# Weather-Dependent AC Power Flow Algorithms

Arif Ahmed*, Tobias Massier†, Fiona Stevens McFadden‡, and Ramesh Rayudu§
*†TUMCREATE Ltd., Singapore 138602, Singapore
‡§Victoria University of Wellington, Wellington 6140, New Zealand
Email: *arif.ahmed@tum-create.edu.sg, †tobias.massier@tum-create.edu.sg,
‡fiona.stevensmcfadden@vuw.ac.nz, and §ramesh.rayudu@vuw.ac.nz

***Abstract*—The weather-dependent power flow (WDPF) algorithm performs more accurate power flow analysis (PFA) due to the utilisation of the heat balance model of conductors. It is explicitly parameterised in terms of typically available measured weather parameters (ambient temperature, solar irradiance, wind speed, and wind angle) and performs more accurate (PFA) due to the utilisation of the heat balance model of conductors. It is presented in rectangular form in the extant literature. The WDPF algorithm accurately estimates the branch resistances, the system states (current and voltages), the power losses, the branch flows, and the branch loadings via PFA. In this manuscript, we propose and investigate a group of weather-dependent AC power flow algorithms. Namely, the partially decoupled WDPF, the fast decoupled WDPF, and the sequential WDPF algorithm. In addition, we also present the derivation of the WDPF algorithm in polar form. An analysis of the convergence characteristic and the computational complexity of the proposed algorithms is presented via extensive simulations.**

***Index Terms*—power flow analysis, power system modelling, weather-dependent power flow**

## I. Notation

| | |
|---|---|
| $q_c$ | convective heat loss rate (in $\mathrm{W/m}$). |
| $q_r$ | radiative heat loss rate (in $\mathrm{W/m}$). |
| $q_s$ | solar heat gain rate (in $\mathrm{W/m}$). |
| $q_j$ | heat gain rate from Joule losses (in $\mathrm{W/m}$). |
| $H$ | heat balance equation of a conductor (in pu). |
| $T_a$ | ambient temperature (in $^\circ$C). |
| $V_w$ | wind speed (in $\mathrm{m/s}$). |
| $\phi$ | wind angle (in $^\circ$). |
| $Q_s$ | global solar irradiance (in $\mathrm{W/m^2}$). |
| $T_c$ | branch conductor temperature (in $^\circ$C). |
| $P_k$ | active power injection (in pu) at bus $k$. |
| $Q_k$ | reactive power injection (in pu)at bus $k$. |
| $V_k\angle\delta_k$ | complex voltage (in pu) at bus $k$. |
| $v$ | iteration number. |
| $\boldsymbol{\Delta P}$ | active power mismatch (in pu) vector. |
| $\boldsymbol{\Delta Q}$ | reactive power mismatch (in pu) vector. |
| $\boldsymbol{\Delta H}$ | heat balance mismatch (in pu) vector. |
| $y_{ij}$ | admittance (in pu) of conductor between bus $i$ and bus $j$. |
| $\boldsymbol{Y(T_c)}$ | weather-dependent network admittance matrix. |
| $T_a$ | ambient temperature (in $^\circ$C). |
| $K_{\text{angle}}$ | wind direction factor |
| $N_{\text{Re}}$ | Reynold's number. |
| $D$ | conductor diameter (in $\mathrm{m}$). |
| $\rho_f$ | air density (in $\mathrm{kg/m^3}$). |
| $V_w$ | wind speed (in $\mathrm{m/s}$). |
| $\mu_f$ | dynamic viscosity of air $\left(\text{in } \frac{\mathrm{kg}}{\mathrm{m\cdot s}}\right)$. |
| $\phi$ | wind angle (in $^\circ$). |
| $k_f$ | thermal conductivity of air at the boundary layer temperature $\left(\text{in } \frac{\mathrm{W}}{\mathrm{m\cdot^\circ C}}\right)$. |
| $\varepsilon$ | emissivity constant. |
| $\alpha$ | solar absorptivity. |
| $P_{\text{loss}_{ij}}$ | power loss (in pu) in a conductor connecting bus $i$ and bus $j$. |
| $I$ | current (in pu). |
| $R(T_c)$ | conductor resistance per meter ($\Omega/\mathrm{m}$) at conductor temperature $T_c$. |
| $n$ | number of system buses. |
| $g_{ij}$ | conductance (in pu) of conductor between bus $i$ and bus $j$. |
| $b_{ij}$ | susceptance (in pu) of conductor between bus $i$ and bus $j$. |

Matrices and vectors will be represented in bold in this manuscript.

## II. Introduction

Power flow analysis (PFA) is a fundamental tool in the study of steady-state operation of power systems, and is a basis for many essential analyses like planning and design, contingency analysis, stability analysis, security analysis, etc. [1], [2]. Conventional PFA is based on the assumption of constant branch impedance. However, the branch impedance is a function of current flow, conductor characteristics, and weather conditions, making the branch impedance dynamic. Therefore, improvements in PFA are achievable by modelling the weather-dependent network characteristics utilising conductor thermal/heat-balance models in the power flow problem [3], [4].

The importance of incorporating weather information to improve power system studies has been established in the extant literature. In [5], power loss error up to 30% was shown in conventional power flow when conductor temperature and resistance were corrected considering weather conditions. A weather-based optimal power flow was presented in [6], which highlighted the potential for, reducing total generation cost, optimizing power generation according to thermal ratings, and maximizing useable capacity of overhead lines. More recently,

studies investigating the impact of weather on transient stability analysis have been undertaken [7]–[9]. It has been observed mainly that the critical fault clearing times vary depending on the weather conditions, which highlights the importance of considering weather in transient stability studies.

Various AC power flow algorithms that explicitly or implicitly consider weather information have been proposed and presented [3]–[5], [10]. These algorithms present the advantage of more accurate evaluation of power system states, power losses, and power flows in the network [3]–[5], [10]. Recently, a weather-dependent power flow algorithm (WDPF) [4] was presented that is fully coupled with the nonlinear and weather-dependent heat balance model. The WDPF algorithm, in addition to more accurate power flows, also outputs the temperature of the branch conductors. This is beneficial for conductor temperature profiling and evaluating short-term thermal line ratings leading to better network management [4], [11]. The WDPF algorithm models the comprehensive relationship between weather condition and power flow in line conductor by considering the nonlinear steady-state heat balance model, which is well documented in IEEE Std 738$^{\text{TM}}$-2012 [12] and CIGRE Working Group 22.12 [13].

The derivation of the WDPF algorithm presented in [4] was done in the rectangular form. Consequently, in this manuscript we contribute by presenting the derivation in the polar form. In addition, further contributions are made by presenting and investigating different variants of WDPF algorithms. The following distinct algorithms are proposed and investigated in this manuscript:

1) a fully coupled WDPF algorithm in which all the system states are solved and updated simultaneously;
2) a partially decoupled WDPF algorithm in which the power system states and the conductor temperatures are solved and updated separately;
3) a fast decoupled WDPF in which all the power system states and the conductor temperature state are solved and updated separately;
4) a sequential WDPF algorithm in which the power system states and the conductor temperature states are sequentially solved and updated.

The above algorithms are numerically investigated in this manuscript to assess their computational complexities. Simulations are undertaken on a number of power networks to investigate the accuracies and computational times of the proposed algorithms. Since the fully coupled WDPF algorithm inherently has a larger Jacobian matrix in comparison to conventional power flow, the computational effort to obtain the power flow solution is greater. Consequently, exploration of other WDPF algorithm formulations is essential, and therefore undertaken in this manuscript.

The rest of the paper is organized as follows: Section III introduces and gives an overview of the steady-state nonlinear heat balance model for overhead conductors. In Section IV, the proposed weather-dependent AC power flow algorithms are presented and their implementation is discussed. Numerical investigation of the WDPF algorithms are presented in Section V, which includes investigation of convergence characteristics and computational complexity. Finally, the manuscript is concluded in Section VI.

## III. Overview of Nonlinear Heat Balance Model

The steady-state nonlinear heat balance model of an overhead conductor defines its heat equilibrium by relating its temperature, resistance, current, and the weather condition surrounding it. It is based on the assumption that the mean wind speed, wind direction, ambient temperature, solar radiation, and current are fairly constant and, hence, the conductor temperature does not change significantly.

The steady-state nonlinear heat balance equation according to IEEE Std 738$^{\text{TM}}$-2012 [12] is as follows:

$$q_c + q_r = q_s + q_j \tag{1}$$

In Equation (1), $q_c$ is the heat loss rate due to convective cooling, $q_r$ is the heat loss rate due to radiative cooling, $q_s$ is the heat gain rate due to solar radiation, and $q_j$ is the heat gain rate due to Joule heating. The detailed equations for these can be referred to in the standards [12], [13], and Appendix A.

The nonlinear heat balance Equation (1) is essential to the derivation of the WDPF algorithms. Equation (1) can be solved to obtain the temperature of the line conductor ($T_c$) for any amount of power flowing through it under any given weather condition. This is then utilized to correct the resistance and then the impedance of the line conductors yielding a more accurate power flow.

A heat balance function ($H$) is formulated from Equation (1), which can be written as a function of ambient temperature ($T_a$), wind speed ($V_w$), wind incidence angle ($\phi$), solar irradiance ($Q_s$), and conductor temperature ($T_c$), as follows:

$$H(T_a, V_w, \phi, Q_s, T_c) = q_c + q_r - q_s - q_j = 0 \tag{2}$$

The heat balance function in Equation (2) is utilized in the next section to derive the proposed algorithms.

## IV. Weather-Dependent Power Flow Algorithms

This section presents the derivation of the proposed algorithms, starting with the fully coupled WDPF algorithm in polar form.

### A. Fully Coupled Weather-Dependent Power Flow

The fully coupled WDPF algorithm in polar form is derived considering the power injection equations and heat balance function as presented in Equations (3), (4), and (5). In Equation (5), the Joule heating term ($q_{j_{ij}}$) is substituted by the power loss expression of a conductor between bus $i$ and bus $j$. This enables derivation of the partial terms $\frac{\partial \boldsymbol{H}}{\partial \boldsymbol{\delta}}$ and $\frac{\partial \boldsymbol{H}}{\partial \boldsymbol{V}}$ for the weather-dependent branch conductors in the network.

$$P_k = \sum_{i=1}^{n} V_k V_i Y_{ki} \cos\left(\delta_k - \delta_i - \delta_{ki}\right) \tag{3}$$

$$Q_k = \sum_{i=1}^{n} V_k V_i Y_{ki} \sin\left(\delta_k - \delta_i - \delta_{ki}\right) \tag{4}$$

$$H_{ij}(T_{c_{ij}}) = q_{r_{ij}} + q_{c_{ij}} - q_{s_{ij}} - \left[(V_i^2 + V_j^2)y_{ij}\cos\delta_{ij} - 2V_iV_jy_{ij}\cos\left(\delta_i - \delta_j - \delta_{ij}\right)\right] = 0 \tag{5}$$

Application of Newton's method to Equations (3), (4), and (5) in order to solve for the solution states ($V$, $\delta$, $T_c$) yields:

$$\begin{bmatrix}\boldsymbol{\delta}\\ \boldsymbol{V}\\ \boldsymbol{T_c}\end{bmatrix}^{(v+1)} = \begin{bmatrix}\boldsymbol{\delta}\\ \boldsymbol{V}\\ \boldsymbol{T_c}\end{bmatrix}^{(v)} + \begin{bmatrix}\frac{\partial \boldsymbol{P}}{\partial \boldsymbol{\delta}} & \frac{\partial \boldsymbol{P}}{\partial \boldsymbol{V}} & \frac{\partial \boldsymbol{P}}{\partial \boldsymbol{T_c}}\\ \frac{\partial \boldsymbol{Q}}{\partial \boldsymbol{\delta}} & \frac{\partial \boldsymbol{Q}}{\partial \boldsymbol{V}} & \frac{\partial \boldsymbol{Q}}{\partial \boldsymbol{T_c}}\\ \frac{\partial \boldsymbol{H}}{\partial \boldsymbol{\delta}} & \frac{\partial \boldsymbol{H}}{\partial \boldsymbol{V}} & \frac{\partial \boldsymbol{H}}{\partial \boldsymbol{T_c}}\end{bmatrix}^{-1} \begin{bmatrix}\boldsymbol{\Delta P}\\ \boldsymbol{\Delta Q}\\ \boldsymbol{\Delta H}\end{bmatrix}^{(v)} \tag{6}$$

In Equation (6), $\boldsymbol{\Delta P}$, $\boldsymbol{\Delta Q}$, and $\boldsymbol{\Delta H}$ are the mismatch vectors while $\boldsymbol{V}$, $\boldsymbol{\delta}$, and $\boldsymbol{T_c}$ are the solution state vectors. The derivation of partial derivative terms in the Jacobian are presented in Appendix B. Algorithm 1 summarizes the fully coupled WDPF algorithm in polar form and is referred to as the FC-WDPF algorithm in this manuscript. The admittance matrix $\boldsymbol{Y}(\boldsymbol{T_c})$ here is weather-dependent as it is calculated based on the branch temperature estimate.

**Algorithm 1:** Weather-Dependent Power Flow

```
Read all input data (conductor, bus, load, weather);
Set error tolerance;
Initialize voltages and conductor temperatures;
while error > error tolerance do
    Form/update admittance matrix Y(T_c);
    Calculate power injection and heat balance
      mismatch (Equation (6));
    Calculate Jacobian matrix (Equation (6));
    Calculate state mismatch (Equation (6));
    Update states V, δ, T_c (Equation (6));
    Update network resistances based on T_c
      (Equation (A.7));
    Update network impedances;
end
Output results;
```

### B. Partially Decoupled Weather-Dependent Power Flow

In the partially decoupled weather-dependent power flow (PD-WDPF) algorithm, the power system states ($\boldsymbol{V}$, $\boldsymbol{\delta}$) and the branch conductor temperature states ($\boldsymbol{T_c}$) are solved and updated separately every iteration as presented in Equation (7) and (8). The inversion of the Jacobians in Equation (7) and (8) is expected to be faster in comparison to the fully coupled WDPF algorithm due to the smaller sizes of the matrices. However, the solution is expected to require a higher number of iterations. Algorithm 2 summarizes the PD-WDPF algorithm.

$$\begin{bmatrix}\boldsymbol{\delta}\\ \boldsymbol{V}\end{bmatrix}^{(v+1)} = \begin{bmatrix}\boldsymbol{\delta}\\ \boldsymbol{V}\end{bmatrix}^{(v)} + \begin{bmatrix}\frac{\partial \boldsymbol{P}}{\partial \boldsymbol{\delta}} & \frac{\partial \boldsymbol{P}}{\partial \boldsymbol{V}}\\ \frac{\partial \boldsymbol{Q}}{\partial \boldsymbol{\delta}} & \frac{\partial \boldsymbol{Q}}{\partial \boldsymbol{V}}\end{bmatrix}^{-1} \begin{bmatrix}\boldsymbol{\Delta P}\\ \boldsymbol{\Delta Q}\end{bmatrix}^{(v)} \tag{7}$$

$$\left[\boldsymbol{T_c}\right]^{(v+1)} = \left[\boldsymbol{T_c}\right]^{(v)} + \left[\frac{\partial \boldsymbol{H}}{\partial \boldsymbol{T_c}}\right]^{-1} \left[\boldsymbol{\Delta H}\right]^{(v)} \tag{8}$$

**Algorithm 2:** Partially Decoupled Weather-Dependent Power Flow

```
Read all input data (conductor, bus, load, weather);
Set error tolerance;
Initialize voltages and conductor temperatures;
while error > error tolerance do
    Form/update admittance matrix Y(T_c);
    Calculate power injection mismatch (Equation (7));
    Calculate Jacobian matrix (Equation (7));
    Calculate heat balance mismatch (Equation (8));
    Calculate Jacobian matrix (Equation (8));
    Calculate state mismatch (Equation (7));
    Calculate state mismatch (Equation (8));
    Update states V, δ (Equation (7));
    Update states T_c (Equation (8));
    Update network resistances based on T_c
      (Equation (A.7));
    Update network impedances;
end
Output results;
```

### C. Fast Decoupled Weather-Dependent Power Flow

The fast decoupled weather-dependent power flow algorithm (FD-WDPF) is executed by calculating the voltage magnitudes ($\boldsymbol{V}$), voltage angles ($\boldsymbol{\delta}$), and the branch conductor temperatures ($\boldsymbol{T_c}$) separately as presented in Equations (9), (10), and (11). The Jabocians in the FD-WDPF algorithm are much smaller, indicating faster computation times but a higher number of iterations. Algorithm 3 summarizes the FD-WDPF algorithm.

$$\left[\boldsymbol{\delta}\right]^{(v+1)} = \left[\boldsymbol{\delta}\right]^{(v)} + \left[\frac{\partial \boldsymbol{P}}{\partial \boldsymbol{\delta}}\right]^{-1} \left[\boldsymbol{\Delta P}\right]^{(v)} \tag{9}$$

$$\left[\boldsymbol{V}\right]^{(v+1)} = \left[\boldsymbol{V}\right]^{(v)} + \left[\frac{\partial \boldsymbol{Q}}{\partial \boldsymbol{V}}\right]^{-1} \left[\boldsymbol{\Delta Q}\right]^{(v)} \tag{10}$$

$$\left[\boldsymbol{T_c}\right]^{(v+1)} = \left[\boldsymbol{T_c}\right]^{(v)} + \left[\frac{\partial \boldsymbol{H}}{\partial \boldsymbol{T_c}}\right]^{-1} \left[\boldsymbol{\Delta H}\right]^{(v)} \tag{11}$$

### D. Sequential Weather-Dependent Power Flow

In the sequential weather-dependent power flow (S-WDPF) algorithm, the power flow and the heat balance are solved separately and sequentially for power flow analysis. The algorithm first forms the weather-dependent admittance matrix $\boldsymbol{Y}(\boldsymbol{T_c})$ based on the initial conductor temperatures. The conventional power flow is then solved followed by the solution of the nonlinear heat balance to obtain a correct estimate of the conductor temperature state. This is then utilized to update the conductor resistance and form the updated weather-dependent admittance matrix $\boldsymbol{Y}(\boldsymbol{T_c})$. This process is repeated until all the mismatches are within the error tolerance. The main advantage of the sequential algorithm is the usability of any existing conventional power flow approach that can be followed by

**Algorithm 3:** Fast Decoupled Weather-Dependent Power Flow

Read all input data (conductor, bus, load, weather);
Set error tolerance;
Initialize voltages and conductor temperatures;
**while** *error* $>$ *error tolerance* **do**
  Form/update admittance matrix $\boldsymbol{Y}(\boldsymbol{T_c})$;
  Calculate active power injection mismatch (Equation (9));
  Calculate reactive power injection mismatch (Equation (10));
  Calculate heat balance mismatch (Equation (11));
  Calculate Jacobian matrix (Equation (9));
  Calculate Jacobian matrix (Equation (10));
  Calculate Jacobian matrix (Equation (11));
  Calculate state mismatch (Equation (9));
  Calculate state mismatch (Equation (10));
  Calculate state mismatch (Equation (11));
  Update states $\boldsymbol{V}$ (Equation (9));
  Update states $\boldsymbol{\delta}$ (Equation (10));
  Update states $\boldsymbol{T_c}$ (Equation (11));
  Update network resistances based on $\boldsymbol{T_c}$ (Equation (A.7));
  Update network impedances;
**end**
Output results;

**Algorithm 4:** Sequential Weather-Dependent Power Flow

Read all input data (conductor, bus, load, weather);
Set error tolerance;
Initialize voltages and conductor temperatures;
**while** *error* $>$ *error tolerance* **do**
  Form/update admittance matrix $\boldsymbol{Y}(\boldsymbol{T_c})$;
  **while** *error_PQ* $>$ *error tolerance* **do**
    Calculate power injection mismatch (Equation (7));
    Calculate Jacobian matrix (Equation (7));
    Calculate state mismatch (Equation (7));
    Update states $\boldsymbol{V}$, $\boldsymbol{\delta}$ (Equation (7));
  **end**
  **while** *error_H* $>$ *error tolerance* **do**
    Calculate heat balance mismatch (Equation (8));
    Calculate Jacobian matrix (Equation (8));
    Calculate state mismatch (Equation (8));
    Update states $\boldsymbol{T_c}$ (Equation (8));
  **end**
  Update network resistances based on $\boldsymbol{T_c}$ (Equation (A.7));
  Update network impedances;
**end**
Output results;

the solution of the branch conductor temperature states from the heat balance equation. However, it may understandably be computationally complex when scaled up to larger systems requiring more number of power flow solution executions. Algorithm 4 summarizes the S-WDPF algorithm.

## V. Numerical Investigation

Extensive simulations are performed to observe the convergence computational and assess the computational complexity of the proposed WDPF algorithms. Equitable comparisons between the algorithms can become difficult due to differences in computers, programming methods, and test problems [14]. In order to minimize these uncertainties, every algorithm presented here was coded in MATLAB® [15] on a standard desktop computer for fair investigation. A number of power networks from the MATPOWER [16] library were extracted for the investigation. MATPOWER networks do not accompany conductor-specific data or weather-specific data. Therefore, in our simulations, all branch conductors for the simulated networks were assumed to be replaced by the 795 kcmil 26/7 Drake ACSR conductor [12] and lossless branches were excluded from the analysis.

For simulation purposes, the hottest weather condition in Bismarck, North Dakota, USA for the year 2017 was collected from the National Renewable Energy Laboratory's National Solar Radiation Data Base (NSRDB) [17]. The selected location has one of the largest temperature differences between the coldest and the hottest day of the year. The weather parameters and their values utilized in the simulations are presented in Table I. It is assumed that the entire network experiences the same weather condition in the simulations.

TABLE I: Hottest weather conditions in 2017

| Weather parameter | Value |
|---|---|
| Ambient temperature (°C) | 39 |
| Wind speed (m/s) | 5 |
| Wind incidence angle (°) | 20.9 |
| Solar irradiance (W/m$^2$) | 699 |

### A. Convergence Characteristics

The convergence characteristic of all the presented WDPF algorithms is investigated first. Figure 1 presents the convergence of all four WDPF algorithms for the 30-bus network. A zoomed-in view of the first 20 iterations is included in the figure. The FD-WDPF converges after around 140 iterations, requiring the highest number of iterations while the FC-WDPF algorithm converges in 5 iterations requiring the lowest number of iterations. Both PD-WDPF and S-WDPF algorithms converge to a maximum mismatch closer to the FC-WDPF algorithm after 10 iterations.

Similarly, the convergence characteristics for the 89-bus network is presented in Figure 2. A similar trend in convergence is observed, however the number of iterations required by FD-WDPF algorithm was lower.

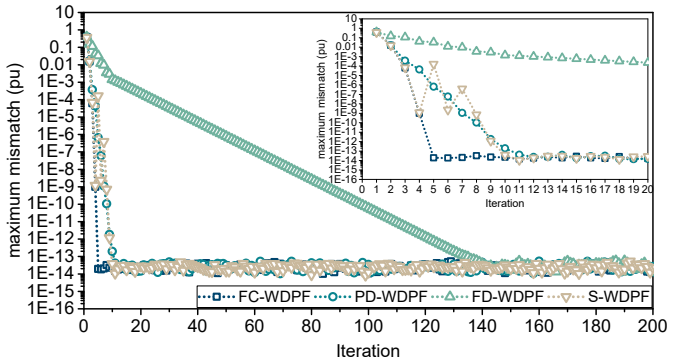



Fig. 1: Maximum mismatch (pu) versus number of iterations for the 30-bus network [16].

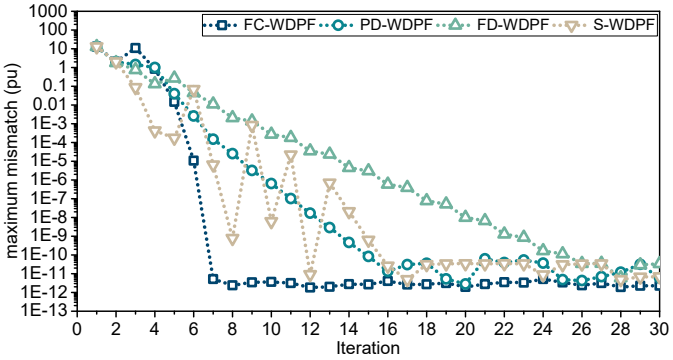



Fig. 2: Maximum mismatch (pu) versus number of iterations for the 89-bus network [16].

It should be noted that in our investigations, the FD-WDPF algorithm showed the worst convergence characteristics and diverged for some of the tested networks. This is attributed to

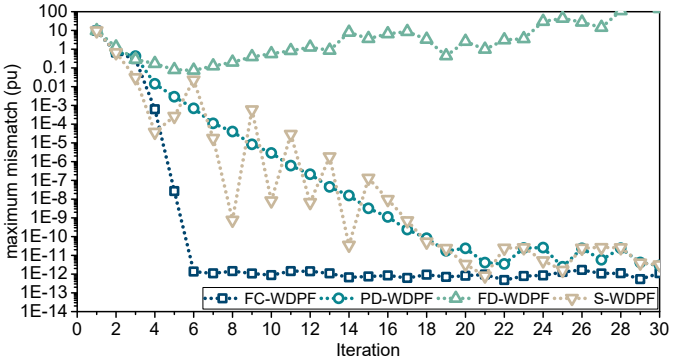



Fig. 3: Maximum mismatch (pu) versus number of iterations for the 300-bus network [16].

the oversimplification of the nonlinear relationships by decoupling the power flow solution. As an example, the convergence characteristic for the 300-bus network is presented in Figure 3. It is observed that the maximum mismatch for the FD-WDPF algorithm reduces until iteration 5, and then diverges without reaching a solution.

An interesting observation in Figures 1, 2, and 3 is the convergence of the PD-WDPF and the S-WDPF algorithm. The convergence of the S-WDPF algorithm appears to be a convergence trendline to the PD-WDPF algorithm. This is logical as Equation (7) of the PD-WDPF algorithm represents the conventional power flow, which is used in the S-WDPF algorithm. However, the differing solution and update procedure utilized in these two algorithms lead to the differences observed in the convergence in Figure 3.

### *B. Computational Complexity*

The most computationally complex step in any power flow algorithm is the inversion of the Jacobian matrix. However, due to advancements in the field of numerical techniques, there are various efficient methods of inverting the Jacobian matrix (LU decomposition, Gauss elimination, etc.) [18], [19]. In our codes, we employ the efficient internal algorithms of MATLAB® for the Jacobian inversion. It should be noted that all the algorithms were coded in a similar fashion (including the usage of Jacobian inversion in MATLAB®) for fair comparison. To compare the computational complexity of the different algorithms, the total time of execution is compared rather than the total number of operations.

Figure 4 presents the relative computation time for each of the algorithms on a number of power networks ranging from a 4-bus network to a 2736-bus network. All computation times

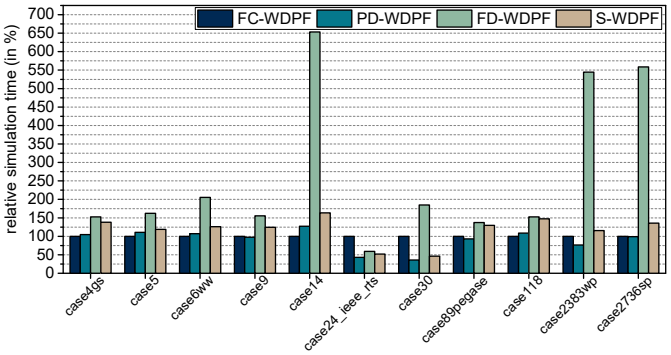



Fig. 4: Relative simulation time versus various power networks [16].

are normalized and presented in percentage by considering the computation time of FC-WDPF as the base. The results presented in Figure 4 are calculated by taking the average time of 100 separate simulation runs for each network at the same load under the weather conditions presented in Table I. However, a fewer simulations were performed for the larger network sizes as the FD-WDPF was taking excessively longer times for these networks. For example, the FD-WDPF algorithm took 281 seconds on average to complete one simulation of the 2383-bus network while it took 329 seconds on average to simulate the 2736-bus network (refer to Table II).

The FD-WDPF and the S-WDPF require more time than the other algorithms for most of the networks. This can be correlated to the convergence characteristics as observed in Section V-A. Since the S-WDPF executes multiple load flows

to reach the solution, the computational time is higher. The FD-WDPF, however, converges slowly towards the solution requiring more time. For the largest networks, the FD-WDPF becomes impractical due to very high computation time. An interesting observation is that the PD-WDPF does not provide a substantial advantage over the FC-WDPF.

Simulation times for the largest three of the networks are presented in Table II. Our investigation reveals that the FC-WDPF and the PD-WDPF are more promising as the network size increases.

TABLE II: Computational time (s) for 118-bus, 2383-bus, 2736-bus network.

| Network | FC-WDPF | PD-WDPF | FD-WDPF | S-WDPF |
|---|---|---|---|---|
| **118-bus** | 0.14 | 0.15 | 0.21 | 0.21 |
| **2383-bus** | 51.74 | 39.50 | 281.67 | 59.77 |
| **2736-bus** | 58.98 | 58.52 | 329.49 | 79.97 |

### C. Jacobian Characteristics

As a clear differentiating pattern in the computation time of the algorithms was not achieved, a look into the Jacobians of the proposed algorithms is important to understand the algorithms.

The Jacobian of the FC-WDPF algorithm for the 4-bus network is presented in Figure 5. The numbers on the axes

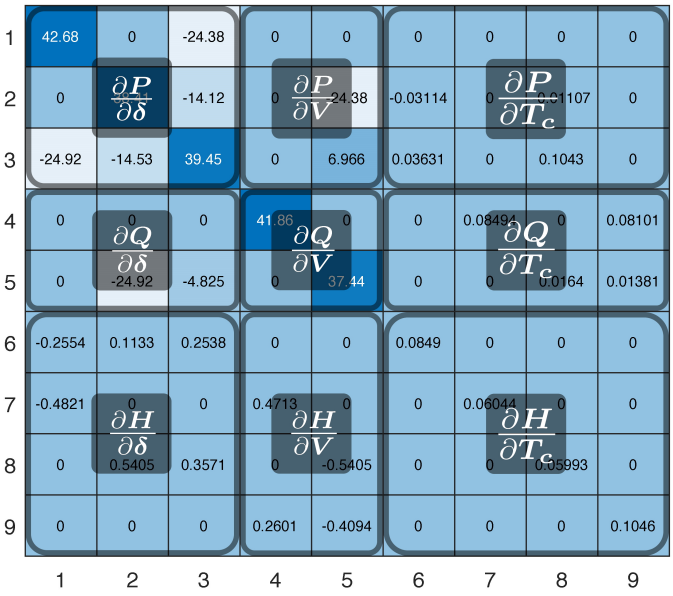
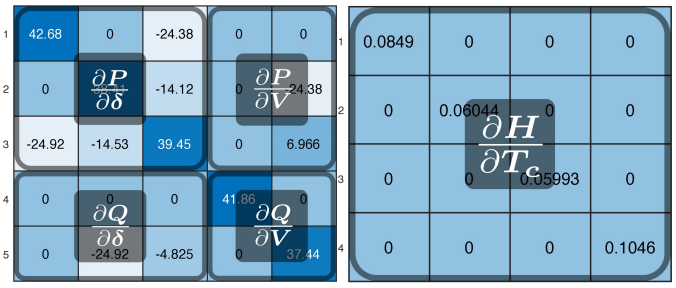



Fig. 5: Jacobian matrix of FC-WDPF algorithm (Equation (6)) for the 4-bus network [16].

represent row and column numbers of the Jacobian matrix. The matrices and their corresponding partial derivative terms are highlighted in Figure 5. As observed, the Jacobian matrix is larger than conventional power flow Jacobian (four quadrants on the top-left of the Jacobian). Although sparsity is observable, sparsity is dependent on the number of weather-dependent branches in the network. As the number of weather-dependent branches increases, the size of matrices $\frac{\partial \boldsymbol{P}}{\partial \boldsymbol{T_c}}$, $\frac{\partial \boldsymbol{Q}}{\partial \boldsymbol{T_c}}$, $\frac{\partial \boldsymbol{H}}{\partial \boldsymbol{\delta}}$, $\frac{\partial \boldsymbol{H}}{\partial \boldsymbol{V}}$, and $\frac{\partial \boldsymbol{H}}{\partial \boldsymbol{T_c}}$ increases as well. The nonzero elements in these matrices adjust the power system states for the weather-dependent impact, thereby resulting in a more accurate power flow.

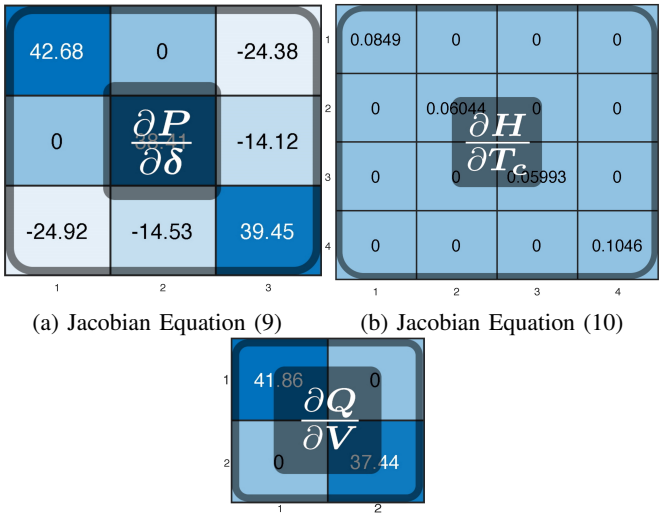
The Jacobians for the PD-WDPF algorithm and the FD-WDPF algorithm are presented in Figures 6 and 7, respectively. The Jacobians for the S-WDPF algorithm are the same as the ones for the PD-WDPF algorithm.




(a) Jacobian Equation (7) (b) Jacobian Equation (8)

Fig. 6: Jacobian matrices of PD-WDPF algorithm (Equations (7) and (8)) for the 4-bus network [16].

The PD-WDPF algorithm has Jacobians that are smaller in size in comparison to the FC-WDPF algorithm. This yields shorter iteration times, but requires a higher number of iterations to achieve the solution. In contrast, the FC-WDPF algorithm has a larger iteration time due to inversion of a larger Jacobian, but requires a lower number of iterations. Overall, the trade-off of faster iteration versus number of iterations between these two algorithms seem to balance out as the overall computation time difference is negligible for most of the cases.




(a) Jacobian Equation (9) (b) Jacobian Equation (10)

(c) Jacobian Equation (11)

Fig. 7: Jacobian matrices of FD-WDPF algorithm (Equations (9), (10), and (11)) for the 4-bus network [16].

The FD-WDPF algorithm however neglects more matrices due to decoupling, and has much smaller Jacobians as observed in Figure (7). This does mean shorter iteration times, but requiring a higher number of iterations. As a result, the states update to the solution with smaller jumps resulting in a large increase in the computation time as the network size increases.

## VI. Conclusion

This manuscript presented the derivation and comparison of various novel weather-dependent AC power flow (WDPF) algorithms. In addition, the FC-WDPF algorithm was derived and presented in polar form. Investigation of the algorithms presented reveals different convergence characteristics and computational complexity. Our investigation suggests the FC-WDPF algorithm offers a better balance between computation time and number of iterations. However, depending on the use case, the choice of algorithm may vary. It is also suggested that the FD-WDPF algorithm is more suitable for smaller networks with solvable conditions. Overall, the WDPF algorithms presented allow more accurate power flow analysis and are expected to be used as a replacement to conventional power flow algorithms. Future work entails derivation of other weather-dependent power flow algorithms like the fixed-point power flow, current injection power flow, etc. Investigation of power system planning and operation utilizing the proposed algorithms is also anticipated.

## Acknowledgement

This work was financially supported by the Singapore National Research Foundation under its Campus for Research Excellence And Technological Enterprise (CREATE) programme.

## References

[1] H. Saadat, *Power System Analysis*. PSA Publishing, 2010.

[2] J. Glover, T. Overbye, and M. Sarma, *Power System Analysis and Design*. Cengage Learning, 2016.

[3] S. Frank, J. Sexauer, and S. Mohagheghi, "Temperature-dependent power flow," *IEEE Transactions on Power Systems*, vol. 28, no. 4, pp. 4007–4018, Nov 2013.

[4] A. Ahmed, F. J. S. McFadden, and R. Rayudu, "Weather-dependent power flow algorithm for accurate power system analysis under variable weather conditions," *IEEE Transactions on Power Systems*, vol. 34, no. 4, pp. 2719–2729, July 2019.

[5] J. R. Santos, A. G. Exposito, and F. P. Sanchez, "Assessment of conductor thermal models for grid studies," *IET Generation, Transmission Distribution*, vol. 1, no. 1, pp. 155–161, January 2007.

[6] J. Cao, W. Du, and H. Wang, "Weather-based optimal power flow with wind farms integration," *IEEE Trans. Power Syst*, vol. 31, no. 4, pp. 3073–3081, 2016.

[7] A. Ahmed, F. S. McFadden, and R. Rayudu, "Transient stability study incorporating weather effects on conductors," in *2018 IEEE Power Energy Society General Meeting (PESGM)*, Aug 2018, pp. 1–5.

[8] A. Ahmed, F. S. McFadden, R. Rayudu, and T. Massier, "Weather-dependent transient stability analysis of single-machine infinite-bus system," in *2019 54th International Universities Power Engineering Conference (UPEC)*, Sep. 2019, pp. 1–6.

[9] ——, "Investigation of the impact of weather on multi-machine transient stability analysis," in *2020 IEEE Power Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, Feb 2020, pp. 1–5.

[10] V. Cecchi, M. Knudson, and K. Miu, "System impacts of temperature-dependent transmission line models," *IEEE Transactions on Power Delivery*, vol. 28, no. 4, pp. 2300–2308, Oct 2013.

[11] A. Ahmed, F. S. McFadden, R. Rayudu, and T. Massier, "Dynamic conductor temperature modelling and analysis of a distribution network with pv," in *2019 IEEE Innovative Smart Grid Technologies - Asia (ISGT Asia)*, May 2019, pp. 1889–1894.

[12] "IEEE standard for calculating the current-temperature relationship of bare overhead conductors," *IEEE Std 738-2012*, pp. 1–72, Dec 2013.

[13] R. Stephen, G. Pirovano, M. Tunstall, Y. Ojala, A. McCulloch, F. Jakl, K. Bakic, L. Varga, T. Seppa, H. Pohlman *et al.*, "The thermal behaviour of overhead conductors sections 1 and 2 mathematical model for evaluation of conductor temperature in the steady state and applications thereof," *Electra*, no. 144, pp. 107–125, 1992.

[14] M. A. Laughton and M. W. Humphrey Davies, "Numerical techniques in solution of power-system load-flow problems," *Proceedings of the Institution of Electrical Engineers*, vol. 111, no. 9, pp. 1575–1588, Sep. 1964.

[15] MATLAB, *version 9.7.0.1296695 (R2019b) Update 4*. The MathWorks Inc., 2019, the MathWorks, Natick, MA, USA.

[16] R. D. Zimmerman, C. E. Murillo-Sánchez, and R. J. Thomas, "Matpower: Steady-state operations, planning, and analysis tools for power systems research and education," *IEEE Transactions on Power Systems*, vol. 26, no. 1, pp. 12–19, Feb 2011.

[17] National Renewable Energy Laboratory, "The NSRDB Data Viewer," https://maps.nrel.gov/nsrdb-viewer/, 2019, online; accessed September 2019.

[18] S. Rosloniec, *Fundamental numerical methods for electrical engineering*. Springer Science & Business Media, 2008, vol. 18.

[19] B. Stott, "Review of load-flow calculation methods," *Proceedings of the IEEE*, vol. 62, no. 7, pp. 916–929, July 1974.

## Appendix A
## Nonlinear Heat Balance Model of Conductors

The nonlinear heat balance Equation (1) comprises $q_c$, $q_r$, $q_s$, and $q_j$. The convective heat loss rate ($q_c$) of a conductor is of two types [12], [13]: natural convection and forced convection. The convective heat loss rate is calculated utilising the following equations:

$$\begin{aligned} q_{c_1} &= K_{\text{angle}}[1.01 + 1.35{N_{\text{Re}}}^{0.52}]k_f(T_c - T_a) \\ q_{c_2} &= 0.754K_{\text{angle}}{N_{\text{Re}}}^{0.6}k_f(T_c - T_a) \\ q_{c_n} &= 3.645{\rho_f}^{0.5}D^{0.75}(T_c - T_a)^{1.25} \end{aligned} \tag{A.1}$$

In Equation (A.1), $q_{c_1}$ and $q_{c_2}$ represent forced convection, while natural convection is represented by $q_{c_n}$. The expression in Equation (A.1) that outputs the maximum value is selected as suggested by the standards [12], [13]. $N_{\text{Re}}$ in Equation (A.1) is given by:

$$N_{\text{Re}} = \frac{D\rho_f V_w}{\mu_f} \tag{A.2}$$

The radiated heat loss rate $q_r$ represents the rate at which the heat energy of a conductor is radiated to its surroundings. The equation to calculate the radiated heat loss rate [12] is given as:

$$q_r = \frac{17.8}{100^4}D\varepsilon[(T_c + 273)^4 - (T_a + 273)^4] \tag{A.3}$$

The solar heat gain rate of a conductor depends on its diameter ($D$), the absorptivity ($\alpha$), and the global solar irradiance ($Q_s$) [13]. This relationship is presented in Equation (A.4).

$$q_s = \alpha Q_s D \tag{A.4}$$

The heat gain rate due to Joule heating ($q_j$) in Equation (1) is given by:

$$q_j = P_{\text{loss}} = I^2 R(T_c) \tag{A.5}$$

The power loss ($P_{\text{loss}}$) in a conductor between two buses, bus $i$ and bus $j$, can be derived as:

$$P_{\text{loss}_{ij}} = \left[(V_i^2 + V_j^2) y_{ij} \cos\delta_{ij} - 2V_i V_j y_{ij} \cos\left(\delta_i - \delta_j - \delta_{ij}\right)\right] \tag{A.6}$$

Further details and assumptions regarding the heat balance model should be referred to in the IEEE Std 738-2012 [12].

The nonlinear heat balance Equation (1) is solved to get the temperature of the branch conductors ($T_c$) for any amount of power flowing through it under any given weather conditions. This is then utilized to update the branch resistance and then the impedance, followed by the update of the admittance matrix. Consequently, a weather-dependent admittance matrix $\boldsymbol{Y(T_c)}$ that is corrected for the power flow and weather conditions is achieved.

In Equation (A.5), $R(T_c)$ is the resistance of the conductor at the conductor temperature $T_c$, which is calculated as [12]:

$$R(T_c) = \left[\frac{R(T_{\text{high}}) - R(T_{\text{low}})}{T_{\text{high}} - T_{\text{low}}}\right](T_c - T_{\text{low}}) + R(T_{\text{low}}) \tag{A.7}$$

In Equation (A.7), $R(T_c)$ is the AC resistance at conductor temperature $T_c$, $R(T_{\text{high}})$ is the AC resistance at conductor temperature $T_{\text{high}}$, and $R(T_{\text{low}})$ is the AC resistance at conductor temperature $T_{\text{low}}$ such that $T_{\text{high}} > T_{\text{low}}$. The use of Equation (A.7) means that the magnetic effects, skin effects, and lay ratios are already included in the calculation [12].

## APPENDIX B
## DERIVATION OF THE JACOBIAN ELEMENTS

In this subsection, the partial derivative equations are derived. The equations of $P$, $Q$, and $H$ are differentiated with respect to the states $\delta$, $V$, and $T_c$.

*1)* $\frac{\partial P_k}{\partial \delta_i}$*:* Active power $P_k$ partially differentiated by $\delta_i$ yields:

$$\frac{\partial P_k}{\partial \delta_i} = \begin{cases} -\sum\limits_{\substack{j=1 \\ j\neq k}}^{n} V_k V_j Y_{kj} \cos(\delta_k - \delta_j - \delta_{kj}) & \text{for } i = k \\ V_k V_i Y_{ki} \cos\left(\delta_k - \delta_i - \delta_{ki}\right) & \text{for } i \neq k \end{cases} \tag{B.1}$$

*2)* $\frac{\partial P_k}{\partial V_i}$*:* Active power $P_k$ partially differentiated by $V_i$ yields:

$$\frac{\partial P_k}{\partial V_i} = \begin{cases} 2V_k Y_{kk} \cos\delta_{kk} + \sum\limits_{\substack{j=1 \\ j\neq k}}^{n} V_j Y_{kj} \cos\left(\delta_k - \delta_j - \delta_{kj}\right) & \text{for } i = k \\ V_k Y_{ki} \cos\left(\delta_k - \delta_i - \delta_{ki}\right) & \text{for } i \neq k \end{cases} \tag{B.2}$$

*3)* $\frac{\partial P_k}{\partial T_{c_{ij}}}$*:* Active power $P_k$ partially differentiated by branch temperature $T_{c_{ij}}$ yields:

$$\frac{\partial P_k}{\partial T_{c_{ij}}} = \begin{cases} (V_k^2 - V_k V_j \cos\left(\delta_k - \delta_j\right)) \frac{\partial g_{ij}}{\partial T_{c_{ij}}} - (V_k V_j \sin\left(\delta_k - \delta_j\right)) \frac{\partial b_{ij}}{\partial T_{c_{ij}}} & \text{for } i = k \\ (V_k^2 - V_k V_i \cos\left(\delta_k - \delta_i\right)) \frac{\partial g_{ij}}{\partial T_{c_{ij}}} - (V_k V_i \sin\left(\delta_k - \delta_i\right)) \frac{\partial b_{ij}}{\partial T_{c_{ij}}} & \text{for } j = k \\ 0 & \text{for } i, j \neq k \end{cases} \tag{B.3}$$

*4)* $\frac{\partial Q_k}{\partial \delta_i}$*:* Reactive power $Q_k$ partially differentiated by $\delta_i$ yields:

$$\frac{\partial Q_k}{\partial \delta_i} = \begin{cases} \sum\limits_{\substack{j=1 \\ j\neq k}}^{n} V_k V_j Y_{kj} \cos\left(\delta_k - \delta_j - \delta_{kj}\right) & \text{for } i = k \\ -V_k V_i Y_{ki} \cos\left(\delta_k - \delta_i - \delta_{ki}\right) & \text{for } i \neq k \end{cases} \tag{B.4}$$

*5)* $\frac{\partial Q_k}{\partial V_i}$*:* Reactive power $Q_k$ partially differentiated by $V_i$ yields:

$$\frac{\partial Q_k}{\partial V_i} = \begin{cases} -2V_k Y_{kk} \sin\delta_{kk} + \sum\limits_{\substack{j=1 \\ j\neq k}}^{n} V_j Y_{kj} \sin\left(\delta_k - \delta_j - \delta_{kj}\right) & \text{for } i = k \\ V_k Y_{ki} \sin\left(\delta_k - \delta_i - \delta_{ki}\right) & \text{for } i \neq k \end{cases} \tag{B.5}$$

*6)* $\frac{\partial Q_k}{\partial T_{c_{ij}}}$*:* Reactive power $Q_k$ partially differentiated by branch temperature $T_{c_{ij}}$ yields:

$$\frac{\partial Q_k}{\partial T_{c_{ij}}} = \begin{cases} -V_k V_j \sin\left(\delta_k - \delta_j\right) \frac{\partial g_{ij}}{\partial T_{c_{ij}}} + \left(V_k V_j \cos\left(\delta_k - \delta_j\right) - V_k^2\right) \frac{\partial b_{ij}}{\partial T_{c_{ij}}} & \text{for } i = k \\ -V_k V_i \sin\left(\delta_k - \delta_i\right) \frac{\partial g_{ij}}{\partial T_{c_{ij}}} + \left(V_k V_i \cos\left(\delta_k - \delta_i\right) - V_k^2\right) \frac{\partial b_{ij}}{\partial T_{c_{ij}}} & \text{for } j = k \\ 0 & \text{for } i, j \neq k \end{cases} \tag{B.6}$$

*7)* $\frac{\partial H_{ij}}{\partial \delta_k}$*:* Heat balance, $H_{ij}$, partially differentiated by $\delta_k$ yields:

$$\frac{\partial H_{ij}}{\partial \delta_k} = \begin{cases} -2V_i V_j y_{ij} \sin\left(\delta_i - \delta_j - \delta_{ij}\right) & \text{for } k = i \\ 2V_i V_j y_{ij} \sin\left(\delta_i - \delta_j - \delta_{ij}\right) & \text{for } k = j \\ 0 & \text{for } k \neq i, j \end{cases} \tag{B.7}$$

*8)* $\frac{\partial H_{ij}}{\partial V_k}$*:* Heat balance, $H_{ij}$, partially differentiated by $V_k$ yields:

$$\frac{\partial H_{ij}}{\partial V_k} = \begin{cases} -2(V_i y_{ij} \cos\delta_{ij} - V_j y_{ij} \cos\left(\delta_i - \delta_j - \delta_{ij}\right)) & \text{for } k = i \\ -2(V_j y_{ij} \cos\delta_{ij} - V_i y_{ij} \cos\left(\delta_i - \delta_j - \delta_{ij}\right)) & \text{for } k = j \\ 0 & \text{for } k \neq i, j \end{cases} \tag{B.8}$$

*9)* $\frac{\partial H_{ij}}{\partial T_{c_{kn}}}$*:* Heat balance, $H_{ij}$, partially differentiated by $T_{c_{kn}}$ yields:

$$\frac{\partial H_{ij}}{\partial T_{c_{kn}}} = \begin{cases} \frac{\partial q_{c_{ij}}}{\partial T_{c_{ij}}} + \frac{\partial q_{r_{ij}}}{\partial T_{c_{ij}}} - \left[V_i^2 + V_j^2 - 2V_i V_j \cos(\delta_i - \delta_j)\right] \frac{\partial g_{ij}}{\partial T_{c_{ij}}} & \text{for } kn = ij \\ 0 & \text{for } kn \neq ij \end{cases} \tag{B.9}$$

The admittance matrix utilized in the above equations is the weather-dependent admittance matrix $\boldsymbol{Y(T_c)}$. $y_{ij}$, $g_{ij}$, and $b_{ij}$ in the above equations are the branch admittance, branch conductance, and branch susceptance, respectively. The equations of $\frac{\partial g_{ij}}{\partial T_{c_{ij}}}$ and $\frac{\partial b_{ij}}{\partial T_{c_{ij}}}$ should be referred to in [4].