Applied Mathematics and Nonlinear Sciences 8(1) (2023) 1037-1044

Applied Mathematics and Nonlinear Sciences

http://journals.up4sciences.org

Power Flow Calculation in Smart Distribution Network Based on Power Machine Learning Based on Fractional Differential Equations

Yuan Ji [1,*], Jie Yuan [1] , Junfeng Qian [1], Liya Huang [1], Moaiad Ahmad Khder [2]

1. Information Center of Guizhou Power Grid Co., Ltd. , Guiyang, 550002, China
2. College of Arts & Science, Applied Science University, Bahrain

Submission Info

Communicated by Juan Luis García Guirao
Received February 20th 2022
Accepted April 10th 2022
Available online July 15th 2022

Abstract

Based on the theory of fractional differential equations, this paper proposes a simple recursive, iterative scheme for power flow calculation in pure radial networks. The paper determines the network hierarchy formed by the ADT stack through breadth theory. This helps us define the branch sequence of the forward and backward generation in the power flow calculation of the smart distribution network. We ensure that the Jacobian matrix remains unchanged in the smart distribution grid power flow calculation. The interval model is more practical and computationally simpler than the point model. The research results show that the power flow calculation method is efficient based on the fractional differential equation.

Keywords: Fractional differential equation; Power distribution network; Power flow calculation; Smart grid; Machine learning

AMS 2010 codes: 34A08

1. Introduction

Newton's method is a common method for performing power flow calculations. The convergence and robustness of Newton's method have been unsatisfactory when the system load is heavy [1].

The optimal multiplier method is considered a relatively successful algorithm for solving ill-conditioned power flow problems. The optimal multiplier method combines the calculation of the optimal multiplier with the conventional Newton's method to make the algorithm's convergence controllable [2]. However, this method does not overcome the problem of sensitivity to the initial value. Some scholars have proposed a load flow algorithm for load model admittance of heavy-load nodes. They replace reactive loads with ground admittance at heavily loaded nodes. This method changes the minimum norm eigenvalue of the Jacobian matrix near the critical point. At the same time, this method can better solve the convergence problem of heavy-load ill-conditioned power flow. However, due to the impedance

*Corresponding author.
Email address: yetevv@163.com

ISSN 2444-8656 DOI: 10.2478/amns.2022.2.0187

transformation of the load, the convergence result is a certain distance from the initial load condition.

The results obtained by this algorithm can only be regarded as approximate solutions. Some scholars have proposed two different parameterization methods to solve the ill-conditioned trend [3]. These two methods make the power flow calculation converge at the critical point by shifting the singular point of the Jacobian correction matrix. But its calculation accuracy is low, and the calculation time is relatively long. Some scholars have conducted in-depth research on the convergence of the tensor method for solving nonlinear equations. Some scholars have introduced two methods for solving the power flow equation based on the tensor method. But it is only applicable to the case where the tensor equation has zero roots and the case where Cartesian coordinates represent the power flow equation. Some scholars have compared the optimal multiplier method for Cartesian and polar coordinates. They believe that the optimal multiplier method in polar coordinates can solve the power flow problem better.

The optimal multiplier method is the same as some load flow algorithms that preserve nonlinearity. These algorithms all make use of higher-order information about the step size. Therefore, the above algorithm has better convergence than Newton's method, which only retains the first-order term. The optimal multiplier method obtains the Newton direction by performing a Newton iteration [4]. Then find the best step size. The step size is restricted to search only in the Newtonian direction. When the Jacobian matrix of the modified equation is singular or close to singular, the optimal multiplier method is very unreliable because the Newton step size or Newton direction cannot be obtained. Its iterations are also non-convergent. In this paper, two new methods based on the tensor method to solve the power flow of the power system are proposed. Method 1 is a power flow calculation method for obtaining a tensor correction amount based on the least-squares method. Method 2 is a power flow calculation method based on the direct tensor method in polar coordinates. The calculation results of several examples show that the method in this paper is effective.

2. Tensor Model

$J(x)$ tensor is a generalization of a vector: the derivative of a scalar concerning a multidimensional variable is a tensor of order 1 (usually a vector). The derivative of a vector concerning a multidimensional variable forms a tensor of order 2. The usual Jacobian matrix A is a tensor of order 2. A rank 2 tensor is derived from a multidimensional variable to form a rank 3 tensor T. By analogy, we can obtain the n dimensional r rank tensor. Accordingly, it has n^r components. make

$$g(x)=0 \tag{1}$$

Equation (1) is a nonlinear equation system. The incremental $d(k)$ for the k iteration can be obtained by solving the following system of linear equations:

$$J(x^{(k)})d^{(k)}=-g(x^{(k)}) \tag{2}$$

Where $J(x^{(k)})$ is the Jacobian matrix at the current point $x^{(k)}$. And then we get $x^{(k+1)}$ as $x^{(k+1)}=x^{(k)}+d^{(k)}$. The m dimensional tensor T of order 3 can be derived from the vector $g(x)$ twice concerning the variable x. After T is obtained, $(d^{T}Td)/2$ can be regarded as a quadratic term about d. Solving nonlinear equations based on formula (2) is called the tensor method. The main difficulty of the tensor method is how to calculate the quadratic term. There are 2 ways to calculate the quadratic term: the interpolation method and the direct method [5]. The interpolation method does not directly solve the quadratic differential of the function but obtains the quadratic term employing interpolation. Another way is to get the quadratic term by directly solving the quadratic differential of the function. This article calls it the direct method.

3. Power flow calculation method based on interpolation tensor method

The interpolation method is to uses the calculation results of several previous iterations. T is obtained by interpolation, so that the corresponding increment can be obtained by formula (2). If the Jacobian matrix $J(x^{(k)})$ is not singular, we multiply both sides of equation (2) by $s_i^T J^{-1}, i = 1, 2, \cdots, p$. We let $\beta_i = s_i^T d$, then the equation becomes

$$s_i^T J^{-1} g(x^{(k)}) + \beta_i + \frac{1}{2}\sum_{k=1}^{p}(s_i^T J^{-1} a_k)\beta_k^2 = 0 \tag{3}$$

When the quadratic equation of equation (3) about β_i has real roots, we can obtain the real roots β_i through equation (3). We refer to equation (2) or equation (3) as a tensor equation for short. d is obtained from the following equation:

$$g(x^{(k)}) + Jd + \frac{1}{2}\sum_{k=1}^{p} a_k \beta_k^2 = 0 \tag{4}$$

If we use two-point interpolation, its accuracy is sufficient in many cases. $p = 1, \beta$ is a scalar. It can be obtained by formula (4)

$$d = J^{-1}(-g(x^{(k)}) - \frac{1}{2}a\beta^2) \tag{5}$$

From formula (5), it can be seen that the increment d consists of two parts: the first part $d_a = -J^{-1}g(x^{(k)})$. It is Newton's correction and it is the main part of d . Part 2 $d_b = -J^{-1}[(a\beta^2)/2]$ is the tensor modifier. It is also a correction to d_a. We assume that the initial point is $x^{(0)}$, and the implementation steps of the method when iterating at step k are as follows:

1) Calculate the function value $g(x^{(k)})$ of the current point.
2) If the $g(x^{(k)})$ element with the largest amplitude satisfies the convergence accuracy condition, the algorithm ends.
3) Calculate the Jacobian matrix $J(x^{(k)})$ of the current point.
4) We solve for Newton's correction $d_a = -J^{-1}g(x^{(k)})$.
5) If β has a real root in equation (3), solve β and get the tensor correction $d_b = -J^{-1}[1/(2a\beta^2)]$. At this time, the total correction amount is $d = d_a + d_b$. If equation (3) has no real root, we use Newton's correction $d = d_a$.

The above basic algorithm requires that the expression (3) is a quadratic equation of one variable with respect to β_i with real roots. Equation $g(x^{(k)} + d) = 0$ in equation (2) has real roots with respect to β_i . The tensor correction cannot be obtained when it has no real roots, so we use a simple Newton correction [6]. When the quadratic equation of one variable about $g(x^{(k)} + d) = 0$ has no real roots β_i, it just means that the tensor correction cannot be obtained directly from the tensor equation. But we can turn it into a least-squares problem concerning d . We solve $d \in R^m$ such that $\| g(x^{(k)} + d) \|_2$ is minimal. At this point, we can find an approximate tensor correction. Its least squares problem can be expressed as $\min_{d \in R^n} \| Qg(x^{(k)} + d) \|_2$.

Where Q is an orthogonal matrix of $n \times n$. $Q = \begin{bmatrix} U^T \\ Z^T \end{bmatrix} \circ U = J^{-T} S[S^T (J^T J)^{-1} S]^{-1/2} \in R^{n \times p}, S$ is a matrix of $n \times p$. Each column is $s_i, i = 1, 2, \mathrm{L}, p$. $Z \in R^{n \times (n-p)}$ is a set of the orthonormal basis for the space spanned by the columns of matrix $J^{-T} S$.

If remember $W = [S^T (J^T J)^{-1} S], \beta = S^T d, q(\beta) = S^T J^{-T} g(x^{(k)}) + \beta + (S^T J^{-T} A\beta^2)/2$. β^2 Each element in is the square of the corresponding element of β, then we have

$$
\begin{gathered}
Qg(x^{(k)} + d) = \begin{bmatrix} U^T (g(x^{(k)}) + Jd + A\beta^2 / 2) \\ Z^T g(x^{(k)} + d) \end{bmatrix} = \\
\begin{bmatrix} W^{-1/2} S^T J^{-1} (g(x^{(k)}) + Jd + A\beta^2 / 2) \\ Z^T g(x^{(k)} + d) \end{bmatrix} = \\
\begin{bmatrix} W^{-1/2} q(\beta) \\ Z^T g(x^{(k)} + d) \end{bmatrix}
\end{gathered} \tag{6}
$$

For any $\beta \in R^p$ there is always $d \in R^n$ such that $Z^T g(x^{(k)} + d) = 0$ and $S^T d = \beta$. We get that the least-squares problem on d is equivalent to the least-squares problem on F: $\min\limits_{\beta \in R^p} \| W^{-\frac{1}{2}} q(\beta) \|_2$.

And because Q is an orthogonal matrix ($Q^{-1} = Q^T$), we can derive from equation (6) to get

$$
g(x^{(k)} + d) = Q^T \begin{bmatrix} W^{-\frac{1}{2}} q(\beta) \\ 0 \end{bmatrix} = UW^{-\frac{1}{2}} q(\beta) \tag{7}
$$

We substitute equation (2) and the expression of U into equation (7) to get

$$
g(x^{(k)}) + Jd + \frac{1}{2} A\beta^2 = J^{-T} SW^{-1} q(\beta) \tag{8}
$$

So have

$$
d = -J^{-1} (g(x^{(k)}) + \frac{1}{2} A\beta^2 = J^{-T} SW^{-1} q(\beta)) \tag{9}
$$

The first part of equation (9) is the Newton correction $d_a = -J^{-1} g(x^{(k)})$, and the second part is the tensor correction: $d_b = -J^{-1} \frac{1}{2} (A\beta^2 = -J^{-T} SW^{-1} q(\beta))$.

So far we have obtained a new solution method for tensor correction. It has nothing to do with whether the quadratic equation of equation (3) about β_i has real roots or not. We only require that the Jacobian matrix J is non-singular [7]. This paper refers to the power flow algorithm as method 1. We assume that the initial point is $x^{(0)}$, and when the k iteration is performed, the steps are as follows:

1) Calculate the function value $g(x^{(k)})$ of the current point.

2) If the $g(x^{(k)})$ element with the largest amplitude satisfies the convergence accuracy condition, the algorithm ends.

3) Calculate the Jacobian matrix $J(x^{(k)})$ of the current point.

4) Solve the Newton correction $d_a = -J^{-1} g(x^{(k)})$.

5) If the tensor equation of equation (3) has a real number solution, solve β to get the tensor correction $d_b = -J^{-1}[1/(2a\beta^2)]$. If equation (3) has no real solution, then we solve the least-

squares problem about β to get β . Then we get the tensor method correction $d_b, d_b = -J^{-1}(A\beta^2 - J^{-T}SW^{-1}q(\beta))/2$.

6) Find the total correction amount, $d = d_a + d_b$.

4. Power flow calculation based on the direct tensor method in polar coordinates

Regardless of whether it is the basic algorithm or method 1, we need to perform interpolation calculation and corresponding $g(x^{(k)} + d)$ matrix calculation before calculating the tensor correction amount [8]. The increased amount of calculation is relatively large. If we can avoid this calculation, we can expect to achieve high convergence of iterations while obtaining tensor corrections. Therefore, this paper proposes a method of quadratic differentiation of the power flow equation in polar coordinates to obtain the quadratic term. In this way, the tensor correction amount can be obtained. This paper calls this method the direct tensor method. It can be known from equation (1) that if the quadratic differential of A can be obtained, then equation (1) can be easily solved. Hence $d = d_a + d_b$. $(d^T Td)/2 \approx (d_a^T Td_a)/2$ where d_a is the Newton correction and d_b is the tensor correction. When $d_a >> d_b$, the equation (1) equals 0 can be transformed into $g(x^{(k)}) + J(x^{(k)})(d_a + d_b) + \frac{1}{2} d_a^T Td_a = 0$. $d_a + d_b = J^{-1}(-g(x^{(k)}) - \frac{1}{2} d_a^T Td_a$.

We can rewrite it as

$$\begin{cases} d_a = -J^{-1} g(x^{(k)}) \\ d_b = -J^{-1}(x^{(k)})(\frac{1}{2} d_a^T Td_a) \end{cases} \tag{10}$$

In the formula, $\Delta P_i^{(1)}, \Delta P_i^{(2)}, \Delta P_i^{(3)}, \Delta Q_i^{(1)}, \Delta Q_i^{(2)}, \Delta Q_i^{(3)}$ is the primary term, quadratic term, and cubic term + quartic term of active power increment and reactive power increment concerning voltage and angle increment, respectively [9]. And $\Delta P_i^{(1)}$ and $\Delta Q_i^{(1)}$ are Newton's increments. $(d^T Td)/2$ is $\Delta P_i^{(2)}$ and $\Delta Q_i^{(2)}$. It can be seen that the i term of $(d^T Td)/2$ is

$$\begin{aligned} \Delta P_i^{(2)} = {} & \Delta U_i \sum_{j \in i} \Delta U_j G_{ij} \cos\theta_{ij} - \sum_{j \in i} \frac{1}{2} U_i \sum_{j \in i} U_j G_{ij} \cos\theta_{ij} (\Delta\theta_{ij})^2 + \\ & \Delta U_i \sum_{j \in i} \Delta U_j B_{ij} \sin\theta_{ij} + U_i \sum_{j \in i} \Delta U_j B_{ij} \cos\theta_{ij} \Delta\theta_{ij} + \Delta U_i \sum_{j \in i} U_j B_{ij} \cos\theta_{ij} \Delta\theta_{ij} - \frac{1}{2} U_i \sum_{j \in i} U_j B_{ij} \sin\theta_{ij} (\Delta\theta_{ij}^3) \end{aligned} \tag{11}$$

When the system is lightly loaded, θ_{ij} is generally small, so $\cos\theta_{ij} = 1, \sin\theta_{ij} = 0$ can be approximately considered. Therefore, the calculation of the quadratic term $(d^T Td)/2$ of equation (1) can be obtained by equation (10) or (11). We can calculate the tensor correction amount by substituting the quadratic term $(d_a^T Td_a)/2$ calculated in equation (10) or (11) into equation (10). In this paper, the power flow algorithm based on the direct tensor method is called method 2. We assume that the initial point is $x^{(0)}$. When step k is iterated, the steps are as follows:

1) Calculate the function value $g(x^{(k)})$ of the current point.

2) If the $g(x^{(k)})$ element with the largest amplitude satisfies the convergence accuracy condition, the algorithm ends.

3) Calculate the Jacobian matrix $J(x^{(k)})$ of the current point.

4) Solve the Newton correction $d_a = -J^{-1} g(x^{(k)})$.

5) We use equation (11) to solve the quadratic term $(d_a^T T d_a)/2$ of the light-load system. We use equation (10) to solve the quadratic term $(d_a^T T d_a)/2$ of the overloaded system. Solution tensor method correction $d_b = -J^{-1}(x^{(k)})(d_a^T T d_a)/2$.
6) Calculate $\| d_b \| / \| d_b \|$, $d = d_a + d_b$ when $\| d_b \| / \| d_b \| < \lambda(\lambda << 1)$; otherwise $d = d_a$.
7) Assume $x^{(k+1)} = x^{(k)} + d, k = k+1$ and go to step 1) for loop iteration.

5. Calculation example and result analysis

We use C language programming. We use the above algorithm to calculate the power flow for multiple systems on a computer with a CPU frequency of 2.0GHz and memory of 1G. Table 1 shows the system scale data of each example [10]. Calculation example 1 and calculation example 2 are the New England and IEEE118 systems, respectively. Calculation examples 3-6 use 39 and 118 node system networks of building systems, respectively. Example 7 is the actual North China network system.

Table 1 Data from the test system

Examples	Total number of nodes	Total number of branches	Examples	Total number of nodes	Total number of branches
Example 1	39	46	Example 5	648	1147
Example 2	118	179	Example 6	810	1434
Example 3	226	293	Example 7	1923	2280
Example 4	324	573	Examples		

Example 2 compares the iteration times and calculation time of the four methods when the system is under normal load. N is the number of calculation iterations. In method 2, take 0.15. It can be seen from Table 2 that the calculation time of the basic tensor method and method 1 is the longest under the rated load [11]. The computation time of method 1 is longer than that of the basic tensor method. Mainly due to its use of optimization algorithms. The calculation time of method 2 is due to directly solving the quadratic term. Its calculation time is the same as Newton's method.

Table 2 Comparison of iteration number and computation time (rated load) for 4 methods

Test	Newton's method		Basic tensor method		Method 1		Method 2	
Example 1	3	0.061	2	0.11	2	0.122	2	0.47
Example 2	3	0.077	3	0.14	3	0.153	3	0.066
Example 3	4	0.147	3	0.151	3	0.179	3	0.104
Example 4	4	0.292	4	0.51	4	0.755	4	0.259
Example 5	4	0.527	4	1	4	1.12	4	0.539
Example 6	4	0.65	4	1.2	4	1.49	4	0.793
Example 7	4	0.949	4	1.421	4	1.97	4	0.955

Method 1 has the advantage of convergence. It can still converge when the basic tensor method and Newton's method do not converge. The computation time of method 1 is longer than that of the basic tensor method and Newton's method. Since the quadratic term can be easily obtained directly, the convergence and convergence speed of method 2 is the best among the four methods.

Tensor-based algorithms, including the basic tensor method, method 1, and method 2, all have better convergence than Newton's method. However, the calculation amount of the basic tensor method and method 1 is larger than that of the Newton method [12]. Compared with the basic tensor method, the calculation amount of method 1 is larger. The tensor method is similar to other nonlinear Newton methods that retain high valence, and they all require the Newton direction. The difference is that the tensor method corrects the Newton direction. And other Newton-like methods that retain higher-order nonlinear terms represented by the optimal multiplier method. Only the Newton step size is optimized, and the Newton direction is not changed. When the system load increases, the Jacobian matrix condition number of the system worsens, making Newton's method and optimal multiplier method not convergent.

6. Conclusion

In this paper, two new methods of power flow calculation based on the tensor method are proposed. Method 1 employs a least-squares optimization algorithm when the tensor equation has no real solution. We obtain its corresponding tensor correction value so that the power flow calculation based on the interpolation tensor method has better convergence. Method 2 is a power flow calculation method called the direct tensor method in polar coordinates. The convergence and calculation speed of power flow calculation are improved because the quadratic term is easily taken into account. The calculation results of several examples show that the two algorithms proposed in this paper are effective.

References

[1] Maulik, A., & Das, D Application of linearised load flow method for droop-controlled DCMGs. IET Generation, Transmission & Distribution.,2020; 14(6):1114-1126

[2] Hammouch, Z., Yavuz, M., & Özdemir, N. Numerical solutions and synchronization of a variable-order fractional chaotic system. Mathematical Modelling and Numerical Simulation with Applications.,2021; 1(1):11-23

[3] Abro, K. A., & Atangana, A. Numerical study and chaotic analysis of meminductor and memcapacitor through fractal–fractional differential operator. Arabian Journal for Science and Engineering., 2021;46(2):857-871

[4] Gómez-Aguilar, J. F., & Atangana, A. Time-fractional variable-order telegraph equation involving operators with Mittag-Leffler kernel. Journal of electromagnetic waves and applications.,2019; 33(2):165-177

[5] Owolabi, K. M., Atangana, A., & Akgul, A. Modelling and analysis of fractal-fractional partial differential equations: application to reaction-diffusion model. Alexandria Engineering Journal.,2020; 59(4):2477-2490

[6] Wei, W., Danman, W. U., Qiuwei, W. U., Shafie-Khah, M., & Catalao, J. P. Interdependence between transportation system and power distribution system: A comprehensive review on models and applications. Journal of Modern Power Systems and Clean Energy.,2019; 7(3):433-448

[7] Abro, K. A., Mirbhar, M. N., & Gomez-Aguilar, J. F.Functional application of Fourier sine transform in radiating gas flow with non-singular and non-local kernel. Journal of the Brazilian Society of Mechanical Sciences and Engineering.,2019; 41(10):1-8

[8] Muhammad, Y., Khan, R., Ullah, F., Aslam, M. S., & Raja, M. A. Z.Design of fractional swarming strategy for solution of optimal reactive power dispatch. Neural Computing and Applications.,2020; 32(14):10501-10518

[9] dos Santos, M. A.Fractional Prabhakar derivative in diffusion equation with non-static stochastic resetting. Physics.,2019; 1(1):40-58

[10] de Assis, R., Pazim, R., Malavazi, M., Petry, P., de Assis, L. & Venturino, E. A Mathematical Model to describe the herd behaviour considering group defense. Applied Mathematics and Nonlinear Sciences.,2020; 5(1): 11-24

[11] Chandoul, A.Proof without words: Periodic continued fractions. Applied Mathematics and Nonlinear Sciences.,2018; 4(1): 57-60

[12] Roohi, R., Heydari, M. H., Bavi, O., & Emdad, H. Chebyshev polynomials for generalized Couette flow of fractional Jeffrey nanofluid subjected to several thermochemical effects. Engineering with Computers.,2021; 37(1):579-595

Funder：Guizhou Power Grid Co., Ltd. technology projects for No. GZKJXM20190715.