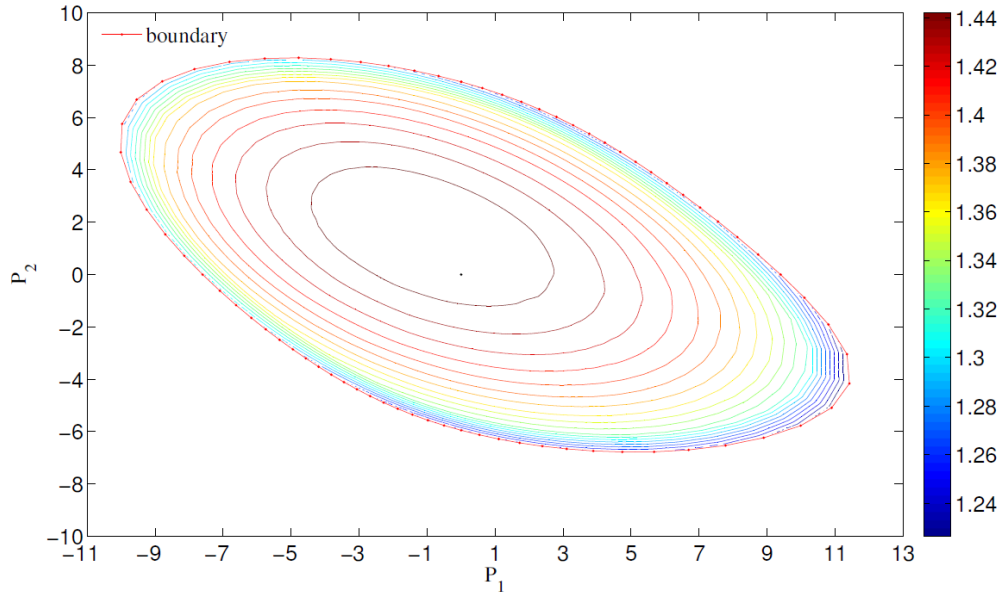


Figure 16. Contour of norm of state variables.

Conclusions:

The norm (2.9) cannot be used as an indicator of voltage collapse. Its rate of change can actually serve as such indicator, except for the case when the trajectory used to approach VSB is tangent to VSB. Unfortunately, in the XR method explained in Section 3.1, the trajectory used to approach the VSB is tangent to the VSB.

2.5 Mapping Stresses in the Parameter and State Spaces

Because of the quadratic properties of the power flow in rectangular coordinates, the mapping between the parameter space (nodal active and reactive power injections represented in Cartesian coordinates) of and state space (nodal voltages represented in Cartesian coordinates) is nonlinear. This section illustrates the mapping property using a two-bus system, as shown in Figure 17. In this example system, bus 1 is a PQ bus and bus 2 is the slack bus.

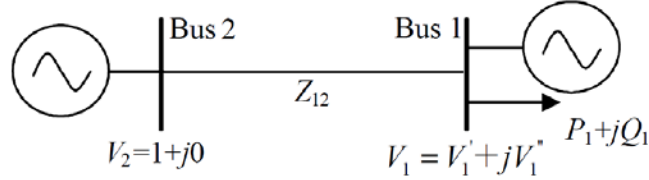


Figure 17. A 2-generator 2-bus system.

One can find the boundary in parameter space. Part of the boundary is shown in Figure 18, together with several example paths. The mapping of the boundary points and paths in Figure 18 to state space are shown in Figure 19. Generally, when a point moves along a straight line to the boundary point in parameter space (e.g., path 1 and 2), the corresponding trajectory in state space is a non-linear curve, which converges to the right eigenvector of the Jacobian matrix at the boundary point. When a point moves along a straight line to the boundary point and crosses the boundary in state space (e.g., path 3), the corresponding points in parameter space first moves toward the boundary, touches the boundary point, and then turns away from boundary to the feasible parameter region. In particular, when a point approaches the boundary a straight line along the right eigenvector of the Jacobian matrix at the boundary point in state space, the corresponding point in parameter space also moves along a straight line.

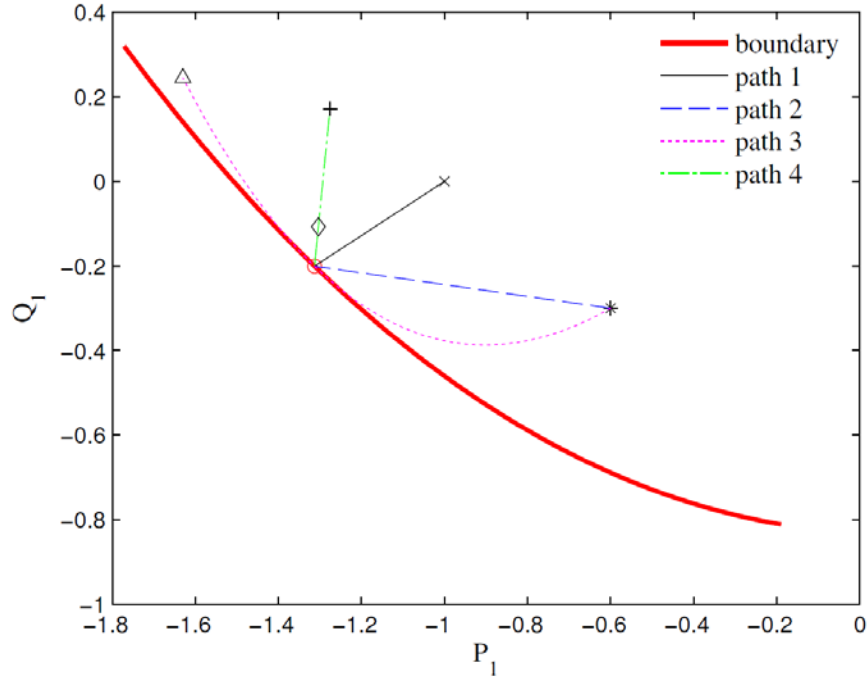


Figure 18. Part of the boundary and some paths in parameter space. (The mapping of Figure 19 to parameter space.)

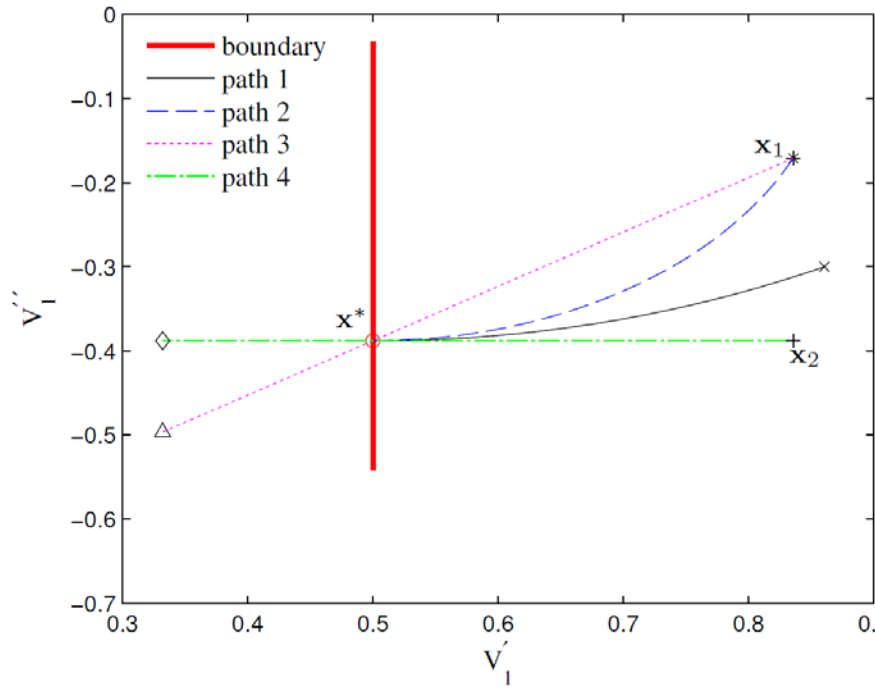


Figure 19. Part of the boundary and some paths in state space. (The mapping of Figure 18 to state space.)

As shown in the previous Section, the rate of change of the norm ($\|\Delta x\|/\|\Delta g\|$) increases as an operating point approaches the boundary and reaches the maximum at the boundary. Does this mean the boundary point is corresponding to the minimum norm of parameter change rate? If so, one can easily identify the boundary point by examining the $\|\Delta x\|/\|\Delta g\|$ once the direction in state space is given.

Let $x = [V_1' \ V_1'']^T$, which denotes any point in state space. Furthermore, let $\Delta x = (x^* - x_2) / 100$, where x^* and x_2 are shown in Figure 19. Any point that moves from x_2 to x^* along path 4 can be expressed as $x = x_2 + \alpha \Delta x$. Let $g(x) = [P_1(x) \ Q_1(x)]^T$. Since x_2 and Δx are fixed, g is a function of α . It should be noted that $\alpha = 100$ is corresponding to the boundary point. Let $\Delta g(\alpha) = g(\alpha + 1) - g(\alpha)$, with norms evaluated for different α , as plotted in Figure 20. For this particular path, where the operating point moves a straight line along the right eigenvector of Jacobian matrix at the boundary point, $\|\Delta g\|$ reaches the minimum value at the boundary point ($\alpha = 100$). However, this property does not hold in general. For example, when an operating point moves along path 3 in state space, the minimum $\|\Delta g\|$ does not occur at the boundary point, as shown in Figure 21.

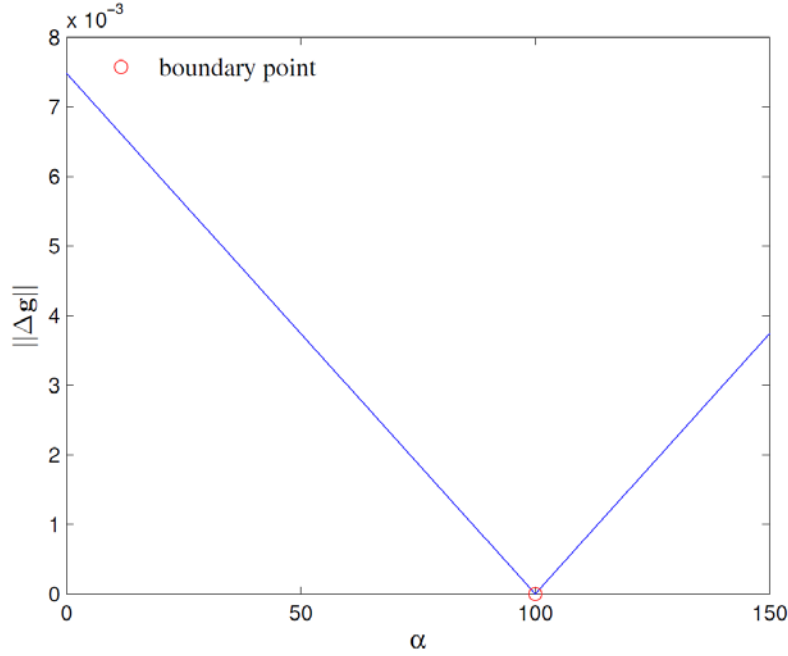


Figure 20. $\|\Delta g\|$ vs. α along path 4.

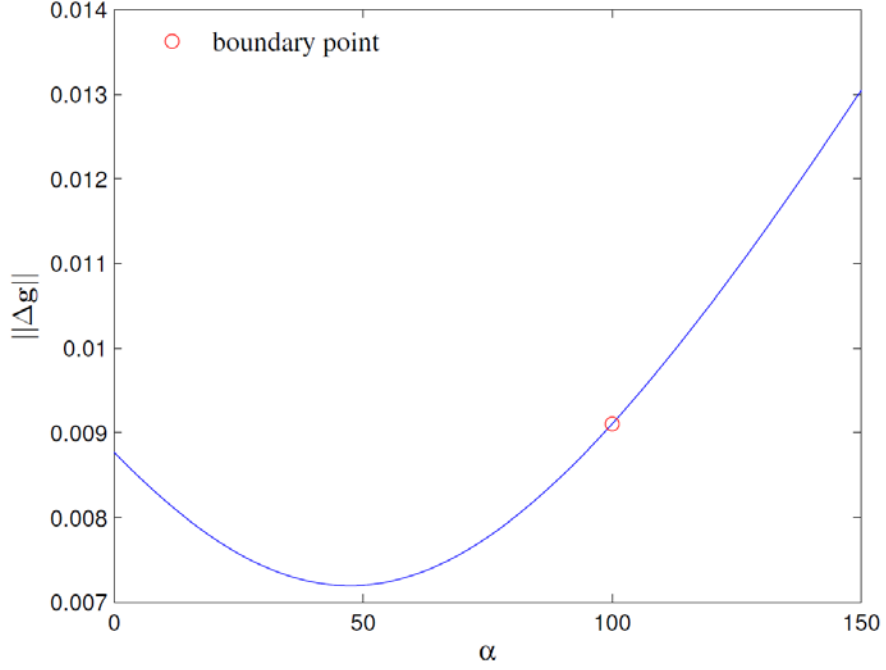


Figure 21. $\|\Delta g\|$ vs. α along path 3.

Contours of $\|x\|$ are shown in Figure 22, which explains what happens in Figure 20 and Figure 21. As explained in the previous section, $\|\Delta x\|/\|\Delta g\|$ at a boundary point is not necessarily bigger than any internal points. One can observe that $\|\Delta x\|/\|\Delta g\|$ monotonically decreases as an operating point approaches the boundary only when the point moves a straight line in parameter space. Because of the quadratic property, when a point moves a straight line in state space, its trajectory in parameter space becomes nonlinear, as shown in Figure 22, path 3. For path 3, $\|\Delta x\|/\|\Delta g\|$ reaches the minimum before arriving at the boundary.

Conclusion:

Criterion $\|\Delta x\|/\|\Delta g\|$ can serve as an indicator of VSB only if it is calculated along a trajectory in the parameter space, which intersects VSB. For a tangent trajectory, this criterion does not work.

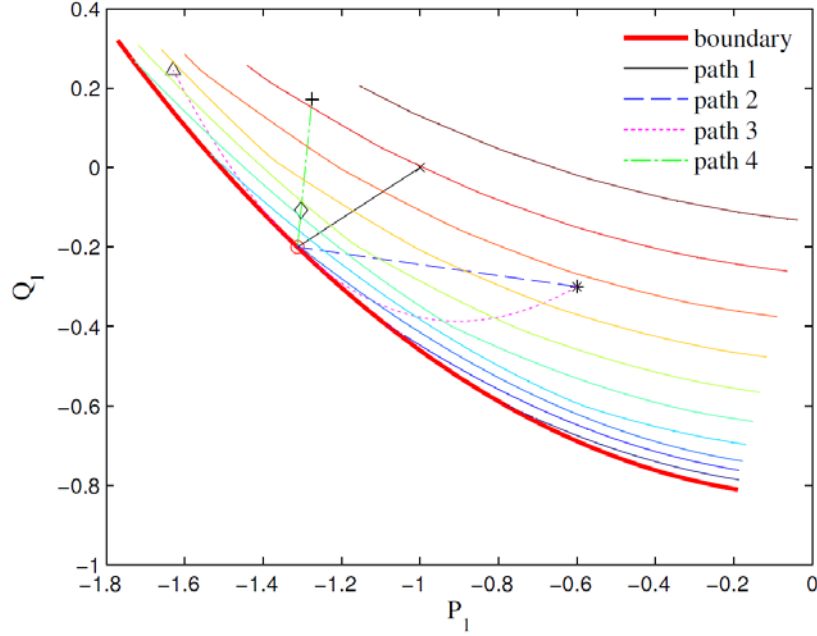


Figure 22. Contour of $\|x\|$ in parameter space.

2.6 Exploring Voltage Stability Boundary Points Using Stresses Defined in Parameter and State Spaces

Usually, it is desirable to find the boundary point along a specified stress direction in parameter space. However, the boundary point can be identified along any given direction in state space, as shown in (2.1), rather than in parameter space. Our question is about the opportunity to detect VSB points along the straight line in the state space, using $\|\Delta x\|/\|\Delta g\|$, as described in the previous section. It has been found, above, this criterion works for the paths that intersect the boundary, such as paths 1 and 2 in Figure 22. It does not work for trajectories that are tangent to VSB (path 3 in Figure 22). A relation is needed to map the straight trajectory in the state space to the trajectory in the parameter space.

Part of the stability boundary and some paths in parameter and state space are shown in Figure 23 and Figure 24, respectively, where the black circle represents the true boundary along the stress direction in the parameter space. As it is seen from the figures, straight stresses in state space correspond to nonlinear trajectories touching VSB in parameter space.

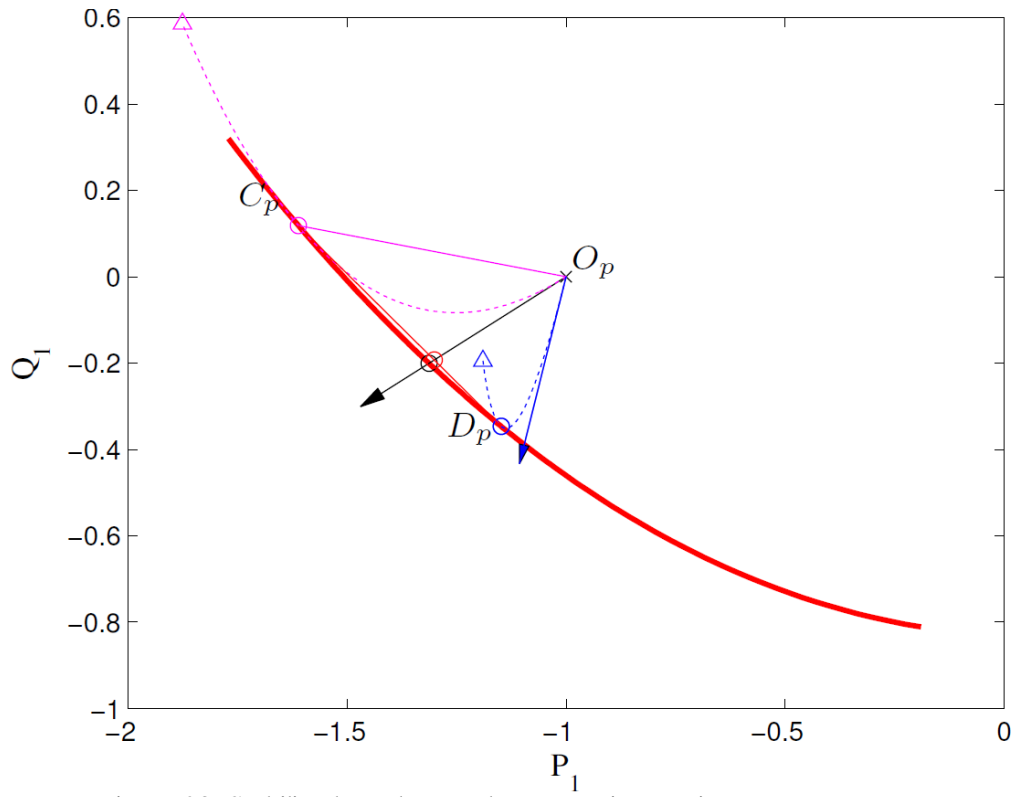


Figure 23. Stability boundary and stress trajectory in parameter space.

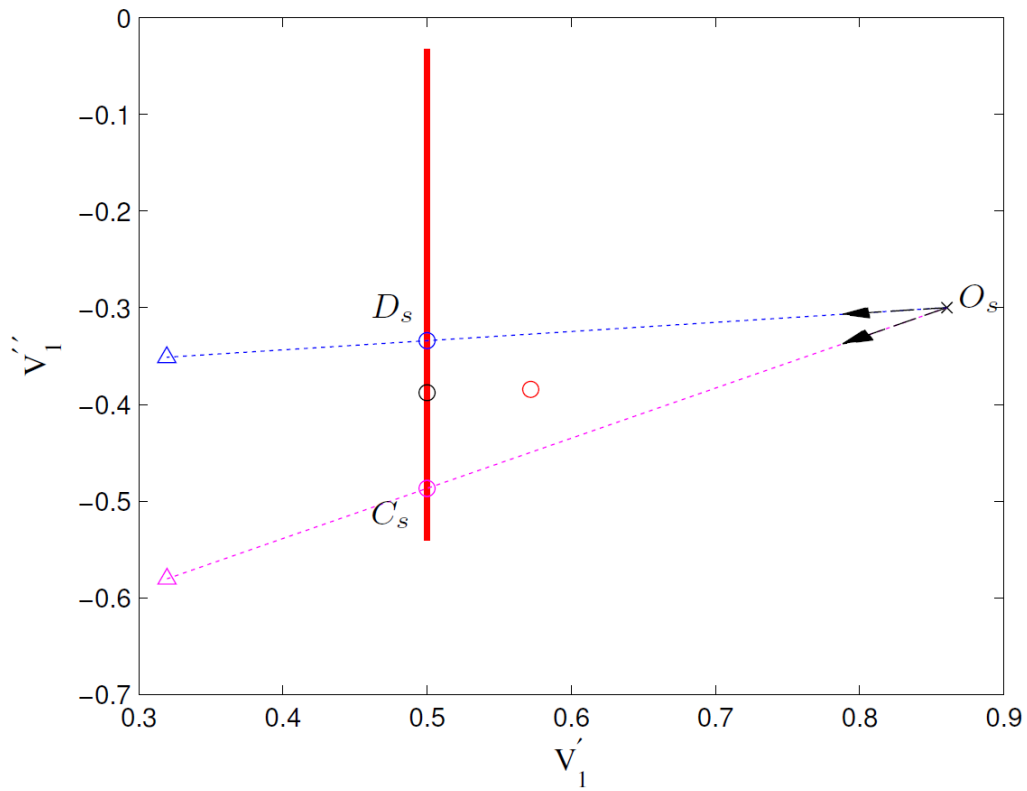


Figure 24. Stability boundary and stress trajectory in state space.

Conclusion:

Criterion $\|\Delta x\|/\|\Delta g\|$ cannot be used to detect VSB points along straight trajectories in the state space.

3.0 New Methodology for Exploring the Voltage Stability Boundary

In the previous section, we have examined several approaches to detect VSB points along a straight trajectory in coordinates of nodal voltages (in rectangular coordinates). Although our effort was not completely successful, it gave some very interesting and not very widely known insights on the characteristic of VSB, its vicinity and stress patterns used to reach it. In particular, in terms of potential practical applications, it is important to realize that the analyzed criteria are not providing good results.

This chapter discusses two innovative methods to explore the static voltage stability boundary in Cartesian coordinates for the bus voltages. The characteristics of the proposed non-iterative method make it novel and important both from theoretical and application point of views. The proposed methods are capable of exploring the steady state stability boundary much faster than current traditional methods. Thus, the proposed methods are promising for real-time application. The ultimate benefit obtained will be early detection of system instability trends and prevention of system blackouts. Based on these perspectives, the proposed methods are not only novel but important for power system stability analysis both theoretically and practically.

3.1 XR Methodology

The proposed XR method is based on the quadratic properties of power flow equations in Cartesian coordinates, i.e. so called x-ray theorem formulated in [Makarov 2000b]. The initial implementation and testing of the method was done in [Makarov 2000a]. In this project, we further developed and improved the method

3.1.1 Power Flow Equations in Cartesian Coordinates

Consider a power system with r PV buses indexed from 1 to r , $(n - r)$ PQ buses indexed from $r+1$ to n , and one slack bus indexed $(n+1)$. In Cartesian coordinates, the voltage at bus i can be expressed as

$$\begin{aligned}\hat{V}_i &= V'_i + jV''_i, \text{ for } i = 1, \dots, n \\ \hat{V}_{n+1} &= 1\end{aligned}\tag{3.1}$$

where V'_i and V''_i are real and imaginary parts of the complex number \hat{V}_i . Let Y denote the bus admittance matrix, with ik^{th} element

$$\hat{Y}_{ik} = G_{ik} + jB_{ik}\tag{3.2}$$

Therefore, the injection current at node $i=1, \dots, n$ is

$$\hat{I}_i = \sum_{k=1}^{n+1} \hat{Y}_{ik} \hat{V}_k = \sum_{k=1}^{n+1} (G_{ik} + jB_{ik})(V'_k + jV''_k)\tag{3.3}$$

The injection power at node i is calculated as

$$\hat{S}_i = P_i + jQ_i = \hat{V}_i \hat{I}_i^* \quad (3.4)$$

where $*$ designates the complex conjugate. The power flow equations—active power P for both PV and PQ buses and reactive power Q only for PQ buses—can be found by replacing \hat{V}_i and \hat{I}_i in (3.4) by (3.7) and (3.3), respectively, and separating the real and imaginary parts:

$$P_i = V_i' \sum_{k=1}^{n+1} (G_{ik} V_k' - B_{ik} V_k'') + V_i'' \sum_{k=1}^{n+1} (B_{ik} V_k' + G_{ik} V_k''), \quad (3.5)$$

for $i = 1, \dots, n$.

and

$$Q_i = -V_i' \sum_{k=1}^{n+1} (B_{ik} V_k' + G_{ik} V_k'') + V_i'' \sum_{k=1}^{n+1} (G_{ik} V_k' - B_{ik} V_k''), \quad (3.6)$$

for $i = r + 1, \dots, n$.

The square of the voltage magnitude can be expressed as

$$|V_i|^2 = (V_i')^2 + (V_i'')^2, \text{ for } i = 1, \dots, r. \quad (3.7)$$

3.1.2 Jacobian Matrix in Cartesian Coordinates

For state variables $\mathbf{x} = [V_1', \dots, V_n', V_1'', \dots, V_n'']$, for bus $1, 2, \dots, n$, let $\mathbf{P}(\mathbf{x})$ denote the vector of active power as function of \mathbf{x} as expressed in (3.5), $\mathbf{Q}(\mathbf{x})$ denote the vector of reactive power as function of \mathbf{x} as expressed in (3.6), $\mathbf{V}(\mathbf{x})$ denote the vector of the voltage magnitude square as a function of \mathbf{x} as expressed in (3.7). We define the function $F: \mathbf{R}^n \rightarrow \mathbf{R}^n$

$$\mathbf{F}(\mathbf{x}) = \begin{bmatrix} \Delta \mathbf{P}(\mathbf{x}) \\ \Delta \mathbf{Q}(\mathbf{x}) \\ \Delta \mathbf{V}(\mathbf{x}) \end{bmatrix} = \begin{bmatrix} \mathbf{P}(\mathbf{x}) - \mathbf{P}^s \\ \mathbf{Q}(\mathbf{x}) - \mathbf{Q}^s \\ \mathbf{V}(\mathbf{x}) - \mathbf{V}^s \end{bmatrix} \quad (3.8)$$

where \mathbf{P}^s is the vector of specified active power for PV and PQ buses, \mathbf{Q}^s is the vector of specified reactive power for PQ buses, \mathbf{V}^s is the vector of the square of specified voltage magnitude at PV buses. The Jacobian matrix of (3.8) is:

$$J(\mathbf{x}) = \begin{bmatrix} \frac{\partial P_1}{\partial V_1'} & \cdots & \frac{\partial P_1}{\partial V_n'} & \frac{\partial P_1}{\partial V_1''} & \cdots & \frac{\partial P_1}{\partial V_n''} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \frac{\partial P_n}{\partial V_1'} & \cdots & \frac{\partial P_n}{\partial V_n'} & \frac{\partial P_n}{\partial V_1''} & \cdots & \frac{\partial P_n}{\partial V_n''} \\ \frac{\partial Q_{r+1}}{\partial V_1'} & \cdots & \frac{\partial Q_{r+1}}{\partial V_n'} & \frac{\partial Q_{r+1}}{\partial V_1''} & \cdots & \frac{\partial Q_{r+1}}{\partial V_n''} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \frac{\partial Q_n}{\partial V_1'} & \cdots & \frac{\partial Q_n}{\partial V_n'} & \frac{\partial Q_n}{\partial V_1''} & \cdots & \frac{\partial Q_n}{\partial V_n''} \\ \frac{\partial V_1}{\partial V_1'} & \cdots & \frac{\partial V_1}{\partial V_n'} & \frac{\partial V_1}{\partial V_1''} & \cdots & \frac{\partial V_1}{\partial V_n''} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \frac{\partial V_r}{\partial V_1'} & \cdots & \frac{\partial V_r}{\partial V_n'} & \frac{\partial V_r}{\partial V_1''} & \cdots & \frac{\partial V_r}{\partial V_n''} \end{bmatrix} \quad (3.9)$$

which, because of the linearity of J in \mathbf{x} , can be written in the form

$$J(\mathbf{x}) = \sum_{i=1}^{2n} A_i x_i + B \quad (3.10)$$

where A_i and B denote constant matrices, x_i denotes the i^{th} element in \mathbf{x} .

3.1.3 Solving Singularity Equations

The key idea of the proposed non-iterative method is to find the stability boundary point in state space by solving a linear system problem. Let \mathbf{x}_0 denote the current operation point and $\Delta \mathbf{x} = \mathbf{x}_1 - \mathbf{x}_0$ denote the incremental change in the state space, where \mathbf{x}_1 is a point close to \mathbf{x}_0 . The Jacobian matrices at the two points described by two voltage vectors are

$$\begin{aligned} J(\mathbf{x}_0) &= \sum_{i=1}^{2n} A_i (x_0)_i + B \\ J(\mathbf{x}_1) &= \sum_{i=1}^{2n} A_i (x_1)_i + B \end{aligned} \quad (3.11)$$

Any point along the direction from \mathbf{x}_0 to \mathbf{x}_1 can be described as

$$\begin{aligned} \mathbf{x} &= \mathbf{x}_0 + \alpha(\mathbf{x}_1 - \mathbf{x}_0) \\ &= (1 - \alpha)\mathbf{x}_0 + \alpha\mathbf{x}_1, \end{aligned} \quad (3.12)$$

and its Jacobian matrix is

$$\begin{aligned}
J(\alpha) &= J(\mathbf{x}) = \sum_{i=1}^{2n} A_i \mathbf{x}_i + \mathbf{B} \\
&= \sum_{i=1}^{2n} A_i [(1-\alpha)(\mathbf{x}_0)_i + \alpha(\mathbf{x}_1)_i] + [(1-\alpha)\mathbf{B}] + \alpha\mathbf{B} \\
&= (1-\alpha)J(\mathbf{x}_0) + \alpha J(\mathbf{x}_1) \\
&\triangleq (1-\alpha)J_0 + \alpha J_1,
\end{aligned} \tag{3.13}$$

i.e., the Jacobian matrix at any point in space can be constructed as a linear combination of two known matrices. The possible boundary point corresponds to the real smallest α that make $J(\alpha)$ singular.

3.1.3.1 Eigenvalue-based detection of singularity

Let \mathbf{v} be a vector in the null space of $J(\alpha)$, i.e. $J(\alpha)\mathbf{v}=0$. Then $(1-\alpha)J_0\mathbf{v} + \alpha J_1\mathbf{v} = 0$, or with $\lambda = \alpha / (\alpha - 1)$, the boundary point can be found by solving a generalized eigenvalue problem

$$J_0\mathbf{v} = \lambda J_1\mathbf{v} \quad)$$

where \mathbf{v} denotes the generalized eigenvector. When the interested boundary point is on the direction from \mathbf{x}_0 to \mathbf{x}_1 and beyond \mathbf{x}_1 , we have $\alpha > 1$. Therefore, the smallest α is corresponding to the largest positive λ . For eigenvalue λ , the boundary can be calculated as

$$\mathbf{x} = \mathbf{x}_0 + \frac{\lambda_L}{\lambda_L - 1}(\mathbf{x}_1 - \mathbf{x}_0) \quad)$$

3.1.3.2 New method for detecting the singularity (Improved XR method)

The new method expands the ideas first reported in [Makarov 2000a]. Let $\mathbf{Y} = [\mathbf{P}^T \mathbf{Q}^T \mathbf{V}^T]^T$ denotes vector of parameters, which can be expressed as a function of α . Using a Taylor expansion and recognizing the quadratic form of \mathbf{P} , \mathbf{Q} , \mathbf{V} in the voltage, we have the expansion

$$\mathbf{Y}(\mathbf{x}) = \mathbf{Y}(\alpha) = \mathbf{Y}(\mathbf{x}_0) + \alpha J(\mathbf{x}_0)\Delta\mathbf{x} + \frac{\alpha^2}{2}\mathbf{w}(\Delta\mathbf{x}) \tag{3.16}$$

The difference between $\mathbf{Y}(\mathbf{x})$ and $\mathbf{Y}(\mathbf{x}_0)$ is

$$\Delta\mathbf{Y}(\alpha) = \alpha J(\mathbf{x}_0)\Delta\mathbf{x} + \frac{\alpha^2}{2}\mathbf{w}(\Delta\mathbf{x}) \tag{3.17}$$

It can be shown that

$$\mathbf{w}(\Delta\mathbf{x}) = [J(\mathbf{x}_1) - J(\mathbf{x}_0)]\Delta\mathbf{x} \tag{3.18}$$

Replacing $\mathbf{w}(\Delta\mathbf{x})$ by (3.17) in (3.18) yields, after manipulations,

$$\Delta\mathbf{Y}(\alpha) = [\alpha(1 - \frac{\alpha}{2})J(\mathbf{x}_0) + \frac{\alpha^2}{2}J(\mathbf{x}_1)]\Delta\mathbf{x} \tag{3.19}$$

Define $\mathbf{R}(\alpha) \triangleq J^{-1}(\alpha)\Delta\mathbf{Y}(\alpha)$, which converges to the right eigenvector as a point approaches the boundary [Makarov 2000b]. Of course, $\mathbf{R}(\alpha)$ is well-defined only as long as we are only near the

boundary, not exactly on it, since an exact coalescing on the boundary will imply that $\mathbf{J}^{-1}(\alpha)$ is not defined. Approaching the boundary, the conditioning of this Jacobian approaches infinity. This property is what we use to detect the singularity. As we approach the boundary, $\mathbf{R}(\alpha)$ will be strong in the null-space vector, and will have a large norm. In fact, defining $\mathbf{R}(\alpha)$ for any ΔY should be sufficient just as long as the projection of it to the null-vector is large. Furthermore, let

$$\tilde{\mathbf{R}}(\alpha) = \frac{\mathbf{R}(\alpha)}{\|\mathbf{R}(\alpha)\|} \quad (3.20)$$

It is expected that $\|\mathbf{J}(\alpha)\tilde{\mathbf{R}}(\alpha)\|$ achieves its minimum value at the boundary point.

4.0 Exploring Abnormal Singularities within the WECC System Stability Region

To demonstrate the effectiveness of our new method, we applied it to search for potential singular holes within a VSR of a real system. The importance of these holes is hard to underestimate. If they are real, a system with apparently sufficient stability margin may become suddenly unstable.

In this research, significant peculiarities of the boundary configuration were identified, including its non-convexity, discontinuity, branching and internal singularities (“holes”) that were not known before and could not be found by traditional methods. Upon discovery of the “holes” (discontinuity in the feasible solution region), we have given our attention on this unanticipated problem. “Holes” are unstable operation points within a stability boundary. Operating a system close to a “hole” will cause an unexpected voltage collapse. The analyses showed that the multi-singularity behaviors were real in the state space. We proposed a new fast approach to explore the voltage stability boundary (VSB) and revealed its detailed multidimensional structure. Several interesting cases were found:

- Internal “holes” inside of the voltage stability (VS) region. The singularities were verified with the help of eigenvalue analysis
- Branching of the VSB
- Open parts of VSB
- Non-convexity of power flow feasibility boundary

We proposed new methods to explore static voltage stability conditions in Cartesian coordinates instead of polar coordinates, where the singularity problem is formulated as a scalar equation with respect to the stress parameter. The methodology for transformation from polar to rectangular coordinates is presented in Appendix C. The proposed methods allow quick exploration of static stability conditions in the state space.

4.1 Details of the Approach to Explore the Voltage Stability Boundary

In order to reveal more VSB details, one typical and one atypical case were selected to investigate the boundary configuration. The selected typical case corresponds to increased generation in Northwest and Sierra zones, and decreased generation in Fortis BC and El Paso zones. The selected atypical case was obtained by increasing generation in Northwest and Sierra zones, and decreasing generation in Fortis BC and Colorado zones. The VSB is then explored on a two-dimensional plane formulated by 1) the base operating point (x_0), 2) the boundary point in the typical case (x_b^A), and 3) the boundary point in the atypical case (x_b^B), as illustrated in Figure 25. Multiple stress directions are repeated between the selected two stress directions. Herein, the linear property of Jacobian matrix is utilized again following the procedure below:

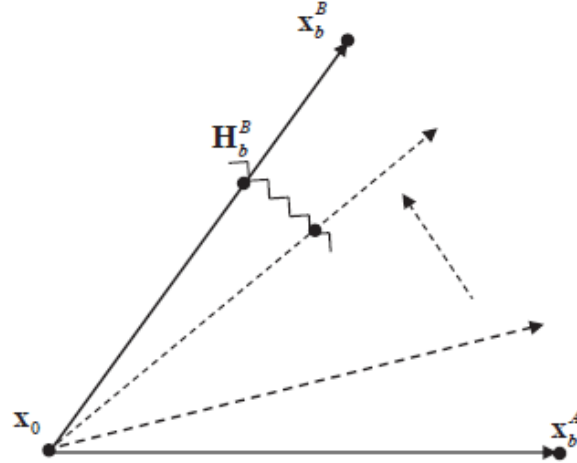


Figure 25. Exploration of voltage stability boundary on a two-dimensional plane.

The proposed methodology is based on the properties of quadratic algebraic equations in Cartesian coordinates and it is detailed in the following steps:

Step 1:

Power flow and Jacobian matrix are derived in Cartesian coordinates. The Jacobian matrix can be represented as a linear function of real and imaginary components of voltage vectors. Let \mathbf{X}_b^A be the voltage vector at a boundary point at normal case and \mathbf{X}_b^B be the voltage vector at a boundary point at exceptional case, both of which are different from the current operating point. \mathbf{X}_0 refers to the voltage vector of the current operating point. \mathbf{X}_1^A is any voltage vector along the direction of \mathbf{X}_0 and \mathbf{X}_b^A . \mathbf{X}_1^B is any voltage vector along the direction of \mathbf{X}_0 and \mathbf{X}_b^B . $\Delta\mathbf{X}_B$ refers to the distance between \mathbf{X}_b^B and \mathbf{X}_0 . Any voltage vector (\mathbf{X}_1^A) along the direction of \mathbf{X}_0 and \mathbf{X}_b^A can be described as

$$\mathbf{X}_1^A = [\mathbf{X}_b^A + (\alpha - 1)\mathbf{X}_0] / \alpha \quad (4.1)$$

and distance between \mathbf{X}_0 and \mathbf{X}_b^A is

$$\Delta\mathbf{X}_b^A = (\mathbf{X}_b^A - \mathbf{X}_0) / \alpha \quad (4.2)$$

Here α is a factor referring to the proportion of the distance from current operating point (\mathbf{X}_0) to any point \mathbf{X} with respect to the distance from current operating point to the given operating point. Similarly, any voltage vector (\mathbf{X}_1^B) along the direction of \mathbf{X}_0 and \mathbf{X}_b^B can be described as

$$\mathbf{X}_1^B = [\mathbf{X}_b^B + (\alpha - 1)\mathbf{X}_0] / \alpha \quad (4.3)$$

and

$$\Delta\mathbf{X}_b^B = (\mathbf{X}_b^B - \mathbf{X}_0) / \alpha \quad (4.4)$$

The Jacobian matrix at voltage \mathbf{X}_1^A and \mathbf{X}_1^B can be given as

$$\mathbf{J}_1^A = [\mathbf{J}_b^A + (\alpha - 1)\mathbf{J}_0] / \alpha \quad (4.5)$$

$$\mathbf{J}_1^B = [\mathbf{J}_b^B + (\alpha - 1)\mathbf{J}_0] / \alpha \quad (4.6)$$

where \mathbf{J}_b^A , \mathbf{J}_b^B , and \mathbf{J}_0 are Jacobian matrices at voltage \mathbf{X}_b^A , \mathbf{X}_b^B , and \mathbf{X}_0 , respectively. \mathbf{J}_1^A and \mathbf{J}_1^B are any Jacobian matrices along the direction of \mathbf{J}_0 and \mathbf{J}_b^A , \mathbf{J}_0 and \mathbf{J}_b^B , respectively.

In this numerical simulations, \mathbf{X}_0 , \mathbf{X}_b^A , \mathbf{X}_b^B , and corresponding Jacobian matrices are obtained from PowerWorld® (PW) simulations. Any voltage vector between \mathbf{X}_1^A and \mathbf{X}_1^B can be estimated as

$$\mathbf{X}_i^{new} = (1 - k)\mathbf{X}_1^A + k\mathbf{X}_1^B \quad (4.7)$$

Similarly, the Jacobian matrix at voltage \mathbf{X}_i^{new} is

$$\mathbf{J}_i^{new} = (1 - k)\mathbf{J}_1^A + k\mathbf{J}_1^B \quad (4.8)$$

and Jacobian matrix at any α can be calculated as

$$\mathbf{J}_{new} = \alpha\mathbf{J}_i^{new} + (1 - \alpha)\mathbf{J}_0 \quad (4.9)$$

where k is a factor that varies from 0 to 1 with a step of 0.01.

$$\Delta\mathbf{X}_i^{new} = (1 - k)\Delta\mathbf{X}_b^A + k\Delta\mathbf{X}_b^B \quad (4.10)$$

and angle between $\Delta\mathbf{X}_i^{new}$ and $\Delta\mathbf{X}_b^A$ is estimated as

$$\theta = \left(\frac{\Delta\mathbf{X}_i^{new} \Delta\mathbf{X}_b^A}{\|\Delta\mathbf{X}_i^{new} \Delta\mathbf{X}_b^A\|} \right) \quad (4.11)$$

Step 2:

Stability limits are estimated by examining the singularity of the matrix in state space. For any \mathbf{x} expressed in Equation (4.1) or (4.3), the corresponding $\mathbf{Y} = [\mathbf{P}^T \mathbf{Q}^T \mathbf{V}^T]^T$ (vector of parameters) can be expressed as a function of α :

$$\mathbf{Y}(\mathbf{x}) = \mathbf{Y}(\alpha) = \mathbf{Y}(\mathbf{x}_0) + \alpha\mathbf{J}_0\Delta\mathbf{x} + \frac{\alpha^2}{2}\mathbf{w}(\Delta\mathbf{x}) \quad (4.12)$$

where $0.5\mathbf{w}(\Delta\mathbf{x})$ is the quadratic term of the Taylor series expansion. The difference between $\mathbf{Y}(\mathbf{x})$ and $\mathbf{Y}(\mathbf{x}_0)$ is

$$\Delta\mathbf{Y}(\alpha) = \alpha\mathbf{J}_0\Delta\mathbf{X}_i^{new} + \frac{\alpha^2}{2}\mathbf{w}(\Delta\mathbf{x}) \quad (4.13)$$

It can also be shown that

$$\mathbf{w}(\Delta\mathbf{x}) = [\mathbf{J}_i^{new} - \mathbf{J}_0]\Delta\mathbf{X}_i^{new} \quad (4.14)$$

$$\Delta Y(\alpha) = \left[\alpha \left(1 - \frac{\alpha}{2}\right) \mathbf{J}_0 + \frac{\alpha^2}{2} \mathbf{J}_i^{new} \right] \Delta \mathbf{X}_i^{new} \quad (4.15)$$

Define $\mathbf{R}(\alpha) \triangleq \mathbf{J}_{new}^{-1}(\alpha) \Delta Y(\alpha)$ which converges to the right eigenvector as a point approaches to boundary. Furthermore, let

$$\tilde{\mathbf{R}}(\alpha) = \frac{\mathbf{R}(\alpha)}{\|\mathbf{R}(\alpha)\|} \quad (4.16)$$

It is expected that $\|\mathbf{J}_{new} \tilde{\mathbf{R}}(\alpha)\|$ achieves its minimum value at the boundary point. The following steps are involved in margin boundary tracing for WECC 2020 system model:

- Boundary tracing program starts at current operating point for the first boundary point. It is assumed that $\alpha = 100$;
- Determine \mathbf{J}_i^{new} using Equation (4.18) at each k , which varies from 0 to 1 with a step size of 0.01;
- Compute $\|\mathbf{J}_{new} \tilde{\mathbf{R}}(\alpha)\|$ using equations (4.9) and (4.16) for all values of α at each k and record α value(s) at which $\|\mathbf{J}_{new} \tilde{\mathbf{R}}(\alpha)\|$ achieves its minimum value
- Calculate θ using Equation (4.11) and compute x and y components using α obtained from step 3 and θ , and plot in two dimensional plane

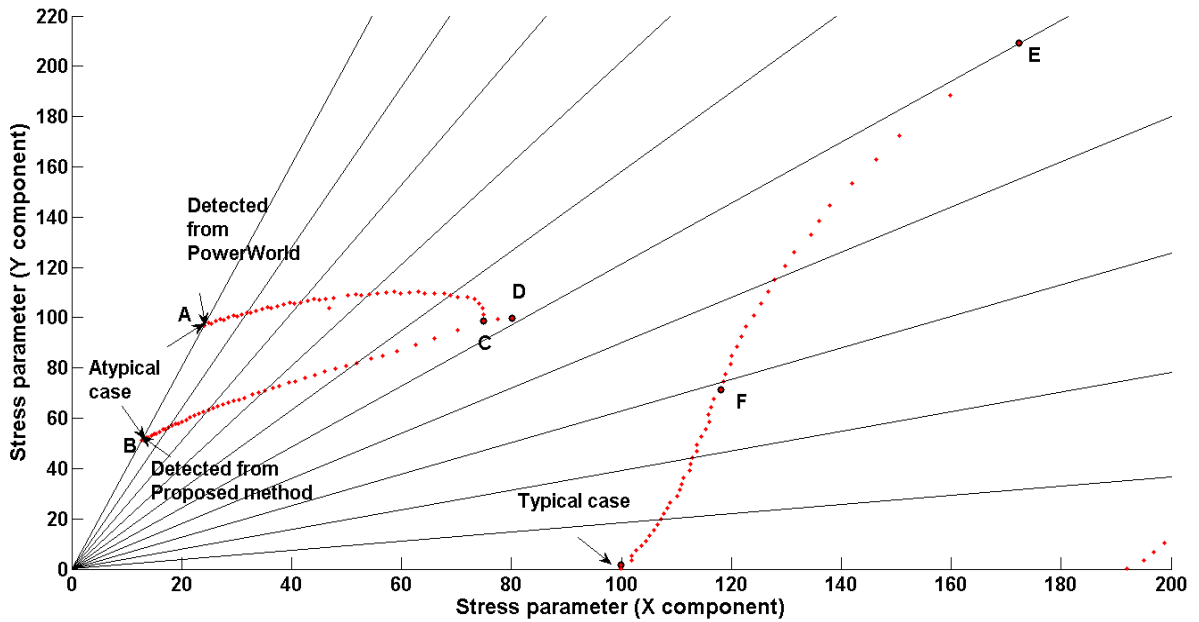


Figure 26. Stability boundary for Western Interconnection system.

With the new method, several unusual cases were found: Internal “holes” inside of the VS region (B-C line in Figure 26). They seem to be layers of closely located multiple internal singularities of the power flow problem. The singularities were verified with the help of eigenvalue analysis. Experiments were conducted with the accuracy of Jacobian matrix to eliminate the impact of inaccuracies on results.

- Branching of the VSB (point C).

- Open parts of VSB (between points D-E).
- Non-convexity of power flow VSB (between points E-F).

Further work in this research includes an enhancement of the method allowing remaining within feasible manifolds located in the state space.

5.0 Overall Implementation of Non-iterative Method

This section explains the implementation of the proposed non-iterative method. This project is based on a solid foundation previously created by PNNL for the CEC -funded project conducted by CERTS, EPG, PNNL, and University of Wisconsin – Madison project “Real-Time Voltage Security Assessment (RTVSA)” [Parashar 2007a], [Parashar 2007b], [Parashar 2007a], [Parashar 2008b]. In this project, we successfully combined the orbiting method with the XR method, modified the XR method to speed up the entire procedure, and perfected the code to handle large real system models. We propose a combination of several effective numerical algorithms to explore the voltage stability boundary in a fast and accurate way. We apply the following methods – Figure 27:

1. Continuation power flow (CPF). This algorithm is used to find an initial point(s) of VSB in certain specified stress direction (approximately). The CPF also produces an approximation of the right eigenvector nullifying the power flow Jacobian matrix on VSB. The right eigenvector is needed to start the direct VSB method in the next step.
2. Direct method to refine the position of VSB point along the stress direction.
3. Inverse iteration method to find the left eigenvector nullifying the power flow Jacobian matrix on the VSB. This vector is used later for finding hyperplanes approximating the VSB.
4. Direct orbiting method (OM) to follow VSB along a specified 2-D slice, starting from the initial stress point. Each step of OM produces a new VSB point.
5. Our new method based on the x-ray theorem [Makarov 2000b] initially implemented in [Makarov 2000a]. The XR method is used to refine the bus type changes during orbiting, and also to restart the orbiting process if the OM diverges due to peculiarities of the VSB shape. The XR method description can be found in section 3.1.3.2.
6. Methods to obtain linear approximations of the boundary (a.k.a. hyperplanes or linear inequalities) – (HPA).
7. High-order numerical methods (HO) for the future implementation.

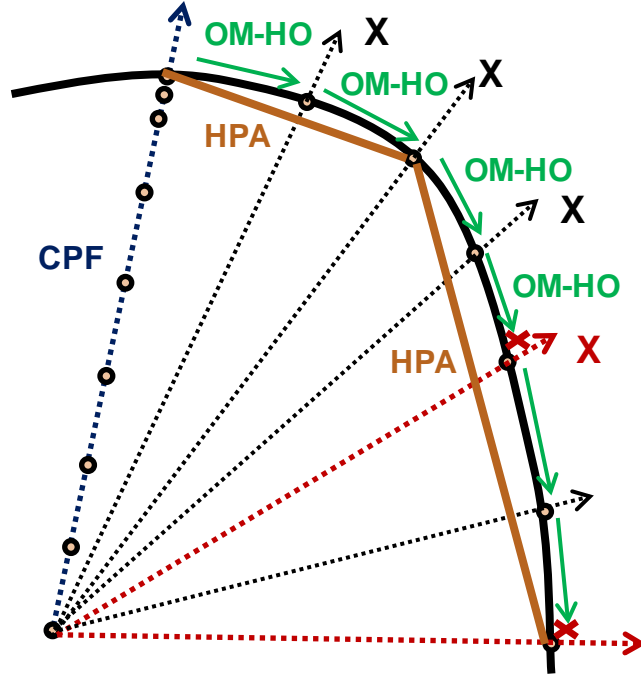


Figure 27. Illustration of the combined method of exploring the VSB.

A flowchart summarizing the key elements of the overall algorithm is provided in Figure 28.

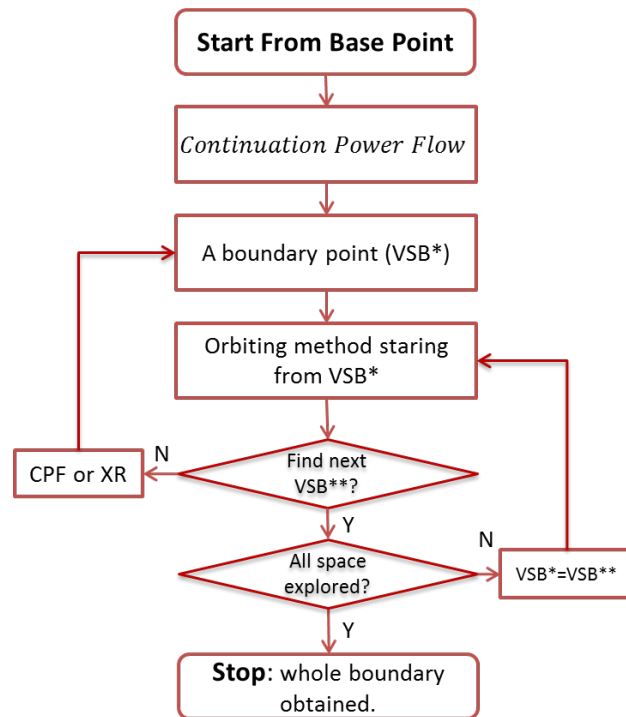


Figure 28. Basic flowchart for the combined method of exploring the VSB.

5.1 Details of Implementation

In this section, we provide a summary of implementation details and stress their importance for the overall performance of our new approach. The material provided in this section can be also partly found in [Parashar 2007a], [Parashar 2007b], [Parashar 2007a], [Parashar 2008b]. It is described here for completeness. The new elements of the methodology developed in this project are also provided below.

5.1.1 Continuation Power Flow (CPF)

CPF is a well-known iterative method with reliable convergence. In our implementation, the CPF algorithm is used to find an initial point(s) of VSB in certain specified stress direction (approximately). This is a multi-step stressing procedure; each step consists of predictor and corrector algorithms – see Figure 29, which will be briefly described below. The CPF is also used in this work to produce an approximation of the right eigenvector nullifying the power flow Jacobian matrix in a VSB point. The right eigenvector is needed to start direct VSB method in the next step.

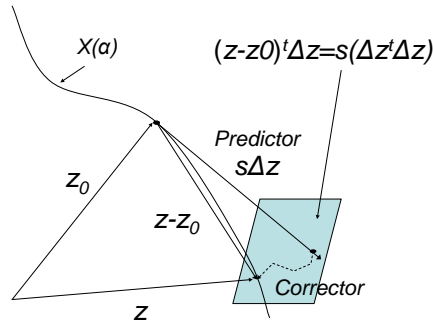


Figure 29. Continuation power flow step: predictor and corrector.

5.1.1.1 Theoretical background of the parameter continuation methods

Suppose that we have two different problems with the same state variables and the same number of equations: $\Phi_1(x) = 0$ and $\Phi_2(x) = 0$ ¹. Suppose that we know a solution of the first problem $x = x_1 : \Phi_1(x_1) = 0$. Based on this known solution, we will be looking for a solution of the second problem by solving the following parametric equation:

$$\Phi(x, \alpha) = (1 - \alpha)\Phi_1(x) + \alpha\Phi_2(x) \equiv 0 \quad (5.1)$$

¹ Here $\Phi(x)$ is considered as a “generic” function. Depending on the application, it can be the power flow mismatch function $F(x)$, or some other functions described in the sequel.

At $\alpha = 0$, this problem is solved by substituting $x = x_1$. At $\alpha = 1$, this problem coincides with the second problem under consideration, which solution x_2 is unknown. We want to trace a smooth transition from problem 1 to problem 2 by changing α . In this case, the full differential of (5.1) must be equal to zero:

$$d\Phi = \frac{\partial \Phi}{\partial x} dx + \frac{d\Phi}{d\alpha} d\alpha = 0 \rightarrow$$

$$\frac{\partial \Phi}{\partial x} \frac{dx}{d\alpha} + \frac{d\Phi}{d\alpha} = \left[(1-\alpha) \frac{\partial \Phi_1(x)}{\partial x} + \alpha \frac{\partial \Phi_2(x)}{\partial x} \right] \frac{dx}{d\alpha} + [\Phi_2(x) - \Phi_1(x)] = 0 \quad (5.2)$$

From (5.2), if matrix $\left[(1-\alpha) \frac{\partial \Phi_1(x)}{\partial x} + \alpha \frac{\partial \Phi_2(x)}{\partial x} \right]$ is not singular,

$$\frac{dx}{d\alpha} = - \left[(1-\alpha) \frac{\partial \Phi_1(x)}{\partial x} + \alpha \frac{\partial \Phi_2(x)}{\partial x} \right]^{-1} [\Phi_2(x) - \Phi_1(x)] \quad (5.3)$$

Differential Equation (5.3) describes how a solution of (5.1) moves while parameter α is changing – see Figure 30.

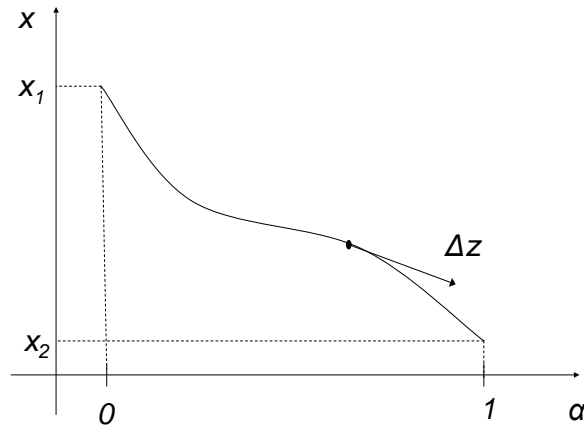


Figure 30. Solution of parametric differential equation for a scalar case.

In some cases, there is no need to add an artificial parameter, as it is done in (5.1), because such a parameter is already a part of the original equation. For instance, if the original system has the following form:

$$\Phi(x) + \beta D = 0 \quad (5.4)$$

where D is a fixed vector and β plays the same role as α in (5.1), differential parametric equation becomes

$$\frac{dx}{d\beta} = - \left[\frac{\partial \Phi(x)}{\partial x} \right]^{-1} D \quad (5.5)$$

By considering (5.4), it is easy to see that solutions of (5.5), $x(\beta): \Phi(x) + \beta D = 0$, move along the straight line βD in the space of mismatches $\Phi(x)$ - see Figure 31.

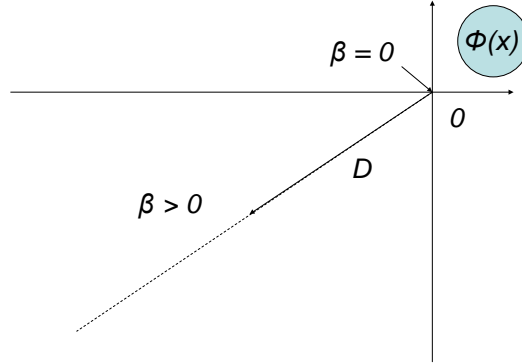


Figure 31. Parametric motion in the space of mismatches.

In finite increments, Equation (5.5) can be rewritten as:

$$\left[\frac{\partial \Phi(x)}{\partial x} \right] \Delta x + D \cdot \Delta \beta = 0 \quad (5.6)$$

and

$$\Delta x = - \left[\frac{\partial \Phi(x)}{\partial x} \right]^{-1} D \cdot \Delta \beta \quad (5.7)$$

5.1.1.2 Parameter continuation method

Equation (5.2) can be expressed in finite differences as follows:

$$\Delta \Phi = \frac{\partial \Phi}{\partial x} \Delta x + \frac{d\Phi}{d\alpha} \Delta \alpha = 0 \quad (5.8)$$

Predictor step

By combining Δx and $\Delta \alpha$ in one vector $\Delta z = [\Delta x^t, \Delta \alpha]^t$, one can obtain the predictor equation used in parameter continuation algorithm:

$$\begin{cases} \frac{\partial \Phi}{\partial z} \Delta z = 0 \\ e_i^t \Delta z = 1 \end{cases} \Leftrightarrow \begin{bmatrix} \frac{\partial \Phi}{\partial z} \\ \text{-----} \\ e_i^t \end{bmatrix} \Delta z = \begin{bmatrix} 0 \\ \text{-----} \\ 1 \end{bmatrix} \quad (5.9)$$

where e_i^t is a zero vector with only i -th element equal to 1. This is the expression which is used in the algorithm.

Note that the generic Equation (5.9) holds also for the form (5.6). The only difference is that, for (5.6), the increment vector $\Delta z = [\Delta x^t, \Delta \beta]^t$.

Solution Δz of (5.9) is a vector tangent to the parametric line $x(\alpha)$ - see Figure 30.

Corrector step

Because of linearization used in (5.9), the parametric curve cannot be followed precisely, and there is a need to execute one more step at each iteration – a corrector step to eliminate the observed deviation – see Figure 29.

The corrector step consists in solving the corrector equation:

$$\begin{cases} \Phi(z) = 0 \\ (z - z_0)^t \Delta z = s(\Delta z^t \Delta z) \end{cases} \quad (5.10)$$

where s is the predictor step, and Δz is the solution of (5.9).

System (5.10) helps to find an intersection of the parametric curve $x(\alpha)$ and a hyperplane that is orthogonal to the predictor vector Δz and includes the predicted point $z_0 + s\Delta z$.

System (5.10) can be solved using the Newton-Raphson method. To provide its reliable convergence, an adequate step correction s must be applied at the predictor stage. By selecting the right value of s , one can make the predicted point $z_0 + s\Delta z$, which is used as an initial guess for solving (5.10), as close to the corrected point z as required for reliable convergence of the Newton-Raphson method.

5.1.2 Direct Method to Refine the Position of VSB Point along the Stress Direction

The exact location of the VSB point can be calculated by solving the following system:

$$\begin{cases} F(x) + \beta D = 0 \\ J(x)R = 0 \\ R^T R = 1 \end{cases} \quad (5.11)$$

where $J(x)$ is the power flow Jacobian matrix and R is the right eigenvector corresponding to the zero eigenvalue of $J(x)$. The loading direction D is exactly the same as the one used in the predictor-corrector procedure.

To solve (5.11), it is important to select good initial guesses for unknown parameters x , β , and R . For x and β , the values produced by the predictor-corrector method near the VSB point are used. For R , a good initial guess would be the increment of state variables Δx near the bifurcation point normalized with respect to the last condition in (5.11) (for example, the difference between two successive iterations close to the VSB point):

$$R_0 \approx \frac{x_i - x_{i-1}}{\|x_i - x_{i-1}\|}, \quad x_i \rightarrow \text{VSB point} \quad (5.12)$$

This recommendation is based on the fact that the trajectory of the state variables tends toward the right eigenvector R in the vicinity of the VSB point.

5.1.3 Inverse Iterations

The inverse iteration method is applied to find the left eigenvector nullifying the power flow Jacobian matrix on the VSB. This vector is used later for finding hyperplanes approximating the VSB.

Suppose J is the Jacobian matrix having an eigenvalue close to zero. Given the initial guess of the left eigenvector: L_0 and $\lambda = 0.000001$, the inverse iteration shown in (5.13) is used to calculate the left eigenvector of J (L_i is normalized in each iteration).

$$\begin{aligned} (J - \lambda I)^T x &= L_i \\ v_{i+1} &= \frac{x}{\|x\|} \end{aligned} \quad (5.13)$$

Here $\|x\|$ represents the norm of x .

5.1.4 Direct Orbiting Method to Follow the VSB

[Hiskens 2001] motivated the orbiting method further developed and implemented in this project for large-size power system models. The direct orbiting method (OM) follows the VSB along a specified 2-D slice, starting from the initial stress point – see Figure 27. Each step of OM produces a new VSB point.

In this procedure, instead of the traditional stressed power flow equation, $F(x) + \beta D = 0$, we will use a modified equation

$$\Phi(z) = \begin{cases} F(x) + \beta_* D + \gamma \cdot e_i + \eta \cdot e_j = 0 \\ J'(x)L = 0 \\ L'L = 1 \end{cases} \quad (5.14)$$

In this equation, parameter $\beta = \beta_*$ is fixed, and two additional unknown parameters γ and η are added. By varying γ and η , one can explore the entire plane (p_i, p_j) . Set (5.14) has one unknown more than the number of equations. Unknown variables z are x, L, γ and η .

System (5.14) can be used to organize the prediction-correction process. For example, The predictor equation becomes:

$$\underbrace{\begin{bmatrix} J(x) & 0 & e_i & e_j \\ \frac{\partial}{\partial x}[J'(x)L] & J'(x) & 0 & 0 \\ 0 & L' & 0 & 0 \\ \hline & & & \end{bmatrix}}_{\underbrace{\begin{bmatrix} \frac{\partial \Phi}{\partial z} \\ \hline E_r' \end{bmatrix}}_{E_r'}} \times \underbrace{\begin{bmatrix} \Delta x \\ \Delta L \\ \Delta \gamma \\ \Delta \eta \end{bmatrix}}_{\Delta z} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad (5.15)$$

where E_r is the extended unit vector.

The corrector equation is given as:

$$\begin{cases} \Phi(z) = 0 \\ (z - z_0)' \Delta z = s(\Delta z' \Delta z) \end{cases} \quad (5.16)$$

The initial guesses of the unknowns and parameters when the first slice is explored should be taken as follows:

D – fixed and equal to the initial loading direction;

β – fixed and equal to the value achieved during the initial predictor-corrector procedure;

x, L – variable, initially set equal to the values achieved during the initial predictor-corrector procedure;
 γ, η – variable, initially zeros.

Note that after each predictor-corrector step, we get a point of the power flow feasibility boundary and the left eigenvector, which is all that is needed for the hyperplane approximation and the “sliced bread” procedure.

5.1.5 Implementation of XR Method

The XR method theory can be found in section 3.1.3.2. In this section, we provide a numerical procedure for solving the XR method equation; additionally, we consider an XR-based procedure to restart the direct orbiting method in case of its divergence.

5.1.5.1 Arnoldi algorithm

The generalized eigenvalue problem $\mathbf{J}_0 \mathbf{v} = \lambda \mathbf{J}_1 \mathbf{v}$ (see section 3.1.1.1) can be solved for the largest real eigenvalue using a shift-and-invert implicit restarted Arnoldi method. In particular, the generalized eigenvalue problem requires the action $(\mathbf{J}_0 - \sigma \mathbf{J}_1)^{-1} \mathbf{J}_1$, where σ is a shift which should be close to the searched eigenvalues of interest. A repeated application of this action requires an LU decomposition of the sparse matrix $(\mathbf{J}_0 - \sigma \mathbf{J}_1)$ once and then a linear solve for each application of the action. This is the inner computation of the outer implicit restart Arnoldi iteration, which determines the approximate generalized eigenvalues close to the shift over a sufficiently large Krylov subspace (a subspace of size 20 was found to be sufficient in the experiments). Since we can assume \mathbf{J}_1 is close to \mathbf{J}_0 , we take a shift close to 1. In our work, we have implemented and tested this procedure. At the same time, we have discovered a different approach to solving the XR problem, which does not require solving the eigenvalue problem. Consequently, this procedure is faster than Arnoldi algorithm.

5.1.5.2 New search procedures for solving XR problem

In section 3.1.1.2, we proposed an expression helping to locate a closest singularity point in a straight line in coordinates of nodal voltages (in rectangular coordinates): $\|\mathbf{J}(\alpha)\tilde{\mathbf{R}}(\alpha)\| \rightarrow \min$. By solving this problem and finding its minimum solution, we can locate the desired point.

Figure 32 shows that $\|\mathbf{J}(\alpha)\tilde{\mathbf{R}}(\alpha)\|$ achieves its minimum value at the boundary point. Therefore, the problem of finding the closest points where the Jacobian matrix becomes singular becomes finding the smallest α that minimizes the “distance” function $z(\alpha) = \|\mathbf{J}(\alpha)\mathbf{R}(\alpha)\|$. Three methods can be used to search the smallest α : 1) the steepest descent approach, 2) the simplex algorithm, and 3) a sampling approach.

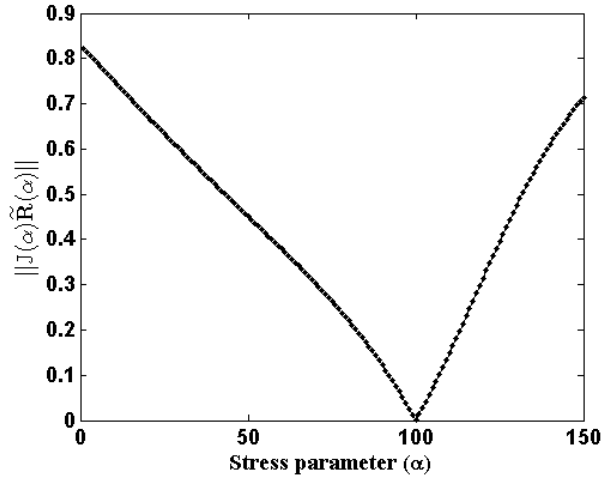


Figure 32. Exploration of distance function (100% denoted a VSB point).

The steepest descent approach

In the steepest descent method (also called the gradient descent method), we start at a point and, as many times as needed, move to the next point by minimizing along the line extending the direction of $-\Delta z(\alpha)$, the local downhill gradient. This searching algorithm has adaptive step sizes and provides one single solution of alpha. A smaller tolerance can be used for higher accuracy.

Algorithm:

- 1) $k = 0$
- 2) $d_k = -\nabla z(\alpha) = -(z_{k+1} - z_k) / (\alpha_{k+1} - \alpha_k)$, the k^{th} step gradient
- 3) Do while $|\alpha_{k+1} - \alpha_k| / \alpha_k > \varepsilon$ and $k \leq m$ and not converge
 1. Use a step size procedure to compute the step size λ_k
 2. $\alpha_{k+1} = \alpha_k + \lambda_k * d_k$
 3. If d_k and d_{k-1} have the same sign, speed up $\lambda_k = \lambda_k * \text{scale}$
 4. If d_k changes sign (turning point), refine step size $\lambda_k = \lambda_k / \text{scale}$
 5. If $|d_k|$ increases significantly, reduce step size $\lambda_k = \lambda_k / 2$
 6. If $\|\alpha_{k+1} - \alpha_k\| \leq \varepsilon$ then $\alpha^* = \alpha + 1$ else $k = k + 1$
 7. $d_k = -\nabla z(\alpha_k)$
- 4) Return.

The step size procedure could be continuous step size reduction, e.g., $\lambda_{k+1} = \lambda_k / \text{scale}$. The scale can be a constant or gradually increasing with the iteration number for stability, or based on the derivatives $z'(\alpha_k)$ and $z''(\alpha_k)$.

$$\lambda_{k+1} = \lambda_k - z'(\alpha_k) / z''(\alpha_k) \quad (5.17)$$

$$\begin{aligned} z'(\alpha_k) &= [z(\alpha_k + \lambda_k * d_k) - z(\alpha_k)] / (\lambda_k * d_k) \\ z''(\alpha_k) &= [z(\alpha_k + \lambda_k * d_k) + z(\alpha_k - \lambda_k * d_k) - 2z(\alpha_k)] / (\lambda_k * d_k)^2 \end{aligned} \quad (5.18)$$

The simplex algorithm

The Nelder–Mead technique is a technique for minimizing an objective function in a multi-dimensional space. Nelder–Mead generates a new test position by extrapolating the behavior of the objective function measured at each test point arranged as a simplex (e.g., a line segment on a line, a triangle on a plane, a tetrahedron in three-dimensional space and so forth). The algorithm then chooses to replace one of these test points with the new test point and so the technique progresses. The simplest step is to replace the worst point with a point reflected through the centroid of the remaining N points. In this study, we tested two simplex approaches: one with absolute error ε (Simplex), and the other with relative error ε (Jsimplex).

Sampling approach

The above approaches involve iterations. Alternatively, we can use a slice sampling (grid search) method. The samples will be generated initially from a uniform distribution of α , but after a series of calibrations of α , we might be able to update the prior pdf of α , e.g., to a normal distribution, and save a number of samples and runs. This approach will have the advantage of being running parallelized, but is much slower if running on a single central processing unit (CPU).

5.1.6 Hyperplane Approximation

Methods to obtain linear approximations of the boundary (a.k.a. hyperplanes or linear inequalities) – (HPA) are explained in this section.

In the parameter space, the boundary surfaces consist of hyper-curved surfaces, which need high computation cost to be determined. In order to avoid the high computation cost, we approximate the curved surfaces through numerous hyper-planes as shown in Figure 33. These hyper-planes can be used to monitor the distance between the current operating point and the boundary surfaces.

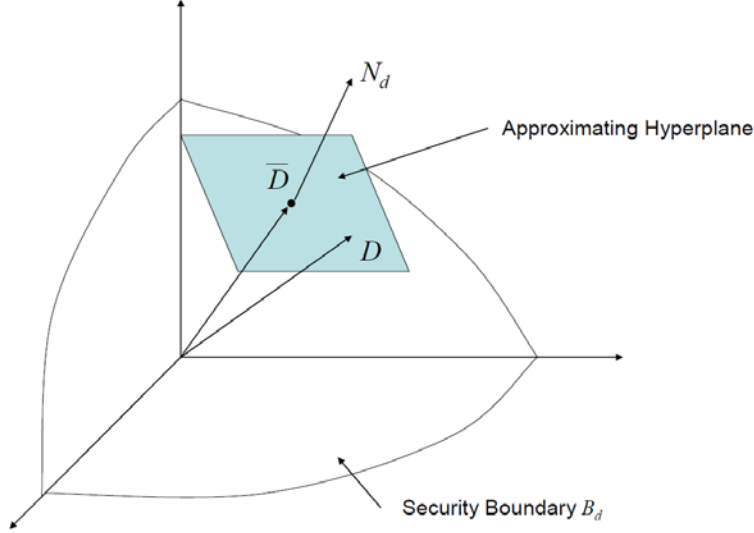


Figure 33. Illustration of an approximating hyperplane in the parameter space.

5.1.6.1 Hyper-plane building procedure

This section demonstrates the procedure proposed in [Parashar 2007a] in the two dimensional parameter space (P space), in which the boundary is represented by a curve shown in Figure 34.

- (1) Suppose we determined the first point of collapse PoC_1 , the normalized left eigenvector L_1 , $\|L_1\| = 1$, and the corresponding tangent hyperplane $L_1^t \cdot (p - PoC_1) = 0$.
- (2) The approximating hyperplane is obtained by parallel shifting the tangent hyperplane along vector L_1 by the distance $(d+m)$, there d and m are the user specified distances. Distance d regulates the accuracy of approximation and the number of required hyperplanes, distance m introduces an additional security margin. The approximating hyperplane equation becomes

$$L_1^t \cdot (p - PoC_1) = d + m \quad (5.19)$$

- (3) Now we start moving along the intersection boundary and the cut set plane (P_i, P_j) . As it will be described below, this motion can be implemented as another type of the parameter continuation procedure, where the intermediate points of collapse are available.
- (4) For each intermediate PoC , we will check the distance r to the tangent hyperplane determined at PoC_1 . We are looking for a point PoC_2 where this distance is slightly less or equal to the user specified distance d :

$$r = \frac{\|L_1^t \cdot (PoC_2 - PoC_1)\|}{\|L_1^t\|} \leq d, \quad d \approx r \quad (5.20)$$

- (5) Continue moving in the same direction checking the distance r from the tangent hyperplane to the PoC_2 . We are looking for the PoC_3 where

$$r = \frac{\|L'_3 \cdot (PoC_3 - PoC_2)\|}{L'_3} \leq d, \quad d \approx r \quad (5.21)$$

- (6) Calculate the new approximating hyperplane $L'_3 \cdot (p - PoC_3) = d + m$.
- (7) Repeat the procedure by continuing the motion along the slice and measuring the distance of the hyperplane from the PoC_3 , and so on.

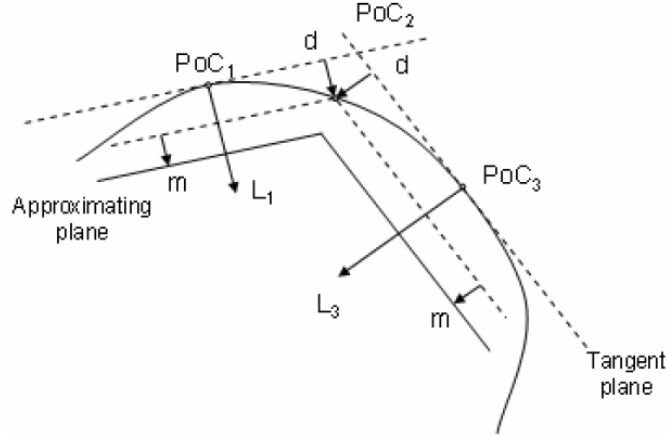


Figure 34: Tangent and approximating hyperplanes.

5.1.7 High-order Numerical Methods (HO)

The high order method description can be found in [Makarov 1995]. A nonlinear predictor can help to minimize or even eliminate corrector steps in the CPF and orbiting method. This section focuses on the CPF method.

The nonlinear predictor may be implemented using high-order numerical methods.

5.1.7.1 Solution motion and its Taylor series expansion

Let us consider a general set of smooth nonlinear power flow equations

$$F(x, \beta) = F(x) + \beta D = 0 \quad (5.22)$$

where x is a vector of dependent variables, β is a scalar parameter, and D is a vector of increments.

If $(\partial F / \partial x)$ is nonsingular, then the function $F(x, \beta)$ can be considered as an implicit function which defines the dependence $x(\beta)$. Differentiation of (5.22) yields

$$((\partial F / \partial x)(dx/d\beta) + (dF/d\beta) = (\partial F / \partial x)(dx/d\beta) + D = 0 \quad (5.23)$$

If the Jacobian matrix $(\partial F/\partial x)$ is nonsingular,

$$(dx/d\beta) = -(\partial F/\partial x)^{-1} D \quad (5.24)$$

Equation (5.24) defines motion of a solution of (5.22) as parameter β varies. A solution of (5.24) can be represented as the Taylor series expansion

$$x(\beta) = x^0 + \sum_{k=1}^{\infty} (k!)^{-1} (d^k x/d\beta^k) (\beta - \beta^0)^k \quad (5.25)$$

$\beta = \beta^0$, $z = z^0$ is a solution of (5.22). Substituting $\alpha = (\beta^0 - \beta)$ gives equation

$$x(\alpha) = x^0 + \sum_{k=1}^{\infty} (\alpha^k/k!) \Delta x_k \quad (5.26)$$

where Δx_k is $(d^k x/d\alpha^k)$ calculated at the point $\alpha = 0$, $x = x^0$. Expansion of Equation (5.26) represents the solution function $x(\alpha)$ as a polynomial of scalar parameter α .

If $\beta^0 = 1$, and D is thought of as a mismatch vector of (5.22) at the point $x = x^0$, then if the series expansion (5.26) converges for $\alpha = 1$, it will give a solution of the problem $F(x) = 0$.

Due to the impracticality of computing a large number of Δx_k , the summation (5.26) must be restricted to a finite number of terms K . Accordingly, (5.26) becomes an iterative procedure

$$x_{i+1} = x_i + \sum_{k=1}^K (\alpha_i^k/k!) \Delta x_{k,i} \quad (5.27)$$

where i is the iteration number and $\Delta x_{k,i}$ is the k th correction vector. The α_i has sense of a *correction coefficient* which influences convergence reliability. It can be easily shown that for $K = 1$, (5.27) corresponds to the Newton-Raphson method with a step correction. If $K > 1$, then (5.27) becomes a generalization of the Newton-Raphson method which takes into account nonlinear terms of the Taylor series expansion. The linear approximation of $F(x)$ that is used in the Newton-Raphson method is replaced by an approximation that is nonlinear.

5.1.7.2 Computation of the correction vectors Δx_k

Expressions for correction vectors Δx_k can be obtained by successive differentiation of (5.23) with respect to β . We set $\alpha = (\beta^0 - \beta) = 1$ in (5.22) and express $F(x)$ as a Taylor series, so giving equation:

$$(1 - \beta^0)D = F(x) = F(x^0) + J(x^0)\Delta x + \sum_{l=2}^{\infty} \frac{1}{l!} W_l(\underbrace{\Delta x, \dots, \Delta x}_l) \quad (5.28)$$

where $J(\cdot)$ is the Jacobian matrix, and $W_l(\cdot)$ is the l -th order term of the Taylor series. It can be shown that by substituting $\Delta x = \sum_{k=1}^K (\Delta x_k / k!)$ (5.28), the following expressions can be obtained:

$$\begin{aligned} \Delta x_1 &= -J^{-1}(x^0)[(\beta^0 - 1)D + F(x^0)]; \\ \Delta x_2 &= -J^{-1}(x^0)[W_2(\Delta x_1, \Delta x_1)]; \\ \Delta x_3 &= -J^{-1}(x^0)[3W_2(\Delta x_1, \Delta x_2) + W_3(\Delta x_1, \Delta x_1, \Delta x_1)]; \\ &\dots \\ \Delta x_i &= -i!J^{-1}(x^0) \sum_{l=2}^i \sum_{\substack{s_1, s_2, \dots, s_K=0, \dots, l \\ s_1 + s_2 + \dots + s_K = l \\ s_1 + 2s_2 + \dots + Ks_K = i}} \prod_{k=1}^K (1/k!)^{-s_k} (s_k!)^{-1} W_l(\underbrace{\Delta x_1, \dots, \Delta x_1}_{s_1}, \dots, \underbrace{\Delta x_K, \dots, \Delta x_K}_{s_K}) \end{aligned} \quad (5.29)$$

The high order terms $W_l(\cdot)$ in (5.29) can be expressed through the values of the function $F(x)$. For example, if (5.22) was a set of quadratic equations, then for $K = 5$ we obtain the following recurrent equalities:

$$\begin{aligned} \Delta x_1 &= -J^{-1}(x^0)[(\beta^0 - 1)D + F(x^0)]; \\ \Delta x_2 &= -J^{-1}(x^0)[W_2(\Delta x_1, \Delta x_1)]; \\ \Delta x_3 &= -J^{-1}(x^0)[3W_2(\Delta x_1, \Delta x_2)]; \\ \Delta x_4 &= -J^{-1}(x^0)[3W_2(\Delta x_2, \Delta x_2) + 4W_2(\Delta x_1, \Delta x_3)]; \\ \Delta x_5 &= -J^{-1}(x^0)[5W_2(\Delta x_1, \Delta x_4) + 10W_2(\Delta x_2, \Delta x_3)] \end{aligned} \quad (5.30)$$

where

$$W_2(\Delta x_i, \Delta x_j) = F(\Delta x_i + \Delta x_j) - F(\Delta x_i) - F(\Delta x_j) + F(0) \quad (5.31)$$

Expressions (5.30) and (5.31) are used at each iteration (5.27) of the method.

5.1.7.3 Correction coefficients

To provide reliable convergence of the method, it is necessary to use appropriate values of the correction coefficients α_i in (5.27). The correct choice of α_i gives direct motion in the space of

mismatches $F(x)$. It was shown that the deviation of the method from the direct line $(\beta^0 - \beta)D$ can be evaluated by the following Euclid norm

$$\left\| (\beta^0 - \alpha)D + F(x^0 + \sum_{k=1}^K \alpha^k k! \Delta x_k) \right\| \leq \varepsilon_F \quad (5.32)$$

It is clear that for $\alpha = 0$ the norm (5.32) is equal to zero. Increasing α results in the method taking larger steps, but the deviation (5.32) can also increase. However, having calculated the correction vectors $x_{k,i}$ at the i -th iteration, and knowing the specified maximum deviation ε_F , it is not difficult to obtain the corresponding value of α_i which keeps the deviation (5.32) within the desired accuracy ε_F . If the value of ε_F is small enough, the method will converge up to a singular point of the original problem (5.32).

Because of the consideration of the high-order Taylor expansion terms, this modification does not require the corrector steps, and theoretically allows to reach the boundary with a fewer number of iterations. At the same time, the computational effort at each iteration is more significant comparing with the predictor-corrector method. Besides, the high order method is more difficult for implementation.

5.1.7.4 Nonlinear approximation of the VSB

In this project, we explored the opportunities for substituting the hyperplane approximation approach by a nonlinear VSB approximation approach. Similar efforts are reflected in references [Perninge 2011] and [Perninge 2013]. The methodology for a quadratic VSB approximation can be found in Appendix D. We have found that the piecewise nonlinear quadratic approximation of VSB meets significant difficulties with finding where the approximating quadrics intersect; this makes practically impossible to quantify the actual voltage stability margin. This approach cannot be recommended for a practical implementation.

5.1.7.5 High-order numerical methods (HO) for the future implementation

At the same time, the non-linear approximation used in the form of a high-order numerical method used to solve VSB equations could be useful for tracing the VSB. It can be implemented in the next phase of the project.

5.2 Algorithm Implementation and Software Code

The PSERC continuation power flow program and MATLAB® programming language were selected as a primary power flow engine. The algorithm is based on methods that were originally used in the NSF-PSERC CPF algorithm¹ [Greene 1997], [Dobson 2001], [Alvarado 2001]. Significant modifications have been made to the PSERC program to meet the objectives of this project.

¹ <http://www.pserc.cornell.edu/tcc/>

5.3 Connectivity with PowerWorld®

SimAuto code is developed to extract field information from PowerWorld and to prepare input file to run the developed MATLAB® code, which generates voltage stability boundary of a given system. This input file is to run the code that consists of powerful methods to explore the voltage stability boundary.

Appendix F contains:

- (1) SimAuto code, and
- (2) MATLAB® code used to calculate several input parameters (needed to generate voltage stability boundary) using the input file generated by SimAuto code.

6.0 Experimental Results and Improvements

The MATLAB® code implementing the non-iterative voltage stability analysis (VSA) has been comprehensively tested using large-scale real system models.

6.1 Test Cases

To test the performance of the developed MATLAB® codes, we used the following three real system models.

6.1.1 Central America System

This model includes an interconnection of the power grids of Panama, Costa Rica, Honduras, Nicaragua, El Salvador, and Guatemala. The model contains 1,985 buses and 2,298 branches.

6.1.2 California ISO Model

California ISO is a balancing authority serving most of California and a small section of Nevada. The model used in this project includes 3,535 buses and 4,402 branches.

6.1.3 Western Electricity Coordinating Council (WECC) Model

The WECC system is an interconnection of the power grids of Canadian provinces of Alberta and British Columbia, the northern portion of Baja California, Mexico, and 14 U.S. states. It includes 19,331 buses and 22,946 branches.

6.2 Transformation from Polar Coordinates to Rectangular Coordinates

To implement the XR method, Jacobian matrices and voltages must be transformed from polar coordinates into rectangular coordinate. We developed a procedure for the Jacobian matrix transformation from polar to rectangular coordinates and it is presented in Appendix E.

6.3 Improvements and Tests: Discrete Events

In this section, we discuss a problem related to discrete events (such as transformations of bus types from P-V to P-Q and vice versa) in power flow calculations, their dependence on the stressing trajectory and potential non-uniqueness of power flow solutions, and a possible remedy to this problem based on the XR method. We also test the performance of XR method when reactive power constraints are added to the problem.

6.3.1 Comparing Bus Types at Boundary Points Obtained From CPF and OM

First, we wanted to compare the bus types (P-V and P-Q) obtained along the voltage stability boundary by using the CPF method and the bus types produced by the orbiting method (OM). Figure 35 below shows the boundary points (A, B and C) that were obtained from the CPF and OM. It was observed that generators' bus types that obtained from these methods are not matching.

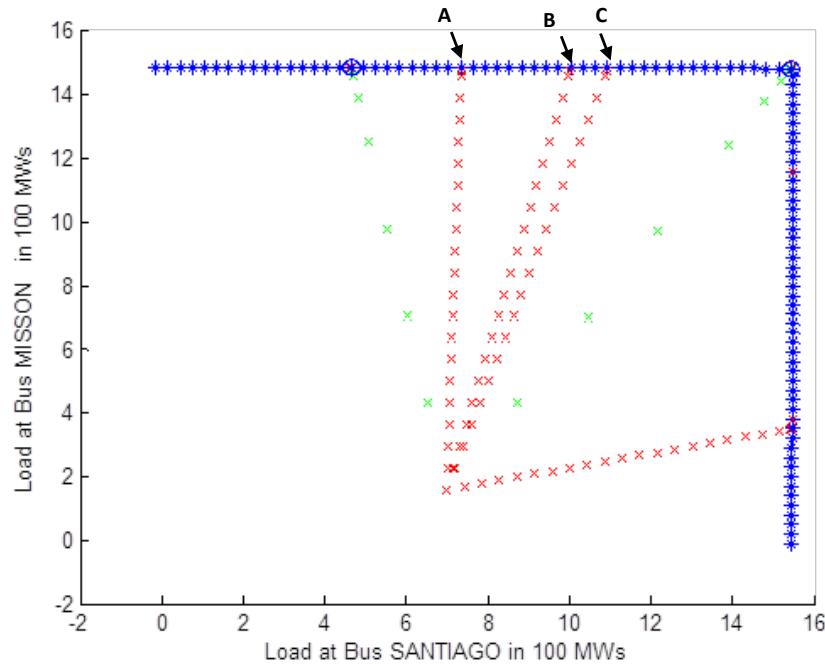


Figure 35. Comparing bus types obtained by different methods in California system model.

An explanation of this fact can be found in the next section.

6.3.2 Sensitivity of Bus Types to Changes in Stress Trajectory

To investigate the sensitivity of bus types to changes in stress trajectory, we stressed the system in three different directions as shown in Figure 36:

- Direction 1: **D1** from 1 to 2
- Direction 2: **D2** from 1 to 3
- Direction 3: **D3** from 2 to 3

The boundary points obtained by stress D1 and D2 - D3 are almost the same (near point 2). However, bus types in these two cases are not matching.

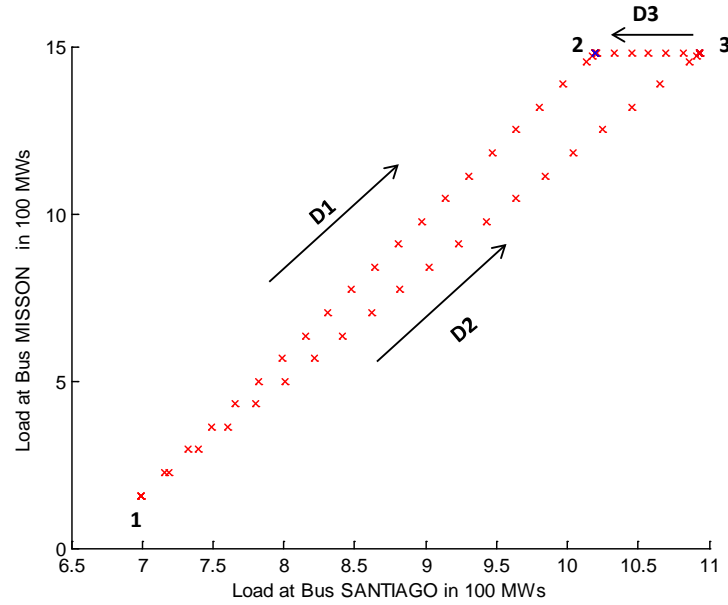


Figure 36. Boundary points from different stressing direction (California system model).

Table 1 below shows the number of PV buses at the point 2 obtained along two different stress directions. This number is significantly different. This means that the number and sequence of discrete events such as $P-V \rightarrow Q-V$ and $P-Q \rightarrow P-V$ changes is a function of stress trajectory. This happens because discrete events change system characteristic and system parameters (e.g., nodal voltage) are sensitive to these changes. This is a common problem in power flow calculations.

Table 1. Number of PV buses in each direction

Trajectory	No. of PV buses at point 2
D1	423
D2 - D3	413

6.3.3 Proposed Approach to Directly Detect Discrete Events and Generation Bus Types in the XR Method

In this section, we analyze approaches to directly detect events such as voltage violations, power transfer violations and P-V \rightarrow P-Q / P-Q \rightarrow P-V changes on XR trajectory.

The proposed procedure uses power flow equations in rectangular coordinates, where the voltage phasors are represented as $X = \{V_i' + jV_i''\}$. In these coordinates, the power flow mismatch can be represented as a quadratic vector-function $F(X) - Y = \{P_i, Q_i\} - Y$ for P-Q buses or $F(X) - Y = \{P_i, V_i^2\} - Y$ for P-V buses, where Y is a vector of nodal power injections or function of fixed voltage magnitudes V_{i*}^2 .

Assume that X_0 is the operating point, and X_* is any other power flow point along the stress trajectory in the state space, $F(X_0) - Y_0 = 0$, $F(X_*) - Y_* = 0$. Let us connect these two points with a parametrically-defined line:

$$X(\alpha) = X_0 + \alpha(X_* - X_0) \quad (6.1)$$

where α is a scalar parameter positioning point X along this line. Under these assumptions, functions $P(\alpha) = P[X(\alpha)]$, $Q(\alpha) = Q[X(\alpha)]$ and $V^2(\alpha)$ become quadratic functions of α .

It is important to stress the following difference between the stress trajectory in injection space $Y(\beta) = Y_0 + \beta \cdot \Delta Y$ and trajectory $Y(\alpha) = F[X(\alpha)]$. As shown in Figure 37, in the Y-space, trajectory $Y(\beta)$ is a straight line, whilst $Y(\alpha)$ is a quadratic curve connecting two points of $Y(\beta)$.

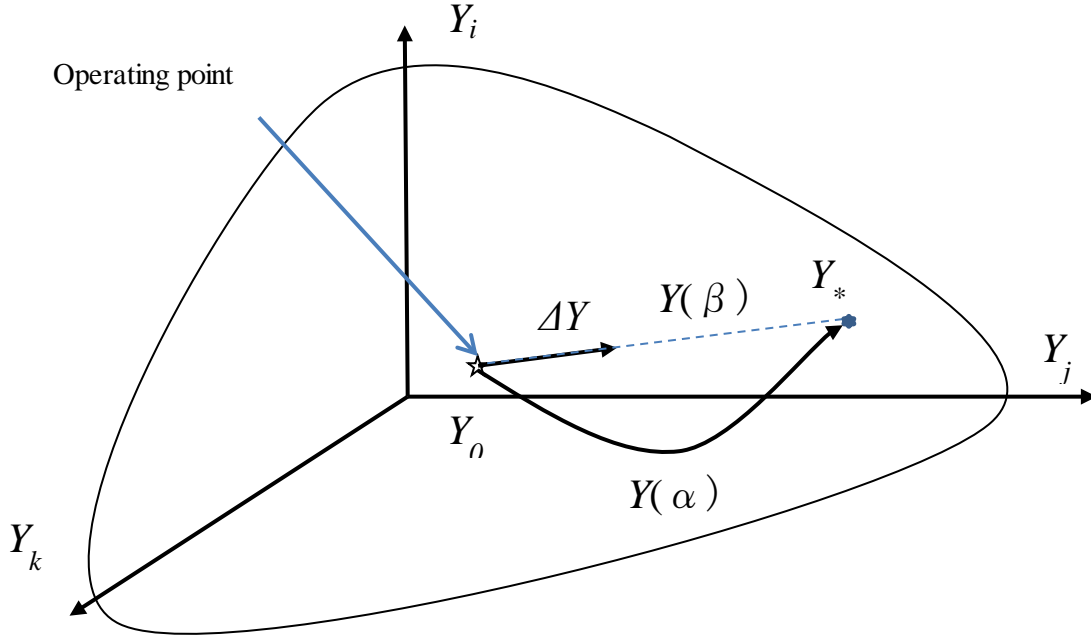


Figure 37. The difference between trajectories $Y(\alpha)$ and $Y(\beta)$.

6.3.3.1 Finding voltage violations along $Y(\alpha)$

Assume that the point $Y_* = Y(\alpha_*) = Y(\beta_*) = F[X(\alpha_*)]$, we have several voltage violations, for instance, $V_{*v} < V_{\min v}$, $v \in \Upsilon$, where Υ are buses with violations. Normally this task is solved by a multi-step procedure, such as step-by-step stressing or binary search, with multiple power flow calculations involved. Our task is to locate these violations along the trajectory $Y(\alpha)$ directly.

Here we propose a better approach. To locate voltage violations, the following scalar quadratic equations can be solved with respect to α_v :

$$V_v^2(\alpha_v) - V_{fixed v}^2 = \left[V_{0v}' + \alpha_v (V_{*v}' - V_{0v}') \right]^2 + \left[V_{0v}'' + \alpha_v (V_{*v}'' - V_{0v}'') \right]^2 - V_{fixed v}^2 = 0 \rightarrow \alpha_v, v \in \Upsilon \quad (6.2)$$

If Equation (6.2) has real solutions for certain bus numbers $v \in \Upsilon$, they help to directly locate violations along the trajectory (6.1). If these violations are related to discrete events such as capacitor switching or under-voltage generation protection operation, we can assume that the sequence of these events is repeating the sequence of events along the trajectory (6.1). This assumption provides uniqueness of a discrete events sequence during a power flow solution process, because this sequence does not depend anymore on the trajectory shape in the Y -space.

6.3.3.2 Finding thermal violations along $Y(\alpha)$

The approach is similar to the one proposed for voltage violations.

A power transfer T_{ij} can be expressed as a function of state variables X :

$$T_{ij} = T_{ij}(\alpha) = T_{ij}[X_0 + \alpha(X_* - X_0)], ij \in IJ \quad (6.3)$$

where IJ is the set of critical power transfers. Assume that at the point $X_* = X(\alpha_*)$ we observe violations on several critical paths $ij \in IJ_* \subset IJ$:

$$T_{ij}(\alpha_*) = T_{ij}(\beta_*) > T_{\max ij}, ij \in IJ_* \quad (6.4)$$

The sequence of power transfer violations can be evaluated by the sequence of these violations along $Y(\alpha)$. To find this sequence, the following scalar quadratic equations can be solved with respect to α :

$$T_{ij} = T_{ij}(\alpha) = T_{\max ij} \rightarrow \alpha, ij \in IJ_* \quad (6.5)$$

If Equation (6.5) has real solutions for certain branches $ij \in IJ_*$, they help to directly locate violations along the trajectory (6.1). If these violations are related to discrete events such as transmission line tripping, we can assume that the sequence of these events is repeating the sequence of events along the trajectory (6.1). This assumption provides uniqueness of a discrete events sequence during a power flow solution process, because this sequence does not depend anymore on the trajectory shape in the Y-space.

6.3.3.3 Finding generation bus type changes along $Y(\alpha)$

To locate generation bus type changes, the following scalar quadratic equations can be solved with respect to α_v for each bus v with voltage control from a generator:

$$Q_v(\alpha_v) - Q_{v\max(\min)} = 0 \rightarrow \alpha_v, v \in \Psi \quad (6.6)$$

If Equation (6.6) has real solutions for certain bus numbers $v \in \Psi$, they help to directly locate points, where the generators' types change along the trajectory (6.1). This approach provides uniqueness of bus types for any power flow point, because this sequence does not depend anymore on the trajectory shape in the Y-space.

6.3.4 Incorporating Constraints and Discrete Events into the XR Method

In this project, we incorporated discrete events into the XR method (such as changes of generators' types from PV to PQ type and vice versa). Here we report on experimental results helping to evaluate the performance of XR method when the reactive power constraints are added to the problem. These experiments are to check if our methodology obtains the same boundary points as found in step-by-step method using CPF or not. Figure 38 shows exploration of $\|\mathbf{R}(\alpha)\|$ and distance function $\|\mathbf{J}(\alpha)\tilde{\mathbf{R}}(\alpha)\|$ with without reactive power constraints. The point "100" indicates the voltage stability boundary point (100% loading), obtained by CPF method. It can be seen that the XR finds the same VSB point.

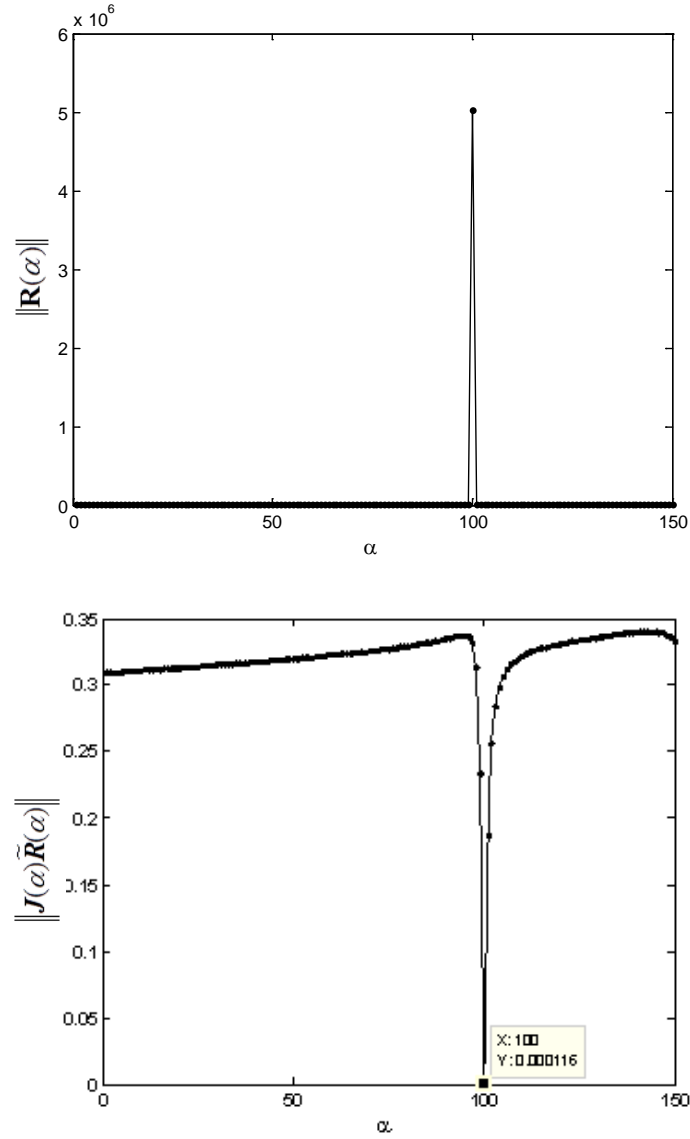


Figure 38. Exploration of $\|R(\alpha)\|$ and distance function ($\|J(\alpha)R(\alpha)\|$) without reactive power constraints.

Figure 39 shows exploration of $\|R(\alpha)\|$ and distance function ($\|J(\alpha)\tilde{R}(\alpha)\|$) with reactive power constraints α . From the results, it is observed that distance function achieves its minimum value at $\alpha = 100$, which matches the boundary point found by CPF method.

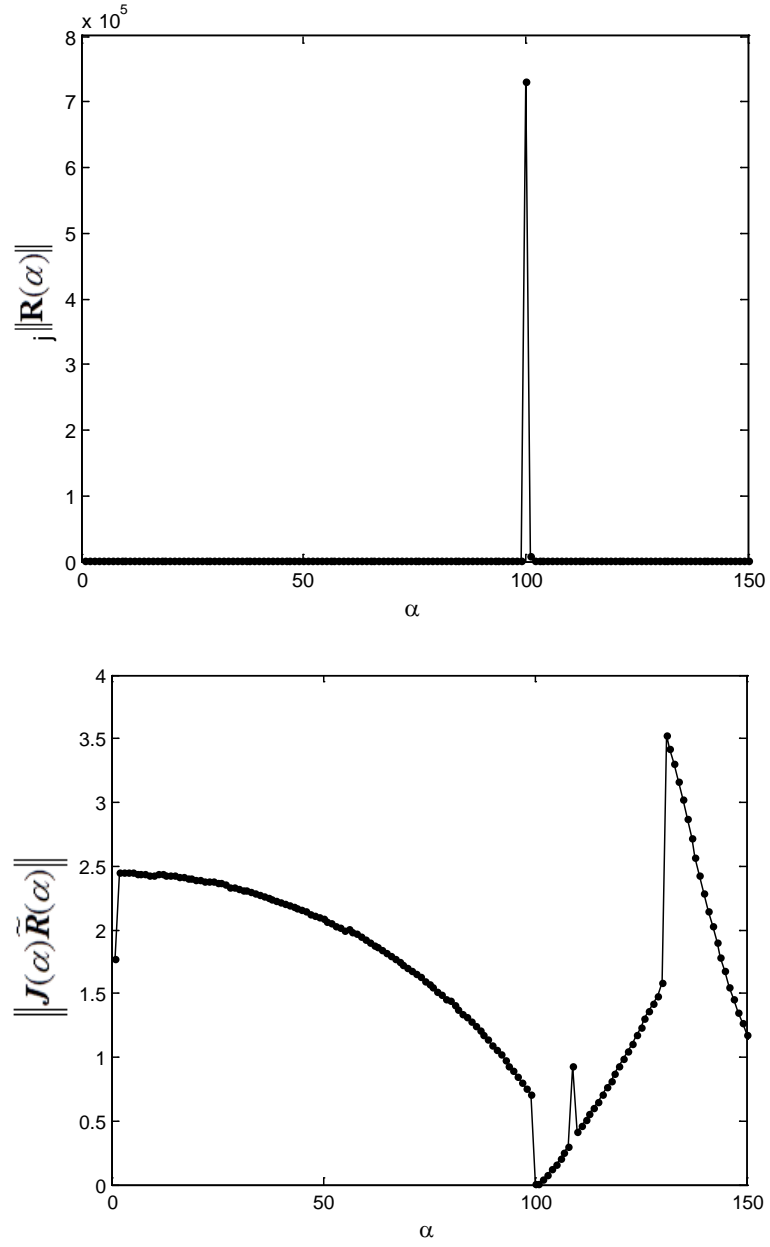


Figure 39. Exploration of $\|R(\alpha)\|$ and distance function ($\|J(\alpha)\tilde{R}(\alpha)\|$) without fixed reactive power constraints.

6.4 Hyperplane Method Test

In this section, we will take California system model as an example to illustrate how to build its VSB approximation by hyperplanes.

In this case, $p = [p_1, p_2]$, where p_1 is the real power load at bus 1 (SANTIAGO) and p_2 is the real power load at bus 2 (MISSION). The VSB plot is shown as dotted blue line in Figure 40. By applying our approximation approach described above, we get the following equation for the approximating line (the tangent set shown as solid pink line), which approximates VSB (in p.u.; 1 p.u. is 100 MW):

$$-0.0004(p_1 - 8.1825) - 0.0955(p_2 - 14.8293) = 0 \quad (6.7)$$

Similarly, we can get another line (in p.u.):

$$0.6023(p_1 - 15.4932) - 0.0014(p_2 - 12.4785) = 0 \quad (6.8)$$

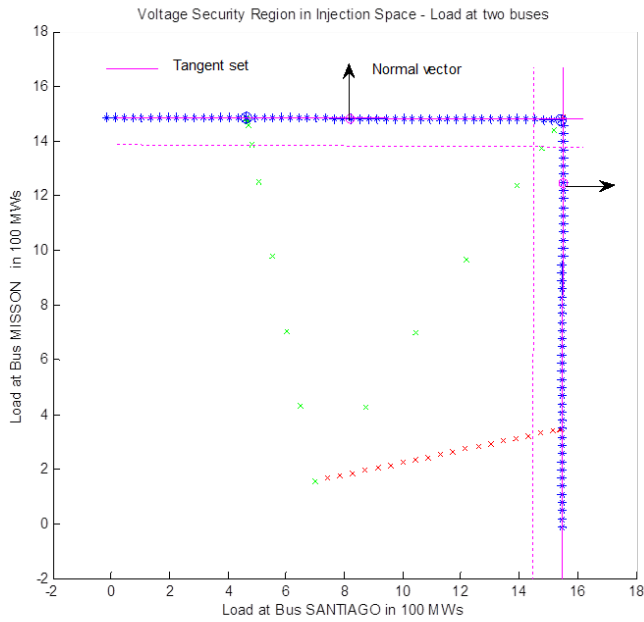


Figure 40 also shows two lines plotted at certain distance from the tangent set. This distance reflects a specially added voltage stability margin (100 MW).

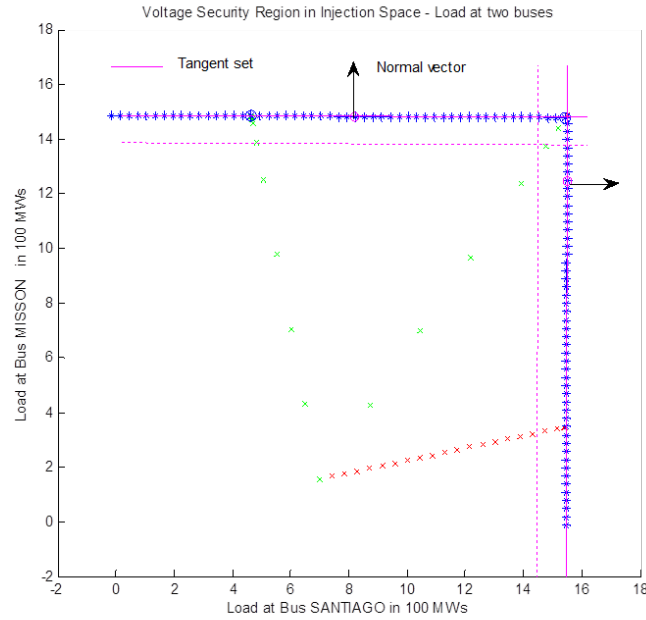


Figure 40. Approximating lines for the California test case (pink lines).

6.5 Examples of VSB Plots for Test Systems

For the WECC, California and Central America systems, we tested the robustness, computational time, and the accuracy of our algorithm against the conventional CPF method and PowerWorld simulator widely used in the industry. VSB plots shown below are produced using our VSA tool and verified using CPF simulations.

6.5.1 Central America System

A VSB plot for the Central America System is shown in Figure 41. Two loads were selected in this system to plot a slice of VSB. The boundary was plotted using the proposed method. The initial stress has been done by applying the CPF method and step reduction to reach the vicinity of VSB (the sequence of red “x” symbols in the center of Figure 41). The CPF process has been repeated two more times due to divergence of the orbiting method (sequences of green “x” symbols in the center of Figure 41). The divergence was caused by singularities of VSB around these points. The rest of boundary points were obtained using the orbiting method. Each iteration of this method produced a new VSB point.

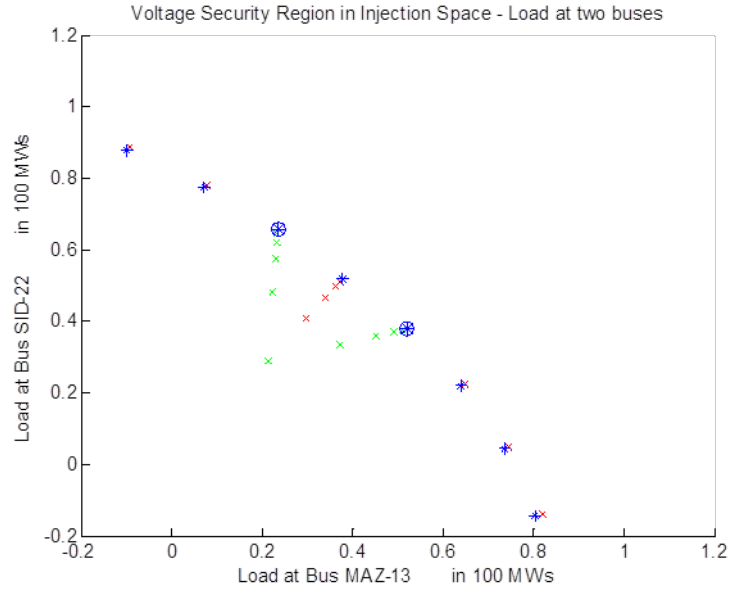


Figure 41. Central America System VSB.

6.5.2 California System

Similarly to the Central America case, two loads were selected to produce a slice of VSB - Figure 42. Three CPF stresses were needed to obtain an initial VSB point and restart the orbiting process in cases of divergence. The rest of VSB points were calculated using the orbiting method. It is interesting that VSB has rectangular shape, which indicates that the selected loads were independent in terms of their impact on voltage stability.

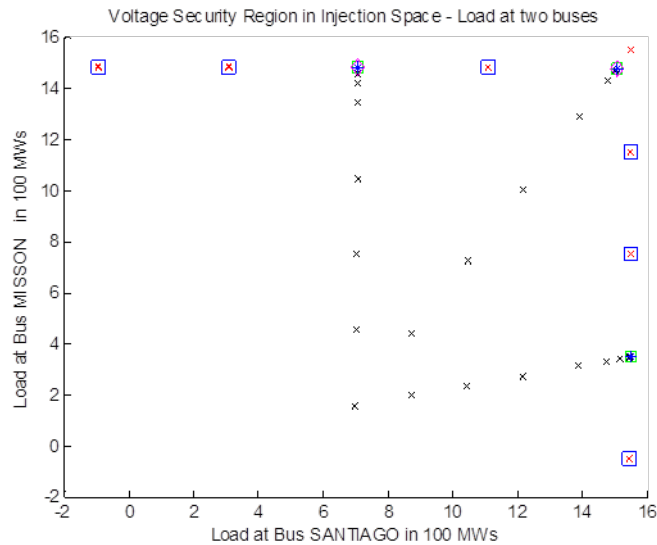


Figure 42. California system VSB.

6.5.3 WECC System

Only one CPF run was needed to initiate orbiting method in the WECC case. The rest of the points were obtained using non-iterative method - Figure 43.

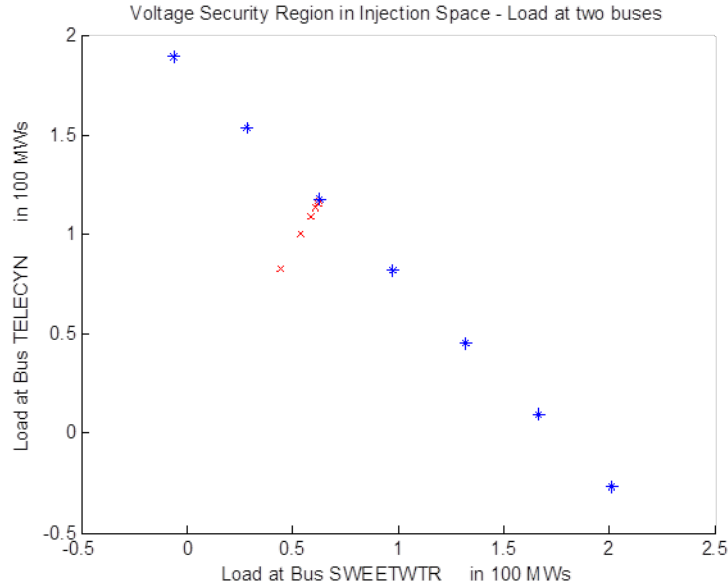


Figure 43. WECC system VSB.

6.6 Computational Time

We compared the computational time of our method against the traditional CPF method.

6.6.1 Central America System

Table 2 contains a comparison of computational time of our method against the CPF method. The total time to produce 8 boundary points was 24.8 sec (3.1 sec per point average). The CPF method was repeated 3 times and took 23.1 seconds (7.7 sec per run). The non-iterative method was repeated 7 times (including 2 divergent cases) with the average time of 0.17 sec. Therefore:

- For the Central America System, one step of the orbiting method is 46 times faster than the CPF method.
- To get 8 points of VSB using CPF method will take about 61.6 sec. The overall process of our method took 24.8 sec. The resulting speed-up ratio is 2.1 times.

Table 2. Computational time for the Central America system

System	Total simulation time		Continuation power flow (CPF)			Boundary orbiting method (BOM)		
	Boundary points	Time(s)	runs	Time(s)	Average Time Per Run (s)	runs	Time(s)	Average Time Per Run (s)
Central America (1985 buses)	8	24.841	3	23.0889	7.6963	7	1.1584	0.1655

6.6.2 California System

Table 3 contains a comparison of computational time of our method against the CPF method. The total time to produce 9 boundary points was 32.6 sec (3.6 sec per point average). The CPF method was repeated 3 times and took 28.9 seconds (9.6 sec per run). The non-iterative method was repeated 8 times (including 2 divergent cases) with the average time of 1.1 sec. Therefore:

- For the California system, one step of the orbiting method is 8.76 times faster than the CPF method.
- To get 8 points of VSB using CPF method will take about 260 sec. The overall process of our method took 38.6 sec. The resulting speed-up ratio is 6.74 times.

Table 3. Computational time for the California system

System	Total simulation time		Continuation power flow (CPF)			Boundary orbiting method (BOM)		
	Boundary points	Time(s)	runs	Time(s)	Average Time Per Run (s)	runs	Time(s)	Average Time Per Run (s)
CAISO (3535 buses)	9	38.622	3	28.8918	9.6306	8	8.8076	1.1010

6.6.3 WECC System

Table 4 contains a comparison of computational time of our method against the CPF method. The total time to produce 9 boundary points was 209.5 sec (29.9 sec per point average). The CPF method was repeated 1 time and took 152.5 seconds. The non-iterative method was repeated 6 times with the average time of 9.5 sec. Therefore:

- For the WECC system, one step of the orbiting method is 16 times faster than the CPF method.

- To get 7 points of VSB using CPF method will take about 1068 sec. The overall process of our method took 206.5 sec. The resulting speed-up ratio is 5.17 times.

Table 4. Computational time for the WECC system

System	Total simulation time		Continuation power flow (CPF)			Boundary orbiting method (BOM)		
	Boundary points	Time(s)	runs	Time(s)	Average Time Per Run (s)	runs	Time(s)	Average Time Per Run (s)
WECC2014 (19331 buses)	7	209.5	1	152.5	152.5	6	57.02	9.5

6.6.4 Comparison of Results and Conclusion

Figure 44 shows a comparison of the traditional computational methods and the new method in terms of computational time needed to get one slice of the voltage stability boundary depending on the system size. Note that for the WECC system (19,331-bus system) the computational time of our method remains within 4 minutes.

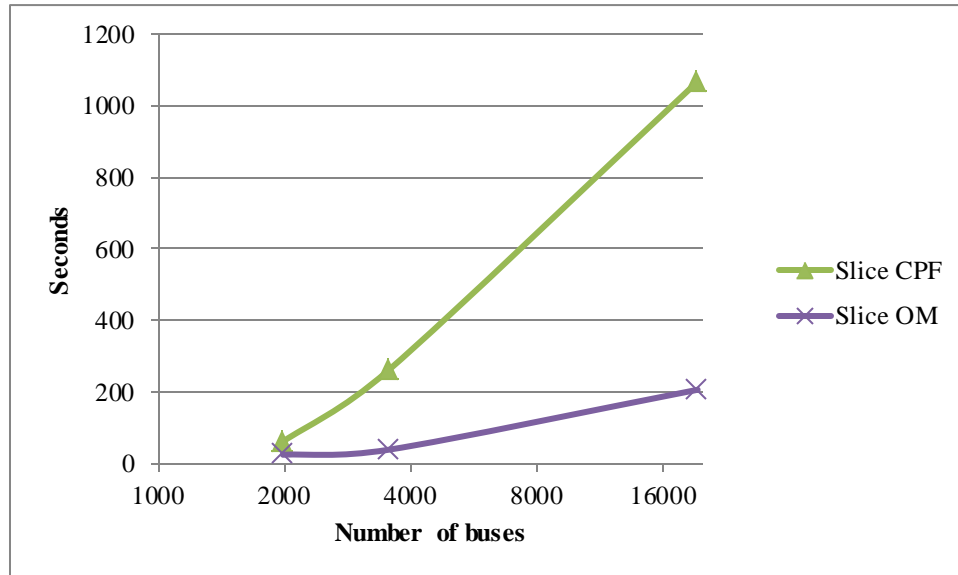


Figure 44. Computational time to get one slice of VSB depending on the system size.

6.7 Accuracy

We compared VSB points obtained using our method against the points calculated using PowerWorld. To run this experiment, we had to make some assumptions and adjustments in the models related to phase shifting transformers, switching capacitors, etc. They are listed in Appendix F. The reason for those changes is as follows. The MATLAB® code that has been developed is research software, which cannot simulate a detailed system model typical for the production-grade software tools.

Appendix G contains a SimAuto and MATLAB® codes developed by the project team to read information directly from PowerWorld formats.

Table 5 shows results of comparison between the non-iterative methods and PowerWorld for several stress directions in Central America system. The developed method provides results within 1% from the PowerWorld results.

Table 5. Comparison of the VSB points obtained by two methods

Stress direction	Non-iterative Method		PowerWorld		Accuracy
	Sink Load (MW)	Sum(Sink) (MW)	Sink Load(MW)	Sum(Sink) (MW)	
1	740.08	740.08	738.11	738.11	0.28
2	1188.9	1188.9	1192.8	1192.8	0.82
3	260.58	260.58	261.61	261.61	0.4
4	744.56	1002.4	746.95	1007.0	0.46

7.0 Project Benefits and Expected Impacts

The project helps to achieve the following benefits for system operations:

- Enhanced situational awareness, early detection of potential system instability, and improved reliability achieved by providing real-time dynamically adjusted information to power system dispatchers and VSB visibility for multiple stress directions and contingencies.
- Actionable information and prevention of system blackouts achieved by providing advisory information using the wide area security region approach.
- Better utilization of transmission assets due to near-real-time adjustment of voltage stability limits, reflecting mutual dependences of power flows on critical transmission paths.
- Developing real-time and HPC applications.
- Potential for predictive/preventive control.
- Potential for close-loop automatic emergency control systems.

8.0 Conclusion

Voltage stability is becoming an increasing source of concern in the secure operation of present-day power systems. It is important to address voltage stability problems effectively in the planning, design and operation of electric power systems. In this report, a new non-iterative method to determine the voltage stability boundary is proposed and examined. The stability theory in terms of the stability boundary in state space and parameter space and their relationship is examined. A few test power systems are used to evaluate the performance of the proposed non-iterative method. The non-iterative procedure associated with the quadratic approximation of power flow equations in rectangular coordinates allows the method to explore the static voltage stability boundary very quickly, without iteration steps. Thus the proposed method is extremely suitable for parallel processing.

9.0 Recommendations for Next Phase

The proposed effort for FY15 will include the following major tasks:

(1) Algorithm Enhancement: Increase the robustness of the VSA software tool for a variety of system models and conditions and demonstrate its trustworthiness for solving real industry cases.

(2) Tool and UI Refinement: Further improve the performance of the VSA software tool and improve the GUI to make it attractive for adoption by vendors.

(3) Vendor Engagement: We will collaborate with a selected vendor or vendors to transfer the technology. This will include presentations at industry forums and conferences. To date, PNNL has received expressions of interest from the following vendors: Powerworld, V&R Energy Systems Research, and PowerTech.

(4) Non-iterative Small Signal Stability Method Development: This task will develop methods and prototypes for non-iterative small signal analysis. The method will enable plotting of the small-signal stability boundary in coordinates of critical system parameters in near real-time.

10.0 References

- [Adjarapu 1992] V. Ajarapu and C. Christy, "The continuation power flow: a tool for steady state voltage stability analysis," *Power Syst. IEEE Trans.*, vol. 7, no. 1, pp. 416–423, Feb. 1992.
- [Alvarado 2001] F.L. Alvarado, "Solving Power Flow Problems with a MATLAB Implementation of the Power System Applications Data Dictionary," *Decision Support Systems*, Volume 30, Issue 3, January 2001, Pages 243–254.
- [Anderson 2003] P. M. Anderson and A. A. Fouad, *Power System Control and Stability*, 2nd ed. New York: IEEE Press, 2003.
- [Dobson 1992] I. Dobson and L. Lu, "Computing an optimum direction in control space to avoid stable node bifurcation and voltage collapse in electric power systems," *Autom. Control. IEEE Trans.*, vol. 37, no. 10, pp. 1616–1620, Oct. 1992.
- [Dobson 1993] I. Dobson and L. Lu, "New methods for computing a closest saddle node bifurcation and worst case load power margin for voltage collapse," *Power Syst. IEEE Trans.*, vol. 8, no. 3, pp. 905–913, Aug. 1993.
- [Dobson 2001] I. Dobson, S. Greene, R. Rajaraman, C.L. DeMarco, F.L. Alvarado, M. Glavic, J. Zhang, R. Zimmerman, *Electric power transfer capability: Concepts, applications, sensitivity, uncertainty*, PSERC publication 01-34, November 2001, available from <http://www.pserc.wisc.edu/>.
- [Dong 2012] X. Dong, J. Liang, X. Zhang, and H. Sun, "Computation of closest steady state voltage stability bifurcation using PSO approach," in *Innovative Smart Grid Technologies - Asia (ISGT Asia)*, 2012 IEEE, 2012, pp. 1–4.
- [Eto 2008] J. H. Eto, M. Parashar, and N. J. Lewis, "Real-time system operations (RTSO) 2006-2007," Consortium for Electric Reliability Technology Solutions (CERTS), 2008, California Energy Commission, PIER Transmission Research Program.
- [Greene 1997] S. Greene, I. Dobson, F.L. Alvarado, P.W. Sauer, "Initial concepts for applying sensitivity to transfer capability," NSF Workshop on Available Transfer Capability, Urbana, IL USA, June 1997.
- [Hiskens 2001] I. A. Hiskens and R. J. Davy, "Exploring the Power Flow Solution Space Boundary," *IEEE Transactions on Power Systems*, Vol. 16, No. 3, August 2001.
- [Ibsais 1996] A. Ibsais and V. Ajarapu, "Voltage stability-limited interchange flow," *Elec. Power Sys. Res.*, vol. 38, no. 2, pp. 91–95, 1996.

- [Kwatny 1995] H. G. Kwatny, R. F. Fischl, and C. O. Nwankpa, "Local bifurcation in power systems: theory, computation, and application," *Proc. IEEE*, vol. 83, no. 11, pp. 1456–1483, Nov. 1995.
- [Lu 1995] [16] J. Lu, C.-W. Liu, and J. S. Thorp, "New methods for computing a saddle-node bifurcation point for voltage stability analysis," *Power Syst. IEEE Trans.*, vol. 10, no. 2, pp. 978–989, May 1995.
- [Makarov 1995] Y. V. Makarov, V. A. Maslennikov, and D. J. Hill, "Calculation of oscillatory stability margins in the space of power system controlled parameters", *Proc. of the International Symposium on Electric Power Engineering Stockholm Power Tech: Power Systems*, Stockholm, Sweden, 18-22 June, 1995, pp. 416-422.
- [Makarov2000a] Y. V. Makarov, D. J. Hill, and Z. Dong, "Computation of bifurcation boundaries for power systems: a new Delta-plane method," *Circuits Syst. I Fundam. Theory Appl. IEEE Trans.*, vol. 47, no. 4, pp. 536–544, Apr. 2000.
- [Makarov 2000b] Y. V. Makarov, D. J. Hill, and I. A. Hiskens, "Properties of Quadratic Equations and Their Application to Power System Analysis", *Electrical Power and Energy Syst.*, Vol.22 (2000), pp. 313-323.
- [Makarov 2007] Y. V. Makarov, J. Ma, and Z. Dong, "Determining Static Stability Boundaries Using A Non-Iterative Method," in *Power Engineering Society General Meeting, 2007. IEEE*, 2007, pp. 1–9.
- [Makarov 2009] Y. V. Makarov, P. Du, S. Lu, T. B. Nguyen, and X. Guo, *Wide Area Power System Security Region, Final Report PNNL-19049*, Pacific Northwest National Laboratory, Richland, WA, November 2009.
- [Nguyen 2012] T. V. Nguyen, Y. M. Nguyen, and Y. T. Yoon, "A new method for static voltage stability assessment based on the local loadability boundary," *Int. J. Emerg. Elec. Power Sys.*, vol. 13, pp. 1–14, 2012.
- [Perninge 2011] M. Perninge and L. Soder, "On the Validity of Local Approximations of the Power System Loadability Surface," *Power Syst. IEEE Trans.*, vol. 26, no. 4, pp. 2143–2153, Nov. 2011.
- [Perninge 2013] M. Perninge, "Approximating the loadability surface in the presence of SNB–SLL corner points," *Int. J. Elec. Power Energy Sys.*, vol. 22, pp. 313–323, 2013.
- [Parashar 2007a] Parashar, M., A. Agarwal, Y. V. Makarov, and I. Dobson, "Real-time Voltage Security Assessment (RTVSA) Algorithms & Framework," Prepared For: California Independent System Operator, Prepared by: Consortium for Electric Reliability Technology Solutions, Funded By: California Public Interest Energy Research Transmission Research Program, Contract No. 500-02-004, MR-041, December 2007. Available at <http://certs.lbl.gov/certs-rt-pubs.html>.

- [Parashar 2007b] Parashar, M., A. Agarwal, Y. V. Makarov, and I. Dobson, , “Real-Time Voltage Security Assessment Tool (RTVSA) Functional Specifications For Commercial Grade Application”, Prepared For: California Independent System Operator, Prepared by: Consortium for Electric Reliability Technology Solutions, Funded By: California Public Interest Energy Research Transmission Research Program, Contract No. 500-02-004, MR-041, April 9, 2007. Available at <http://certs.lbl.gov/pdf/rts0-06-07-appe.pdf>.
- [Parashar 2008a] Parashar, M., A. Agarwal, Y. V. Makarov, and I. Dobson, "Real-Time Voltage Security Assessment (RTVSA): Summary Report", Prepared For: California Independent System Operator, Prepared by: Consortium for Electric Reliability Technology Solutions, Funded By: California Public Interest Energy Research Transmission Research Program, Contract No. 500-02-004, MR-041, January 21, 2008. Available at: <http://certs.lbl.gov/pdf/task3-rtvsa-summary-rpt.pdf>.
- [Parashar 2008b] Parashar, M., A. Agarwal, and Y. V. Makarov, “Real-Time Voltage Security Assessment (RTVSA): Algorithm’s Simulation & Validation Results, Prepared For: California Independent System Operator, Prepared by: Consortium for Electric Reliability Technology Solutions, Funded By: California Public Interest Energy Research Transmission Research Program, Contract No. 500-02-004, MR-041, January 15, 2008. [Online.] Available: <http://certs.lbl.gov/pdf/task3-rtvsa-scal-demo-results.pdf>.
- [Trias 2012] A. Trias, "The Holomorphic Embedding Load Flow method," IEEE Power and Energy Society General Meeting, 22-26 July 2012.
- [Van Cutsem 1991] T. Van Cutsem, “A method to compute reactive power margins with respect to voltage collapse,” *Power Syst. IEEE Trans.*, vol. 6, no. 1, pp. 145–156, Feb. 1991.
- [Venkatasubramanian 1995a] V. Venkatasubramanian, H. Schattler, and J. Zaborsky, “Dynamics of large constrained nonlinear systems-a taxonomy theory [power system stability],” *Proc. IEEE*, vol. 83, no. 11, pp. 1530–1561, Nov. 1995.
- [Venkatasubramanian 1995b] V. Venkatasubramanian, H. Schattler, and J. Zaborszky, “Local bifurcations and feasibility regions in differential-algebraic systems,” *Autom. Control. IEEE Trans.*, vol. 40, no. 12, pp. 1992–2013, Dec. 1995.
- [Zhang 2011] Y.-S. Zhang and H.-D. Chiang, “Local bifurcation boundary and steady-state security boundary in large electric power systems: Numerical studies,” *Int. J. Bifurc. Chaos*, vol. 21, no. 03, pp. 647–662, Mar. 2011.

Appendix A: Voltage Stability Related Blackouts Worldwide

Many system blackouts and “near misses” were related to voltage instability and voltage collapse problems. Among them, the following major events are of interest – See Appendix A.1 references for more details:

1. 1965: Eastern US and Canada
2. 1970: Japan and New York Power Pool
3. 1977: New York City and Jacksonville, Florida
4. 1978: France
5. 1982: Belgium and Florida
6. 1983: Sweden
7. 1987: Tokyo, Japan, Western Tennessee, France
8. 1989, 1993-1994: Italy
9. 1996: US Western Interconnection (2 events)
10. 1997: The Netherlands
11. 1998: Atlanta, San Francisco Area, Upper Midwest U.S.
12. 1999: Northeast U.S. (2 events)
13. 2003: Croatia - Bosnia Herzegovina, Eastern U.S. - Canada, Denmark - Sweden, and Italy
14. 2004: Western Norway and Southern Greece
15. 2005: Moscow region, Russia

A.1 References for Appendix A

- [A1] P. Kundur, Power System Stability and Control, New York: McGraw-Hill, 1994.
- [A2] S. Lindahl, “Case studies of recent blackouts”, CRIS International Workshop on Power System Blackouts ~ Causes, Analyses, and Countermeasures, Lund, Sweden, May 3, 2004.
- [A3] Final Report on the August 14, 2003 Blackout in the United States and Canada: Causes and Recommendations, U.S. – Canada Power System Outage Task Force, April 5, 2004. Available online at <http://www.nerc.com/~filez/blackout.html>.

- [A4] G. Doorman, G. Kj lle, K. Uhlen, E. S. Huse, and N. Flatab , “Vulnerability of the Nordic Power System”, Report to the Nordic Council of Ministers, SINTEF Energy Research, May 2004. Available online at <http://sparky.harvard.edu/hepg/Papers/Doorman.vul.nordic.system.0504.pdf>
- [A5] A. Kurita and T. Sakurai, “The Power System Failure on July 23, 1987 in Tokyo”, in Proceedings of the 27th Conference on Decision and Control, December 1988.
- [A6] G.C. Bullock, “Cascading Voltage Collapse in West Tennessee, August 22, 1987”, Georgia Institute of Technology 44th Annual Protective Relaying Conference, May 2--4 1990.
- [A7] “The Electric Power Outages In The Western United States, July 2-3, 1996”, DOE Report to the President, August 2, 1996. Available online at ftp://www.nerc.com/pub/sys/all_updl/docs/pubs/doerept.pdf.
- [A8] “Western Systems Coordinating Council Disturbance Report for the Power System Outage that Occurred on the Western Interconnection on August 10, 1996”, October 18, 1996. Available online at ftp://www.nerc.com/pub/sys/all_updl/docs/pubs/AUG10FIN.pdf.
- [A9] “California ISO System Disturbance Report: December 8, 1998 - San Francisco Area”. Available online at <http://www.caiso.com/docs/1999/03/31/1999033116341717931.pdf>
- [A10] N. Dizdarevic, M. Majstrovic, S. Cudjic Coko, N. Mandic, and J. Bonovic, “Causes, Analyses and Countermeasures with Respect to Blackout in Croatia on January 12, 2003”, CRIS International Workshop on Power System Blackouts ~ Causes, Analyses, and Countermeasures, Lund, Sweden, May 3, 2004.
- [A11] “Technical Analysis of the August 14, 2003, Blackout: What Happened, Why, and What Did We Learn?”, Report to the NERC Board of Trustees by the NERC Steering Group, July 13, 2004. Available online at ftp://www.nerc.com/pub/sys/all_updl/docs/blackout/NERC_Final_Blackout_Report_07_13_04.pdf
- [A13] “Power failure in Eastern Denmark and Southern Sweden on 23 September 2003”, Final report on the course of events, Ekraft System, November 4, 2003. Available online at <http://www.elkraft-system.dk>
- [A14] “The Black-Out In Southern Sweden and Eastern Denmark”, 23 September, 2003, Preliminary Report, PM 2003-10-02, Svenska Kraftn t. Available at http://www.svk.se/upload/3195/Disturbance_Sweden_DenmarkSept23.pdf
- [A15] “Interim Report of The Investigation Committee on the 28 September 2003 Blackout in Italy”, Press Release, Union for the Coordination of Transmission of Electricity UCTE, Brussels, Belgium, October 3, 2003. Available online at <http://www.energie-schweiz.ch/imperia/md/content/medienmitteilungen/mm06-122003/82.pdf>
- [A16] “Black-Out: The Events Of 28 September 2003”, Press Release, Gestore Rete Trasmissione Nazionale, Italy, October 1, 2003. Available online at <http://www.grtn.it/eng/documentinewsstatiche/blackout28set03.pdf>

- [A17] C. Vournas, “Technical Summary on the Athens and Southern Greece Blackout of July 12, 2004”. Available online at http://www.pserc.org/Greece_Outage_Summary.pdf.
- [A18] Отчет по расследованию аварии в ЕЭС России, произошедшей 25.05.2005. Председатель комиссии по расследованию аварии В.К.Паули, 18 июня 2005 г. [Report on failure inquiry in UES of Russia that took place at May 25,2005 – in Russian]. Available online at: http://www.mosenergo.ru/download/r410_account.zip.
- [A19] “Voltage Stability of Power Systems: Concepts, Analytical Tools, and Industry Experience”, IEEE Special Publication 90TH0358-2-PWR, 1990.
- [A20] “Modeling of Voltage Collapse Including Dynamic Phenomena”, CIGRE Task Force 38-02-10, 1993.

Appendix B: Power Flow in Cartesian Coordinates

In Cartesian coordinates, the voltage at bus i can be expressed as

$$\hat{V}_i = V_i' + jV_i'' \quad (B1)$$

where \hat{V}_i represents complex number, V_i' and V_i'' are real and imaginary parts, respectively. Let Y denote the bus admittance matrix, its element at i th row and k th column can be expressed as

$$\hat{Y}_{ik} = G_{ik} + jB_{ik} \quad (B2)$$

Therefore, the injection current at bus i can be expressed as

$$\hat{I}_i = \sum_{k=1}^{N+1} \hat{Y}_{ik} \hat{V}_k = \sum_{k=1}^{N+1} (G_{ik} + jB_{ik})(V_k' + jV_k'') \quad (B3)$$

where N is the number of buses. The injection power at node i is calculated as

$$\hat{S}_i = P_i + jQ_i = \hat{V}_i \hat{I}_i^* \quad (B4)$$

replacing \hat{V}_i and \hat{I}_i in (B4) by (B1) and (B3), respectively, and resolving it into real and imaginary part yields

$$P_i = V_i' \sum_{k=1}^{N+1} (G_{ik} V_k' - B_{ik} V_k'') + V_i'' \sum_{k=1}^{N+1} (B_{ik} V_k' + G_{ik} V_k''), \quad (B5)$$

and

$$Q_i = -V_i' \sum_{k=1}^{N+1} (B_{ik} V_k' + G_{ik} V_k'') + V_i'' \sum_{k=1}^{N+1} (G_{ik} V_k' - B_{ik} V_k'') \quad (B6)$$

For a system with r PV buses, $n-r$ PQ buses, and one slack bus whose index is set to be $n+1$, the corresponding power balance and voltage equations are shown in **Error! Reference source not found.**, where P_i^s is the specified value of active power for bus i (either PQ or PV bus), Q_i^s is the specified value

of reactive power for bus i (PQ bus), V_i^s is the specified set-point voltage at the PV buses, and $N = n+1$. Therefore, we have $2n$ equations and $2n$ state variables.

Appendix C: Formulation of the Quadratic Approximation

The objective of this task is to formulate the quadratic approximation for the non-iterative method. In this task, a quadratic surface has been determined around a boundary point, which is used to approximate voltage stability boundaries. Calculation of the shortest distance between current operating point and the quadratic surface has been formulated.

Let's take the following equation as an example to illustrate the first order and the second order approximation [C1]:

$$\phi(z) = \begin{cases} F(x) + \gamma e_i + \eta e_j = 0 \\ J'(x)L = 0 \\ L'L = 1 \end{cases} \quad (C1)$$

where $z = [x, L, \gamma, \eta]$.

C.1 First Order Approximation

According to (C1a), we have

$$J(x)\Delta x + \frac{\partial F}{\partial \gamma} \Delta \gamma + \frac{\partial F}{\partial \eta} \Delta \eta = 0, \quad (C2)$$

where $J(x) = \frac{\partial F}{\partial x}$. Multiplying L' on both side of (C2), we have

$$L'J(x)\Delta x + L' \frac{\partial F}{\partial \gamma} \Delta \gamma + L' \frac{\partial F}{\partial \eta} \Delta \eta = 0 \quad (C3)$$

According to (C1b), we have $L'J(x) = 0$, and Equation (C3) turns into

$$L' \frac{\partial F}{\partial \gamma} \Delta \gamma + L' \frac{\partial F}{\partial \eta} \Delta \eta = 0 \quad (C4)$$

Let $\Delta\gamma = \gamma - \gamma_*$, $\Delta\eta = \eta - \eta_*$, $n_\Sigma = \begin{bmatrix} L' \frac{\partial F}{\partial \gamma} & L' \frac{\partial F}{\partial \eta} \end{bmatrix}$, then Equation (C4) can be written as follows:

$$n_\Sigma \begin{bmatrix} \gamma - \gamma_* \\ \eta - \eta_* \end{bmatrix} = 0 \quad (\text{C5})$$

Equation (C5) is the tangent plane at the point (γ_*, η_*) where n_Σ represent the normal vector.

C.2 Quadratic Approximation

Based on the normal vector and the tangent plane which we already obtained in previous section, the second-order Taylor expansion of a boundary surface Σ around a point (γ_*, η_*) can be expressed as the following:

$$L(\Delta\gamma) = \begin{bmatrix} \gamma_* \\ \eta_* \end{bmatrix} + C\Delta\gamma + \frac{1}{2} \Pi_{(\gamma_*, \eta_*)} n_\Sigma(\gamma_*, \eta_*) \quad (\text{C6})$$

Here $C = [c_1 \cdots c_{m-1}]$ is a basis for the tangent plane; in two dimensional space $C = [c_1]$, $c_1 = \begin{bmatrix} 1 \\ k \end{bmatrix}$ is the basis for this tangent set, k is the slop of the tangent line; $C\Delta\gamma$ is a displacement away from (γ_*, η_*) in the tangent hyperplane, and $\Pi_{(\gamma_*, \eta_*)}$ is the second fundamental form of Σ at point (γ_*, η_*) as shown in Figure 45.

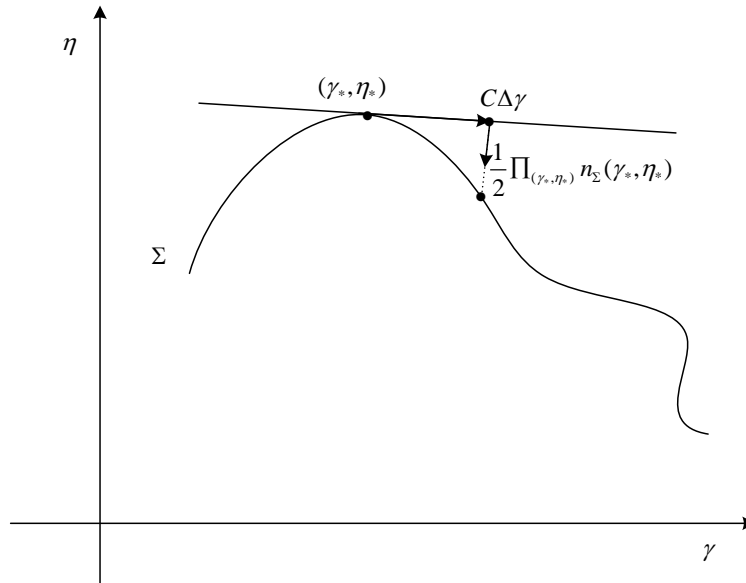


Figure 45: Illustration for quadratic approximation.

For three dimensional space, Equation (C1) will turns into

$$\phi(z) = \begin{cases} F(x) + \gamma e_i + \eta e_j + \kappa e_k = 0 \\ J^t(x)L = 0 \\ L^t L = 1 \end{cases} \quad (C7)$$

and the tangent set in (C5) is a plane shown in

$$n_\Sigma \begin{bmatrix} \gamma - \gamma_* \\ \eta - \eta_* \\ \kappa - \kappa_* \end{bmatrix} = 0 \quad (C8)$$

Here
$$n_\Sigma = \begin{bmatrix} L^t \frac{\partial F}{\partial \gamma} & L^t \frac{\partial F}{\partial \eta} & L^t \frac{\partial F}{\partial \kappa} \end{bmatrix}.$$

According to the plane equation, we can find a vector perpendicular to n_Σ in the tangent

plane, say $c_1 = \begin{bmatrix} c_{11} \\ c_{21} \\ c_{31} \end{bmatrix}$, and another vector $c_2 = \begin{bmatrix} c_{12} \\ c_{22} \\ c_{32} \end{bmatrix}$ which satisfies $c_2 \perp c_1$. Then c_1 and c_2 are two basis of the plane.

Let $C = [c_1, c_2]$, the quadratic approximation in the three dimensional space is shown in

$$L(\Delta\gamma, \Delta\eta, \Delta\kappa) = \begin{bmatrix} \gamma_* \\ \eta_* \\ \kappa_* \end{bmatrix} + C \begin{bmatrix} \Delta\gamma \\ \Delta\eta \end{bmatrix} + \frac{1}{2} \Pi_{(\gamma_*, \eta_*, \kappa_*)} n_\Sigma(\gamma_*, \eta_*, \kappa_*) \quad (C9)$$

The above procedure can be extended to the space with more than three dimensions (Using Gram-Schmidt process to compute C [D2]).

Now let's continue to demonstrate how to calculate Equation (C9). In (C9), according to [D2], we have

$$\Pi_{(\gamma_*, \eta_*)}(\Delta\gamma) = -\langle dN_{(\gamma_*, \eta_*)}(\Delta\gamma), \Delta\gamma \rangle \quad (C10)$$

where

$$dN = \phi_{\lambda_p}^t L^t F_{xx} \phi_{\lambda_p} + \phi_{\lambda_p}^T L^t F_{x\lambda} \phi_{\lambda_p} C + C^T L^t F_{\lambda x} \phi_{\lambda_p} \quad (C11)$$

and

$$x = \phi(\mu) \quad (C12)$$

$$\frac{\partial}{\partial \mu_p} = C^T \frac{\partial}{\partial \mu} \quad (C13)$$

$$\begin{bmatrix} F_x \\ R^T L^T F_{xx} \end{bmatrix} \varphi_{\mu_p} = - \begin{bmatrix} F_\mu \\ R^T L^T F_{x\mu} \end{bmatrix} C \quad (C14)$$

Here F_x is the Jacobian matrix J , $\mu = [\gamma, \eta]$, L and R are the left and right eigenvectors, respectively.

$$L^T F_{xx} = \frac{\partial(L^T F_x)}{\partial x} \quad (C15)$$

Remember

$$F_x = \frac{\partial F}{\partial x} = J \quad (C16)$$

which is the Jacobian matrix.

Substituting (C16) into (C15), we have

$$L^T F_{xx} = \frac{\partial(L^T J)}{\partial x} = \frac{\partial(J^T L)^T}{\partial x} = \left[\frac{\partial(J^T L)}{\partial x} \right]^T \quad (C17)$$

which can be calculated through Hessian matrices (see[19]). Here L is the left eigenvector.

$$L^T F_{x\mu} = \frac{\partial(L^T F_x)}{\partial \mu} = \frac{\partial(L^T J)}{\partial \mu} = 0 \quad (C18)$$

$$L^T F_{x\mu} = \frac{\partial(L^T F_\mu)}{\partial x} = 0 \quad (C19)$$

Substituting (C17), (C18) and (C19) into (C14), we can obtain φ_{μ_p} . And then substitute the obtained φ_{μ_p} , (C17), (C18) and (C19) into (C11), dN can be obtained. Finally,

$\Pi_{(\gamma_*, \eta_*)}(\Delta \gamma) n_\Sigma(\gamma_*, \eta_*)$ can be obtained through solving (C12), and quadratic approximation gets achieved.

C.3 The Shortest Distance between the Current Operating Point and the Surface from Quadratic Approximation

Suppose (γ_0, η_0) represents current operation point, and (γ_a, η_a) represents the points on the approximated surface (the red curve in Figure 46), which can be expressed in

$$\begin{bmatrix} \gamma_a \\ \eta_a \end{bmatrix} = \begin{bmatrix} \gamma_* \\ \eta_* \end{bmatrix} + C\Delta\gamma + \frac{1}{2}\Pi_{(\gamma_*, \eta_*)} n_{\Sigma}(\gamma_*, \eta_*) \quad (C20)$$

according to (C6).

The distance between the current operating point (γ_0, η_0) and the point (γ_a, η_a) on the surface can be expressed as follows.

$$d = \sqrt{(\gamma_a - \gamma_0)^2 + (\eta_a - \eta_0)^2} \quad (C21)$$

$$D = d^2 = (\gamma_a - \gamma_0)^2 + (\eta_a - \eta_0)^2 \quad (C22)$$

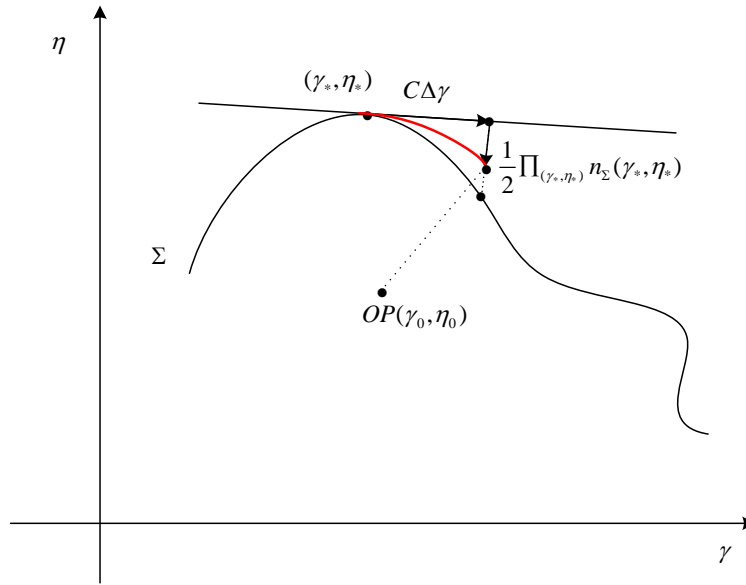


Figure 46: Shortest distance between current operating point and the surface.

Remember the location of point (γ_a, η_a) is a function of $\Delta\gamma$ (see Equation (C20)), the shortest distance can be obtained by solving the following equations.

$$\frac{dD}{d\Delta\gamma} = \frac{\partial D}{\partial \gamma_a} \frac{d\gamma_a}{d\Delta\gamma} + \frac{\partial D}{\partial \eta_a} \frac{d\eta_a}{d\Delta\gamma} = 0 \quad (C23)$$

According to (C22), we have

$$\frac{\partial D}{\partial \gamma_a} = 2(\gamma_a - \gamma_0) \quad (C24)$$

$$\frac{\partial D}{\partial \eta_a} = 2(\eta_a - \eta_0) \quad (\text{C25})$$

According to reference [D3], in (C20) we have

$$\frac{1}{2} \prod_{(\gamma_*, \eta_*)} n_{\Sigma}(\gamma_*, \eta_*) = \frac{1}{2} \rho (\Delta \gamma)^2 \quad (\text{C26})$$

Here ρ is a constant which is calculated by (C11).

According to (C20), we have

$$\gamma_a = \gamma_* + c_{11} \Delta \gamma + \frac{\rho}{2} n_{\Sigma 11} (\Delta \gamma)^2 \quad (\text{C27})$$

$$\eta_a = \eta_* + c_{21} \Delta \gamma + \frac{\rho}{2} n_{\Sigma 21} (\Delta \gamma)^2 \quad (\text{C28})$$

$$\frac{d\gamma_a}{d\Delta \gamma} = c_{11} + \rho n_{\Sigma 11} \Delta \gamma \quad (\text{C29})$$

$$\frac{d\eta_a}{d\Delta \gamma} = c_{21} + \rho n_{\Sigma 21} \Delta \gamma \quad (\text{C30})$$

By substituting (C27), (C28) into (C24) and (C25) respectively, we have

$$\frac{\partial D}{\partial \gamma_a} = 2[\gamma_* + c_{11} \Delta \gamma + \frac{\rho}{2} n_{\Sigma 11} (\Delta \gamma)^2 - \gamma_0] \quad (\text{C31})$$

$$\frac{\partial D}{\partial \eta_a} = 2[\eta_* + c_{21} \Delta \gamma + \frac{\rho}{2} n_{\Sigma 21} (\Delta \gamma)^2 - \eta_0] \quad (\text{C32})$$

Substitute (C29) ~ (C32) into (C23), $\Delta \gamma$ can be obtained, and then substitute the obtained $\Delta \gamma$ into (C30) and (C31), the shortest distance can be calculated. The above algorithm can be easily extended to multi-parameter space.

C.4 References

- [C1] M. Perninge and L. Soder, "On the Validity of Local Approximations of the Power System Loadability Surface," Power Syst. IEEE Trans., vol. 26, no. 4, pp. 2143–2153, Nov. 2011.
- [C2] J. Lavenius, "An algorithm for find minimal load interruption costs," dissertation, Royal Institute of Technology Stockholm, Sweden 2012

- [C3] C. Hamon, M. Perninge, and L. Soder, “A Stochastic Optimal Power Flow Problem With Stability Constraints—Part I: Approximating the Stability Boundary,” *Power Syst. IEEE Trans.*, vol. 28, no. 2, pp. 1839–1848, May 2013.

Appendix D: Development of Input File Using SimAuto

This appendix details the method and code used to convert a PowerWorld system model to a format compatible with the MATLAB code used in this project.

Step1: Connect PowerWorld SimAuto and open an input system that is saved as PowerWorld file.

Commands used in the script:

```
A = actxserver('pwrworld.SimulatorAuto');  
pwFile = strcat(pwd, '\14HS_c4800_d3102_n8000_m-400_new_V4.PWB');  
simOutput = A.OpenCase(pwFile);
```

Step2: Find fields contained within a given object type for which the fields are requested. The fields contained within branch, bus, generator and line shunts are found by using the following functions:

```
%% Object type to obtain BRANCH field list  
objecttype = 'branch';  
% Make the GetField call  
Output_branch = A.GetFieldList(objecttype);  
  
%% Object type to obtain BUS field list  
objecttype = 'bus';  
% Make the GetField call  
Output_bus = A.GetFieldList(objecttype);  
  
%% Object type to obtain GEN field list  
objecttype = 'gen';  
% Make the GetField call  
Output_gen = A.GetFieldList(objecttype);  
%% Object type to obtain GEN field list  
  
objecttype = 'lineshunt';  
% Make the GetField call  
Output_ls = A.GetFieldList(objecttype);  
%% Object type to obtain GEN field list
```

Step3: Extract specified parameters from “Bus” field array.

Bus field list:

```
fieldarray = {'BusNum ', 'BusCat', 'BusLoadMW', 'BusLoadMVR', 'BusG', 'BusB',
'AreaNum', 'BusPUVolt', 'BusAngle', 'BusNomVolt', 'ZoneNum',
'BusVoltLimHigh', 'BusVoltLimLow', 'BusName', 'BusSS'};
```

The following adjustments were made after extracting the specified parameters from object type Bus:

- Discard ‘Dead’ busses
- Enumerate bus types
- Replace ‘NaN’ with zero
- Accumulate generator data having same bus number
- Incorporate line shunt and switched shunt information

Step4: Extract specified parameters from “Gen” field array.

Generator field list:

```
fieldarray = {'BusNum', 'GenMW', 'GenMVR', 'GenMVRMax', 'GenMVRMin',
'GenVoltSet', 'GenID', 'GenStatus', 'GenMWMax', 'GenMWMin'};
```

All the inactive generators are excluded after extracting the specified parameters from the object type Gen.

Step5: Extract specified parameters from “Branch” field array.

Branch field list:

```
fieldarray = {'BusNum', 'BusNum:1', 'LineR', 'LineX', 'LineC', 'LineAMVA',
'LineAMVA:1', 'LineAMVA:2', 'LineTap', 'LinePhase',
'LineStatus', 'LineCircuit'};
```

The following adjustments were made after extracting the specified parameters from object type Branch:

- Exclude open branches
- Update R, X values for those branches having phase shift with new values saved in “impedance correction table

Appendix E: Jacobian Matrix Transformation from Polar to Rectangular Coordinates

The following bus numbers are used:

PV_l - l -th PV bus number (nPV buses total);

RB - Reference Bus, and

PSERC Variables $[2*\text{nbus}+1+\text{nPV}+1]$:

$$\begin{aligned}
 \boldsymbol{\theta} & \quad - \text{Voltage Phase Angles [nbus]} \\
 \mathbf{V} & \quad - \text{Voltage Magnitudes [nbus]} \\
 \beta_1 & \quad - \text{Source Factors (Distributed Slack Bus Factors) [1]} \\
 \mathbf{Q}_{PV} & \quad - \text{PV Bus Reactive Power Injections [nPV]} \\
 \beta_2 & \quad - \text{Sink Factors (Stress Factors) [1]}
 \end{aligned} \tag{E1}$$

These variables are described by a common notation \mathbf{x} .

$$\mathbf{F}(\mathbf{x}) = \begin{cases} \mathbf{P}(\boldsymbol{\theta}, \mathbf{V}) - \mathbf{P}_D - \mathbf{P}_G \beta_1 = 0 & - \text{Active Power Balance [nbus]} \\ \mathbf{Q}(\boldsymbol{\theta}, \mathbf{V}) + \mathbf{Q} = 0 & - \text{Reactive Power Balance [nbus]} \\ \Delta \boldsymbol{\theta}_{RB} = 0 & - \text{Reference Bus Equation [1]} \\ \Delta \mathbf{V}_{PV} = 0 & - \text{PV Bus Equations [nPV]} \end{cases} \tag{E2}$$

where $\mathbf{x} = [\boldsymbol{\theta}; \mathbf{V}; \beta_1; \mathbf{Q}_{PV}]$.

The Jacobian matrix $\mathbf{J}(\mathbf{x})$ of is the PSERC “full Jacobian”

$$\mathbf{J}(\mathbf{x}) = \begin{bmatrix} \frac{\partial \mathbf{P}}{\partial \boldsymbol{\theta}} & \frac{\partial \mathbf{P}}{\partial \mathbf{V}} & -\mathbf{P}_G & 0 \\ \frac{\partial \mathbf{Q}}{\partial \boldsymbol{\theta}} & \frac{\partial \mathbf{Q}}{\partial \mathbf{V}} & 0 & \mathbf{I}_{PV} \\ 100 \cdot \mathbf{e}_{RB}^T & 0 & 0 & 0 \\ 0 & 100 \cdot \mathbf{I}_{PV}^T & 0 & 0 \end{bmatrix} \quad (\text{E3})$$

where $\mathbf{I}_{PV} = \frac{\partial \mathbf{Q}}{\partial \mathbf{Q}_{PV}} = \left(\frac{\partial \mathbf{V}_{PV}}{\partial \mathbf{V}} \right)^t$ is a (nbus \times nPV) matrix which consists of 0s and 1s. The “100” multiplier is introduced to improve the Jacobian matrix condition.

Let $\mathbf{y} = [\mathbf{V}'; \mathbf{V}''; \beta_1; \mathbf{Q}_{PV}]$,

(E4)

$$\begin{aligned}
\text{where } \frac{\partial \boldsymbol{\theta}}{\partial \mathbf{V}'} &= \begin{bmatrix} \frac{\partial \boldsymbol{\theta}_1}{\partial \mathbf{V}_1'} & 0 & \dots & 0 \\ 0 & \frac{\partial \boldsymbol{\theta}_2}{\partial \mathbf{V}_2'} & \dots & 0 \\ \vdots & \dots & \ddots & \vdots \\ 0 & 0 & \dots & \frac{\partial \boldsymbol{\theta}_{nbus}}{\partial \mathbf{V}_{nbus}'} \end{bmatrix}, \quad \frac{\partial \boldsymbol{\theta}}{\partial \mathbf{V}''} = \begin{bmatrix} \frac{\partial \boldsymbol{\theta}_1}{\partial \mathbf{V}_1''} & 0 & \dots & 0 \\ 0 & \frac{\partial \boldsymbol{\theta}_2}{\partial \mathbf{V}_2''} & \dots & 0 \\ \vdots & \dots & \ddots & \vdots \\ 0 & 0 & \dots & \frac{\partial \boldsymbol{\theta}_{nbus}}{\partial \mathbf{V}_{nbus}''} \end{bmatrix}, \\
\frac{\partial \mathbf{V}}{\partial \mathbf{V}'} &= \begin{bmatrix} \frac{\partial \mathbf{V}_1}{\partial \mathbf{V}_1'} & 0 & \dots & 0 \\ 0 & \frac{\partial \mathbf{V}_2}{\partial \mathbf{V}_2'} & \dots & 0 \\ \vdots & \dots & \ddots & \vdots \\ 0 & 0 & \dots & \frac{\partial \mathbf{V}_{nbus}}{\partial \mathbf{V}_{nbus}'} \end{bmatrix}, \quad \frac{\partial \mathbf{V}}{\partial \mathbf{V}''} = \begin{bmatrix} \frac{\partial \mathbf{V}_1}{\partial \mathbf{V}_1''} & 0 & \dots & 0 \\ 0 & \frac{\partial \mathbf{V}_2}{\partial \mathbf{V}_2''} & \dots & 0 \\ \vdots & \dots & \ddots & \vdots \\ 0 & 0 & \dots & \frac{\partial \mathbf{V}_{nbus}}{\partial \mathbf{V}_{nbus}''} \end{bmatrix}, \quad \frac{\partial \boldsymbol{\theta}_i}{\partial \mathbf{V}_i'} = \frac{-\mathbf{V}_i''}{(\mathbf{V}_i')^2 + (\mathbf{V}_i'')^2}, \\
\frac{\partial \boldsymbol{\theta}_i}{\partial \mathbf{V}_i''} &= \frac{\mathbf{V}_i'}{(\mathbf{V}_i')^2 + (\mathbf{V}_i'')^2}, \quad \frac{\partial \mathbf{V}_i}{\partial \mathbf{V}_i'} = \frac{\mathbf{V}_i'}{\sqrt{(\mathbf{V}_i')^2 + (\mathbf{V}_i'')^2}}, \quad \frac{\partial \mathbf{V}_i}{\partial \mathbf{V}_i''} = \frac{\mathbf{V}_i''}{\sqrt{(\mathbf{V}_i')^2 + (\mathbf{V}_i'')^2}}
\end{aligned}$$

Appendix F: SimAuto and MATLAB® Scripts

The scripts provided in this appendix automate reading of PowerWorld input files to make adjustments to the input parameters needed in the MATLAB code.

F.1 SimAuto Script

```
clc
clear all

%% Establish a connection with PowerWorld / SimAuto

disp('>> Connecting to PowerWorld Simulator / SimAuto...')
A = actxserver('pwrworld.SimulatorAuto');
disp('Connection established')

%% Open case

disp('>> Opening case');
pwFile = strcat(pwd, '\14HS_c4800_d3102_n8000_m-400_new_V4.PWB');
simOutput = A.OpenCase(pwFile);
if ~(strcmp(simOutput{1}, ''))
    disp(simOutput{1})
else
    disp('Case opened');
    validcase = true;
end

% This example loads all buses in the case, and then gets
% some parameters of the last bus in the list
%% Object type to obtain BRANCH field list

objecttype = 'branch';

% Make the GetField call

Output_branch = A.GetFieldList(objecttype);

%% Object type to obtain BUS field list
```

```

objecttype = 'bus';

% Make the GetField call
Output_bus = A.GetFieldList(objecttype);

%% Object type to obtain GEN field list
objecttype = 'gen';

% Make the GetField call
Output_gen = A.GetFieldList(objecttype);
%% Object type to obtain GEN field list

objecttype = 'lineshunt';

% Make the GetField call
Output_ls = A.GetFieldList(objecttype);

%% Object type to obtain GEN field list

% validcase is a global variable - check case is open
if validcase

% Gets all buses in the case

output = A.ListOfDevices('bus','');

    if ~(strcmp(output{1},''))

        disp(output{1});

        validbusarray = false;

    else

% Puts the buses in row vector busarray

        for i=size(output{2}{1},1):size(output{2}{1},2)

            busarray(i,1) = output{2}{1}(i);

        end

        disp('Successful ListOfDevices')
%         disp(busarray);
        validbusarray = true;
    end
end

```

```

% validbusarray is a global variable - check buses are loaded

if validcase && validbusarray

    % Gets parameters for last bus of busarray

    fieldarray = {'BusNum ', 'BusCat', 'BusLoadMW', 'BusLoadMVR', 'BusG',
'BusB', 'AreaNum', 'BusPUVolt', 'BusAngle', 'BusNomVolt', 'ZoneNum',
'BusVoltLimHigh', 'BusVoltLimLow', 'BusName', 'BusSS'};

    for bus_count=1:size(busarray,1);

        valuearray =
{busarray(bus_count), '0', '0', '0', '0', '0', '0', '0', '0', '0', '0', '0', '0', '0', '0'}
;

        output(:,bus_count) =
A.GetParametersSingleElement('bus',fieldarray,valuearray);

        if ~(strcmp(output{1}, ''))

            disp(output{1}) ;

        else

            % Puts the buses in row vector busparam

            paramlist = transpose(output{2});

            for i=size(paramlist,1):size(paramlist,2)

                busparam(i,1) = paramlist(i);

            end

            disp('Succesful GetParameters for Bus')

%         disp(fieldarray);

            disp(busparam) ;

        end

    end
end

count=0;
for ij=1:size(busarray,1)
    buscon=strcmp(output{2,ij}{2,1}, 'Dead');
    if buscon==0
        count=count+1;
        bus_data (count,1:14)=[ output{2,ij}(1,1) output{2,ij}(2,1)
output{2,ij}(3,1) output{2,ij}(4,1) output{2,ij}(5,1) output{2,ij}(6,1)

```

```

output{2,ij}(7,1) output{2,ij}(8,1) output{2,ij}(9,1) output{2,ij}(10,1)
output{2,ij}(11,1) output{2,ij}(12,1) output{2,ij}(13,1) output{2,ij}(15,1)];
    end
end

count=0;
for ij=1:size(bus_data,1)
    busconshunt=strcmp(num2str(bus_data{ij,14}), 'NaN');
    if busconshunt==0

%         bus_data_shunt(ij,1)=bus_data(ij,14);
        bus_data_shunt(ij)=num2cell(str2num(bus_data{ij,6})+
str2num(bus_data{ij,14}));
    else
%         bus_data_shunt(ij,1)=bus_data(ij,14);
        bus_data_shunt(ij)=bus_data(ij,6);
    end
end
bus_data(1:ij,6)=bus_data_shunt(1:ij)';

for i=1:size(bus_data,1)
    PQcheck=strcmp(bus_data(i,2), 'PQ');
    PQcheckRR=strcmp(bus_data(i,2), 'PQ (Remote Reg Slave)');
    slackcheck=strcmp(bus_data(i,2), 'Slack');
    PVcheck=strcmp(bus_data(i,2), 'PV');

    PQcheckgen=strcmp(bus_data(i,2), 'PQ (Gens at Var Limit)');

    PVcheckRR=strcmp(bus_data(i,2), 'PV (Local/Remote Reg Master)');

    PQcheckRRV=strcmp(bus_data(i,2), 'PQ (Remotely Regulated at Var Limit)');

    PQcheckCSV=strcmp(bus_data(i,2), 'PQ (Continuous Shunts at Var Limit)');

%     PQcheck=strcmp(bus_data(1:ij,1), 'PQ');

if (PQcheck==1 || PQcheckRR==1 || PQcheckRRV==1)
    bus_data(i,2)=num2cell(1);
elseif (PVcheck==1 || PVcheckRR==1 || PQcheckgen==1 || PQcheckCSV==1)
    bus_data(i,2)=num2cell(2);
elseif slackcheck==1
    bus_data(i,2)=num2cell(3);

end

end

for ik=1:size(bus_data,1)
    nanchek=strcmp(str2num(bus_data{ik,1}), 'NaN');
    bus_data_new(ik,1)=str2num(bus_data{ik,1});
    if nanchek==1
        bus_data_new(ik,1)=0;
    end
end

```



```

end

for ik=1:size(bus_data,1)
    bus_data_new(ik,2)=(bus_data{ik,2});
end

for ik=1:size(bus_data,1)
    nanchek=strcmp(num2str(bus_data{ik,3}), 'NaN');
    if nanchek==1
        bus_data_new(ik,3)=0;
    else
        bus_data_new(ik,3)=str2num(bus_data{ik,3});
    end
end

for ik=1:size(bus_data,1)
    nanchek=strcmp(num2str(bus_data{ik,4}), 'NaN');
    if nanchek==1
        bus_data_new(ik,4)=0;
    else
        bus_data_new(ik,4)=str2num(bus_data{ik,4});
    end
end

for ik=1:size(bus_data,1)
    for il=5:13

        if il==6
            zerocheck=strcmp(num2str(bus_data{ik,6}), ' 0.0000000');

            if zerocheck==1
                bus_data_new(ik,6)=0;
            else
                if isstr(bus_data{ik,6})==1

                    bus_data_new(ik,6)=str2num(bus_data{ik,6});
                else
                    bus_data_new(ik,6)=(bus_data{ik,6});
                end
            end
        end

        else
            bus_data_new(ik,il)=str2num(bus_data{ik,il});
        end
    end
end

end

% IF(B3="PQ",1,IF(B3="PV",2,IF(B3="Slack",3,IF(B3="PQ (Gens at Var
Limit)",1,IF(B3="PQ (Remote Reg Slave)",1,IF(B3="PV (Local/Remote Reg
Master)",2,IF(B3="Dead",4,"ERROR")))))
count=0;

for ij=1:size(busarray,1)
    buscon=strcmp(output{2,ij}{2,1}, 'Dead');
    if buscon==0
        count=count+1;
    end
end

```

```

        bus_names (count,1)= output{2,ij}(14);
    end
end

%%Branch information
% validbusarray is a global variable - check buses are loaded

% validcase is a global variable - check case is open

if validcase

% Gets all buses in the case

output_gen = A.ListOfDevices('gen','');

    if ~(strcmp(output_gen{1}, ''))

        disp(output_gen{1});

        validbusarray = false;

    else

% Puts the buses in row vector busarray

        for i=size(output_gen{2}{1},1):size(output_gen{2}{1},2)

            genarray(i,1) = output_gen{2}{1}(i);
            genarrays(i) = output_gen{2}{2}(i);

        end
        disp('Successful ListOfDevices')
%         disp(genarray);
        validgenarray = true;
    end
end

% validbusarray is a global variable - check buses are loaded
% genarrays = output_gen{2}{2}(1:939);
if validcase && validbusarray

% Gets parameters for last bus of busarray

fieldarray = {'BusNum', 'GenMW', 'GenMVR', 'GenMVRMax', 'GenMVRMin',
'GenVoltSet', 'GenID', 'GenStatus', 'GenMWMax', 'GenMWMin'};

for gen_count=1:size(genarray,1);

    valuearray =
{genarray(gen_count), '0', '0', '0', '0', '0', char(output_gen{2}{2}(gen_count)), '0
', '0', '0'};

```

```

        output_gen_final(:,gen_count) =
A.GetParametersSingleElement('gen',fieldarray,valuearray);

        if ~(strcmp(output_gen_final{1},''))

            disp(output_gen_final{1}) ;

        else

            % Puts the buses in row vector genparam

            paramlist = transpose(output_gen_final{2});

            for i=size(paramlist,1):size(paramlist,2)

                genparam(i,1) = paramlist(i);

            end

            disp('Successful GetParameters for Gen')

%            disp(fieldarray);

            disp(genparam) ;

        end

    end
end
count=0;
for ij=1:size(genarray,1)
    gencon=strcmp(output_gen_final{2,ij}{8},'Open');

    if gencon==0
        count=count+1;
        gen_data (count,1:10)=[ output_gen_final{2,ij}(1,1)
output_gen_final{2,ij}(2,1) output_gen_final{2,ij}(3,1)
output_gen_final{2,ij}(4,1) output_gen_final{2,ij}(5,1)
output_gen_final{2,ij}(6,1) 100 output_gen_final{2,ij}(8)
output_gen_final{2,ij}(9,1) output_gen_final{2,ij}(10,1)];
    end
end

for ik=1:size(gen_data,1)
    for il=1:size(gen_data,2)
        if il==7
            gen_data_new(ik,7)=cell2mat(gen_data(ik,7));
        elseif il==8
            gencheck=strcmp(gen_data(ik,il),'Closed');
            if gencheck==1
                gen_data_new(ik,8)=1;%this should be 'closed'
            else
                gen_data_new(ik,8)=0;
            end
        else
    end
end

```

```

        gen_data_new(ik,il)=str2num(gen_data{ik,il});
    end

end

end

if validcase

% Gets all buses in the case

output_br = A.ListOfDevices('branch','');

    if ~(strcmp(output_br{1},''))

        disp(output_br{1})

        validbusarray = false;

    else

% Puts the buses in row vector busarray

        for i=size(output_br{2}{1},1):size(output_br{2}{1},2)

            brancharray(i,1) = output_br{2}{1}(i);
            brancharray(i,2) = output_br{2}{2}(i);

        end
        disp('Successful ListOfDevices')
%        disp(brancharray);
        validbrancharray = true;
    end
end

if validcase && validbrancharray

% Gets parameters for last bus of busarray

    fieldarray = {'BusNum', 'BusNum:1', 'LineR', 'LineX', 'LineC', 'LineAMVA',
'LineAMVA:1', 'LineAMVA:2', 'LineTap', 'LinePhase',
'LineStatus', 'LineCircuit'};

    for branch_count=1:size(brancharray,1);

        valuearray =
[output_br{2}{1}(branch_count),output_br{2}{2}(branch_count),'0','0','0','0',
'0','0','0','0','0',char(output_br{2}{3}(branch_count))];

        output_br_final(:,branch_count) =
A.GetParametersSingleElement('branch',fieldarray,valuearray);

        if ~(strcmp(output_br_final{1},''))

```

```

disp(output_br_final{1})

else

% Puts the buses in row vector branchparam

paramlist_br = transpose(output_br_final{2});

    for i=size(paramlist_br,1):size(paramlist_br,2)

        branchparam(i,1) = paramlist_br(i);

    end

disp('Successful GetParameters for Branch')

%    disp(fieldarray);

disp(branchparam) ;

end

end
end
%end of branch data
count=0;
for ij=1:size(brancharray,1)
    branchcon=strcmp(output_br_final{2,ij}{11}, 'Open');
    if branchcon==0
        count=count+1;
        branch_data (count,1:13)=[ output_br_final{2,ij}(1,1)
output_br_final{2,ij}(2,1) output_br_final{2,ij}(3,1)
output_br_final{2,ij}(4,1) output_br_final{2,ij}(5,1)
output_br_final{2,ij}(6,1) output_br_final{2,ij}(7,1)
output_br_final{2,ij}(8,1) output_br_final{2,ij}(9,1)
output_br_final{2,ij}(10,1) output_br_final{2,ij}(11,1) -180 180];
        branch_data_ckt (count,1:3)=[ output_br_final{2,ij}(1,1)
output_br_final{2,ij}(2,1) output_br_final{2,ij}(12,1)];
    end
end
for ik=1:size(branch_data,1)
    for il=1:size(branch_data,2)
        if il==11
            brnachchk=strcmp(branch_data(ik,il), 'Closed');
            if brnachchk==1;
                branch_data_new(ik,il)=1;%this should be 'closed'
            else
                branch_data_new(ik,il)=0;
            end
        elseif il==12
            branch_data_new(ik,il)=-180;
        elseif il==13
            branch_data_new(ik,il)=180;
        else
            branch_data_new(ik,il)=str2num(branch_data{ik,il});
        end
    end
end

```

```
end
end
```

F.2 MATLAB® Script Used to Calculate Several Input Parameters

```
% clear,clc
mpc=Data_Input_WECC_2014_v2;

Bus.NAME=mpc.busnames;
Bus.NUMBER=mpc.bus(:,1);
Bus.Spuload=(mpc.bus(:,3)+1i*mpc.bus(:,4))/mpc.baseMVA;
% calculate pu values for YL

mpc.bus(:,5)= mpc.bus(:,5)/mpc.baseMVA;
mpc.bus(:,6) = mpc.bus(:,6)/mpc.baseMVA;

% mpc.bus(:,6) = mpc.bus(:,6)./abs(mpc.Bus.V)^2

Bus.YL=mpc.bus(:,5)+1i*mpc.bus(:,6);
Bus.type=mpc.bus(:,2);
Bus.slack_factor=zeros(length(mpc.bus(:,1)),1);
Bus.slack_factor(Bus.type==3)=1;
Bus.Vlow=mpc.bus(:,13);
Bus.Vmax=mpc.bus(:,12);
Bus.V=mpc.bus(:,8).*exp(1i*mpc.bus(:,9)/180*pi);
Bus.KV=mpc.bus(:,10);
Bus.AREA=mpc.bus(:,7);
Bus.ZONE=mpc.bus(:,11);

index=zeros(length(mpc.gen(:,1)),1);
for i=1:length(mpc.gen(:,1))
    index(i)=find(Bus.NUMBER==mpc.gen(i,1));
    index_num(i)=mpc.gen(i,1);
end
number.count=unique(index);
number.rep=histc(index,number.count);

number.count_bus=unique(index_num);
number.rep_bus=histc(index_num,number.count_bus);

count_rep=[number.count number.rep];
[gen_f,gen_b,gen_r] = xlsread('Gendata.xlsx','Sheet2','A1:C2333');

Bus.Spugen=zeros(length(mpc.bus(:,1)),1);
Bus.Qgmax=zeros(length(mpc.bus(:,1)),1);
Bus.Qgmin=zeros(length(mpc.bus(:,1)),1);
Bus.Pgmax=zeros(length(mpc.bus(:,1)),1);
Bus.Pgmin=zeros(length(mpc.bus(:,1)),1);

for gc=1:length(number.count)

    if gc>1
        Bus.Spugen(count_rep(gc,1))=(nansum(mpc.gen(gen_f(gc-1,3)+1:gen_f(gc,3),2))+1i*nansum(mpc.gen(gen_f(gc-1,3)+1:gen_f(gc,3),3)))/mpc.baseMVA;

        Bus.Qgmax(count_rep(gc,1))=nansum(mpc.gen(gen_f(gc-1,3)+1:gen_f(gc,3),4))/mpc.baseMVA;
```

```

        Bus.Qgmin(count_rep(gc,1))=nansum(mpc.gen(gen_f(gc-
1,3)+1:gen_f(gc,3),5))/mpc.baseMVA;

        Bus.Pgmax(count_rep(gc,1))=nansum(mpc.gen(gen_f(gc-
1,3)+1:gen_f(gc,3),9))/mpc.baseMVA;

        Bus.Pgmin(count_rep(gc,1))=nansum(mpc.gen(gen_f(gc-
1,3)+1:gen_f(gc,3),10))/mpc.baseMVA;
    else

Bus.Spugen(count_rep(gc,1))=(mpc.gen(gen_f(gc,3),2)+1i*mpc.gen(gen_f(gc,3),3)
)/mpc.baseMVA;

        Bus.Qgmax(count_rep(gc,1))=mpc.gen(gen_f(gc,3),4)/mpc.baseMVA;

        Bus.Qgmin(count_rep(gc,1))=mpc.gen(gen_f(gc,3),5)/mpc.baseMVA;

        Bus.Pgmax(count_rep(gc,1))=mpc.gen(gen_f(gc,3),9)/mpc.baseMVA;

        Bus.Pgmin(count_rep(gc,1))=mpc.gen(gen_f(gc,3),10)/mpc.baseMVA;
    end

end

Bus.mismatch=zeros(length(mpc.bus(:,1)),1);
%%
Branch.I=mpc.branch(:,1);%by Di
Branch.J=mpc.branch(:,2);% by Di

for i=1:length(Branch.I)
    index1=find(Bus.NUMBER==Branch.I(i));
    index2=find(Bus.NUMBER==Branch.J(i));
    Branch.NAME(i,:)=[mpc.busnames(index1,:) ' - ' mpc.busnames(index2,:)];
end
Branch.Z=mpc.branch(:,3)+1i*mpc.branch(:,4);
Branch.B=mpc.branch(:,5);
Branch.YI=zeros(length(Branch.I),1);
Branch.YJ=zeros(length(Branch.I),1);
Branch.TAP=mpc.branch(:,9).*exp(1i*mpc.branch(:,10)/180*pi); %??? how about
the angle in column 10
Branch.STATUS=mpc.branch(:,11);
Branch.RATES=mpc.branch(:,6:8)/mpc.baseMVA; % previously it was
Branch.RATES=mpc.branch(:,6:8);Modified by Bharat
Branch.IpuAmps=zeros(length(Branch.I),1);
Branch.JpuAmps=zeros(length(Branch.I),1);
Branch.IMVA=zeros(length(Branch.I),1);
Branch.JMVA=zeros(length(Branch.I),1);

for i=1:length(mpc.branch)%Modified by Bharat
    branchI(i,1)=find(mpc.bus==mpc.branch(i,1));%Modified by Bharat
    branchJ(i,2)=find(mpc.bus==mpc.branch(i,2));%Modified by Bharat

```



```
end%Modified by Bharat

Branch.I=branchI(:,1);%by Bharat
Branch.J=branchJ(:,2);% by Bharat
clear i index index1 index2
```


Appendix G: Additional Procedures and Results

The experimental program for exploring the VSB was tested on the a modified version of the WECC 2020 planning model, which contains 21 areas, 17,939 buses, 3,525 generators, and 8,596 loads. The original system was modified so that the proposed method could be applied, such as by removing constant impedance loads and disabling remotely regulated buses.

G.1 Performance Comparison with PowerWorld

The boundary points were found using: (1) a step-by-step approach using the PowerWorld software package , (2) eigenvalue-based detection of singularities, and (3) our new approach developed in this project to detect the singularities.

A traditional step-by-step approach was implemented using PowerWorld in order to explore the VSB points. The following steps were used in generating boundary points for a total of 200 cases using the modified WECC planning model:

- **Step 1:** Start from an initial operating point. Select two different areas from the model such that one area has more generation than the total load (Area 1) and the other area has less generation than the total load (Area 2).
- **Step 2:** Increase the generation in Area 1 and decrease the generation in Area 2 with an initial step size.
- **Step 3:** Run a power flow. Repeat Step 2 and Step 3 until the power flow diverges. If the power flow diverges, go to Step 4.
- **Step 4:** Divide the step size by two and repeat Step 2 and Step 3 until the size becomes smaller than a specified precision. If the step size is smaller than the specified precision and the power flow diverges, then save: (1) the Jacobian and Y-bus matrices, (2) voltage magnitudes and angles of each bus of the system, and (3) simulation time to reach the boundary point.
- **Step 5:** Repeat Steps 1 through, selecting different areas of the system.

The total average simulation time was calculated using simulation times of these 200 cases, shown in Table 6. The information obtained using this method is used in the other two methods.

Table 6. Comparison of simulation time (in seconds) of different methods

Method		Step-by-step iterative method	Eigenvalue-based non-iterative method	PNNL method	Time to solve a linear system
Single processor machine	Single point	133.57	11.39	2.61	0.1-0.12
	200 point	26714	2278	522	
200-processor machine	Extrapolated time	--	2278	522	--
	Actual time for 200 point	--	27.4	3.5	--

G.2 Impacts of Numerical Precision

In Phase I of this project, we performed some experiments to confirm whether the multi-singularity behaviors we found with the WECC 2020 planning model were due to numerical precision issues. We evaluated the impacts of numerical precision on the function (JRnorm) behavior for the two test cases with and without storing the Jacobian matrices from PowerWorld in scientific notation.

At our request, PowerWorld modified the source code and added extra features to save the Jacobian in scientific notation as opposed to fixed decimal notation. Previously, PowerWorld allowed us to save the Jacobian matrix with five digits to the right of decimal. The new feature allows scientific notation, but only to seven significant digits. This is because PowerWorld internally stores the values as single precision. That means that it can only show seven significant digits in scientific notation. If it shows more digits (for example, 2.1345679845), the extra digits will be a random value because it will be whatever value the computer decided to put after the 7th significant digit. For small numbers, the impact of the additional significant digits is proportionally significant; for example, with fixed decimal notation a number might be truncated to 0.0001223, while with 7 significant digits of scientific notation it is 1.223546e-4.

Figure 47, Figure 48, and Figure 49 show the same Jacobian data for the same computation, but with values stored to seven, six, and five significant digits respectively. Figure 49 looks substantially different, and shows “singularities” in the data not present in Figure 47 and Figure 48.

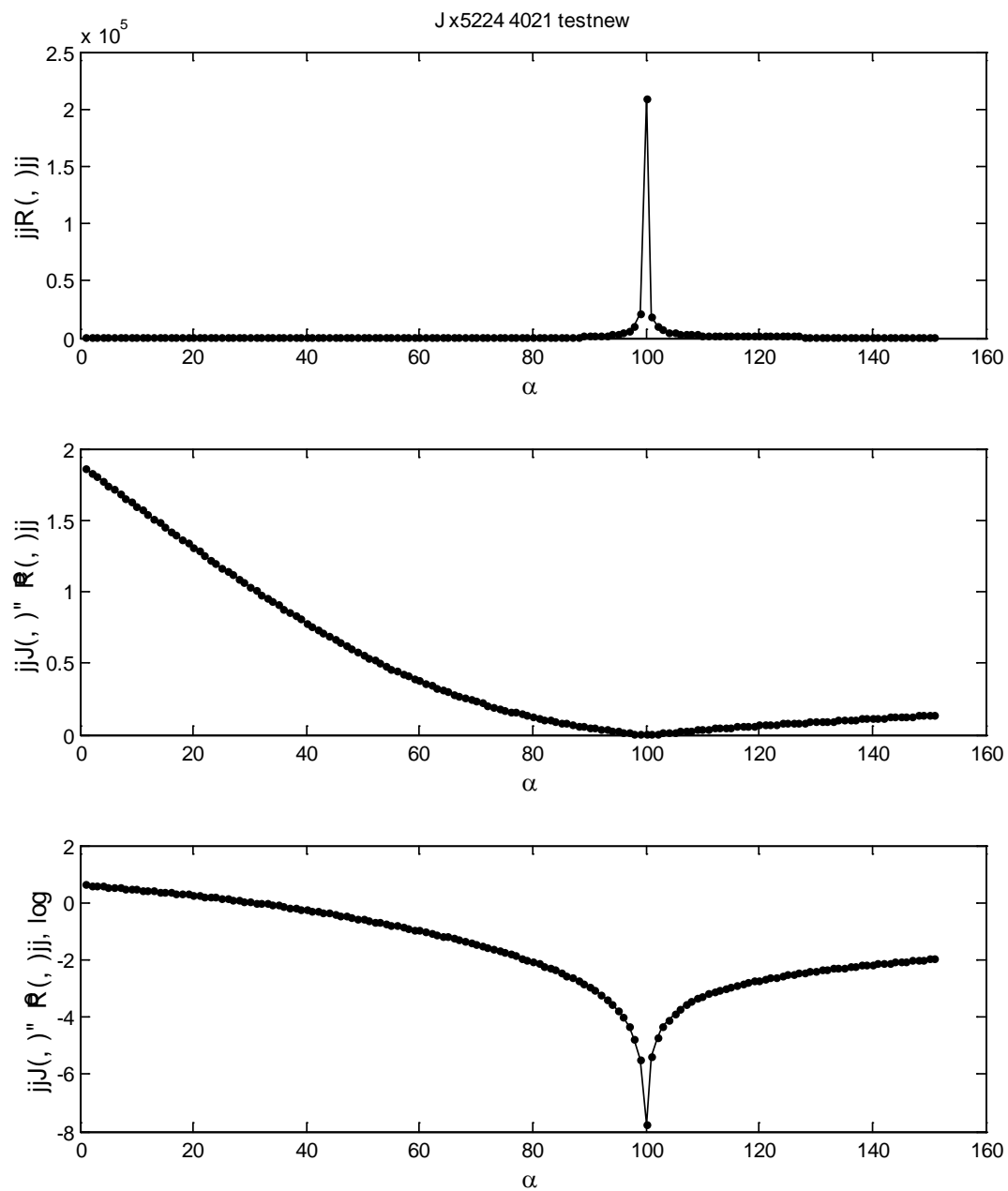


Figure 47: Original precision with seven significant digits.

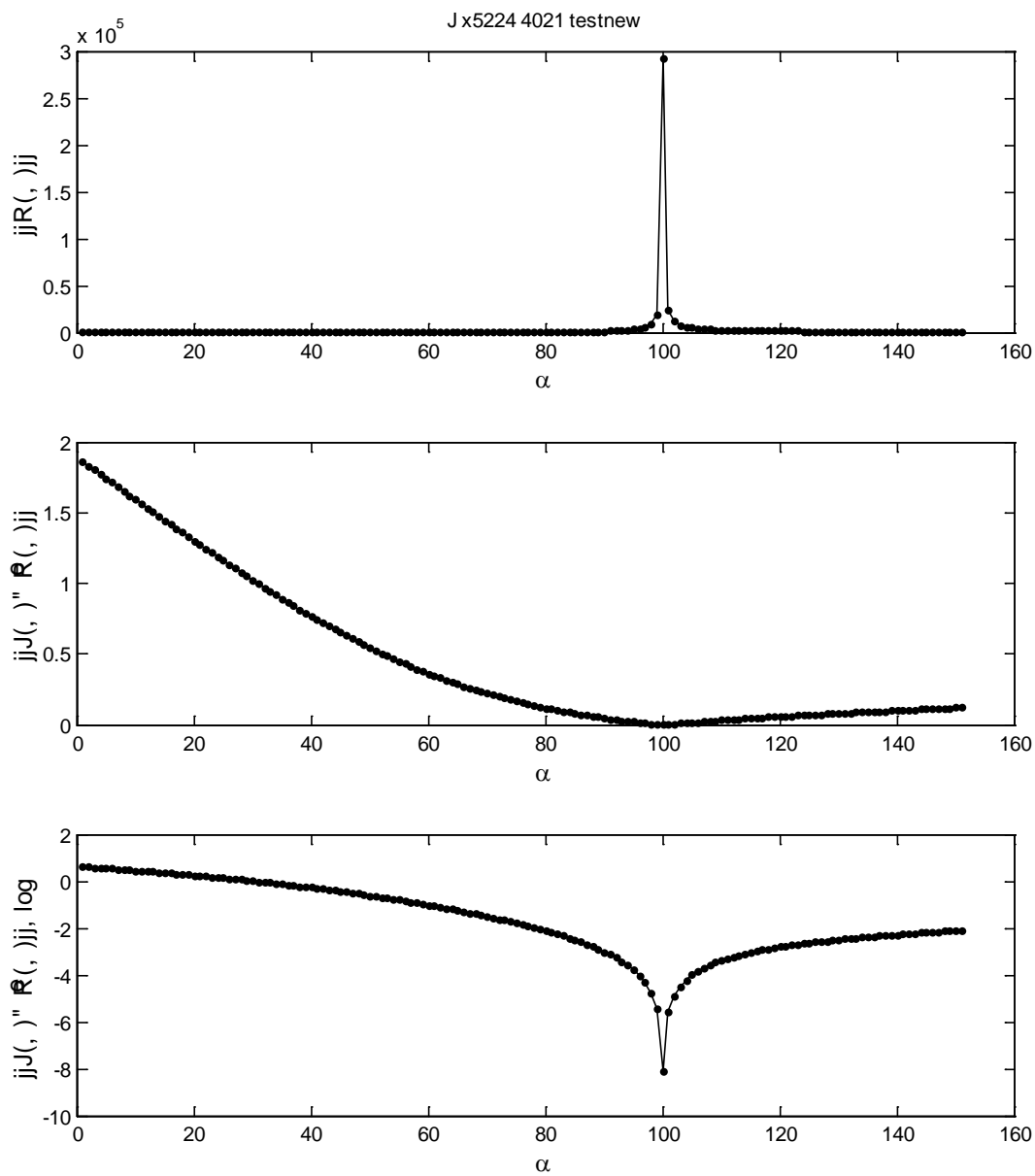


Figure 48: Original precision with six significant digits.

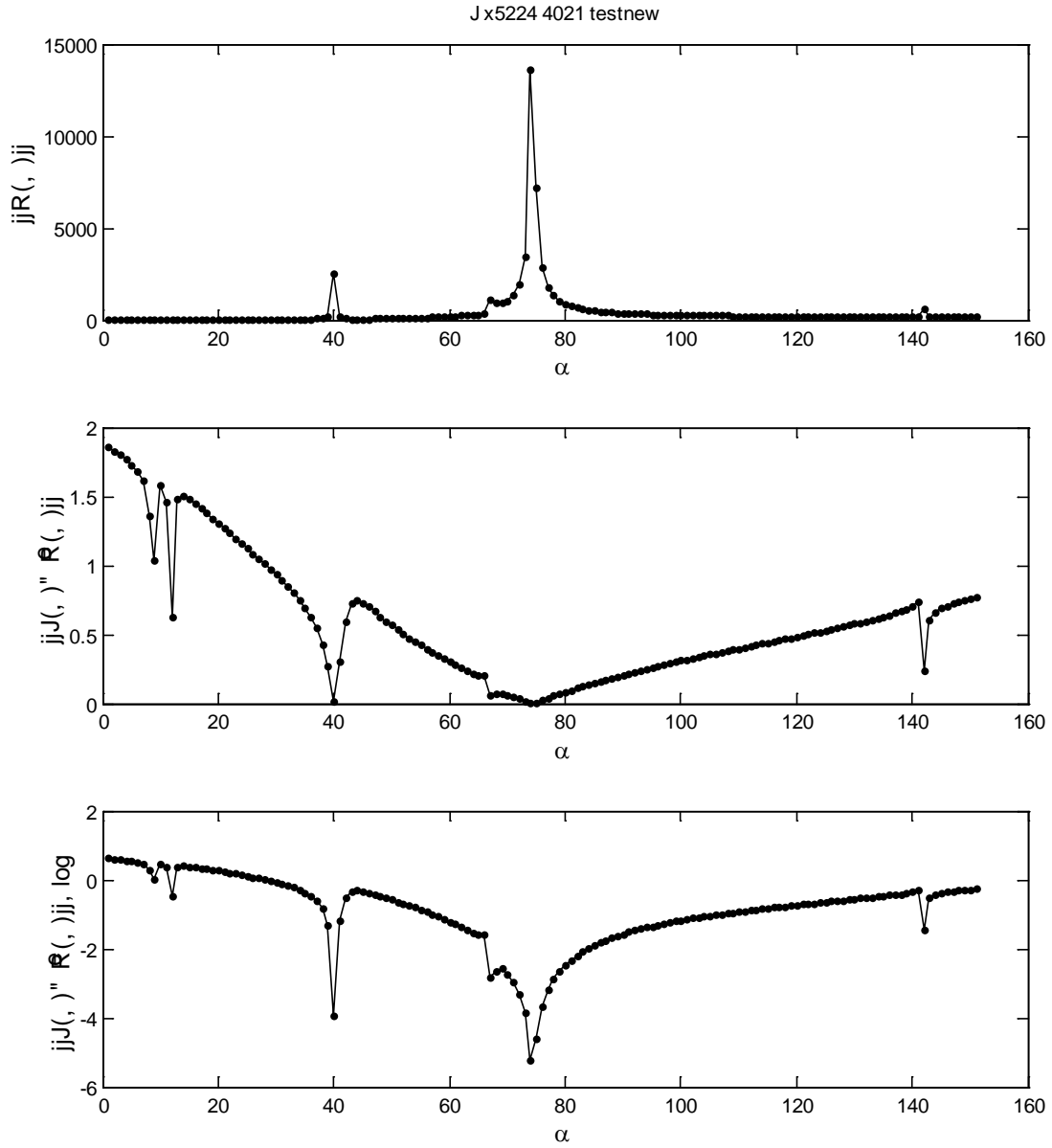


Figure 49: Original Precision with five significant digits.

Figure 50, Figure 51, and Figure 52 also show the same Jacobian data for the same computation, but with values stored to seven, six, and five significant digits respectively. Again Figure 52 shows different behavior from Figure 50 and Figure 51.

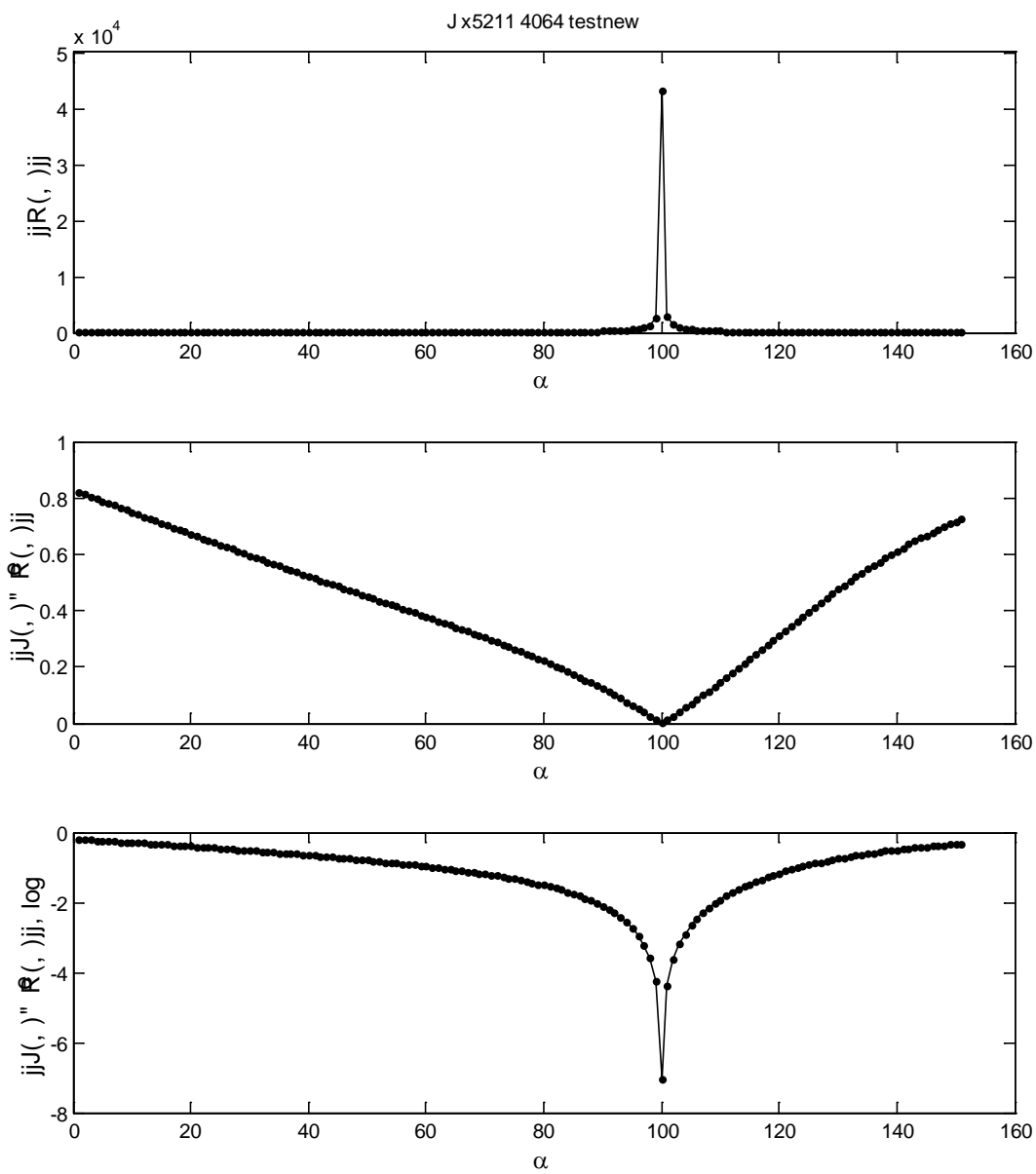


Figure 50: Original precision with seven significant digits.

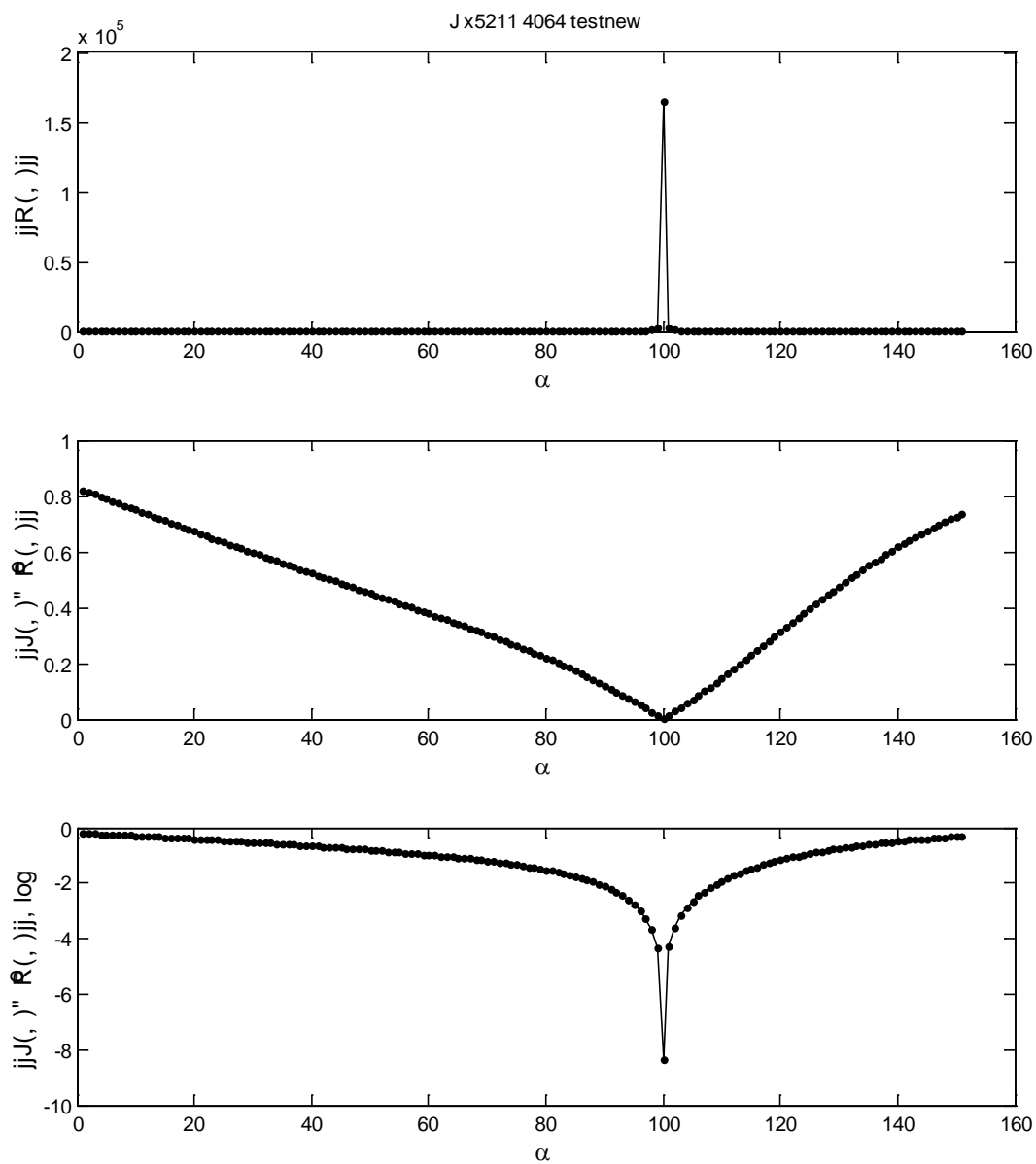


Figure 51: Original precision with six significant digits.

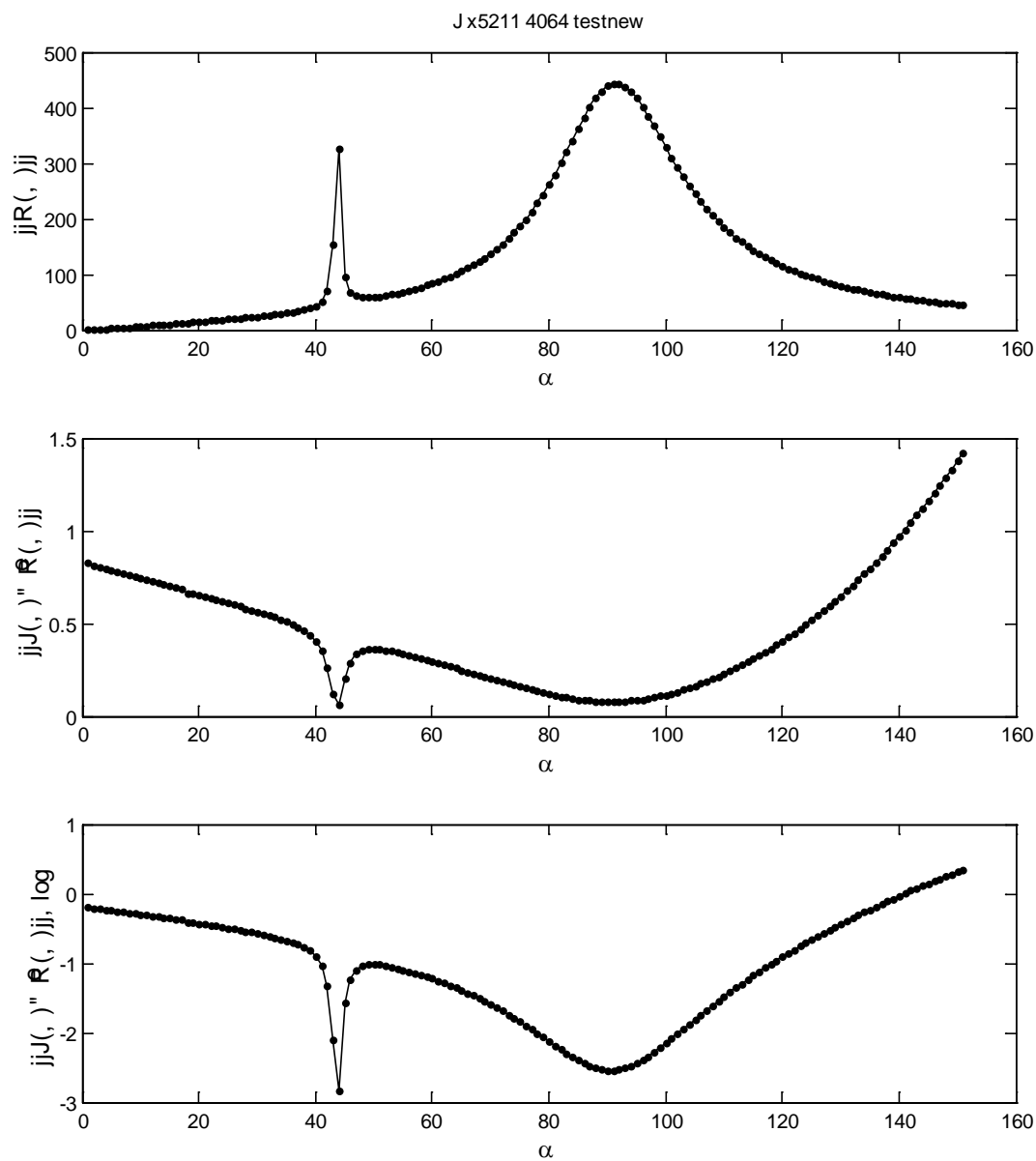


Figure 52: Original precision with five significant digits.

While these test show that precision reduction can introduce additional ‘singularities’, the analysis does not prove that the multi-singularity behaviors that were found elsewhere in the project were due to precision issues alone.



Proudly Operated by Battelle Since 1965

902 Battelle Boulevard
P.O. Box 999
Richland, WA 99352
1-888-375-PNNL (7665)

www.pnnl.gov



U.S. DEPARTMENT OF
ENERGY