APPLICATION OF OPTIMAL MULTIPLIER METHOD IN WEIGHTED LEAST-SQUARES STATE ESTIMATION PART II: SIMULATION

Jianping Meng   Christopher L. DeMarco
Department of Electrical and Computer Engineering
University of Wisconsin-Madison
Madison, Wisconsin 53706

Abstract: Standard algorithms for state estimation may be viewed as quasi-Newton's methods applied to the first order optimality conditions of a least squares minimization problem. Previous work in the literature has documented the (somewhat surprising) fact that when a full Newton's method is applied to the same formulation, convergence properties are far worse than the quasi-Newton's method, until the iterates reach an EXTREMELY small neighborhood of the solution. Motivated by these results, and by availability of efficient algorithms to compute higher order derivatives necessary in an exact Newton formulation, the companion paper [2] proposes several Newton's method variants to improve state estimator convergence. In this paper Benchmarks for the IEEE 118 and 300 bus test systems are provided, with comparisons against classical normal equations, Hatchel's method, and QR algorithms. In these benchmark examples, the new algorithms developed show more reliable convergence for ill-conditioned cases, while making minimal sacrifices in computational efficiency for well-conditioned cases.

Keywords: Weighted Least-Squares(WLS) State Estimation, Optimal Multiplier, Exact Newton's Method, Quasi-Newton's Method

I. Introduction

Traditionally WLS power system state estimation problems are solved by normal equation methods as summarized in the companion paper [2]. If the system is well-conditioned, the normal equation method can converge very fast. But given different measurement set, the conditioning of the system will change. Usually the ill-conditioning can be brought on by zero injections, fewer measurements and less observable measurement set. For the ill-conditioned state estimation problems, normal equation method will often display numerical instability, e.g. oscillation and divergence. Many ideas have been proposed to solve the ill-conditioned estimation problems. Among them are Peters-Wilkinson method [7], orthogonal factorization(QR) method [9], and Hachtel's augmented matrix method [8][10]. The Hachtel's augmented matrix technique usually is used to handle the ill-conditioning produced by zero injection or constrained estimation problems. The QR method is claimed to improve properties of numerical stability. But its slow speed limits its utilization in utility state estimation computation. Some ideas, such as fast Givens rotation method [12] has been proposed to speed up orthogonal factorization. It remains rather slow compared with the normal equation method.

In the companion paper [2] three new Newton algorithm variants are proposed to solve WLS power system state estimation problems. Algorithm 1 uses quasi-Newton's method combined with the optimal multiplier method. Algorithm 2 uses exact Newton's method combined with optimal multiplier method. But during the first several iterates it applies the quasi-Newton's method to drive the iterate to a very small neighborhood of the solution; then it switches to exact Newton's method to achieve quadratic convergence. Algorithm 3 switches between quasi-Newton's method and exact Newton's method by evaluating the two different cost function values produced by the different iterate step sizes obtained by optimal multiplier method. All the three algorithms try to inheritate the fast convergence speed from Newton's method and overcome ill-conditioning by using optimal multiplier method or exact gain matrix or both.

As for the test systems, IEEE 118 and 300 bus systems are used to check the effectiveness of the new algorithms with comparisons against classical normal equation method, Hatchel's method, and QR method. The full sparse techniques and efficient vectorized MATLAB code are used to improve all the test algorithms.

II. Some Key Derivation

In this section we will give the derivations in vectorized MATLAB format, to compute branch power vector, bus injection vector, branch power Jacobian, injection power Jacobian and the corresponding higher order derivatives which are the core in our MATLAB code implementation. We will use MATLAB notation and the notations in Part I of the two companion papers in our following formulation.

Notice that the voltage at both ends of a branch is $\mathbf{A}' * \vec{v}$, the branch current at both ends is $\mathbf{Y}_p * (\mathbf{A}' * \vec{v})$. So The branch complex power h at both ends can be computed by

$$\begin{aligned} \underline{h}(\underline{x}) &= \mathbf{A}' * \underline{\vec{v}}.*(\mathbf{Y}_p^* * \mathbf{A}' * \underline{\vec{v}}^*) \\ &= \mathbf{A}' * (\underline{e} + j\underline{f}).*\left[\mathbf{Y}_p^* * \mathbf{A}' * (\underline{e} - j\underline{f})\right] \end{aligned} \quad (1)$$

where the superscript * denotes the conjugate, and the interpretation of h(x) here is slightly different from that in [2].

The first derivatives of the branch complex power function w.r.t. the state variables are (note: here we keep the slack bus voltage imaginary part as a variable)

$$\frac{\partial \underline{h}}{\partial \underline{e}} = diag\left(\mathbf{Y}_p^* * \mathbf{A}' * \underline{\vec{v}}^*\right) * \mathbf{A}' + diag\left(\mathbf{A}' * \underline{\vec{v}}\right) * \mathbf{Y}_p^* * \mathbf{A}' \quad (2)$$

$$\frac{\partial \underline{h}}{\partial \underline{f}} = j * diag\left(\mathbf{Y}_p^* * \mathbf{A}' * \underline{\vec{v}}^*\right) - j * diag\left(\mathbf{A}' * \underline{\vec{v}}\right) * \mathbf{Y}_p^* * \mathbf{A}' \quad (3)$$

Given a constant vector w which matches the dimension of branch power vector h, second order derivatives of the branch power tensor Jacobian are given by:

$$\frac{\partial}{\partial \underline{e}}\left[\frac{\partial \underline{h}}{\partial \underline{e}}\underline{w}\right] = diag(\mathbf{A}' * \underline{w})\mathbf{Y}_p^* * \mathbf{A}' + diag(\mathbf{Y}_p^* * \mathbf{A}' * \underline{w}) * \mathbf{A}' \quad (4)$$

$$\frac{\partial}{\partial \underline{f}}\left[\frac{\partial \underline{h}}{\partial \underline{e}}\underline{w}\right] = j * diag(\mathbf{Y}_p^* * A' * \underline{w}) * \mathbf{A}' - j * diag(\mathbf{A}' * \underline{w}) * \mathbf{Y}_p^* * \mathbf{A}' \quad (5)$$

$$\frac{\partial}{\partial \underline{f}}\left[\frac{\partial \underline{h}}{\partial \underline{e}}\underline{w}\right] = -\frac{\partial}{\partial \underline{e}}\left[\frac{\partial \underline{h}}{\partial \underline{f}}\underline{w}\right] \quad (6)$$

$$\frac{\partial}{\partial \underline{f}}\left[\frac{\partial \underline{h}}{\partial \underline{f}}\underline{w}\right] = \frac{\partial}{\partial \underline{e}}\left[\frac{\partial \underline{h}}{\partial \underline{e}}\underline{w}\right] \quad (7)$$

$$\frac{\partial}{\partial \underline{e}}\left[\left(\frac{\partial \underline{h}}{\partial \underline{e}}\right)^T \underline{w}\right] = \mathbf{A} * diag(\underline{w}) * \mathbf{Y}_p^* * \mathbf{A}' + \mathbf{A} * \mathbf{Y}_p^* * diag(\underline{w}) * \mathbf{A}' \quad (8)$$

$$\frac{\partial}{\partial \underline{f}}\left[\left(\frac{\partial \underline{h}}{\partial \underline{e}}\right)^T \underline{w}\right] = j * \mathbf{A} * \mathbf{Y}_p^* * diag(\underline{w}) * \mathbf{A}' - j * \mathbf{A} * diag(\underline{w}) * \mathbf{Y}_p^* * \mathbf{A}' \quad (9)$$

$$\frac{\partial}{\partial \underline{e}}\left[\left(\frac{\partial \underline{h}}{\partial \underline{f}}\right)^T \underline{w}\right] = -\frac{\partial}{\partial \underline{e}}\left[\left(\frac{\partial \underline{h}}{\partial \underline{f}}\right)^T \underline{w}\right] \quad (10)$$

$$\frac{\partial}{\partial \underline{f}}\left[\left(\frac{\partial \underline{h}}{\partial \underline{f}}\right)^T \underline{w}\right] = \frac{\partial}{\partial \underline{e}}\left[\left(\frac{\partial \underline{h}}{\partial \underline{e}}\right)^T \underline{w}\right] \quad (11)$$

Similarly the first order derivatives of bus injection power $\underline{s} = \underline{\vec{v}}.*\underline{i}^* = (\underline{e}+j\underline{f}).*\left[Y_b^* * (\underline{e} - j\underline{f})\right]$ w.r.t. the state variables are

$$\frac{\partial \underline{s}}{\partial \underline{e}} = diag\left(\mathbf{Y}_b^* * \underline{\vec{v}}^*\right) + diag(\underline{\vec{v}}) * \mathbf{Y}_b^* \quad (12)$$

$$\frac{\partial \underline{s}}{\partial \underline{f}} = j * diag\left(\mathbf{Y}_b^* * \underline{\vec{v}}^*\right) - j * diag(\underline{\vec{v}}) * \mathbf{Y}_b^* \quad (13)$$

Given a constant vector w which matches the dimension of injection power vector s, the injection power tensor Jacobian is given by:

$$\frac{\partial}{\partial \underline{e}}\left[\frac{\partial \underline{s}}{\partial \underline{e}}\underline{w}\right] = diag(\underline{w}) * \mathbf{Y}_b^* + diag(\mathbf{Y}_b^* * \underline{w}) \quad (14)$$

$$\frac{\partial}{\partial \underline{f}}\left[\frac{\partial \underline{s}}{\partial \underline{e}}\underline{w}\right] = j * diag(\mathbf{Y}_b^* * \underline{w}) - j * diag(\underline{w})\mathbf{Y}_b^* \quad (15)$$

$$\frac{\partial}{\partial \underline{e}}\left[\frac{\partial \underline{s}}{\partial \underline{f}}\underline{w}\right] = -\frac{\partial}{\partial \underline{f}}\left[\frac{\partial \underline{s}}{\partial \underline{e}}\underline{w}\right] \quad (16)$$

$$\frac{\partial}{\partial \underline{f}}\left[\frac{\partial \underline{s}}{\partial \underline{f}}\underline{w}\right] = \frac{\partial}{\partial \underline{e}}\left[\frac{\partial \underline{s}}{\partial \underline{e}}\underline{w}\right] \quad (17)$$

$$\frac{\partial}{\partial \underline{e}}\left[\left(\frac{\partial \underline{s}}{\partial \underline{e}}\right)^T \underline{w}\right] = diag(\underline{w}) * \mathbf{Y}_b^* + \mathbf{Y}_b^* * diag(\underline{w}) \quad (18)$$

$$\frac{\partial}{\partial \underline{f}}\left[\left(\frac{\partial \underline{s}}{\partial \underline{e}}\right)^T \underline{w}\right] = j * \mathbf{Y}_b^* * diag(\underline{w}) - j * diag(\underline{w}) * \mathbf{Y}_b^* \quad (19)$$

$$\frac{\partial}{\partial \underline{e}}\left[\left(\frac{\partial \underline{s}}{\partial \underline{f}}\right)^T \underline{w}\right] = -\frac{\partial}{\partial \underline{f}}\left[\left(\frac{\partial \underline{s}}{\partial \underline{e}}\right)^T \underline{w}\right] \quad (20)$$

$$\frac{\partial}{\partial \underline{f}}\left[\left(\frac{\partial \underline{s}}{\partial \underline{f}}\right)^T \underline{w}\right] = \frac{\partial}{\partial \underline{e}}\left[\left(\frac{\partial \underline{s}}{\partial \underline{e}}\right)^T \underline{w}\right] \quad (21)$$

III. Simulation Results

In this section we will compare our proposed new algorithms with the traditional polar form and rectangular form normal equation methods, polar form orthogonal decomposition method, and polar form Hachtel's augmented matrix method. The computation examples are IEEE 118 and 300 bus systems. The fictitious measurements including line flows, bus injections and bus voltages are obtained from load flow computation with Gaussian noise added. For the virtual measurements the added noise should be zero. The standard deviation for active line flow, reactive line flow, active bus injection, reactive bus injection and bus voltage are 0.0125, 0.0143, 0.02, 0.025 and 0.01(base 100 MVA) respectively. The standard deviation for virtual measurements are 0.00001. Before moving into the comparison we briefly describe the QR method [9] and Hachtel's method [8][10] as follows.

Orthogonal Decomposition Method:
Consider the following linear WLS problem at each iteration, which seeks to minimize

$$\begin{aligned} J(\Delta \underline{x}) &= [\Delta \underline{\hat{z}} - \mathbf{H}\Delta \underline{x}]^T \mathbf{R}^{-1} [\Delta \underline{\hat{z}} - \mathbf{H}\Delta \underline{x}] \\ &= \left[\Delta \underline{\tilde{z}} - \tilde{\mathbf{H}}\Delta \underline{x}\right]^T \left[\Delta \underline{\tilde{z}} - \tilde{\mathbf{H}}\Delta \underline{x}\right] \\ &= \|\Delta \underline{\tilde{z}} - \tilde{\mathbf{H}}\Delta \underline{x}\|^2 \end{aligned} \quad (22)$$

where $\Delta \underline{\tilde{z}} = \mathbf{R}^{-1/2}\Delta \underline{\hat{z}}$, $\tilde{\mathbf{H}} = \mathbf{R}^{-1/2}\mathbf{H}$ and $\|\cdot\|$ denotes the Euclidean norm.
Let $\mathbf{Q}$ be an orthogonal matrix, such that

$$\mathbf{Q}\tilde{\mathbf{H}} = \begin{bmatrix} \mathbf{U} \\ \mathbf{0} \end{bmatrix} \quad (23)$$

where $\mathbf{U}$ is an upper triangular matrix.

$$\mathbf{Q} = \begin{bmatrix} \mathbf{Q}_1 \\ \mathbf{Q}_2 \end{bmatrix} \quad (24)$$

and $\mathbf{Q}_1\tilde{\mathbf{H}} = \mathbf{U}$, $\mathbf{Q}_2\tilde{\mathbf{H}} = \mathbf{0}$
Then we have

$$J(\Delta \underline{x}) = \|\Delta \underline{\tilde{z}} - \tilde{\mathbf{H}}\Delta \underline{x}\|^2 = \|\mathbf{Q}\Delta \underline{\tilde{z}} - \mathbf{Q}\tilde{\mathbf{H}}\Delta \underline{x}\|^2 \quad (25)$$

and the solution of the WLS problem is

$$\mathbf{U}\Delta\underline{x} = \mathbf{Q}_1\Delta\underline{\tilde{z}} \tag{26}$$

Another hybrid approach of the QR method is proposed in [9], to solve the following normal equation

$$\mathbf{U}^T\mathbf{U}\Delta\underline{x} = \mathbf{H}^T\mathbf{R}^{-1}\Delta\underline{z} \tag{27}$$

The advantage of QR method is its numerical stability. The drawback is the very slow computation speed. Usually the orthogonal decomposition is proceeded by Householder and Givens transformations [12]. In this paper the sparse QR factorization software package developed by Matstoms[11] is used to solve the linear least squares problems. The multifrontal method is used to do the sparse QR factorization. Before orthogonal decomposition, a fill-in minimizing column ordering is first computed. Compared with the standard MATLAB QR factorization routine it is much faster.

Hachtel's Augmented Matrix Method[8][10]
If there are zero injections(virtual measurements) at some load buses, the injection measurements for these buses become constraints

$$\underline{c}(\underline{x}) = \underline{0} \tag{28}$$

To accommodate these constraints with the cost function J, we form the following Lagrangian

$$L = \frac{1}{2}(\Delta\underline{z} - \mathbf{H}\Delta\underline{x})^T(\Delta\underline{z} - \mathbf{H}\Delta\underline{x}) - \underline{\lambda}^T(\Delta\underline{c} - \mathbf{C}\Delta\underline{x}) \tag{29}$$

where $\underline{\lambda}$ is the Lagrange multiplier vector. $\mathbf{C}$ is the Jacobian of the constraint function, i.e. $\mathbf{C} = \partial\underline{c}/\partial\underline{x}$
If we treat the residual vector $\underline{r}$ as an unknown vector and let

$$\underline{r} = \Delta\underline{z} - \mathbf{H}\Delta\underline{x}\,, \tag{30}$$

then the solution of the WLS problem can be obtained by solving the following matrix function

$$\begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{C} \\ \mathbf{0} & \alpha\mathbf{I} & \mathbf{H} \\ \mathbf{C}^T & \mathbf{H}^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} -\alpha^{-1}\underline{\lambda} \\ \alpha^{-1}\underline{r} \\ \Delta\underline{x} \end{bmatrix} = \begin{bmatrix} \Delta\underline{c} \\ \Delta\underline{z} \\ \mathbf{0} \end{bmatrix} \tag{31}$$

where α is a scalar parameter to be chosen. A special 2×2 pivoting is used to improve numerical stability[10]. In this paper, for simplicity, we will not use this 2×2 pivoting technique to implement Hachtel's method. Thus the performance of Hachtel's method in our tests may be underestimated.

The computation was done on the SUN ULTRA 10 workstation. We use the flat voltage profile as our initial guess. The tolerance 10^{-5} is used for each algorithm. The computation results are shown in the following tables. In order to ensure fair and consistent comparison, we convert all the intermediate state variables in each algorithm into polar coordinates. The stopping criterion is that the maximum component of the difference between the two consecutive $\underline{x}$ vectors, in this consistent coordinate system, is less than the tolerance. After the estimated result is obtained, the estimation error is computed. Here the estimation error means the Euclidean norm of the difference between the power flow solution and the state variable vector obtained by the state estimator algorithm.

A. **Test 1:** *IEEE 118 bus test system*

In this section we will use IEEE 118 bus system as the example to compare our proposed new algorithms with polar form NE(normal equation) method, rectangular form NE method, polar form QR method and polar form Hachtel's augmented matrix method. Since all the test cases are generated randomly, some cases may be well conditioned and some cases may be very ill-conditioned. In order to get statistics on the results, we test a large number of cases. Table 1 describes some test cases and Table 2 shows the algorithm comparison of those cases in computation flops and estimation error. From Table 1 and Table 2 we can see the three optimal multiplier algorithms are generally better than the other 4 algorithms. Among the three optimal multiplier algorithms method 2 and method 3 are the best. If the system is well-conditioned (usually the condition number of the initial iteration gain matrix is not big small), the convergence characteristics of all 7 algorithms are very close. But if the system is very ill-conditioned (doesn't mean the condition number of the initial iteration gain matrix must be big [1]) generally the optimal multiplier method 2 and method 3 have relatively nice convergence characteristics. This is due to the contribution of the optimal multiplier and the exact gain matrix. We have to mention the QR method and Hachtel's augmented matrix method are claimed to have better convergence properties than NE method. However we didn't observe this. The reason may be that the QR method software package still needs to be improved and our implementation of Hachtel's method is not efficient enough.

B. **Test 2:** *IEEE 300 bus test system*

We now test our proposed three optimal multiplier methods against rectangular form NE method, polar form QR method and polar form Hachtel's method on the bigger system, IEEE 300 bus system. Since the performance of polar form NE method and rectangular form NE method are close, we ignore the polar form NE method here. In order to compare the computation speed, the approximate number of flops are given for each algorithm. The description of the test cases and the algorithm comparison results are shown in Table 3 and Table 4. For 300 bus system we still have similar conclusions. The results in Table 4 clearly show that the optimal multiplier methods have better convergence. As for the computation speed, when the system is well conditioned, the optimal multiplier method 1 may be the best because its computation flops in each iterate is close to NE method and its convergence is better that NE method. If the system is very ill-conditioned generally the optimal multiplier method 2 and 3 are the best. The flops of each iterate for optimal multiplier method 2 is less than that for optimal multiplier 3. But the overall convergence characteristics of method 3 seems better than that of method 2. The four plots on the last three pages

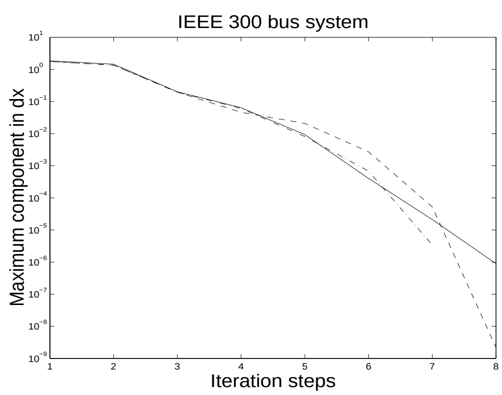
[1] In the simulation we observe that in some cases although the condition number of the initial gain matrix is not big, the convergence characteristics for each algorithm is not good. So in this paper the meaning of ill conditioned systems is not the same as big condition number of initial gain matrix.

Table 1: Data for IEEE 118 bus test system

case	Line flows active/reactive	Bus injections active/reactive	Zero injections active/reactive	Voltages	Total	Redundancy	Condition* Number
1	297/297	54/64	10/11	59	792	3.37	1.6×10^{8}
2	223/223	54/53	9/8	47	617	2.63	3.2×10^{8}
3	186/148	64/53	5/5	47	508	2.16	1.0×10^{9}
4	223/223	21/10	5/5	47	534	2.27	5.0×10^{8}
5	223/223	75/74	10/11	35	651	2.77	2.4×10^{8}
6	186/186	75/74	10/11	35	577	2.46	3.0×10^{8}
7	223/223	54/53	9/8	59	629	2.68	2.5×10^{13}
8	186/148	54/53	9/9	47	506	2.15	6.6×10^{12}
9	111/111	91/90	9/9	35	456	1.94	7.7×10^{8}
10	111/111	75/85	10/9	35	436	1.86	8.8×10^{9}
11	111/111	54/53	10/11	47	397	1.69	6.2×10^{12}
12	93/93	75/74	10/11	35	427	1.82	1.1×10^{23}
13	186/148	64/64	5/5	59	531	2.26	3.7×10^{20}
14	111/111	43/37	10/11	59	382	1.63	1.2×10^{11}
15	186/186	32/32	10/11	59	516	2.20	4.0×10^{11}
16	260/260	10/10	10/11	59	620	2.64	3.8×10^{41}
17	297/297	0/0	10/11	59	674	2.87	6.0×10^{12}
18	111/111	75/74	10/11	35	674	2.87	6.0×10^{12}
19	111/111	64/64	10/11	35	406	1.73	5.3×10^{23}
20	111/111	54/53	10/11	47	397	1.69	2.1×10^{16}
21	111/111	54/53	10/11	47	397	1.69	2.7×10^{15}
22	223/223	21/21	10/11	59	568	2.42	8.8×10^{24}

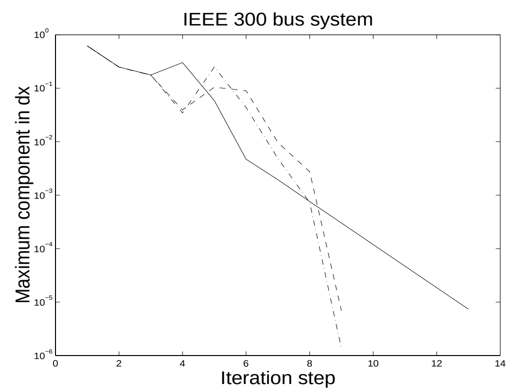
* Condition number of initial gain matrix of rectangular form normal equation method

show the convergence characteristics for rectangular form NE method, optimal multiplier method 2 and optimal multiplier method 3. In these plots solid line stands for rectangular form NE method, dash line stands for optimal multiplier method 2 and dash dot line stands for optimal multiplier method 3.



solid line: Rectangular form NE method
−−: Proposed algorithm 2, −.−: Proposed algorithm 3

Fig.1: *Convergence characteristics of 300 bus system(Plot 1)*



solid line: Rectangular form NE method
−−: Proposed algorithm 2, −.−: Proposed algorithm 3

Fig.2: *Convergence characteristics of 300 bus system(Plot 2)*

IV. Conclusion

In the companion paper [2] three new algorithms, optimal multiplier method 1, 2 and 3 are proposed to solve WLS state estimation problems. They are based on optimal multiplier method which was originally presented in [6] and exact Newton's method which incorporates the second order derivative information of the measurement function with respect to state variables. Among those, method 1 does not use the second order derivative information of measurement function. The detailed formulation in rectangular coordinates for the above three algorithms are given. In order to check the effectiveness of the proposed new algorithms IEEE 118 and 300 systems are used in this paper as the computation examples to compare against the traditional polar form and rectangular form normal equation methods, polar form orthogonal transformation method and polar form Hachtel's augmented matrix method. The simulation results show that by combining optimal multiplier method and exact Newton's method the new algorithms have very nice convergence characteristics. The power of the new algorithm lies in its ability to

Table 2: Algorithm comparison for IEEE 118 bus test system

case	NE (polar)	NE (Cartesian)	QR (polar)	Hachtel (polar)	Optimal 1	Optimal 2	Optimal 3
1	2.3E6/3.4153 *	2.0E6/3.4153	9.0E7/3.6252	4.1E6/3.9661	2.0E6/3.4153	2.1E6/3.4153	3.9E6/3.4153
2	2.2E6/2.9687	2.0E6/2.8956	1.0E8/2.9687	3.7E6/2.8264	2.1E6/2.8956	2.2E6/2.8956	3.8E6/2.8956
3	2.5E6/9.0895	2.2E6/8.7268	1.1E8/9.0895	4.4E6/9.2977	2.3E6/8.7268	2.5E6/8.7268	4.6E6/8.7268
4	2.2E6/3.0280	2.2E6/2.9409	9.5E7/3.0280	3.6E6/3.0170	2.1E6/2.9409	2.2E6/2.9409	4.1E6/2.9409
5	2.5E6/3.1941	1.8E6/3.3165	1.2E8/3.1941	4.2E6/3.4722	2.1E6/3.3165	2.1E6/3.3165	4.0E6/3.3165
6	2.8E6/6.6831	2.4E6/6.5065	1.4E8/6.6831	5.1E6/7.0418	2.7E6/6.5065	2.4E6/6.5065/	4.2E6/6.5065
7	1.1E7/4.6075	6.2E6/4.5854	3.8E8/4.6075	diverge	5.9E6/4.5854	5.5E6/4.5855	8.0E6/4.5855
8	1.6E7/11.7449	1.3E7/12.0939	6.5E8/11.7449	oscillate	1.5E7/12.0939	5.7E6/12.0939	8.5E6/12.0939
9	3.1E7/2.6243	3.6E6/2.5431	1.1E9/2.6243	oscillate	3.6E6/2.5431	2.4E6/2.5431	4.6E6/2.5431
10	3.1E7/7.8755	4.1E6/7.8122	1.1E9/7.8755	oscillate	4.2E6/7.8122	2.5E6/7.8122	4.8E6/7.8122
11	8.3E6/5.9088	6.1E6/7.1427	3.6E8/5.9088	1.1E7/5.6770	4.0E6/7.1427	5.1E6/7.1427	8.4E6/7.1427
12	7.3E6/14.0990	5.9E6/13.7031	3.7E7/14.0990	8.0E6/14.5070	6.6E6/13.7031	4.3E6/13.7030	9.4E6/13.7030
13	diverge	oscillate	diverge	diverge	oscillate	1.8E7/3.9995	1.3E7/3.9995
14	oscillate	oscillate	oscillate	oscillate	oscillate	4.4E6/12.8743	8.3E6/12.8743
15	oscillate	oscillate	oscillate	oscillate	oscillate	4.2E6/7.7939	8.2E6/7.7939
16	diverge	diverge	diverge	diverge	slow convergence	8.0E6/5.0709	5.0709/8.3E6
17	oscillate	oscillate	oscillate	oscillate	oscillate	4.9E6/1.5521	9.7E6/1.5521
18	oscillate	oscillate	oscillate	diverge	oscillate	8.5E6/7.6785	1.2E7/7.6785
19	diverge	oscillate	diverge	diverge	diverge	oscillate	1.3E7/4.5566
20	diverge	oscillate	diverge	diverge	oscillate	5.2E6/10.1508	1.2E7/80.6746
21	diverge	oscillate	diverge	diverge	oscillate	oscillate	1.1E7/7.4338
22	diverge	diverge	diverge	diverge	4.1E6/19.4506	oscillate	9.5E6/15.6902

* column entries: approximate computation flops/estimation error at solution, E means the power of 10.

Table 3: Data for IEEE 300 bus test system

case	Line flows active/reactive	Bus injections active/reactive	Zero injections active/reactive	Voltages	Total	Redundancy	Cond* Number
1	493/452	163/172	67/84	120	1551	2.59	1.3×10^{12}
2	493/411	163/172	67/84	120	1510	2.52	1.1×10^{13}
3	411/369	186/172	67/84	150	1439	2.40	4.4×10^{11}
4	369/369	209/194	67/84	150	1442	2.41	7.1×10^{12}
5	493/493	163/151	67/84	90	1541	2.57	3.2×10^{12}
6	493/493	163/172	67/84	120	1592	2.66	1.6×10^{26}
7	411/411	163/172	67/84	150	1458	2.43	8.0×10^{27}
8	369/369	209/151	67/84	210	1459	2.44	5.5×10^{13}
9	452/369	186/151	67/84	210	1519	2.54	3.5×10^{12}
10	493/493	186/151	67/84	90	1564	2.61	1.0×10^{16}

* Condition number of initial gain matrix of rectangular form normal equation method

solve the very ill-conditioned power system state estimation problems. As we can see from the test results even if the system is extremely ill-conditioned the algorithms can still converge very fast. The drawback of the optimal multiplier algorithm 2 is that it needs use of the quasi-Newton's method for the first several step warming up iterates before switching to exact Newton method. The drawback of optimal multiplier algorithm 3 is its need for more computation in each iteration. With the speed of the computers becoming faster and faster, the convergence characteristics of the algorithm may be viewed as more important.

References

[1] Robert A.M. van Amerongen, "On Convergence Analysis and Convergence Enhancement of Power System Least-Squares State Estimators", *IEEE Transactions on Power Systems*, Vol. 10, No. 4, November 1995.

[2] Jianping Meng, Chtistopher L. DeMarco "Application of Optimal Multiplier Method in Weighted Least-Squares State Estimation, Part I: Theory", *The 31st North American Power Symposium*, San Luis Obispo, California, October, 1999.

[3] J.M. Ortega and W.C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, New York and London, 1970.

[4] Gene H. Golub, Charles F. Van Loan, *Matrix Computation*, The Johns Hopkins University Press, 1985.

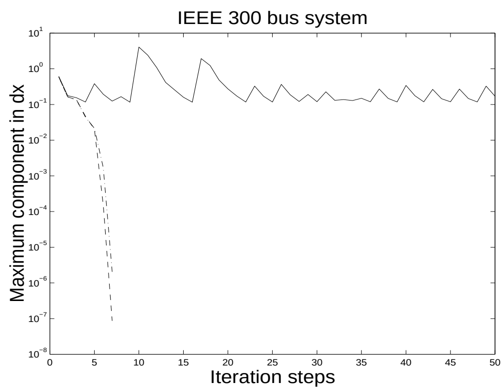
[5] Allen J. Wood, Bruce F. Wollenberg, *Power Generation, Operation, and Control*, 2nd ed., John Wiley & Sons, Inc,. New York, 1996.

[6] S. Iwamoto, Y. Tamura, "A Load Flow Calculation Method for Ill-conditioned Power Systems", *IEEE Transactions on Power Apparatus and Systems*, Vol. PAS-100, No.4, April 1981.

Table 4: Algorithm comparison for IEEE 300 bus test system

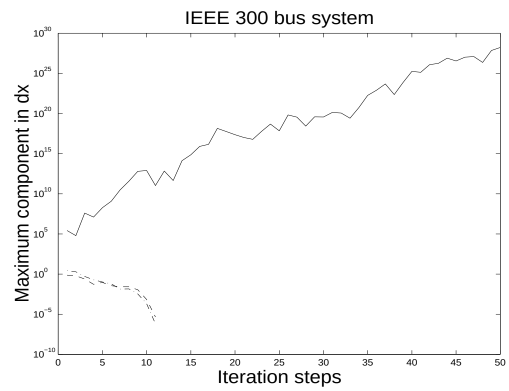
case	NE (Cartesian)	QR (polar)	Hachtel (polar)	Optimal 1	Optimal 2	Optimal 3
1	7.9×10^7/6.92*	2.2×10^9/6.74	2.0×10^8/7.01	9.2×10^7/6.92	8.7×10^7/6.92	1.6×10^8/6.92
2	3.4×10^8/14.21	1.0×10^{10}/12.56	9.5×10^8/12.81	4.0×10^8/14.21	1.9×10^8/14.21	4.1×10^8/14.21
3	1.1×10^8/ 6.88	1.7×10^{10}/7.94	oscillate	1.2×10^8/6.88	9.3×10^7/6.88	1.7×10^8/6.88
4	1.4×10^8/14.57	6.6×10^9/15.04	4.6×10^8/17.02	1.6×10^8/14.57	2.3×10^8/14.57	4.0×10^8/14.57
5	2.1×10^8/12.56	6.2×10^9/12.96	2.9×10^8/14.68	2.2×10^8/12.56	1.7×10^8/12.56	2.9×10^8/12.56
6	diverge	diverge	diverge	1.4×10^8/8.07	6.6×10^8/8.07	2.8×10^8/8.07
7	diverge	diverge	diverge	1.7×10^8/14.71	2.0×10^8/16.45	4.6×10^8/16.45
8	oscillate	oscillate	oscillate	oscillate	2.2×10^8/9.05	3.4×10^8/9.05
9	oscillate	oscillate	oscillate	oscillate	1.2×10^8/12.92	2.3×10^8/12.92
10	1.2×10^8/10.27	diverge	diverge	1.4×10^8/10.27	1.5×10^8/10.27	2.4×10^8/10.27

* column entries: approximate computation flops/estimation error at solution



solid line: Rectangular form NE method
−−: Proposed algorithm 2, −.−: Proposed algorithm 3

Fig.3: *Convergence characteristics of 300 bus system(Plot 3)*



solid line: Rectangular form NE method
−−: Proposed algorithm 2, −.−: Proposed algorithm 3

Fig.4: *Convergence characteristics of 300 bus system(Plot 4)*

[7] J.W. Gu, K.A. Clements, G.R. Krumpholz, P.W. Davis, "The Solution of Ill-conditioned Power System State Estimation Problems via The Method of Peters and Wilkinson", *IEEE Transactions on Power Apparatus and Systems*, Vol. PAS-102, No.10, October 1983.

[8] Anders Gjelsvik, Sverre Aam, Lars Holten, "Hachtel's Augmented Matrix Method - A Rapid Method Improving Numerical Stability in Power System Static State Estimation", *IEEE Transactions on Power Apparatus and Systems*, Vol. PAS-104, No.11, November 1985.

[9] A. Monticelli, C.A.F. Murari, F.F. Wu, "A Hybrid State Estimator: Solving Normal Equations by Orthogonal Transformations", *IEEE Transactions on Power Apparatus and Systems*, Vol. PAS-104, No.12, December 1985.

[10] F.F. Wu, Wen-Hsing E. Liu, Lars Holten, Anders Gjelsvik, Sverre Aam, "Observability Analysis and Bad Data Processing for State Estimation Using Hachtel's Augmented Matrix Method", *IEEE Transactions on Power Systems*, Vol.3, No.2, May 1988.

[11] P. Matstoms, *Sparse QR Factorization with Application to Linear Square Problems*,PH.D Thesis, Dept. of Mathematics, Linkoping University, S-581 83 Linkoping, Sweden, 1994.

[12] Slutsker I.W. Vempatin, W.F. Tinney, "Orthogonal Sparse Vector Methods", *IEEE Transactions on Power Systems*, Vol. 7, No.2, pp926-932, May 1992.

[13] *Using MATLAB Version 5*, The MAth Works Inc., 1996.

Biographies

Jianping Meng obtained his BS degree and two MS degrees all in electrical engineering from Tsinghua University, North China Institute of Electric Power and University of Wisconsin-Madison in 1991, 1994 and 1997 respectively. He is currently a Ph.D. candidate with the Department of Electrical and Computer Engineering at University of Wisconsin-Madison.

Christopher L. DeMarco obtained the BS degree from Masachussets Institute of Technology in 1980 and the MS and Ph.D. degrees from University of California-Berkeley in 1982 and 1985, all in electrical engineering. He is currently a professor with the Department of Electrical and Computer Engineering at University of Wisconsin-Madison.