Energy and Power Engineering, 2016, 8, 99-115
Published Online February 2016 in SciRes. http://www.scirp.org/journal/epe
http://dx.doi.org/10.4236/epe.2016.82009

Solving Load Flow Problems of Power System by Explicit Pseudo-Transient Continuation (E-Ψtc) Method

Tianmin Han[1], Yuhuan Han[2]

[1]Electric Power Research Institute, Beijing, China
[2]Hedge Fund of America, San Jose, USA
Email: hantianmin@126.com, Yuhuan.Han@gmail.com

Received 24 November 2015; accepted 23 February 2016; published 26 February 2016

Abstract

This paper is a further study of two papers [1] and [2], which were related to Ill-Conditioned Load Flow Problems and were published by IEEE Trans. PAS. The authors of this paper have some different opinions, for example, the 11-bus system is not an ill-conditioned system. In addition, a new approach to solve Load Flow Problems, E-Ψtc, is introduced. It is an explicit method; solving linear equations is not needed. It can handle very tough and very large systems. The advantage of this method has been fully proved by two examples. The authors give this new method a detailed description of how to use it to solve Load Flow Problems and successfully apply it to the 43-bus and the 11-bus systems. The authors also propose a strategy to test the reliability, and by solving gradient equations, this new method can answer if the solution exists or not.

Keywords

Load Flow Problems of Power System, Ill-Conditioned System, Normal Newton Method, Optimal Multiplier Method, Brown Method, Explicit Pseudo-Transient Continuation (E-Ψtc), 11-Bus and 43-Bus Systems, Reliability, Gradient Equation

1. Introduction

In the early 1980's, there were two papers which studied ill-conditioned Load Flow Problems ([1] and [2]). Different numerical method was used to analyze the ill-conditioned equations and got some useful results. Because of the importance of this problem in power system, it is necessary to study this problem again by using new numerical method.

How to cite this paper: Han, T.M. and Han, Y.H. (2016) Solving Load Flow Problems of Power System by Explicit Pseudo-Transient Continuation (E-Ψtc) Method. *Energy and Power Engineering*, **8**, 99-115.
http://dx.doi.org/10.4236/epe.2016.82009

Since 1960's, people have taken the direct method as a major tool to solve load flow equations; among them, Newton-Gaussian elimination (LU) is a typical representation [3]. Despite the fact that some people proposed nonlinear programming method [4], this method was not widely used.

Recently people have turned their attention to iterative method, e.g., Inexact-Newton-GMRES method, ([5] and [6]), but none of them is involved with the ill-conditioned problems.

The extent of ill-conditioned problem is determined by the conditioned number of the system. From [2], for 11-bus and 43-bus systems, they are 0.1086 × 10^4 and 0.2560 × 10^5 respectively. For double precision calculation, it is not a severe problem; we can determine the existence of the solutions of the two systems by the normal Newton method without any trouble.

For the conclusion (3) of [1] and the statements "Supposing that the correction vector Δx is obtained by some way…". For those statements, the authors of this paper have a different opinion. The details of discussion can be found in Appendix.

Brown method was used by [2]. Ref. [7] did a deep and detailed study for this method and pointed out: "The method did not have a clear algorithmic description and hence was hard to implement". Maybe even more important is that for each iteration, it needs $O\left(n^4\right)$ arithmetic operations. [8] improved Brown method, and also gave a clearer algorithmic description and reduced arithmetic operations from $O\left(n^4\right)$ to $O\left(n^3\right)$. [2] did not take advantage of this point, and still used original Brown method.

Considering that the two papers were published in 1980's, we should not make excessive demands for authors of [2], but the conclusion for 11-bus and 43-bus systems, Newton-Raphson method is "divergent" (**Table 2** on page 3653 of Ref. [2]), which is incorrect and conflicts with [1].

2. Description of the New Algorithm

Before the description of the new method, let us first review the derivation process of the two classical methods:

1) Newton method:

From $F\left(x_n+\Delta x\right)=F\left(x_n\right)+F'\left(x_n\right)*\Delta x=0$.

We have $\Delta x=-F'\left(x_n\right)^{-1}*F\left(x_n\right)$.

Or, for $\dfrac{\mathrm{d}x}{\mathrm{d}t}=-F'\left(x\right)^{-1}F\left(x\right)$ using explicit Euler numerical integration method:

$$x_{n+1}=x_n+h*\left(-F'\left(x_n\right)^{-1}\right)*F\left(x_n\right).$$

Take $h=1$, hence we have:

$$x_{n+1}-x_n=\Delta x=-F'\left(x_n\right)^{-1}*F\left(x_n\right).$$

That is to say, we can use both Static State and Dynamic State approaches to derive Newton method. Despite the Newton method can be recognized as an explicit ODE Euler method, the right side function includes an inverse matrix of $F'\left(x\right)$. It still belongs to implicit method.

If the Newton method fails to converge, two approaches can be taken to improve the convergence:

a) Using damp factor: Replacing Δx by $\left(h*\Delta x\right)\left(h<1\right)$. This is equivalent to reducing the step size of Euler method, making $\left(h*\Delta x*\rho\right)$ to fall in the stability region of Euler method. Here ρ is the spectral radius of the Jacobian matrix of $F'\left(x_n\right)^{-1}*F\left(x_n\right)$.

b) Altering initial values: If the initial value $x\left(0\right)$ closes to the solution x^*, the Jacobian matrix of $F'\left(x_n\right)^{-1}*F\left(x_n\right)$ will close to unit matrix. Such that taking large step size h = 1, the stability of numerical integration can be kept.

2) About the fixed iteration

For $F\left(x\right)=0$, put $\phi\left(x\right)=h*F\left(x\right)+x$, here h is the parameter. Fixed iteration method:

$$x_{n+1}=\phi\left(x_n\right)$$

will have the same solution of $h*F\left(x\right)=0$ and $F\left(x\right)=0$. We can still interpret it as an explicit ODE Euler method. In fact, for differential equation $\dfrac{\mathrm{d}x}{\mathrm{d}t}=F\left(x\right)$ using Euler method:

$$x_{n+1} = x_n + h * F(x_n) = x_n + (\phi(x_n) - x_n) = \phi(x_n).$$

Take $h = 1$, we can get classical fixed iteration:

$$x_{n+1} = \phi(x_n) = F(x_n) + x_n.$$

It is well known that the convergence condition of fixed iteration is:

$$\left\|\frac{\partial \phi}{\partial x}\right\| = \left\| h * \frac{\partial F(x)}{\partial x} + 1 \right\| < 1.$$

If $h = 1$, the convergence condition of classical Fixed Point Iteration Method does not satisfy. We can reduce h, if only all of the eigenvalues of the Jacobian matrix of $F(x)$ is located in the left-half plane, when h is small enough, the inequality $\left\|\frac{\partial \phi}{\partial x}\right\| < 1$ must hold.

We know that explicit ODE method only has very small stability region. If the Jacobian matrix of $F(x)$ has large condition number, or the system is stiff, the conventional explicit ODE method will be very inefficient.

3) Does the explicit ODE method with the large stability region exist? The answer is positive. In [9] the authors of this paper proposed a new ODE method which has a very large stability region. Choosing proper parameter, the region in real axis direction can reach ∞ point. It seems strange, the textbook tells us the stability region of the explicit ODE method must be bounded, however our method includes a parameter ε. It plays a very important role for stability and [11] gives it a reasonable explanation.

Our purpose is to reach the steady point of the dynamic system $\frac{\mathrm{d}x}{\mathrm{d}t} = F(x)$ as quickly as possible. We do not require the trajectory produced by our method approximates the solution curve of differential equation of $\frac{\mathrm{d}x}{\mathrm{d}t} = F(x)$. Or we can say that our method is allowed to have large errors, if only the trajectory produced by our method is within the attracting domain of the steady point. So our method can take the very large step size, and goes to steady point quickly. The traditional explicit ODE methods cannot take large step size. Usually, their trajectory is very close to the solution of the system. In a sense the trajectory approximately reflects the real transient process of the system, however the trajectory produced by our method does not. So, our method can be recognized as "Pseudo-Transient Continuation."

The existing "Pseudo-Transient Continuation" method can be written as follows:

$$\begin{aligned} x_{n+1} &= x_n + \left(h^{-1} - F'(x_n)\right)^{-1} * F(x_n) && [11] \\ &= x_n + h * \left(I - h * F'(x_n)\right)^{-1} * F(x_n) && [12] \end{aligned}$$

Various algorithms adopt different strategies to control h. They are known as the "LINEARLY IMPLICIT EULER DISCRETIZATION", [12].

Because each iteration step needs to solve a system of linear equations, so all of them are implicit methods.

Our method is the unique Explicit "Pseudo-Transient Continuation" method. In order to express this difference, our method is named by [11] as "Explicit Pseudo-Transient Continue" method.

From [9] and [13], we can write our scheme as follows:

For $\frac{\mathrm{d}x}{\mathrm{d}t} = F(x)$,

1) $x_{n+1}^0 = x_n + z_n$.

2) $z_{n+1} = \omega * \left(\varepsilon * F\left(x_{n+1}^0\right) + z_n\right)$.

3) $x_{n+1} = x_n + z_{n+1}$.

Here $\omega = \frac{h}{(h+\varepsilon)}$, ε is a parameter, h is the step size.

For $n = 0$, $z_0 = h * F(x_0)$, from 1), 2) and 3) we can get x_1^0, z_1, x_1, Then for $2, 3, \cdots, n \cdots x_2^0, z_2, x_2; x_3^0, z_3, x_3; \cdots x_n^0, z_n, x_n; \cdots;$

If the steady point x^* of $F(x)$ exists, for $n = 1,2,\cdots,$ we can get certain x_n which satisfies $\left\| F(x_n) \right\| < \text{Tol}$, then x_n is the approximation of x^*, the solution we expect.

3. The Advantage of the New Method

1) Our method is an explicit stiff ODE method, so it has a very strong capability to deal with tough problems. Brown's almost linear equation is as follows: [7] [14]

$$f_i = x_i + \Sigma x_{j,\, j=1,n} - (n + 1) = 0, \text{ for } i = 1,2,\cdots n-1$$

$$f_n = \left(\Pi x_{j,\, j=1,n} \right) - 1 = 0$$

$$x_0 = (0.5, 0.5, \cdots 0.5)^{\mathrm{T}}$$

Obviously, $x^* = (1.0, 1.0, \cdots 1.0)^{\mathrm{T}}$ is a solution.

For n = 10, 30, Newton method diverges drastically, but Brown method converges well.

If n > 35, according to [7], Brown method will not work. This has been proved by [14], for n = 40.

What happens to our method? For n = 40, 100, 200, 400, we can get the solution easily. In this paper, we just show the result of n = 400:

Virtually, the differential equation to deal with is:

$$\frac{\mathrm{d}x}{\mathrm{d}t} = D(x)^{-1} F(x).$$

Here the diagonal matrix $D(x)$ has elements $d_i(x) = 2.0$, $i = 1,2,\cdots,n-1$, $d_n(x) = \prod x_{j,\, j=1,n-1}$. If $d_n(x) < 1.0$, the division has been avoided, or we put $d_n(x) = 1.0$.

As we did in [9], we separated the calculation into three stages (in every stage we expect the 2-Norm of $F(x)$ is descending about 5 magnitude order) and took parameters $\varepsilon = \frac{2.0}{400.0}$, Tol1 = 1D−0, Tol2 = 1D−5, Tol3 = 1D−10; h1 = 0.015, h2 = 0.15, h3 = 0.8. After 826 iterations, we got the solution: $x_i = 1.00000000$, $I = 1,2,\cdots,399$; $x_{400} = 1.00000002$. The initial value of 2-Norm of $F(x)$ is 0.4005D+04. The results are listed in **Table 1**. In the table, NFE denotes Number of Function Evaluations. Double precision was used.

Incidentally, the efficiency of Brown method is strongly related to the order of the equations. It is more efficient if the equations are ordered by increasing the degree of nonlinearity. In our method, each component can be calculated independently, so there is no ordering problem.

An experiment has been done: We place the nonlinear equation on the top of the $F(x)$, *i.e.* the first equation. Both of them have exactly the same results.

2) Our method is as simple as fixed iteration, so it can handle very large systems. Here we give an ill-conditioned, non-symmetric linear equations with dimension n = 1,000,000:

$$Ax = b.$$

Here

$$A = UDU,\ U = U^{-1} = U^T = I - \left(\frac{2}{u^T u} \right) uu^T$$

$$u = (1.0, 1.0 \cdots, 1.0)^{\mathrm{T}}$$

Table 1. (Brown) $\varepsilon = \frac{2.0}{400.0} = 0.5D{-}02$.

Tolerance	Step size	NFE	2-Norm of $F(x)$
1D−0	0.015	9	0.5258D−00
1D−5	0.15	55	0.7321D−05
1D−10	0.8	826	0.8798D−10

$$D = \begin{pmatrix} D_1 & & & \\ . & D_2 & & \\ & . & & \\ & & & D_{500000} \end{pmatrix}.$$

$$D_i = \begin{pmatrix} i & 1 \\ -1 & i \end{pmatrix}$$

We ask the solution is $x^* = (1.0, 1.0, \cdots, 1.0)^{\mathrm{T}}$, so that we can determine the right side vector *b*. The iterative initial $x(0) = (0.5, 0.5, \cdots, 0.5)^{\mathrm{T}}$. The differential equation to deal with is:

$$\frac{\mathrm{d}x}{\mathrm{d}t} = \frac{-Ax + b}{200000.0}.$$

If E-Ψtc method is used, taking $\varepsilon = 0.5$. For any step size the numerical integration is stable.

The only possible comparing method is fixed iteration (Euler) method. The maximum step size is $h = 0.8$. In order to get a good stability, we take $h = 0.75$. The initial Euclid norm is 0.7217D+03 After 150,000 fixed iterations, we have the following results:

Iteration number	Euclid norm
0	0.7217D+03
50,000	0.2238D−04
100,000	0.8389D−05
150,000	0.4845D−05

We want to make the Euclid norm less than 1D–08, it seems impossible.

How about the Explicit Pseudo-Transient Continue” method. (E-Ψtc)?

The following data can give the answer: (Variable step size)

Iteration number	Euclid norm	Step size
0	0.7217D+03	
100	0.1721D−01	5.0
200	0.6852D−02	25.0
650	0.8323D−03	50.0
1600	0.8600D−04	80.0
2850	0.8609D−05	100.0
4550	0.9758D−06	120.0
8350	0.9320D−07	150.0
14800	0.9473D−08	200.0

3) Another outstanding feature is that our method is very suitable for parallel calculation.

4. Load Flow Calculation

Now we turn our focus to the main topic of this paper—Load Flow Calculation.

1) The Construction of the Differential Equations.

The load flow equations can be formulated by a system of nonlinear.

Equations: $F(X) = 0$.

Here $X=(V_1,V_2,\cdots,V_n)^{\mathrm{T}}$, $V_i=e_i+j*f_i,\ j=\sqrt{-1}$.

In rectangular coordinate system, for PQ buses, the power equations are:

$$\Delta P_i=P_i-e_i\Sigma_{j=1,n}\left(G_{ij}e_j-B_{ij}f_j\right)-f_i\Sigma_{j=1,n}\left(G_{ij}f_j+B_{ij}e_j\right),\tag{3.1}$$

$$\Delta Q_i=Q_i-f_i\Sigma_{j=1,n}\left(G_{ij}e_j-B_{ij}f_j\right)+e_i\Sigma_{j=1,n}\left(G_{ij}f_j+B_{ij}e_j\right).\tag{3.2}$$

For PV buses, the voltage magnitude values V_{is} are specified and have the relation:

$$\Delta V_i^2=V_{is}^2-e_i^2-f_i^2\tag{3.3}$$

(3.1) and (3.3) constitute the power equations for PV buses.

For elimination method, the optimal ordering of the buses is the first task we need to solve. In our method, this problem does not need to be considered, because the memory of space only depends on the number of buses and not related to the ordering. For the calculating process, as we have pointed out in Section 3, in our method, each component can be calculated independently.

But for every bus, which corresponds to two real number equations. If the nonlinear equations $F(x)=0$ is ordered as following:

$$\left(\Delta P,\Delta Q,\cdots\Delta P,\Delta V^2,\cdots\Delta P,\Delta Q,\cdots\Delta P,\Delta V^2\right)^{\mathrm{T}}.$$

The corresponding differential equation should be arranged as follows:

$$\left(\frac{\mathrm{d}f}{\mathrm{d}t},\frac{\mathrm{d}e}{\mathrm{d}t},\cdots\frac{\mathrm{d}f}{\mathrm{d}t},\frac{\mathrm{d}e}{\mathrm{d}t},\cdots\frac{\mathrm{d}f}{\mathrm{d}t},\frac{\mathrm{d}e}{\mathrm{d}t}\cdots\frac{\mathrm{d}f}{\mathrm{d}t},\frac{\mathrm{d}e}{\mathrm{d}t}\right)^{\mathrm{T}}.$$

Because in the i^{th} bus, the active power P_i is mainly concerned with f_i and reactive power Q_i is mainly concerned with e_i ([15] p. 205). The diagonal element of the Jacobian of $F(x)$ will be the one which has the largest absolute value in each row. This is crucial for any iterative method.

In fact, for elimination method, every bus corresponds to a 2-dimension sub-block and which is also arranged just the same.

In order to let the readers understand the discussion above more clearly, let us give a further statement by a simple example:

Consider a 2-dimensional linear equations:

$$x+2*y=4$$
$$10*x+4*y=24$$

The matrix-vector form is:

$$AX=C.$$

Here $A=\begin{pmatrix}1 & 2\\ 10 & 4\end{pmatrix}$, $X=(x,\ y)^{\mathrm{T}}$, $C=(4,\ 24)^{\mathrm{T}}$.

If we use a fixed point iteration method to solve this linear equations $C-AX=0$, which will not converge. This is because the two eigenvalues of the matrix A $\lambda_1\approx\frac{5+9.434}{2}$ and $\lambda_2\approx\frac{5-9.434}{2}$, they do not satisfy half-plane condition.

If we alter the sequence of x and y, consider equations $C-BY=0$.

Here $B=\begin{pmatrix}2 & 1\\ 4 & 10\end{pmatrix}$, $Y=(y,\ x)^{\mathrm{T}}$, $C=(4,\ 24)^{\mathrm{T}}$.

The two eigenvalues of the matrix B, $\mu_1\approx\frac{12+8.9}{2}$ and $\mu_2\approx\frac{12-8.9}{2}$, so all eigenvalues of matrix-B are located on the left half-plane. Choosing proper parameter, the fixed point iteration will converge.

From the differential equations' point of view:

For differential equations:

$$\frac{\mathrm{d}y}{\mathrm{d}t}=4-(x+2*y)$$

$$\frac{\mathrm{d}x}{\mathrm{d}t}=24-(10*x+4*y)$$

Choosing a proper step size *h*, Euler method will make (y,x) step by step towards the stability point (1, 2).

However, for the differential equations:

$$\frac{\mathrm{d}x}{\mathrm{d}t}=4-(x+2*y)$$

$$\frac{\mathrm{d}y}{\mathrm{d}t}=24-(10*x+4*y)$$

No matter what *h* is, (x,y) cannot converge.

The matrix B is diagonal dominant, but matrix A is not. This is why for every bus, we arrange the sequence according to (f,e). Only by this way, we can guarantee every sub-block of the matrix is diagonal dominant.

Here we would like to emphasize this matter again. The algebraic equations and differential equations were all the same, just the unknown variables have different order—one is (x,y) and another one is (y,x), which causes a completely different iterative process. One is divergent, but another one is convergent.

Now, let us see what is in the reference [1].

[1] Page 1736, right side:

(Ex.) $x=\text{col.}[e_1,f_1,e_2,f_2]$.

(Ex.) $y(x)=\text{col.}\left[P_1(x),Q_1(x),P_2(x),|V_2(x)|^2\right]$.

This expression is not correct. It will make the Jacobian matrix loose diagonal dominant. That is to say, for each row of the Jacobian matrix, the element with the largest absolute value will not be diagonal element. Obviously, for PV-bus, the one of diagonal elements will be "−2f". If flat start is used, then for the first step, the diagonal element will be zero. It will be the worst thing for any algorithm to solve the linear algebraic equations. For elimination method, almost all code, column pivot technique was taken. It automatically adjusts the order of equations so people usually ignore the "order problem". But for any iterative method, the "order problem" is crucial.

The correct expression should be:

(Ex.) $x=\text{col.}[f_1,e_1,f_2,e_2]$.

(Ex.) $y(x)=\text{col.}\left[P_1(x),Q_1(x),P_2(x),|V_2(x)|^2\right]$.

Or the matrix form:

$$\begin{pmatrix}\Delta P\\ \Delta Q\\ \Delta V^2\end{pmatrix}=\begin{pmatrix}H & N\\ J & L\\ R & S\end{pmatrix}=\begin{pmatrix}\Delta f\\ \Delta e\end{pmatrix}.$$

For the details, please see [15] page 184.

We can also change the order of equations;

(Ex.) $x=\text{col.}[e_1,f_1,e_2,f_2]$.

(Ex.) $y(x)=\text{col.}\left[Q_1(x),P_1(x),|V_2(x)|^2,P_2(x)\right]$.

2) Preconditioning

We will give two preconditioning strategies:

a) point diagonal preconditioning: The preconditioner is a diagonal matrix of the Jacobian. If you do not want to calculate the diagonal elements, according to the observation of the calculated data, we can further simplify this strategy.

For PQ bus:

$$\frac{\partial(\Delta P_i)}{\partial f_i} \approx B_{ii}$$

$$\frac{\partial(\Delta Q_i)}{\partial e_i} \approx B_{ii}$$

For PV bus:

$$\frac{\partial(\Delta P_i)}{\partial f_i} \approx B_{ii}$$

$$\frac{\partial(\Delta V_i^2)}{\partial e_i} \approx -2.0$$

The precondition matrix can be simplified as a constant diagonal matrix. The elements are just B_{ii}, and −2.0. It was named as D_1. The differential equation that we deal with is:

$$\frac{\mathrm{d}x}{\mathrm{d}t} = -(D_1)^{-1} F(x).$$

b) 2-dimension block diagonal preconditioning: We know every bus corresponds to a 2-dimension diagonal sub-block of the Jacobian matrix. These sub-blocks constitute a block diagonal matrix D_2:

The differential equation that we deal with is:

$$\frac{\mathrm{d}x}{\mathrm{d}t} = -(D_2)^{-1} F(x).$$

The preconditioning not only raises efficiency, but also helps us to determine the parameter $\varepsilon = 0.5$. Because the Jacobian matrix diagonal elements of $(D_1)^{-1} F(x)$ and $(D_2)^{-1} F(x)$ are very close to 1, and it is fully possible, ∞-Norm of the Jacobian matrix < 3.0 (because the largest absolute value element of the Jacobian matrix is in the diagonal place of each row and the sparse property of the matrix) and $(0.5*\rho) < \frac{4}{3}$, ρ is the spectral radius. (See the reference [10]). By this means, there is only one parameter h which needs be determined.

For ordinary load flow problem, if point diagonal preconditioning is used, *h* is in the range of 5 to 7 is suggested. If an iteration cannot converge, we suggest to reduce *h*. If *h* is small enough, say $h < 1$, the iteration still cannot converge, the iteration should be stopped, then check the whole process to see what mistake was taken.

5. 43-Bus System

For this system, 8 calculating results are given, we take the Largest Absolute Values of the Mismatch (LAVM), less than 1D−05 as the stopping condition (double precision):

1) Normal Newton method: 7 Newton iteration, The LAVM = 0.2123D−08.

2) Optimal Multiplier Newton method: 6 Newton iteration. The LAVM = 0.4134D−08. (The values of optimal multiplier μ are taken from [1]).

3) Euler method: step size $h = 0.00065$ (maximum step-length to meet the stability requirements), 157006 function evaluations. The LAVM = 1D−05.

4) No preconditioning E-Ψct: $\varepsilon = 0.0005$, $h = 0.00085$, 120076 function evaluations. The LAVM = 1D−05.

5) Point diagonal preconditioning E-Ψct: $\varepsilon = 0.5$, $h = 0.35$, 42505 function evaluations. The LAVM = 0.9998D−05.

6) 2-dimension block diagonal preconditioning E-Ψct: 2008 function evaluations. The LAVM = 0.9995D−05.

7) Variable step size 2-dimension block diagonal preconditioning E-Ψct: $\varepsilon = 0.5$, h1 = 6.5, h2 = 7.0, h3 = 7.5, h4 = 8.0, h5 = 9.0, h6 = 10.0.

1335 function evaluations. The LAVM = 0.9968D−05.

What conclusion that we can get based on the 1)-7) results?

From 3) we can see that this system is surely an ill-conditioned system, but from 1), the ill-conditioned extent is not severe, using double precision, only 7 Newton iterations are needed to make the LAVM = 0.2123D−08.

Compare 1) and 2), Optimal Multiplier Newton method saved one iteration, if the stopping condition is 1D−08. This is the contribution of [1].

Compare 3) and 7): the proportion of numbers of function evaluation is 157006/1335 ≈ 118. For 7), besides function evaluations, forty-two 2-dimension sub-matrix need to be inverted in each iteration. The workload of 2-dimension matrix inverse is relatively very small, so we can say that for ill-conditioned Load Flow Problems, using 7) is a good choice, especially when the system is large.

For 7), if someone considers that the workload of 1335 function evaluations is too big to be accepted, here we give another choice. This choice is based on the following observation: For Newton method, the quadratic convergence superiority can be utilized only when the norm of $F(x)$ is small, say it is less than 0.1, but for function iteration methods, at the beginning of the iteration, the norm has faster descending speed. So a hybrid calculation strategy is recommended.

8) At the beginning, we use 6) (ε = 0.5, h = 6.0), after 132 function evaluations, the LAVM = 0.9821D−01, then we switch to normal Newton iteration, after 4 iterations, the LAVM = 0.3366D−12. By directly using the normal Newton method, if LAVM < 1D−12 is required, 8 iterations are needed. That is to say, at the cost of 132 function evaluations, we can save 4 normal Newton iterations.

Here, what we want to say is as following:

For any existing Newton program, you can use our method as a tool to choose a good starting point. The textbook tells us: “The HOMOTOPY method can be used to generate a good starting value”. So our method can be considered as a homotopy method. It can be used independently and can also be embedded into the Newton program as a tool to choose starting value.

There are a lot of references related to “HOMOTOPY”, we merely list three of them [17]-[19].

For ordinary load flow problem, e.g. IEEE-39 IEEE-118, our method is also suitable. You can use the method by two ways: One is directly solve nonlinear equations, like what we did in this paper. [20], and another one is to solve the linear equations arising from the Newton method [21].

For nonlinear equations, the solution is not unique, as a research paper, we should give the final numerical results or some description for large systems, but neither did [1] nor [2]. In this paper, listing all the 8 final results is not practical, because of the limited space. Here we only point out that no matter the LAVM is 1D-05 or 1D-12, the final results are almost the same. For bus 43, all of them are exactly the same: voltage magnitude is 1.0763, angle is −11.176.

6. 11-Bus System

At the beginning, the initial LAVM = 0.1232E+01, we give the following results:

1) As [1] did, the normal Newton method, after 18 iterations, we have the results (Single precision):

0.3046E+00, 0.1330E+00, 0.8979E−01, 0.1883E−01, 0.2195E−02, 0.7789E−03,
0.2557E−03, 0.8268E−01, 0.2074E−01, 0.5251E−02, 0.1383E−02, 0.4276E−03,
0.2818E−03, 0.1217E−02, 0.3891E−03, 0.3055E−03, 0.5773E−03, 0.2670E−03.

2) Optimal multiplier method (single precision), 18 iterations. Results are:

0.2858E+00, 0.1326E+00, 0.3429E−01, 0.1691E−01, 0.5123E−02, 0.4473E−03,
0.2464E−03, 0.2164E−03, 0.2155E−03, 0.2155E−03, 0.2155E−03, 0.2154E−03,
0.2155E−03, 0.2155E−03, 0.2154E−03, 0.2155E−03, 0.2155E−03, 0.2154E−03.

If we use double precision, we have almost the same results as above 1) and 2). Based on this observation, we can see:

a) Oscillation occurred after 7 normal Newton method iterations, but for the optimal multiplier method, oscillation did not appear.

b) The LAVM cannot go down to zero and this is not caused by the defect of numerical precision of the computer and/or programming code.

c) After 7 iterations, the LAVM of the optimal multiplier method is 0.2464E−03. After 7 iterations the LAVM of the normal Newton method is 0.2557E−03.

3) How about the E-Ψtc method? We use point diagonal preconditioning strategy. Taking ε = 0.5, h = 1.5, after 178 function evaluations, the LAVM is less than 10^{-4} (0.995493×10^{-4}).

Unless it has a very small oscillation, the LAVM basically keeps monotone descending. (Both single and double precision have the same results). The LAVM is about 1/2 of the optimal multiplier method.

4) When using Inexact Newton-GMRES (15), we use single and double precision to calculate, the results are almost the same. Here we give the double precision result: (For every GMRES loop, we calculate the initial residue R_0 and the final residue R_1 of the linear equations, and take $R_l/R_0 < 0.1$, or Maximum Number of Restart = 100 as the stop condition for each Newton iteration.)

$\mathrm{NT}=1$	Restart Number $L=1$	$R_l/R_0 = 0.3728\mathrm{D}{-}01$	$\mathrm{LAVM} = 0.3030\mathrm{D}{+}00$
$\mathrm{NT}=2$	Restart Number $L=3$	$R_l/R_0 = 0.4500\mathrm{D}{-}01$	$\mathrm{LAVM} = 0.1291\mathrm{D}{+}00$
$\mathrm{NT}=3$	Restart Number $L=4$	$R_l/R_0 = 0.9350\mathrm{D}{-}01$	$\mathrm{LAVM} = 0.5252\mathrm{D}{-}01$
$\mathrm{NT}=4$	Restart Number $L=100$	$R_l/R_0 = 0.3396\mathrm{D}{+}00$	$\mathrm{LAVM} = 0.1961\mathrm{D}{-}01$
$\mathrm{NT}=5$	Restart Number $L=72$	$R_l/R_0 = 0.9481\mathrm{D}{-}01$	$\mathrm{LAVM} = 0.1044\mathrm{D}{-}01$
$\mathrm{NT}=6$	Restart Number $L=100$	$R_l/R_0 = 0.4000\mathrm{D}{+}00$	$\mathrm{LAVM} = 0.2825\mathrm{D}{-}02$
$\mathrm{NT}=7$	Restart Number $L=100$	$R_l/R_0 = 0.9975\mathrm{D}{+}00$	$\mathrm{LAVM} = 0.2796\mathrm{D}{-}02$
$\mathrm{NT}=8$	Restart Number $L=100$	$R_l/R_0 = 0.1000\mathrm{D}{+}01$	$\mathrm{LAVM} = 0.2796\mathrm{D}{-}02$
$\vdots$	$\vdots$		$\vdots$
$\mathrm{NT}=16$	Restart Number $L=100$	$R_l/R_0 = 0.1000\mathrm{D}{+}01$	$\mathrm{LAVM} = 0.2796\mathrm{D}{-}02$

That is to say, after 7 Newton iterations, there are totally (1 + 3 + 4 + 100 + 72 + 100 + 100 = 380) 380 × 15 = 5700 GMRES iterations, finally the LAVM reaches 0.2796D−02, then it no longer descends.

To summarize the results above:

1) Using the normal Newton method, the oscillation occurs;

2) After 9 optimal multiplier iterations, the LAVM keeps 0.2155×10^{-3};

3) With E-Ψtc method, after 178 function evaluations, the LAVM reaches 0.995493×10^{-4};

4) By using Inexact-Newton-GMRES (15), after 5700 GMRES iterations, LAVM = 0.2796×10^{-2}.

All results we got show that for the 11-bus system, the LAVM cannot go down to zero, in another word, for this system, the solution of $F(x)=0$, does not exist. What we should do is to seek the minimum of the function

$$(1/2) * \Sigma_{i=1,11} f_i^2(x).$$

7. Solving Gradient Equation to Seek the Minimum for Bus-11

The first choice should be the Gauss-Newton method:

$$\Delta x = -\left(J^T * J\right)^{-1}\left(J^T * F(x)\right),$$

if the Jacobi is a $N * N$ nonsingular matrix, Gauss-Newton method will change into Newton method. As we have pointed out after 7 iterations the oscillation occurred.

We can also solve the gradient equations by using Newton method. In order to compare the two results, re-solve it again by LINPACK SUBROUTINE DGECO. By this way, we can get the condition number of the matrix sequence $J^T * J$. The results are as follows in **Table 2** (Rcond means the Reciprocal of the condition number).

Comparing the two results, we can see that before oscillation occurs, except 6th-iteration has tiny difference, both results are exactly same. It shows, even if the condition number of the $J^T * J$ increases to the *J*’s square, we can still get the good results. This fact proves again that bus-11 is not an ill-conditioned system.

What about the Point Diagonal Preconditioning E-Ψct for the gradient equation?

The differential equation should be:

$$\frac{\mathrm{d}x}{\mathrm{d}t} = -(DD)^{-1} J^T * F(x).$$

Here *DD* is the diagonal matrix of $J^T * J$. We have the following results in **Table 3**.

One may think 8858 gradient function evaluations are too expensive, but please note the step size is 20.0 - 600.0. If we use fixed iteration (Euler method), the step size is not greater than 0.76. We try this method, the

Table 2. Results of Newton Method for gradient equation by LINPACK DGECO.

Iteration	Rcond	LAVM
1	0.5843D−06	0.3046D+00.
2	0.6465D−06	0.1330D+00
3	0.3566D−06	0.8979D−01
4	0.1804D−06	0.1883D−01
5	0.4104D−07	0.2195D−02
6	0.9671D−08	0.7329D−03
7	0.1611D−08	0.2557D−03
8	0.1187D−11	0.8330D−01
23	0.9184D−12	0.1070D+00
28	0.1452D−08	0.2654D−03
49	0.3995D−11	0.2469D−01
50	0.1886D−06	0.6238D−02

Table 3. Results of Point Diagonal Preconditioning E-Ψtc for gradient equation.

Iteration	Step size	Norm of gradient	Norm of $F(x)$	LAVM
0		0.2215D+03	0.2621D+01	0.1232D+01
930	20.0	0.9991D−02	0.8234D−02	0.3678D−02
1734	50.0	0.9565D−04	0.2209D−02	0.1125D−02
3955	200.0	0.8513D−05	0.2647D−03	0.1712D−03
8858	600.0	0.9891D−07	0.2089D−03	0.1027D−03

step size is from 0.73 − 0.76, it needs 12,902,751 evaluations. The proportion of the number of the evaluating gradient functions is 12902751/8858 ≈ 1456. In the later period of iteration, these two iterations keep monotone descending. When $\|F\|$ reaches 0.0002089, LAVM reaches 0.0001027. They keep no variation. So we can determine the minimum of $\|F\|$ is 0.0002089. At this time, can we say an efficient seeking minimum method, E-Ψtc, has been achieved? Some people can reject this statement, the reason is conventional Gauss-Newton method only needs 7 iterations, which can make LAVM reach 0.0002557. Ours answer is that, after 7 iterations, the LAVM continues to oscillate and if the system is large, say the dimension is 10,000, then to solve 7 linear equations with dimension of 10,000, the workload is surely much greater than 8858 function evaluations.

8. Discussion

Usually we evaluate the accuracy of a solution by the mismatch of the equation, but for ill-conditioned problem, it might be misleading. Here we give an extreme example.

Consider a 2-dimension linear algebraic equations:

$$0.2161 * x_1 + 0.1441 * x_2 = 0.1440$$
$$1.2969 * x_1 + 0.8648 * x_2 = 0.8642$$

If substitute x_1 by 0.9911, and x_2 by −0.4870, the mismatches will be -10^{-8} and 10^{-8}. People might think the most accurate solution has been obtained. Unfortunately, s/he is wrong. The exact solution of these equations is 2 and −2. From a mathematical theory, discriminating the accuracy of a linear equation, we need to resort the knowledge of the condition number [16], but this is a very difficult task to implement, especially for a real world problem.

We suggest taking the following easy ways to identify if the system is ill-conditioned or not:

Using both single and double precision to do the calculation, if the results basically keep coincidence, we can consider the system is well-conditioned.

If using a Normal Newton method, for 43-bus system, single precision cannot make the LAVM less than 1E−5, but for double precision, the LAVM can go down less than 1D−12 easily, so 43-bus system is surely an ill-conditioned system.

For 11-bus system, we do the same process above, the results of single and double precision are almost the same, the LAVM cannot go down to 1E−4 and 1D−4. After 7 Newton iterations, the LAVM are 0.255709E−03 and 0.255653D−03, then oscillation occurs for both iterations. So we confirmed that the 11-bus system is not an ill-conditioned system, but the LAVM cannot go down to zero.

In order to comprehend this phenomenon, consider the equation:

$$F(x) = x^2 + 0.0001 = 0\,.$$

The solution does not exist, but at $x = 0.0$, $F(x)$ reaches the minimum 0.0001. Let us see what happens if the normal Newton method is used. The initial value of $x_0 = 0.5$, $F(x_0) = 0.2501$, after a sequence of monotone descend, at 6th iteration, $x = 0.2995\text{D}{-02}$, $F(x) = 0.1090\text{D}{-03}$, then oscillation occurs, at 10th iteration, $x = -0.5698\text{d}{-03}$, $F(x) = 0.1003\text{D}{-03}$, then oscillation occurs again. This oscillation continues until $F(x)$ reaches its minimum 0.0001, at this time the value of x is 0.8337D-04.

If $F(x) = x^2 = 0$, the oscillation does not occur for Newton iteration. After 536 iterations, $x = 0.2223\text{D}{-161}$, $F(x) = 0.4941\text{D}{-323}$. After 537 iterations, x and $F(x)$ keep 0.1111D−161, and 0.0000, have no any variation. That is to say, we have an approximate solution, and the accuracy reaches 0.1111×10^{-161}.

In engineering design, making $\|F\| = 0$ or very small, e.g. $\|F\| < 10^{-15}$ is almost impossible, if we can make $\|F\| < 10^{-5}$ or $<10^{-6}$, usually the design is considered acceptable.

Strictly speaking, for the engineering design problem, the solution of $F(x) = 0$, in the sense of mathematics, does not exist. What we can do is to find the minimum. As for the size of the minimum, which is determined by the requirements of the engineering projects. Usually considering $\|F\| < 10^{-10}$ is good enough and $\|F\| < 10^{-5}$ or $<10^{-6}$ is acceptable.

BUT, finding the minimum of $F(x)$ is an even harder task when comparing with solving $F(x) = 0$. In the process of solving $F(x) = 0$, if $\|F(x)\|$ around some small value is oscillating, we can recognize that the minimum has been obtained. For 43-bus system, the minimum is about 1D−12, so this system is well designed. As for the 11-bus system, the E-Ψtc method gives the minimum 1D−04. We consider it is determined by the parameters of design, but it was said to be "operational" [1] pg. 1738.

To summarize our opinion, the 11-bus system is not ill-conditioned, the norm of $F(x)$ does not go down to zero, the LAVM can reach 10^{-4}, but in engineering it is "operational". For all the methods tested by us, the only method which can make LAVM to reach to 1D−04 is the E-Ψtc method.

Now let us discuss the reliability of the algorithm. In many literature, there has been too much emphasis on measuring the efficiency of the algorithm proposed by the author(s) but not enough on reliability. For the nonlinear equations, the non-uniqueness of the solution makes measuring the reliability being an important subject. Reference [14] proposed a strategy to test the reliability: Giving a standard starting point x_s which is regarded as very close to the solution, then choosing another two starting points $10x_s$ and $100x_s$, which are regarded as being far away from the solution. This strategy is totally not working for the power system. Almost all people take flat start ($1.0 + j*0$) as the standard starting point x_s, choosing ($10.0 + j*0$) and ($100.0 + j*0$) as the starting point is unimaginable.

For the power system, the authors of this paper propose the following strategy to test the reliability:

1) Using single and double precision to solve the equation $F(x) = 0$. If the LAVM of both goes down to less than 10^{-5} or 10^{-6} and the results basically keep coincidence, we can believe the results are reliable. If the results do not coincident, like the 43-bus system, which reminds us that this system could be an ill-conditioned system. In order to get reliable results, we can use double precision to make even smaller LAVM.

2) If single and double precision are used, the iterations do not converge, the doubts about the algorithm and the existence of the solution might be raised. We suggest that people change the algorithm first to see what happens. If the iteration still does not converge, the existence problem should be considered. We observe all the iteration processes, if all of the iterations diverges, we can affirm that the solution does not exist. If the solution

around some value oscillates and this value is relatively small, like the 11-bus system, we can recognize the solution exists, in engineering it is called "operational".

We do not suggest readers to change the initial values to improve the convergence, because for the power system, flat start is good enough, these initial values are very close to the solution.

Even though oscillation can also be regarded as divergence, in this paper, we discriminate those two different phenomena. Here, so-called divergence means drastic oscillation happens in the iteration process.

3) As we have mentioned before, the solution of a nonlinear equation is not unique. The small LAVM does not guarantee the solution we searched is the solution what we want to get. The reliability asks us to list the results produced by different methods and to compare them, but none of [1] or [2] gives the final results. Especially [2] should pay more attention to this demand, because to test a new method, the reliability is more important than anything else. On the last page (pg. 3657) of [2], there was a result of the 11-bus system, cited other literature. In order to compare, we list three results (voltage magnitude and angle): 1) page 3657, 2) optimal multiplier (6 iterations) and 3) point diagonal preconditioning E-Ψtc.

The value of the angle of the bus-2 is probably wrong.

From the three results above in **Table 4**, we can see they are basically coincident. The LAVM = 0.995493D−04 for E-Ψtc method, and the LAVM = 0.447367D−03 for 6 Optimal Multiplier iterations. As we pointed out before, after 9 optimal multiplier iterations, the LAVM keeps 0.2155D−03. In order to compare with the data on page 3657, we list the 6 iterations results instead of 9 iterations results.

9. Conclusions

In this paper, we give a review about references [1] and [2], and introduce E-Ψtc method as well as its application for Load Flow Problems.

We find out that the benefits of the optimal multiplier method can only occur in the initial period/stage of Newton iteration. For the 43-bus system, if the stopping condition is LAVM < 10^{-8}, it can save one Newton iteration. For the 11-bus system, it can eliminate the oscillation which occurs in the normal Newton method. During the later period, we should take care of the loss of quadratic convergence. The applicability of the optimal multiplier method is only for the Newton iterative method. Other methods, such as steepest descent method, are not available.

43-bus system is an ill-conditioned, using double precision can overcome "convergence trouble".

11-bus system is not an ill-conditioned. It does not have convergence trouble. The problem is that the LAVM of $F(x)$ is not as small as we usually expect, say 10^{-5} or 10^{-6}. It is likely 10^{-4}. Here we use the word "likely" to express the idea we are not sure. This is because that we resort solving $F(x)=0$ to get the LAVM. From

Table 4. Comparison of three results.

	page 3657			Optimal multiplier			E-Ψtc	
Bus-1	1.024	0.000	1	1.024	0.000	1	1.024	0.000
2	1.0552	1.215	2	1.0546	−2.469	2	1.0568	−2.413
3	1.0444	4.608	3	1.0436	−4.170	3	1.0461	−4.106
4	1.0306	2.830	4	1.0275	−2.877	4	1.0312	−2.841
5	1.0329	4.830	5	1.0320	−4.906	5	1.0349	−4.855
6	1.0501	2.916	6	1.0473	−2.961	6	1.0511	−2.925
7	0.8310	12.066	7	0.7743	−12.569	7	0.8082	−12.566
8	0.9131	14.857	8	0.8588	−15.626	8	0.9072	−15.475
9	1.1264	15.911	9	1.1266	−16.675	9	1.1912	−16.419
Bus-10	0.8266	21.033	10	0.7487	−22.753	10	0.8129	−21.998
Bus-11	1.0186	23.996	11	0.9778	−25.992	11	1.0666	−24.750

Note: For the data on page 3657, in the front of angle should have a negative sign.

strictly mathematical theory, we should solve the gradient equation. We surely did it. The result tells us, for the 11-bus system, LAVM = 0.0001027, the minimum of $\|F(x)\|$ = 0.0002089. We are very positive considering these results are correct.

[2] has tested Brown method, as we pointed out that "The method did not have a clear algorithmic description and hence was hard to implement". The authors of [2] did a hard work, applied this method to Load Flow Problems. Their effort should be appreciated, but the inference of Newton method when it was used to solve the 43 and the 11 bus systems—"DIVERGENT" was wrong.

For the 11-bus system, applying the Newton method cannot make LAVM down to zero. This is determined by the parameters of the system, and is not caused by the defect of Newton method. "Divergent" and "cannot make LAVM down to zero" are two different concepts. [2] had the statement that Brown methods "converge" with only 5 iterations, but did not give any data to describe what the "converge" means.

In this paper, we introduce a new method, E-Ψtc. By two examples, one of them is a tough nonlinear equation; another one is super lager scale ill-conditioned linear equations. These results fully demonstrate the capability of the new method. For Load Flow Problems, the preconditioning technique is needed. If the system is ill-conditioned, using 2-dimension block diagonal preconditioning. For well-conditioned problems, use point diagonal preconditioning. As for the reliability, we compare our results with the results of existing methods and find that they are basically coincident (we did not list the results of the 43-bus).

References

[1] Iwamoto, S. and Tamura, Y. (1981) A Load Flow Calculation Method for ILL-Conditioned Power Systems. *IEEE Transaction on Power Apparatus and Systems*, **PAS-100**, 1736-1743. http://dx.doi.org/10.1109/TPAS.1981.316511

[2] Tripathy, S.C., Durga Prasad, G., Malik, O.P. and Hope, G.S. (1982) Load-Flow Solutions for ILL-Conditioned Power Systems by a Newton-Like Method. *IEEE Transactions on Power Apparatus and Systems*, **PAS-101**, 3648-3657. http://dx.doi.org/10.1109/TPAS.1982.317050

[3] Tinney, W.F. and Hart, C.E. (1967) Power Flow Solution by Newton's Method. *IEEE Transactions on Power Apparatus and Systems*, **PAS-86**, 1449-1456. http://dx.doi.org/10.1109/TPAS.1967.291823

[4] Wallach, Y. (1968) Gradient Methods for Load Flow Problems. *IEEE Transactions on Power Apparatus and Systems*, **PAS-87**, 1314-1318. http://dx.doi.org/10.1109/TPAS.1968.292223

[5] Flueck, A.J. and Chiang, H.-D. (1998) Solving the Nonlinear Power Flow Equations with an Inexact Newton Method Using GMRES. *IEEE Transactions on Power Systems*, **13**, 267-273. http://dx.doi.org/10.1109/59.667330

[6] Chen, Y. and Shen, C. (2006) A Jacobian-Free Newton-GMRES (m) Method with Adaptive Preconditioner and Its Application for Power Flow Calculations. *IEEE Transactions on Power Systems*, **21**, 1096-1103. http://dx.doi.org/10.1109/TPWRS.2006.876696

[7] More, J.J. and Cosnard, M.Y. (1979) Numerical Solution of Nonlinear Equations. *ACM Transactions on Mathematical Software*, **5**, 64-85. http://dx.doi.org/10.1145/355815.355820

[8] Gay, D.M. (1975) Brown's Method and Some Generalizations, with Applications to Minimization Problems. Ph.D. Diss. Cornell U. Ithaca, N.Y.

[9] Han, T. and Han, Y. (2010) Solving Large Scale Nonlinear Equations by a New ODE Numerical Integration Method. *Applied Mathematics*, **1**, 222-229. http://dx.doi.org/10.4236/am.2010.13027

[10] Han, T., Luo, X. and Han, Y. (2011) Solving Large Scale Unconstrained Minimization Problems by a New ODE Numerical Integration Method. *Applied Mathematics*, **2**, 527-532. http://dx.doi.org/10.4236/am.2011.25069

[11] Kelley, C.T. and Liao, L.Z. (2013) Explicit Pseudo-Transient Continuation. *Pacific Journal of Optimization*, **9**, 77-91.

[12] Deuflhard, P. (2004) Newton Methods for Nonlinear Problem. Springer-Verlag, Berlin Heidelberg.

[13] Han, T. and Han, Y. (2013) Numerical Solution for Super Large Scale Systems. *IEEE Access*, **1**, 537-544. http://dx.doi.org/10.1109/ACCESS.2013.2280244

[14] More, J.J., Garbow, B.S. and Hillstrom, K.E. (1981) Testing Unconstrained Optimization Software. *ACM Transactions on Mathematical Software*, **7**, 17-41. http://dx.doi.org/10.1145/355934.355936

[15] Han, Z.X. (1992) Power System Analysis. Zhejiang University Publishing House. (In Chinese)

[16] Cline, A.K., Moler, C.B., Stewart, G.W. and Wilkinson, J.H. (1979) An Estimate for Condition Number of a Matrix. *SIAM Journal on Numerical Analysis*, **16**, 368-375.

[17] Watson, L.T., Sosonkina, M., Meliville, R.C., Morgan, A.P. and Walker, H.F. (1997) Algorithm 777 HOMPACK90: A Suite of FORTAAN 90 for Globally Convergent Homotopy Algorithms. *ACM Transactions on Mathematical Software*,

23, 514-549. http://dx.doi.org/10.1145/279232.279235

[18] Watson, L.T. (1990) Globally Convergent Homotopy Algorithms for Nonlinear Systems of Equations. *Nonlinear Dynamics*, **1**, 143-191. http://dx.doi.org/10.1007/BF01857785

[19] L1 Homotopy: A MATLAB Toolbox for Homotopy Algorithms in L1 Norm Minimization Problems.

[20] Xu, D.C. and Han, T.M. (2012) Applied Research of a New ODE Method in the Power Flow Computation of Power System. 2012 *IEEE Power Engineering and Automatic Conference* (*PEAM*), Wuhan, 18-20 September 2012, 1-4.

[21] Tian, S.M. and Han, T.M. (2012) A New Method Was Used for Solving Linear Algebraic Equations Arising from Inexact Newton Method Applying to Power Flow Problems. 2012 *International Conference on Electronic Information and Electrical Engineering*, Changsha, 15-17 June 2012.

Appendix (About Reference [1])

In conclusion (3) of [1], the authors said: "In the case that the solution continues to oscillate with the normal N-R method…

Applying the optimal multiplier, if the values of μ^* approaches zero with the iteration count increase the solution does not exist from the initial estimate...". The authors of this paper can give an example to show this conclusion is incorrect.

For the 43-bus system, if the voltage initial values of all buses are (0.7, j*0). The normal Newton method (double precision) continues to oscillate. The optimal multipliers of the optimal multiplier method are μ = 0.0411, 0.0218, 0.0217, 0.0000, 0.0009, 0.0003, 0.0048, 0.0005, 0.0000, 0.0000, … According to the conclusion (3), the solution does not exist. But in fact, the solution does exist and we can use Damp Newton method to get it. If the damp factor is small enough, *i.e.*, $\alpha = 0.007$, then after 1957 iterations, the solution can be obtained. The LAVM is less than 10^{-5} (incidentally $\alpha = 0.008$, after 2000 iterations, LAVM = 0.1516D+03). That is to say, for initial value (0.7 + j*0), the solution does exist, but the OPTIMAL MULTIPLIER METHOD cannot get it and gives an incorrect information. This is the reason why the authors of this paper think the conclusion (3) is incorrect.

Now let us discuss the statement of reference [1]: "supposing that the correction vector Δx is obtained by some way". For this statement, the authors of this paper consider that it is also incorrect.

In fact, [1] only give "APPLICATION OF THE OPTIMAL MLTIPLIER TO THE N-R METHOD" (Page 1737 Right Side). According the statement

If $\Delta x = -\nabla F(x)$, the authors of [1] should gave a similar description: "APPLICATION OF THE OPTIMAL MLTIPLIER TO THE STEEPEST DESCENT METHOD" But there is not. In fact the derivation process is only available for N-R METHOD and cannot generalize the derivation to the other methods. So the statement should be "supposing that the correction vector Δx is obtained by N-R METHOD".

What is the benefit of an Optimal Multiplier Method for the 43-bus system? The following figures can answer this question. For flat start, the initial LAVM is 0.5607 × 10^1, the successive iteration values for the normal Newton method and Optimal Multiplier Method are as follows:

0.3243D+01, 0.6209D+00, 0.1395D+00, 0.2084D−01, 0.1559D−2, 0.1657D−04,
0.2123D−08, 0.1656D−12, 0.4639D−12, 0.3622D−12 (Normal Newton Method)

0.1078D+01, 0.2838D+00, 0.3082D−01, 0.1252D−02, 0.3882D−05,
0.4134D−08, 0.5128D−10, 0.2423D−12, 0.3883D−12, 0.2913D−12
(Optimal Multiplier Method, Optimal Multiplier μ was taken from [1])

From the figures above, we can see that if the stopping condition is LAVM < 1D−08, the Optimal Multiplier Method needs 6 iterations, it can save one iteration when compare with normal Newton iteration.

But from another side, we see that at Optimal Multiplier Method's 5th and 6th iteration, the LAVM is 0.3852D−05 and 0.4134D−08 respectively. That is to say, if the values of μ were taken from [1], then Optimal Multiplier Method in the 6th iteration would lose the quadratic convergence nature.

The authors of this paper did the same work of [1] and found the values of μ have small differences than [1], we list them as following:

Ref. [1]:

0.6522, 1.1043, 1.0740, 1.1333, 1.0006, 1.0011, 0.9876, 1.0011, 1.0001, 0.9876.

Ours:

0.6525, 1.1042, 1.0746, 1.1311, 1.0011, 1.0000, 1.0000, 1.0117, 1.0449, 1.01002

The LAVMs of ours are:

0.1073D+01, 0.2840D+00, 0.3088D−01, 0.1296D−02, 0.4415D−05,
0.4944D−10, 0.1385D−12, 0.1818D−12, 0.2634D−12, 0.3000D−12.

In order to check the values of μ, we use another method to do the calculation.

For 3rd polynomial equation, there must be a real root, using the normal Newton method we can find this root, then to solve a 2nd polynomial equation, three roots can be obtained. Among them, the one which most close to 1, should be the optimal multiplier μ.

The results are as follows:

0.6525, 1.1042, 1.0746, 1.1311, 1.0011, 1.0000, 1.0000, 1.0000, 1.0000

We believe these results are even more reasonable.

The corresponding LAVMs are:

0.1073D+01, 0.2840D+00, 0.3088D−01, 0.1296D−02, 0.4415D−05,
0.4944D−10, 0.2855D−12, 0.2423D−12, 0.2307D−12, 0.2394D−12.

The results above show in our calculation, Optimal Multiplier Method does not lose quadratic convergence nature.