Anderson-Acceleration-Based Power Flow Method for Integrated Transmission and Distribution Networks

Yifeng Chen, Kunjie Tang
College of Electrical Engineering
Zhejiang University
Hangzhou, China
cyf97@zju.edu.cn,
tangkunjie1994@zju.edu.cn

Hangyin Mao
State Grid Zhejiang Electric Power Corporation
Hangzhou, China
247477716@qq.com

*Shufeng Dong
College of Electrical Engineering
Zhejiang University
Hangzhou, China
dongshufeng@zju.edu.cn

2021 IEEE Power & Energy Society General Meeting (PESGM) | 978-1-6654-0507-2/21/$31.00 ©2021 IEEE | DOI: 10.1109/PESGM46819.2021.9638035

***Abstract*—Power flow analysis of integrated transmission and distribution networks (I-T&D) is the backbone of coordinated analysis of I-T&D. The traditional method for I-T&D power flow (ITDPF) is the master-slave-splitting-based method (MSSM), which divides an I-T&D into TN part, DN part and boundary part and solves sub-problems with alternating iterations to obtain global solution. This paper abstracts the MSSM as a fixed-point problem. Then, considering the limitations of the MSSM, an Anderson-acceleration-based method (AAM) is proposed to accelerate the ITDPF. This method modifies the iterative variables by solving a least-squares problem in each iteration. Numerical experiment demonstrates the accuracy of the proposed method. Also, better convergence and higher efficiency of the AAM is verified.**

***Index Terms*—integrated transmission and distribution networks, master-slave-splitting-based method, Anderson-acceleration-based method, convergence, least-squares problem.**

I. Introduction

With the continuous expansion of power grids and the increasingly complex operation mode, the traditional independent management of transmission networks (TNs) and distribution networks (DNs) has gradually been unable to meet the needs of the future dispatching and operation of power systems. Therefore, the study on the coordinated analysis for I-T&D has become increasingly prominent.

Theoretically, TNs and DNs should be combined as a global network to manage and analyze without any simplification or equivalence, which is called global model for I-T&D [1]. Without the mismatching between TNs and DNs, a Newton-Raphson method (NRM) based on global model can assure the accuracy of ITDPF. Hence in experiment part, we apply NRM as benchmarks. However, global model cannot be truly applied to real-world application due to privacy issues, numerical stability problems as well as communication burdens [1, 2].

At present, the most commonly used method for ITDPF calculation is the MSSM [1]. In the MSSM, DNs are equivalent as loads when calculating the TN power flow while TNs are equivalent as fixed voltage sources when calculating the DN power flow. Then, TN power flow and DN power flow are calculated alternately until the convergence condition is satisfied. However, the convergence of this method cannot satisfy the rapidly developing DN situation [4]. Also, the convergence rate of the MSSM still needs to be enhanced when it is applied into the future operations of power systems [5, 6].

To deal with the challenge discussed above, the Anderson-acceleration method (AAM) is introduced [7-9] to improve the ITDPF calculation. The main contribution of this paper is that it

i) concludes the traditional MSSM to a fixed-point problem and analyzes its weakness in convergence;

ii) presents a novel AAM for ITDPF calculation and introduces its general theory and specific steps.

Numerical experiments demonstrate the accuracy, convergence and efficiency of the proposed method compared with the MSSM. Besides, the tuning of the key parameter in the AAM is discussed.

The remaining parts of this paper are as follows. Section II presents the MSSM and analyzes its weakness. Section III introduces the theory and implementation of the AAM. Numerical experiments are presented in Section IV. Finally, the conclusions are drawn in Section V.

II. Master-Slave-Splitting-based Method

A. Model of the MSSM

In real-world operations, an I-T&D usually consists of one TN and several DNs. As shown in Figure 1, the master-slave-splitting model divides the global system into three parts, which are composed of TN part, boundary part and DN part [1]. The nodes in the boundary part represent the boundary bus between TNs and DNs.

B. Outline of the MSSM

MSSM is one of the most commonly used methods to calculate ITDPF. This method splits the global power flow calculation problem into a TN sub-problem and a DN sub-problem.

In the DN sub-problem, the boundary nodes are treated as fixed voltage sources, thus the DN sub-problem can be expressed as

$$\boldsymbol{g}_D(\boldsymbol{x}_B) = \boldsymbol{x}_D \tag{1}$$

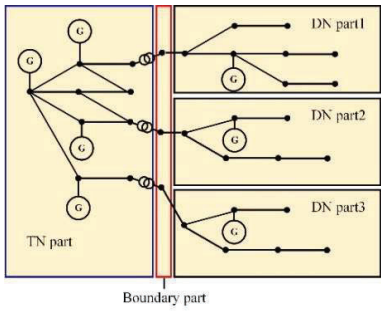


Figure 1. master-slave-splitting Model

where the subscripts B, D represent the boundary part and DN part, respectively. $\boldsymbol{x}_B$, $\boldsymbol{x}_D$ represent the voltage magnitude and angle in the boundary part and DN part, respectively. Thus, $\boldsymbol{g}_D$ is a mapping from the voltage of root nodes in the DN to the voltage of all nodes in the DN, which is equivalent to the distribution power flow calculation. With the obtained $\boldsymbol{x}_B$, $\boldsymbol{x}_D$, the active and reactive power in the boundary part can be computed as

$$\left(\boldsymbol{P}_B, \boldsymbol{Q}_B\right) = \boldsymbol{g}_B\left(\boldsymbol{x}_B, \boldsymbol{x}_D\right) \tag{2}$$

In the TN sub-problem, ($\boldsymbol{P}_B$, $\boldsymbol{Q}_B$) are the boundary power injections. Thus, the TN sub-problem can be expressed as

$$\boldsymbol{g}_T\left(\boldsymbol{P}_B, \boldsymbol{Q}_B\right) = \left(\boldsymbol{x}_T, \boldsymbol{x}_B\right) \tag{3}$$

where the subscript T represents the TN part. $\boldsymbol{x}_T$ represents the voltage magnitude and angle in the TN part. $\boldsymbol{g}_T$ is a mapping from the boundary power injections to the voltage of the nodes in the TN and boundary part.

Based on the formulation above, the traditional MSSM is carried out as following steps.

Step 1: Set the initial value of the boundary state variable $\boldsymbol{x}_B^{(0)}$, iteration number k=0 and convergence tolerance ε.

Step 2: Based on (1), input $\boldsymbol{x}_B^{(k)}$ and obtain $\boldsymbol{x}_D^{(k+1)}$. With $\boldsymbol{x}_B^{(k)}$ and $\boldsymbol{x}_D^{(k+1)}$, $\boldsymbol{P}_B^{(k+1)}$ and $\boldsymbol{Q}_B^{(k+1)}$ can be obtained by (2).

Step *3*: Solve (3) with $\boldsymbol{P}_B^{(k+1)}$ and $\boldsymbol{Q}_B^{(k+1)}$ to obtain $\boldsymbol{x}_B^{(k+1)}$ and $\boldsymbol{x}_T^{(k+1)}$.

Step 4: if $\| \boldsymbol{x}_B^{(k+1)} - \boldsymbol{x}_B^{(k)} \| < \varepsilon$, the algorithm converges and stops. Otherwise, k=k+1, go to **Step 2**.

C. Weakness of the MSSM

As the steps shown above, we can substitute (1) into (2)

$$\left(\boldsymbol{P}_B^{(k+1)}, \boldsymbol{Q}_B^{(k+1)}\right) = \boldsymbol{g}_B\left(\boldsymbol{x}_B^{(k)}, \boldsymbol{g}_D\left(\boldsymbol{x}_B^{(k)}\right)\right) \tag{4}$$

then substitute (4) into (3)

$$\boldsymbol{g}_T\left(\boldsymbol{g}_B\left(\boldsymbol{x}_B^{(k)}, \boldsymbol{g}_D\left(\boldsymbol{x}_B^{(k)}\right)\right)\right) = \left(\boldsymbol{x}_T^{(k+1)}, \boldsymbol{x}_B^{(k+1)}\right) \tag{5}$$

neglecting x_T, (3) can be simplified as

$$\boldsymbol{x}_B^{(k+1)} = \boldsymbol{f}\left(\boldsymbol{x}_B^{(k)}\right) \tag{6}$$

where f represents

$$\boldsymbol{f}(\bullet) = \boldsymbol{g}_T\left(\boldsymbol{g}_B\left(\bullet, \boldsymbol{g}_D(\bullet)\right)\right) \tag{7}$$

This indicates that the iterative procedure of the MSSM can be considered as a fixed-point problem. Therefore, the convergence of the MSSM is similar to the convergence of formulation (6). The convergence condition for formulation (6) is [3]

$$\rho\left(\frac{\partial \boldsymbol{f}\left(\boldsymbol{x}_B\right)}{\partial \boldsymbol{x}_B}\right) < 1 \tag{8}$$

where ρ is the spectral radius of $\boldsymbol{f}$. In addition, the smaller the value of $\rho(\bullet)$ is, the faster the convergence rate is.

According to (7), (8) can be expanded as

$$\rho\left(\frac{\partial \boldsymbol{g}_T(\boldsymbol{t})}{\partial \boldsymbol{t}} \cdot \frac{\partial \boldsymbol{g}_B(\boldsymbol{b})}{\partial \boldsymbol{b}} \cdot \frac{\partial \boldsymbol{g}_D\left(\boldsymbol{x}_B\right)}{\partial \boldsymbol{x}_B}\right) < 1 \tag{9}$$

where $\boldsymbol{t}$ represents $\boldsymbol{g}_B\left(\boldsymbol{x}_B^{(k)}, \boldsymbol{g}_D\left(\boldsymbol{x}_B^{(k)}\right)\right)$, $\boldsymbol{b}$ represents $\left(\boldsymbol{x}_B^{(k)}, \boldsymbol{g}_D\left(\boldsymbol{x}_B^{(k)}\right)\right)$, i.e., $\boldsymbol{b}$ is the formulation of $\boldsymbol{g}_B$.

In formulation (9), three parts are multiplied, which influence whether the MSSM converges or not, corresponding to the three parts of the master-slave-splitting model. In real-world power systems, TN is relatively stable and not sensitive to the fluctuations of $\boldsymbol{x}_B$, so the first two items of (9) have small values that usually will not degrade the convergence of the MSSM.

While for DNs, since the activeness of DNs has been significantly enhanced [3], such as more DGs (especially PV-typed DGs) are accessed into DNs, when $\boldsymbol{x}_B$ fluctuates, the third item of (9) may be very large and the convergence condition cannot hold. Numerical experiments will further demonstrate this point.

III. Anderson-Acceleration-Based Method

A. General Theory of the AAM

As presented in section II-C, the MSSM is abstracted as a fixed-point problem. Hence, some acceleration methods for the fixed-point problem can be taken into account to deal with the weakness of the MSSM.

AAM is a variant of the conventional fixed-point iteration. MSSM only uses the value of boundary state variables in the current iteration to calculate those in the next iteration, while AAM memorizes the values of boundary state variables in several previous iterations to calculate. This modification usually alleviates the fluctuations of boundary values and improves the convergence. Numerical experiments will demonstrate this point.

The basic theory of the AAM is as follows.

First, the general form of a fixed-point problem is

$$\boldsymbol{x} = \boldsymbol{f}\left(\boldsymbol{x}\right) \tag{10}$$

Define $\boldsymbol{g}: \boldsymbol{D} \subset \mathbf{R}^n \to \mathbf{R}^n$,

$$\boldsymbol{g}\left(\boldsymbol{x}\right) = \boldsymbol{x} - \boldsymbol{f}\left(\boldsymbol{x}\right) \tag{11}$$

Thus, for each iteration $k \geq 0$, the following least squares problem with a normalization constraint will be solved:

$$\min \left\| \sum_{j=0}^{m_k} \alpha_j \boldsymbol{g}(\boldsymbol{x}_{k-m_k+j}) \right\|_2 \quad s.t. \sum_{j=0}^{m_k} \alpha_j = 1 \tag{12}$$

where a simple choice of m_k is as (13), given a constant m

$$m_k = \min\{m, k\} \tag{13}$$

and a weight vector $\boldsymbol{\alpha}^k = (\alpha_0^k, \cdots, \alpha_{m_k}^k)$ will be achieved.

The intuition is to minimize the norm of the weighted residuals of the previous m_k+1 iterates. In particular, if $\boldsymbol{g}$ is affine, it is easy to see that $\boldsymbol{\alpha}$ minimizes the residual norm $\|\boldsymbol{g}(\boldsymbol{x}^{k+1/2})\|_2$ among all $\boldsymbol{x}^{k+1/2}$ that can be represented as

$$\boldsymbol{x}^{k+1/2} = \sum_{j=0}^{m_k} \alpha_j \boldsymbol{x}^{k-m_k+j} \tag{14}$$

where $\boldsymbol{x}^{k+1/2}$ represents the intermediate variable between the *k-th* iteration and the (*k*+1)-*th* iteration, which is given by the weighted sum of iterative values in previous m_k+1 iterations.

Hence, the next iteration can be computed

$$\boldsymbol{x}_{k+1} = \boldsymbol{f}\left(\boldsymbol{x}^{k+1/2}\right) = \sum_{j=0}^{m_k} \alpha_j^k \boldsymbol{f}\left(\boldsymbol{x}_{k-m_k+j}\right) \tag{15}$$

B. Practical Implementation

As shown in Subsection A, one constrained least-squares problem needs to be solved in each iteration. A simple and direct method is to solve this problem with mature solvers for solving convex optimization problems, such as Gurobi and Cplex. However, it is time-consuming. Thus, a more efficient method is proposed.

Noting that the minimization problem (12) can be efficiently solved as an unconstrained least squares problem by a simple variable elimination, i.e., (12) can be reformulated as

$$\min \left\| \boldsymbol{g}_k - \boldsymbol{Y}_k \boldsymbol{\gamma} \right\|_2 \tag{16}$$

where $\boldsymbol{\gamma} = (\gamma_0, \cdots, \gamma_{m_k-1})$, $\boldsymbol{g}_k = \boldsymbol{g}(\boldsymbol{x}_k)$, $\boldsymbol{Y}_k = \left[\boldsymbol{y}_{k-m_k} \cdots \boldsymbol{y}_{k-1}\right]$ with $\boldsymbol{y}_i = \boldsymbol{g}_{i+1} - \boldsymbol{g}_i$ for each *i*. Thus,

$$\begin{cases} \boldsymbol{\alpha}_0 = \boldsymbol{\gamma}_0, i = 0 \\ \boldsymbol{\alpha}_i = \boldsymbol{\gamma}_{i+1} - \boldsymbol{\gamma}_i, 1 \le i \le m_k - 1 \\ \boldsymbol{\alpha}_{m_k} = 1 - \boldsymbol{\gamma}_{m_k-1}, i = m_k - 1 \end{cases} \tag{17}$$

A large number of numerical experiments demonstrate that $\boldsymbol{Y}_k$ is always full column rank in ITDPF problems. Thus, the solution $\boldsymbol{\gamma}_k$ to (16) is given by

$$\boldsymbol{\gamma}_k = (\boldsymbol{Y}_k^{\mathrm{T}} \boldsymbol{Y}_k)^{-1} \boldsymbol{Y}_k^{\mathrm{T}} \boldsymbol{g}_k \tag{18}$$

Thus,

$$\begin{aligned} \boldsymbol{x}_{k+1} &= \boldsymbol{f}(\boldsymbol{x}_k) - \sum_{i=0}^{m_k-1} \gamma_i^k [\boldsymbol{f}(\boldsymbol{x}_{k-m_k+i+1}) - \boldsymbol{f}(\boldsymbol{x}_{k-m_k+i})] \\ &= \boldsymbol{f}(\boldsymbol{x}_k) - (\boldsymbol{S}_k - \boldsymbol{Y}_k)(\boldsymbol{Y}_k^{\mathrm{T}} \boldsymbol{Y}_k)^{-1} \boldsymbol{Y}_k^{\mathrm{T}} \boldsymbol{g}_k \end{aligned} \tag{19}$$

where $\boldsymbol{S}_k = \left[s_{k-m_k}, \ldots, s_{k-1}\right]$, $s_i = x_{i+1} - x_i$.

This is the improved iterative form of fixed-point problem.

Thus, based on (6) and (19), the detailed steps of the proposed AAM are as follows.

Step 1: Initialize $\boldsymbol{x}_B^{(0)}$, ε, m, $\boldsymbol{S}_B^{(0)}$ and $\boldsymbol{Y}_B^{(0)}$, where ε is the convergence tolerance; $\boldsymbol{S}_B^{(0)}$ and $\boldsymbol{Y}_B^{(0)}$ are empty matrices. Let iteration count $k = 0$.

Step 2: Calculate $\boldsymbol{x}_B^{(k+1)}$ as

$$\boldsymbol{x}_B^{(k+1)} = \begin{cases} f(\boldsymbol{x}_B^{(k)}), k = 0 \\ f(\boldsymbol{x}_B^{(k)}) - (\boldsymbol{S}_B^{(k)} - \boldsymbol{Y}_B^{(k)})\left(\left(\boldsymbol{Y}_B^{(k)}\right)^{\mathrm{T}} \boldsymbol{Y}_B^{(k)}\right)^{-1} \left(\boldsymbol{Y}_B^{(k)}\right)^{\mathrm{T}} \boldsymbol{g}\boldsymbol{Y}_B^{(k)}, k \ge 1 \end{cases} \tag{20}$$

Step 3: If $\left\| \boldsymbol{x}_B^{(k+1)} - \boldsymbol{x}_B^{(k)} \right\| < \varepsilon$, the algorithm converges and stops. Otherwise, go to **Step 4**.

Step 4: If $k < m$, calculate $\boldsymbol{S}_B^{(k+1)}$ and $\boldsymbol{Y}_B^{(k+1)}$ as

$$\boldsymbol{S}_B^{(k+1)} \leftarrow \left[\boldsymbol{S}_B^{(k)} \quad \boldsymbol{x}_B^{(k+1)} - \boldsymbol{x}_B^{(k)}\right] \tag{21}$$

$$\boldsymbol{Y}_B^{(k+1)} \leftarrow \left[\boldsymbol{Y}_B^{(k)} \quad \boldsymbol{x}_B^{(k+1)} - f(\boldsymbol{x}_B^{(k+1)}) - g(\boldsymbol{x}_B^{(k)})\right] \tag{22}$$

Otherwise,

$$\boldsymbol{S}_B^{(k+1)} \leftarrow \left[\boldsymbol{S}_B^{(k)} \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{I} \end{bmatrix} \quad \boldsymbol{x}_B^{(k+1)} - \boldsymbol{x}_B^{(k)}\right] \tag{23}$$

$$\boldsymbol{Y}_B^{(k+1)} \leftarrow \left[\boldsymbol{Y}_B^{(k)} \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{I} \end{bmatrix} \quad \boldsymbol{x}_B^{(k+1)} - f(\boldsymbol{x}_B^{(k+1)}) - g(\boldsymbol{x}_B^{(k)})\right] \tag{24}$$

where $\boldsymbol{I}$ represents an identity matrix, of which the dimension is m–1.

Step 5: Let $k \leftarrow k+1$ and go to **Step 2**.

The detailed process is shown in Figure 2. Figure 2 elaborates the data exchange between TNs and DNs, and the data updating during process.

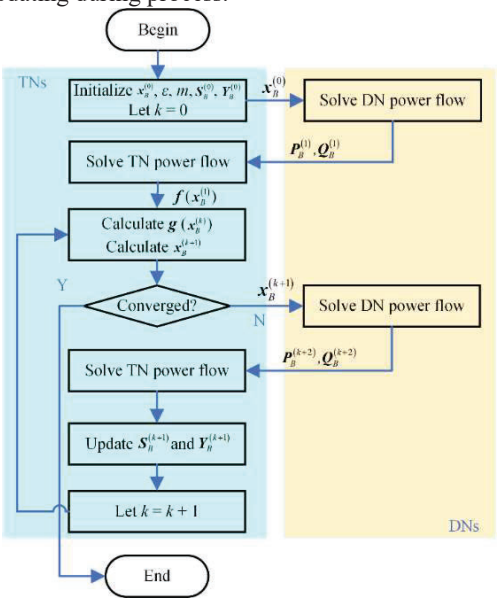


Figure 2. The calculation process for the AAM

In real-world operation, the three-phase imbalance of DNs need to be considered sometimes. Reference [3] presents a heterogeneous model, where the TN is modeled as single-phase while DNs are modeled as three-phase. This model can be applied to extend the proposed AAM into a three-phase form.

IV. NUMERICAL EXPERIMENT

The program for the test run on the Windows 10 of 64 bits. The CPU is Intel Core i7-7700, with 3.60GHz master frequency and 16GB memory. The programming language used is MATLAB R2016b. The convergence tolerance of power flow

methods is set to 1e-6 p.u. In Subsection A and B, the parameter *m* for the AAM is set to 3. Nine I-T&D cases (A1-A3, B1-B3, C1-C3) are constructed for numerical experiments, as shown in TABLE I. Case14, case57, and case118 are IEEE standard cases. Case69 is a distribution network with 69 nodes [10]. Compared to the case69, several PV-typed DGs exist in case 69A and case 69B. And case 69B possesses more DGs. They have a constant active power output 0.5MW. Each feeder is connected to the TN via a transformer with r = 0.002 p.u., x = 0.01 p.u., and ratio = 1.

TABLE I. TEST CASE INFORMATION

case	TN case	DN case	DNs are accessed into the Bus No.	DGs are accessed into the Node No.
A1	Case14	Case69	14	-
A2	Case14	Case69A	14	8,15,20
A3	Case14	Case69B	14	45,61
B1	Case57	Case69	8,9,12,18	-
B2	Case57	Case69A	8,9,12,18	8,15,20
B3	Case57	Case69B	8,9,12,18	45,61
C1	Case118	Case69	11,78,82,118	-
C2	Case118	Case69A	11,78,82,118	8,15,20
C3	Case118	Case69B	11,78,82,118	45,61

Accuracy of the proposed method is tested in Subsection A. Convergence and efficiency are discussed in Subsection B. And the tuning of the parameter *m* is discussed in Subsection C.

A. Accuracy

Taking the NRM based on a global model as benchmarks, TABLE II compares the power flow results of NRM, MSSM and AAM under different cases.

TABLE II. POWER FLOW RESULTS OF THE BOUNDARY PART

Case	Data	Bus No.	NRM	MSSM	AAM
A1	V_B/p.u.	14	1.0250	1.0250	1.0250
	θ_B/ °		-16.7862	-16.7862	-16.7862
	P_B/MW		4.1870	4.1870	4.1870
	Q_B/MVar		2.8673	2.8673	2.8673
B1	V_B/p.u.	8	1.0050	1.0050	1.0050
	θ_B/ °		-4.9888	-4.9888	-4.9888
	P_B/MW		2.7068	2.7068	2.7068
	Q_B/MVar		-1.0404	-1.0404	-1.0404
	V_B/p.u.	12	1.0148	1.0148	1.0148
	θ_B/ °		-10.8154	-10.8154	-10.8154
	P_B/MW		2.6768	2.6768	2.6768
	Q_B/MVar		1.1158	1.1158	1.1158

As the data shown in TABLE II, these three methods get absolutely the same power flow results. i.e., the AAM can accurately calculate the I&T-D power flow, which guarantees the validity of following analysis.

B. Convergence and Efficiency

The convergence and efficiency of the MSSM and the proposed AAM are compared in this subsection. The voltage magnitude set-points are all 1.0 p.u. and the voltage angle set-points are 0. TABLE III lists the iteration number and time cost under different methods and different cases.

Generally, the proposed AAM has better convergence and higher efficiency than the MSSM. With a large number of DGs accessed into DNs, the sensitivity of power injection to voltage in root nodes is significantly enhanced, which leading to the third item of formulation (9) increases and the MSSM diverges under case A3, B3. In comparison, the AAM converges with several iterations. Also, in case A2, B2, it is obvious that the iteration number is decreased by around 2/3. These results demonstrate that the AAM can effectively improve the convergence of ITDPF calculation. As for time, it can be seen that the AAM costs less time than the MSSM generally, which proves the higher efficiency of the AAM.

TABLE III. CONVERGENCE AND EFFICIENCY RESULT

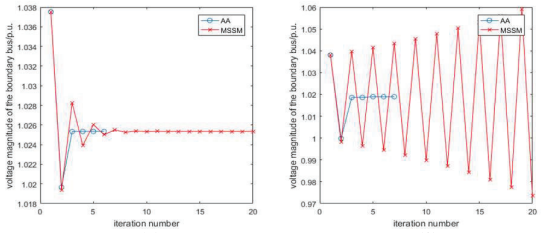
case	MSSM		AAM	
	Iteration Number	Time Cost/ms	Iteration Number	Time Cost/ms
A1	5	46.9	5	31.3
A2	21	103.7	7	25.7
A3	Diverge		8	31.3
B1	5	46.9	5	31.3
B2	21	262.2	8	78.1
B3	Diverge		10	109.2
C1	5	93.8	5	31.3
C2	9	99.7	7	72.7
C3	11	187.5	8	93.8

Further, different initial values of nodal voltage magnitudes are set to evaluate the effect in TABLE IV. Without losing generality, case A2 is selected with the initial value ranging from 1.0 to 1.5. With the initial value ranging from 1.0 to 1.3, the AAM takes less iterations than the MSSM. Then, with the initial value 1.4 and 1.5, the MSSM diverges while the AAM converges. In conclusion, compared to the MSSM, the AAM can converge faster and tolerate a wider range of initial values.

TABLE IV. ITERATION NUMBER UNDER DIFFERENT INITIAL VALUE

Initial Value	Iteration Number of MSSM	Iteration Number of AAM
1.0	21	7
1.1	22	8
1.2	24	9
1.3	26	15
1.4	Diverge	41
1.5	Diverge	33

Besides, the variation of the voltage magnitude of the boundary bus under case A2 and case A3 is shown in Figure 3(*a*) and Figure 3(*b*), respectively. In case A2, the AAM converges faster and in case A3, the MSSM diverges. Meanwhile, the variational curve under the MSSM shows a significant fluctuation at first, and then converges to a stable value or diverges. However, the AAM quickly approaches to the stable value, which further demonstrates the superiority of the AAM.



(*a*) (*b*)

Figure 3. The variation of the voltage magnitude of the boundary bus

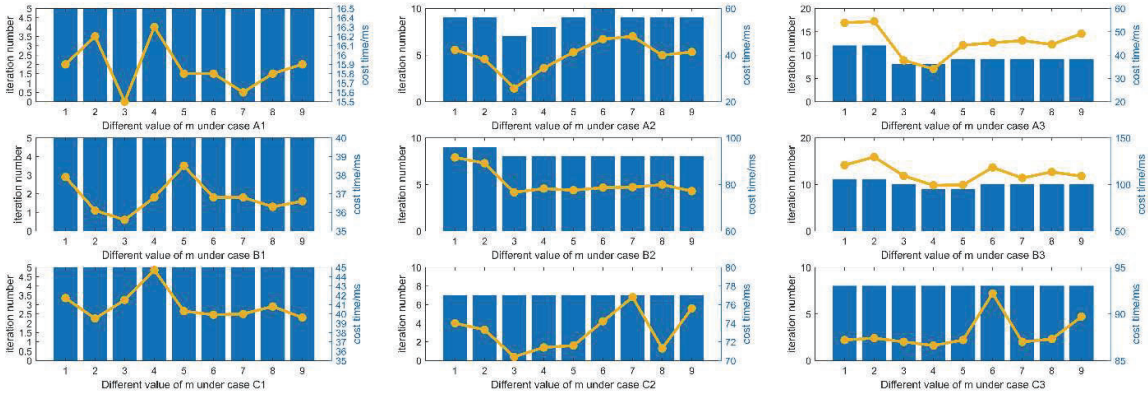


Figure 4. Iteration number and time cost under different value of m

C. Tuning of m

In the AAM, m is an essential parameter that memorizes the past iteration results. In (13), if $m < k$, $m_k = m$, then the update of iterative variables is just based on the previous m iterations.

Otherwise, $m_k = k$, m doesn't work and the acceleration effect is not significant. Hence, the iteration number and time cost with m range from 1 to 9 under each case. The result is shown in Figure 4.

To select the best m for each case, the iteration number and the time cost are compared. For case A1, B1, C1, C2, C3, their iteration numbers are relatively small and stable with the variation of m. For case A2, A3, B2, B3, with the increment of m, the iteration number and time cost first decrease and then increase. In conclusion, different cases require different settings of m to achieve the best acceleration effect. Hence, the selection of m is application-dependent. For the cases studied, $m = 3$ could be taken as an empirical value with stable performance.

V. CONCLUSIONS

Generally, this paper proposes a novel AAM to deal with the weakness of the conventional MSSM. Numerical experiments on several cases demonstrate that

i)The proposed AAM assures the accuracy of ITDPF, which can achieve the same accuracy with a centralized NRM and the conventional MSSM;

ii) The proposed AAM can achieve better convergence and efficiency compared with the conventional MSSM. More specifically, for some cases where the MSSM converges, the AAM can reduce 2/3 iterations and converge with a faster rate. While for some cases where the MSSM diverges, the AAM can still converge within finite iterations;

iii) The selection of key parameter m is application-dependent.

Future work will focus on the variation and wider application of the proposed AAM.

ACKNOWLEDGMENT

This work was supported by the Science and Technology Program of State Grid Corporation of China (52110418000M).

REFERENCES

[1] Z. Li, H. Sun, Q. Guo, J. Wang and G. Liu, "Generalized Master–Slave-Splitting Method and Application to Transmission–Distribution Coordinated Energy Management," *IEEE Trans. Power Syst.*, vol. 34, no. 6, pp. 5169-5183, Nov. 2019.

[2] Z. Li, Q. Guo, H. Sun and J. Wang, "Coordinated Economic Dispatch of Coupled Transmission and Distribution Systems Using Heterogeneous Decomposition," *IEEE Trans. Power Syst.*, vol. 31, no. 6, pp. 4817-4830, Nov. 2016.

[3] H. Sun, Q. Guo, B. Zhang, Y. Guo, Z. Li and J. Wang, "Master–Slave-Splitting Based Distributed Global Power Flow Method for Integrated Transmission and Distribution Analysis," *IEEE Trans. Smart Grid*, vol. 6, no. 3, pp. 1484-1492, May 2015.

[4] K. Tang, S. Dong and Y. Song, "Successive-Intersection-Approximation-Based Power Flow Method for Integrated Transmission and Distribution Networks," *IEEE Trans. Smart Grid*, vol. 35, no. 6, pp. 4836-4846, Nov. 2020

[5] Q. Huang and V. Vittal, "Integrated Transmission and Distribution System Power Flow and Dynamic Simulation Using Mixed Three-Sequence/Three-Phase Modeling," *IEEE Trans. Power Syst.*, vol. 32, no. 5, pp. 3704-3714, Sept. 2017.

[6] K. Li, X. Han, W. Li and R. Ahmed, "Unified power flow algorithm of transmission and distribution network," *2017 IEEE 6th International Conference on Renewable Energy Research and Applications (ICRERA)*, San Diego, CA, 2017, pp. 257-261.

[7] Walker, Homer F., and P. Ni. "Anderson Acceleration for Fixed-Point Iterations." *SIAM Journal on Numerical Analysis*, vol. 49, no. 4, pp. 1715-1735, Aug. 2011.

[8] Fang, Haw Ren, and Y. Saad. "Two classes of multisecant methods for nonlinear acceleration," *Numerical Linear Algebra with Applications*, vol. 16, no. 3, pp. 197-221, Jun. 2007.

[9] Anderson, D.G., "Iterative Procedures for Nonlinear Integral Equations," *Journal of the ACM*, vol. 12, no. 4, pp. 547-560, Oct. 1965.

[10] D. Das, "Optimal placement of capacitors in radial distribution system using a Fuzzy-GA method," *Int. J. Electr. Power & Energy Syst.*, vol. 30, no. 6, pp. 361-367, Jul. 2008.