Optimized LinDistFlow for High-Fidelity Power Flow Modeling of Distribution Networks

Babak Taheri, Rahul K. Gupta, and Daniel K. Molzahn

***Abstract*—The DistFlow model accurately represents power flows in distribution systems, but the model's nonlinearities result in computational challenges for many optimization applications. Accordingly, a linear approximation known as LinDistFlow is commonly employed. This paper introduces an algorithm for enhancing the accuracy of the LinDistFlow approximation, with the goal of aligning the outputs more closely with those from the nonlinear DistFlow model. Using sensitivity information, our algorithm optimizes the LinDistFlow approximation's coefficient and bias parameters to minimize discrepancies in predictions of voltage magnitudes relative to the nonlinear DistFlow model. The algorithm employs the Truncated Newton Conjugate-Gradient (TNC) optimization method to fine-tune coefficients and bias parameters during an offline training phase in order to improve the LinDistFlow approximation's accuracy in optimization applications. Numerical results underscore the algorithm's efficacy, showcasing accuracy improvements in L_1-norm and L_∞-norm losses of up to 92% and 88%, respectively, relative to the traditional LinDistFlow model. We assess how the optimized parameters perform under changes in the network topology and also validate the optimized LinDistFlow approximation's efficacy in a hosting capacity optimization problem.**

***Index Terms*—DistFlow, LinDistFlow, machine learning, parameter optimization, distribution systems, hosting capacity.**

I. Introduction

Power flow models relating power injections, line flows, and voltages are central to the design and operation of electric power systems [1]. The AC power flow equations accurately model these relationships. Incorporating the AC power flow equations into optimization problems can introduce significant computational challenges due to these equations' nonlinearity [2]–[4], even for the radial network topologies that are typical of distribution systems [5]. These challenges are particularly relevant to problems that are run online to inform real-time decisions [6], [7], problems that consider uncertainties [8], and problems that model discrete decisions via mixed-integer nonlinear programming formulations [9]–[12]. To address these challenges, engineers frequently turn to linear power flow approximations that trade accuracy for computational tractability [13].

With the rapid deployment of distributed energy resources, power flow models of distribution systems are of paramount importance. Distribution systems are often modeled using an AC power flow formulation known as "DistFlow" [14]–[16]. A linearization of the DistFlow model known as "LinDistFlow" [14]–[16], or one of its variants [17]–[22], is commonly employed to make distribution system optimization problems tractable. The traditional LinDistFlow approximation linearizes the DistFlow equations by assuming that the active and reactive line losses are much smaller than the active and reactive line flows. LinDistFlow has been used to site and size capacitors [14], [15], optimize the network topology [16], compute inverter setpoints and perform Volt/VAr control [18], [19], [23], [24], formulate stochastic problems [25], ensure fairness in solar photovoltaic curtailments [26], and calculate electricity prices [27], among many other applications. Despite their extensive application in distribution network analysis, the accuracy of such models can vary, particularly outside near-nominal operating regions, as indicated in [19], [28].

Building on ideas from recently developed "adaptive" power flow approximations [29]–[37], this paper proposes an algorithm for optimizing the LinDistFlow parameter values to improve the accuracy of this approximation. Adaptive power flow approximations are linearizations that are tailored to a specific system and operating range of interest. This contrasts with traditional power flow approximations that are often derived using general assumptions about broad classes of systems or are based on a particular nominal operating point. Adaptive power flow approximations invest computing time up front to calculate linearization coefficients in order to achieve increased accuracy when the approximations are deployed in an optimization problem. Thus, adaptive approximations are well suited for settings with both offline and online aspects (e.g., using a day-ahead forecast to compute linearization coefficients that are used for online computations in real-time applications [6], [7]) as well as settings where a nonlinear AC power flow model would lead to an intractable formulation [8]–[12].

For instance, the cold-start DC power flow approximation [38] relies on assumptions of near-nominal voltage magnitudes, negligible line resistances, and small phase angle differences between connected buses. The accuracy of the cold-start DC power flow approximation suffers upon deviations from these assumptions. Hot-start variants of the DC power flow incorporate information from a nominal AC operating point [38], but accuracy still may suffer for deviations away from this nominal point [39]. Our prior work in [40] proposes an adaptive DC power flow approximation based on coefficient and bias parameters that optimize accuracy for a particular system over a specified operating range of interest. Other adaptive power flow approximations minimize linearization error in the worst-case [29] and in expectation [30] or leverage sample-based regression approaches [31]–[33] including techniques for constructing overestimating and underestimating approximations [34]; see [35]–[37] for recent surveys.

While not using traditional machine learning models like neural networks, the approach in [40] for optimizing the DC power flow parameters draws inspiration from methods for training machine learning models. Using analytically calculated parameter sensitivities, an offline training algorithm iteratively updates parameter values to minimize a loss function defined with respect to the AC power flow solutions over a set of sampled operating points. Leveraging parameter

School of Electrical and Computer Engineering, Georgia Institute of Technology. {taheri, rahul.gupta, molzahn}@gatech.edu. Support from NSF award #2145564.

optimization methods developed for training machine learning models, we employ the Truncated Newton Conjugate-Gradient (TNC) method to achieve scalability. The optimized parameters are then used in online calculations for real-time settings or in problems for which a nonlinear AC power flow model would lead to intractability. Using a conceptually similar approach, this paper optimizes the parameters of the LinDistFlow approximation. In contrast to existing adaptive power flow approximations [29]–[37], this approach maintains the structure of the LinDistFlow approximation as dictated by the network topology. Accordingly, the resulting parameter-optimized LinDistFlow approximation has the key advantage of being directly deployable in the many existing applications that rely on LinDistFlow (e.g., [14]–[16], [18], [19], [23]–[27]). As an illustrative application, this paper uses the optimized parameter values in a hosting capacity analysis. Moreover, as this paper also demonstrates, maintaining the underlying network structure enables straightforward modeling of network topology changes. Numerical comparisons demonstrate substantial accuracy advantages of our proposed approach compared to the traditional LinDistFlow approximation as well as several recent LinDistFlow variants that also tune parameter values [20]–[22].

To summarize, the key contributions of this paper are:

- Introducing an algorithm that optimizes the LinDistFlow approximation's coefficient and bias parameters.
- Training the model across diverse operating conditions while employing the TNC optimization method to extend the algorithm's tractability.
- Conducting extensive evaluations of the proposed algorithm in various loading conditions. These tests demonstrate its improved performance over existing linear power flow models.
- Demonstrating the algorithm's capability to compute parameter values that provide improved accuracy across multiple network topologies.
- Illustrating the efficacy of the optimized LinDistFlow approximation in a hosting capacity problem.

The remainder of this paper is organized as follows. Section II reviews the DistFlow model and LinDistFlow approximation. Section III introduces our proposed algorithm. Section IV presents numerical experiments that evaluate the performance of our algorithm. Section V concludes the paper.

II. Power Flow Modeling

This section introduces the DistFlow formulation and its linear approximation, LinDistFlow. We first establish notation. Let operators $(\,\cdot\,)^{-1}$, $|\cdot|$, $(\,\cdot\,)^{\top}$, and $(\,\cdot\,)^{-\top}$ denote the square matrix inverse, the absolute value of a number, the transpose of a matrix/vector, and the transpose of a square matrix inverse, respectively. Let $\mathcal{N} := \{0, 1, \ldots, n\}$ and $\mathcal{E} := \{1, \ldots, n\}$ denote the sets of buses and lines, respectively, in a distribution network, where $|\mathcal{E}| = |\mathcal{N}| - 1$ for radial networks. Let $\mathcal{N}' = \mathcal{N} \setminus ref$ represent the set of all buses excluding the substation bus. For each bus $n \in \mathcal{N}$ and each line $(\pi_n, n) \in \mathcal{E}$, V_n and V_{π_n} represent voltages at the bus and its parent node, $z_n = r_n + jx_n$ is the line impedance with resistance r_n and reactance x_n, where $j := \sqrt{-1}$. Let I_n denote the current of the line, and $S_n = P_n + jQ_n$ the complex power flow of the line. Note that lines are identified by their child nodes to simplify notation. Net power injection at bus n is given by $s_n = p_n + jq_n$. Additional variables include squared voltage and current magnitudes $v_n = |V_n|^2$ and $\ell_n = |I_n|^2$. $\mathbf{D}_r = \mathrm{diag}(\mathbf{r})$ and $\mathbf{D}_x = \mathrm{diag}(\mathbf{x})$ are diagonal matrices of resistances and reactances, $\mathbf{p} = [p_1, \ldots, p_n]^{\top}$, $\mathbf{q} = [q_1, \ldots, q_n]^{\top}$, $\mathbf{P} = [P_1, \ldots, P_n]^{\top}$, and $\mathbf{Q} = [Q_1, \ldots, Q_n]^{\top}$ are vectors of active/reactive power injections and flows. Furthermore, define $\hat{\mathbf{A}} = [\mathbf{a}_0 \;\; \mathbf{A}]$ as the $|\mathcal{E}| \times |\mathcal{N}|$ branch-bus incidence matrix describing the connections between the system's buses and branches, where $\mathbf{a}_0$ is the length-$|\mathcal{E}|$ vector associated with the substation bus and $\mathbf{A}$ is the reduced branch-bus matrix for all buses besides the substation bus. Let $\mathbf{v} := [|V_1|^2, \ldots, |V_n|^2]^{\top}$ represent squared voltage magnitudes, and $v_0 = |V_0|^2$ corresponds to the substation bus. Bias parameters $\boldsymbol{\gamma}$, $\boldsymbol{\rho}$, and $\boldsymbol{\varrho}$ formulated as vectors of length $|\mathcal{N}| - 1$ are optimized using our proposed machine learning-inspired algorithm to enhance voltage prediction accuracy.

A. DistFlow Model

The DistFlow model [14]–[16] accurately represents voltage, current, and power flow relationships in radial distribution networks, enabling analyses of low- and medium-voltage systems where line losses and voltage drops are important factors. The two-bus subsystem depicted in Fig. 1 helps illustrate the DistFlow model. The DistFlow model is:

$$v_n = |V_n|^2, \quad \ell_n = |I_n|^2, \qquad \forall n \in \mathcal{N} \quad \text{(1a)}$$

$$\sum_{k: n \to k} P_k = p_n + P_n - r_n \ell_n, \qquad \forall n \in \mathcal{N} \quad \text{(1b)}$$

$$\sum_{k: n \to k} Q_k = q_n + Q_n - x_n \ell_n, \qquad \forall n \in \mathcal{N} \quad \text{(1c)}$$

$$v_n = v_{\pi_n} - 2(r_n P_n + x_n Q_n) + (r_n^2 + x_n^2)\ell_n, \; \forall n \in \mathcal{N} \quad \text{(1d)}$$

$$v_{\pi_n} \ell_n = P_n^2 + Q_n^2, \qquad \forall n \in \mathcal{N} \quad \text{(1e)}$$

Equation (1a) defines variables for the squared voltage magnitudes v_n and squared current flow magnitudes ℓ_n which are used in place of the voltage phasors V_n and current phasors I_n. Equations (1b) and (1c) correspond to active and reactive power balance at each bus. Equation (1d) models the change in squared voltage magnitudes across lines. Equation (1e) uses the definition of apparent power to relate the squared voltages v_n, squared currents ℓ_n, and squared apparent power $P_n^2 + Q_n^2$. The DistFlow model is nonlinear due to (1e).

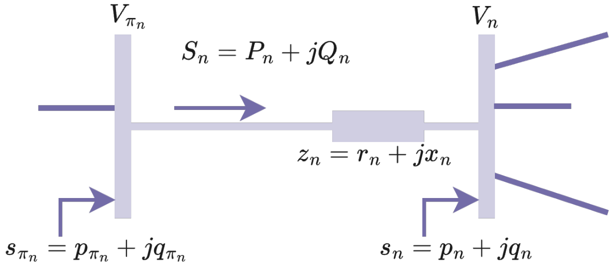


Fig. 1. A two-bus system representing distribution line n feeding bus n from its parent bus π_n

B. Traditional LinDistFlow Approximation

The LinDistFlow approximation linearizes the DistFlow equations by neglecting the active and reactive line loss terms $r_n \ell_n$ in (1b), $x_n \ell_n$ in (1c), and $(r_n^2 + x_n^2)\ell_n$ in (1d) [14]–[16]. Without these, the nonlinear equation (1e) can be dropped, with the remaining equations linearly relating the squared voltage magnitudes, the power injections, and line flows:

$$\sum_{k:n\rightarrow k} P_k \approx p_n + P_n, \qquad \forall n \in \mathcal{N} \tag{2a}$$

$$\sum_{k:n\rightarrow k} Q_k \approx q_n + Q_n, \qquad \forall n \in \mathcal{N} \tag{2b}$$

$$v_n \approx v_{\pi_n} - 2(r_n P_n + x_n Q_n), \qquad \forall n \in \mathcal{N} \tag{2c}$$

The LinDistFlow approximation can be equivalently represented in the following matrix form:

$$\mathbf{D}_r = \text{diag}(\mathbf{r}), \quad \mathbf{D}_x = \text{diag}(\mathbf{x}), \tag{3a}$$

$$\mathbf{p} = \mathbf{A}^\top \mathbf{P}, \tag{3b}$$

$$\mathbf{q} = \mathbf{A}^\top \mathbf{Q}, \tag{3c}$$

$$\mathbf{A}\mathbf{v} + v_0 \mathbf{a}_0 = 2\mathbf{D}_r \mathbf{P} + 2\mathbf{D}_x \mathbf{Q}, \tag{3d}$$

$$\mathbf{v} = v_0 \mathbf{1} + 2\mathbf{A}^{-1}\mathbf{D}_r \mathbf{A}^{-\top}\mathbf{p} + 2\mathbf{A}^{-1}\mathbf{D}_x \mathbf{A}^{-\top}\mathbf{q}. \tag{3e}$$

For radial networks where all lines have positive resistance ($r_n \geq 0$) and reactance ($x_n \geq 0$), reference [41] analytically demonstrates that the LinDistFlow approximation overestimates the voltage magnitudes and underestimates the complex power flows required to supply the loads. The following section presents an approach to reduce this approximation error by optimizing the LinDistFlow parameters.

III. Optimized LinDistFlow Approximation (OLDF)

We first generalize the LinDistFlow approximation by introducing bias parameters $\boldsymbol{\gamma}$, $\boldsymbol{\rho}$, and $\boldsymbol{\varrho}$ that offset the squared voltage magnitudes, active power injections, and reactive power injections, respectively:

$$\mathbf{v} = v_0 \mathbf{1} + 2\mathbf{A}^{-1}\mathbf{D}_r \mathbf{A}^{-\top}(\mathbf{p} + \boldsymbol{\rho}) + 2\mathbf{A}^{-1}\mathbf{D}_x \mathbf{A}^{-\top}(\mathbf{q} + \boldsymbol{\varrho}) + \boldsymbol{\gamma}. \tag{4}$$

Appropriate selection of parameter values for $\mathbf{D}_r$, $\mathbf{D}_x$, $\boldsymbol{\gamma}$, $\boldsymbol{\rho}$, and $\boldsymbol{\varrho}$ can significantly improve the LinDistFlow approximation's accuracy. We next introduce a machine learning-inspired algorithm to optimize the $\mathbf{D}_r$, $\mathbf{D}_x$, $\boldsymbol{\gamma}$, $\boldsymbol{\rho}$, and $\boldsymbol{\varrho}$ parameters. Our proposed algorithm seeks to reduce the discrepancy between the voltages predicted by the LinDistFlow approximation (4) and those from the DistFlow model (1).

A. Parameter Optimization Algorithm Overview

As illustrated in Fig. 2, we propose a two-phase algorithm for parameter optimization with an initial *offline* training phase followed by an *online* application phase. The *offline* phase, conducted once, optimizes the values for the parameters $\mathbf{D}_r$, $\mathbf{D}_x$, $\boldsymbol{\gamma}$, $\boldsymbol{\rho}$, and $\boldsymbol{\varrho}$. This initial computation seeks to align the LinDistFlow approximation with the DistFlow model's behavior across varied operational scenarios. This paves the way for the *online* phase where the optimized parameters are used to improve accuracy in various applications.

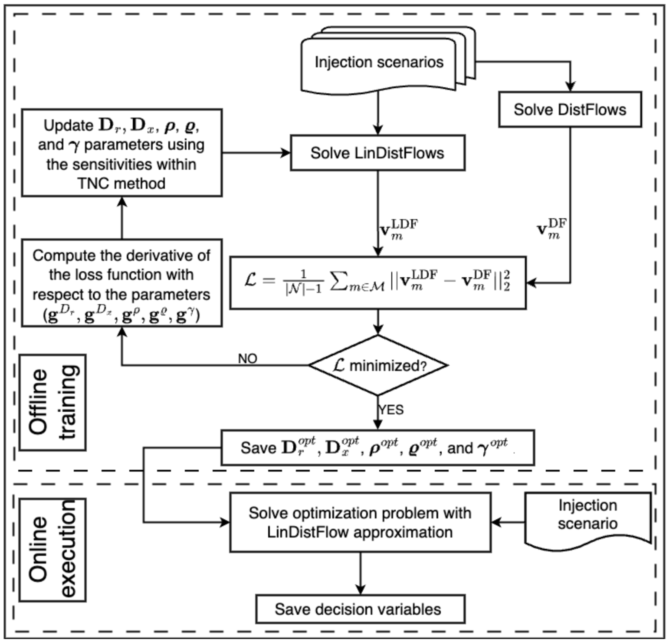

Injection scenarios
Solve DistFlows
Update $\mathbf{D}_r, \mathbf{D}_x$, $\boldsymbol{\rho}$, $\boldsymbol{\varrho}$, and $\boldsymbol{\gamma}$ parameters using the sensitivities within TNC method
Solve LinDistFlows
$\mathbf{v}_m^{\text{LDF}}$
$\mathbf{v}_m^{\text{DF}}$
Compute the derivative of the loss function with respect to the parameters $(\mathbf{g}^{D_r}, \mathbf{g}^{D_x}, \mathbf{g}^{\rho}, \mathbf{g}^{\varrho}, \mathbf{g}^{\gamma})$
$\mathcal{L} = \frac{1}{|\mathcal{N}|-1}\sum_{m\in\mathcal{M}} ||\mathbf{v}_m^{\text{LDF}} - \mathbf{v}_m^{\text{DF}}||_2^2$
NO
$\mathcal{L}$ minimized?
YES
Offline training
Save $\mathbf{D}_r^{opt}$, $\mathbf{D}_x^{opt}$, $\boldsymbol{\rho}^{opt}$, $\boldsymbol{\varrho}^{opt}$, and $\boldsymbol{\gamma}^{opt}$
Online execution
Solve optimization problem with LinDistFlow approximation
Injection scenario
Save decision variables

Fig. 2. Flowchart depicting the proposed optimization algorithm.

To optimize the parameters $\mathbf{D}_r$, $\mathbf{D}_x$, $\boldsymbol{\gamma}$, $\boldsymbol{\rho}$, and $\boldsymbol{\varrho}$, this section first formulates a loss function. The loss function gauges the LinDistFlow approximation's accuracy by comparing its voltage predictions to those from the DistFlow model across varied operational conditions. This section then analytically derives the loss function's sensitivities with respect to the coefficient and bias parameters. Informed by these sensitivities, this section finally applies the TNC optimization method to minimize the loss function. The TNC method approximates the Hessian matrix based on the given gradients to achieve computational scalability. The resulting optimized parameters are used online for various applications of LinDistFlow, as we will discuss and numerically benchmark in Section IV.

B. Formulation of the Loss Function

We define a loss function, $\mathcal{L}$, formulated as the sum of squared discrepancies (in the two-norm sense) between the voltage solutions of the DistFlow model ($\mathbf{v}_m^{DF}$) and LinDistFlow approximation ($\mathbf{v}_m^{LDF}$) for a given set of load scenarios $\mathcal{M} = \{1, 2, \ldots, S\}$. This formulation, akin to strategies prevalent in machine learning, is known for its numerical robustness and analytical tractability. The loss function is:

$$\begin{aligned}\mathcal{L}(\mathbf{D}_r, \mathbf{D}_x, \boldsymbol{\gamma}, \boldsymbol{\rho}, \boldsymbol{\varrho}) &= \frac{1}{|\mathcal{N}|-1}\sum_{m\in\mathcal{M}} ||\mathbf{v}_m^{LDF} - \mathbf{v}_m^{DF}||_2^2, \\ &= \frac{1}{|\mathcal{N}|-1}\sum_{m\in\mathcal{M}} ||v_0\mathbf{1} + 2\mathbf{A}^{-1}\mathbf{D}_r\mathbf{A}^{-\top}(\mathbf{p}+\boldsymbol{\rho}) + \\ &\qquad 2\mathbf{A}^{-1}\mathbf{D}_x\mathbf{A}^{-\top}(\mathbf{q}+\boldsymbol{\varrho}) + \boldsymbol{\gamma} - \mathbf{v}_m^{DF}||_2^2,\end{aligned} \tag{5}$$

where normalization by $\frac{1}{|\mathcal{N}|-1}$ adjusts for the system size. This equation underscores how $\mathbf{v}_m^{LDF}$, and consequently $\mathcal{L}$, are influenced by the coefficient and bias parameters. Using two-norm discrepancies ensures that larger deviations are more heavily penalized, aligning with typical operational priorities where minimizing the most significant errors is often more important than reducing numerous smaller inaccuracies.

The optimal LinDistFlow parameters are computed by minimizing this loss function:

$$\min_{\mathbf{D}_r, \mathbf{D}_x, \boldsymbol{\gamma}, \boldsymbol{\rho}, \boldsymbol{\varrho}} \mathcal{L}(\mathbf{D}_r, \mathbf{D}_x, \boldsymbol{\gamma}, \boldsymbol{\rho}, \boldsymbol{\varrho}). \tag{6}$$

C. Parameter Sensitivity Analysis

We use the TNC optimization method to solve (6). This method relies on the gradients of the loss function with respect to the parameters, which we present next. We start with sensitivities for $\mathbf{D}_r$ and $\mathbf{D}_x$, represented by $\mathbf{g}^{D_r}$ and $\mathbf{g}^{D_x}$. These sensitivities are calculated via the partial derivatives of the loss function $\mathcal{L}$ with respect to these parameters:

$$\mathbf{g}^{D_r} = \frac{2}{|\mathcal{N}|} \sum_{m \in \mathcal{M}} \frac{\partial \mathbf{v}^{LDF}}{\partial \mathbf{D}_r}\bigg|_{\mathbf{v}_m^{LDF}} \left(\mathbf{v}_m^{LDF} - \mathbf{v}_m^{DF} \right), \tag{7a}$$

$$\mathbf{g}^{D_x} = \frac{2}{|\mathcal{N}|} \sum_{m \in \mathcal{M}} \frac{\partial \mathbf{v}^{LDF}}{\partial \mathbf{D}_x}\bigg|_{\mathbf{v}_m^{LDF}} \left(\mathbf{v}_m^{LDF} - \mathbf{v}_m^{DF} \right), \tag{7b}$$

where $\frac{\partial \mathbf{v}^{LDF}}{\partial \mathbf{D}_r}$ and $\frac{\partial \mathbf{v}^{LDF}}{\partial \mathbf{D}_x}$ are obtained from the derivatives of (4) with respect to the coefficient parameters $\mathbf{D}_r$ and $\mathbf{D}_x$:

$$\frac{\partial \mathbf{v}^{LDF}}{\partial \mathbf{D}_r} = 2 \Big(\mathbf{A}^{-\top} (\mathbf{p} + \boldsymbol{\rho}) \Big)^{\top} \mathbf{A}^{-1}, \tag{7c}$$

$$\frac{\partial \mathbf{v}^{LDF}}{\partial \mathbf{D}_x} = 2 \Big(\mathbf{A}^{-\top} (\mathbf{q} + \boldsymbol{\varrho}) \Big)^{\top} \mathbf{A}^{-1}. \tag{7d}$$

The gradients of the loss function $\mathcal{L}$ with respect to the bias parameters $\boldsymbol{\gamma}$, $\boldsymbol{\rho}$, and $\boldsymbol{\varrho}$ are represented by $\mathbf{g}^{\gamma}$, $\mathbf{g}^{\rho}$, and $\mathbf{g}^{\varrho}$, respectively:

$$\mathbf{g}^{\gamma} = \frac{2}{|\mathcal{N}| - 1} \sum_{m \in \mathcal{M}} \frac{\partial \mathbf{v}^{LDF}}{\partial \boldsymbol{\gamma}}\bigg|_{\mathbf{v}_m^{LDF}} \left(\mathbf{v}_m^{LDF} - \mathbf{v}_m^{DF} \right), \tag{8a}$$

$$\mathbf{g}^{\rho} = \frac{2}{|\mathcal{N}| - 1} \sum_{m \in \mathcal{M}} \frac{\partial \mathbf{v}^{LDF}}{\partial \boldsymbol{\rho}}\bigg|_{\mathbf{v}_m^{LDF}} \left(\mathbf{v}_m^{LDF} - \mathbf{v}_m^{DF} \right), \tag{8b}$$

$$\mathbf{g}^{\varrho} = \frac{2}{|\mathcal{N}| - 1} \sum_{m \in \mathcal{M}} \frac{\partial \mathbf{v}^{LDF}}{\partial \boldsymbol{\varrho}}\bigg|_{\mathbf{v}_m^{LDF}} \left(\mathbf{v}_m^{LDF} - \mathbf{v}_m^{DF} \right), \tag{8c}$$

where $\frac{\partial \mathbf{v}^{LDF}}{\partial \boldsymbol{\gamma}}$, $\frac{\partial \mathbf{v}^{LDF}}{\partial \boldsymbol{\rho}}$, and $\frac{\partial \mathbf{v}^{LDF}}{\partial \boldsymbol{\varrho}}$ are calculated by taking the derivatives of (4) with respect to bias parameters $\boldsymbol{\gamma}$, $\boldsymbol{\rho}$, and $\boldsymbol{\varrho}$:

$$\frac{\partial \mathbf{v}^{LDF}}{\partial \boldsymbol{\gamma}} = \mathbf{I}, \tag{8d}$$

$$\frac{\partial \mathbf{v}^{LDF}}{\partial \boldsymbol{\rho}} = 2 \mathbf{A}^{-1} \mathbf{D}_r \mathbf{A}^{-\top}, \tag{8e}$$

$$\frac{\partial \mathbf{v}^{LDF}}{\partial \boldsymbol{\varrho}} = 2 \mathbf{A}^{-1} \mathbf{D}_x \mathbf{A}^{-\top}, \tag{8f}$$

where $\mathbf{I}$ is the identity matrix. These sensitivities enable gradient-based methods such as TNC for optimizing the parameters $\mathbf{D}_r$, $\mathbf{D}_x$, $\boldsymbol{\gamma}$, $\boldsymbol{\rho}$, and $\boldsymbol{\varrho}$, as we will describe next.

D. Implementation of the Optimization Solution

The gradients in Section III-C enable the application of gradient-based optimization methods such as TNC [42], [43] to solve the parameter optimization problem (6). The choice of TNC as opposed to other gradient-based optimization methods such as Broyden-Fletcher-Goldfarb-Shanno (BFGS) and limited-memory BFGS [42], [43] is based on the TNC method's superior scalability in our empirical testing compare to these alternatives. TNC iteratively approximates the Hessian matrix's inverse, efficiently managing memory and computation even in high-dimensional spaces. TNC operates by approximating the solution of Newton's equations for a given function's local minimum, truncating the process early to save computational resources while still moving significantly towards the minimum. This method is particularly suited for problems where the evaluation of the full Hessian matrix is impractical due to computational constraints. We use SciPy's `scipy.optimize.minimize` TNC implementation. Our termination criteria combine gradient norm thresholds and iteration limits.

IV. Numerical Analysis

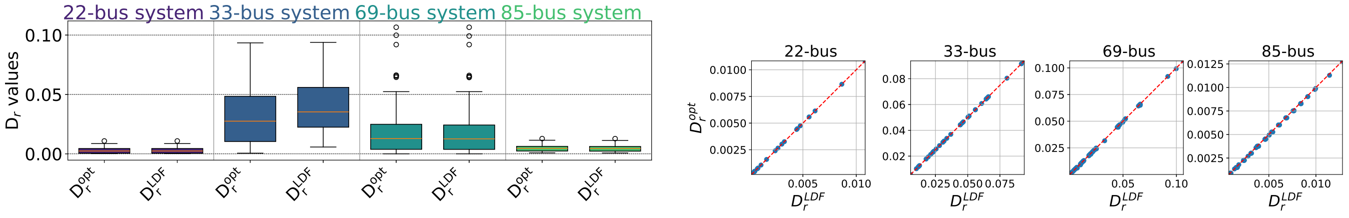
This section empirically benchmarks the proposed algorithm "optimized LinDistFlow" (OLDF) against several other related power flow linearizations. Specifically, in addition to the nonlinear DistFlow model (1) that provides the ground truth via an AC power flow solution, we also benchmark our optimized LinDistFlow against the traditional LinDistFlow approximation (LDF) [14]–[16], the parameterized linear power flow (PLPF) approximation from [22], the Lossy DistFlow (LoDF) approximation from [21], and the decoupled linear power flow (DLPF) approximation from [20]. We use various test systems and loading scenarios to replicate the methodologies adopted in the referenced literature. The test cases include the `IEEE 33-bus`, `IEEE 69-bus`, and modified `IEEE 123-bus` [44] systems as well as the `22-bus`, `85-bus`, and `141-bus` distribution test cases from Matpower [45].

A. Algorithm Training

We used 20 power injection scenarios during our algorithm's training phase. These scenarios were derived by scaling the nominal power injections by multipliers at each bus that follow normal distributions with a mean of one and a standard deviation of 35%. The DistFlow problem solutions were calculated using the `PowerModels.jl` software [46] on a computing node within the Partnership for an Advanced Computing Environment (PACE) at Georgia Tech with a 24-core CPU and 32 GB of RAM. The training algorithm is implemented in Python 3 in a Jupyter Notebook environment using the TNC method from the `scipy.optimize.minimize` library with (5) as the objective function, $\mathbf{g} = [\mathbf{g}^{D_r}{}^{\top}, \mathbf{g}^{D_x}{}^{\top}, \mathbf{g}^{\rho\top}, \mathbf{g}^{\varrho\top}, \mathbf{g}^{\gamma\top}]^{\top}$ as the Jacobian, 1×10^{-6} per unit as the convergence tolerance, and 100 as the iteration limit.

B. Performance Metrics

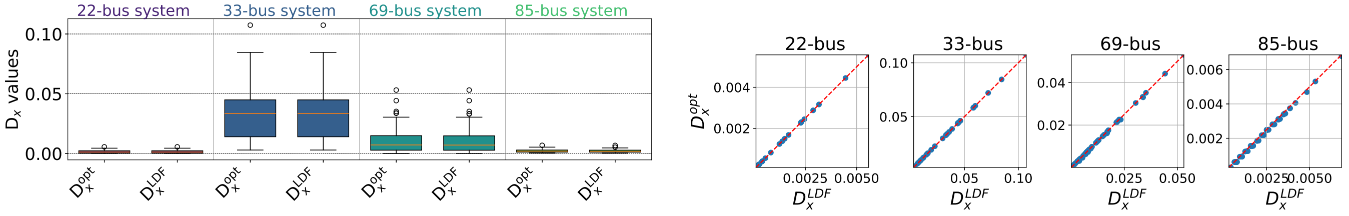
The linear power flow approximations' accuracy was quantified by comparing their voltage magnitude predictions (i.e.,



(a)

(b)

Fig. 3. (a) Boxplots showing the distributions of the $\mathbf{D}_r$ parameter values for multiple test cases. Each test case is represented by two boxplots indicating the cold-start and optimal $\mathbf{D}_r$ parameter values. (b) Scatter plots comparing the coefficient values $\mathbf{D}_r^{LDF}$ and $\mathbf{D}_r^{opt}$ for various test cases.



(a)

(b)

Fig. 4. (a) Boxplots showing the distributions of the $\mathbf{D}_x$ parameter values for multiple test cases. Each test case is represented by two boxplots indicating the cold-start and optimal $\mathbf{D}_x$ parameter values. (b) Scatter plots comparing the coefficient values $\mathbf{D}_x^{LDF}$ and $\mathbf{D}_x^{opt}$ for various test cases.

$\mathbf{v}^{[\text{model}]}$, where [model] is OLDF, PLPF, LoDF, LDF, or DLPF for the various power flow approximations) against the true values from the nonlinear DistFlow solutions returned by `PowerModels.jl` (i.e., $\mathbf{v}^{DF}$). The discrepancies were measured using maximum and mean error metrics in the per unit (p.u.) system, as defined by the following equations:

$$\varepsilon_{\text{max}}^{[\text{model}]} = \|\mathbf{v}^{[\text{model}]} - \mathbf{v}^{DF}\|_\infty \tag{9}$$

$$\varepsilon_{\text{avg}}^{[\text{model}]} = \frac{1}{|\mathcal{M}|(|\mathcal{N}|-1)}\|\mathbf{v}^{[\text{model}]} - \mathbf{v}^{DF}\|_1 \tag{10}$$

where $|\mathcal{M}|$ denotes the number of testing samples, $|\mathcal{N}|-1$ represents the number of non-root nodes in the distribution systems, $\|\cdot\|_\infty$ is the L_∞-norm, and $\|\cdot\|_1$ is the L_1-norm.

The results presented across Fig. 6 and Tables I, II, and III demonstrate the efficacy of our parameter optimization algorithm in enhancing the LinDistFlow approximation's accuracy under various load scenarios. This section next provides detailed examinations of these outcomes.

C. Parameter Optimization Analysis

We next present the results of the parameter optimization across various test cases by plotting the parameter values from the traditional LinDistFlow approximation and the LinDistFlow with optimized parameter values.

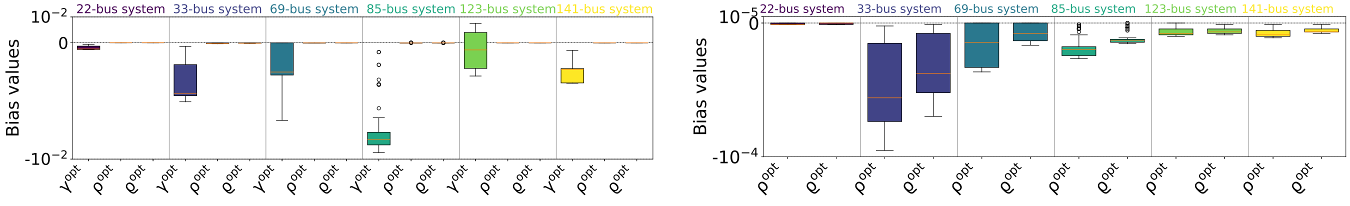
Figs. 3 and 4 utilize box plots to illustrate the distributions of $\mathbf{D}_r$ and $\mathbf{D}_x$ parameter values, respectively, for the traditional LinDistFlow and our optimized parameters. Each box plot captures the interquartile range (IQR) with the middle 50% of the data shown with a median line. The whiskers extend to 1.5 times the IQR, with outliers represented as individual points. The horizontal lines at the whiskers' ends indicate the 90^{th} percentile of the data. For each test system, the box plot figures display two distributions: the LinDistFlow ($\mathbf{D}_r^{LDF}$ or $\mathbf{D}_x^{LDF}$) and the results from our optimization algorithm ($\mathbf{D}_r^{opt}$ and $\mathbf{D}_x^{opt}$). These boxplots reveal that the distributions of the optimized parameter values align closely with those from existing heuristics for selecting $\mathbf{D}_r$ and $\mathbf{D}_x$. This indicates that our algorithm yields parameter values in a reasonable range with values that are consistent with conventional well-established heuristics.

In addition, scatter plots accompanying these box plots compare LinDistFlow ($\mathbf{D}_r^{LDF}$, $\mathbf{D}_x^{LDF}$) and optimized ($\mathbf{D}_r^{opt}$, $\mathbf{D}_x^{opt}$) parameter values. The red dashed line at 45° in each subplot signifies a one-to-one correlation in the parameter values. These plots show that the optimized parameters are broadly similar to those from existing heuristics, suggesting an alignment with longstanding power engineering intuition that the line resistances and reactances are key parameters in dictating power flows. Despite the overall consistency with traditional LinDistFlow parameter choices, our numerical results show that the optimized parameters result in significant accuracy improvements.

Fig. 5 showcases the distribution of bias parameters via box plots across several test cases. Illustrated in Fig. 5a, the optimal values of $\boldsymbol{\rho}^{opt}$ and $\boldsymbol{\varrho}^{opt}$ are notably smaller in magnitude when compared to those of $\boldsymbol{\gamma}^{opt}$. Consequently, Fig. 5b presents the $\boldsymbol{\rho}^{opt}$ and $\boldsymbol{\varrho}^{opt}$ parameters separately to highlight their distinct distributions. This disparity suggests the feasibility of simplifying the model by focusing on optimizing only three parameters: $\mathbf{D}_r$, $\mathbf{D}_x$, and $\boldsymbol{\gamma}$, for the LinDistFlow approximation enhancement. Nonetheless, our experimental results indicate that while this simplification yields comparable outcomes for base and random load conditions, it falls short under high-load scenarios. This underscores the importance of including $\boldsymbol{\rho}$ and $\boldsymbol{\varrho}$ to better address high-load conditions.

D. Algorithm Testing

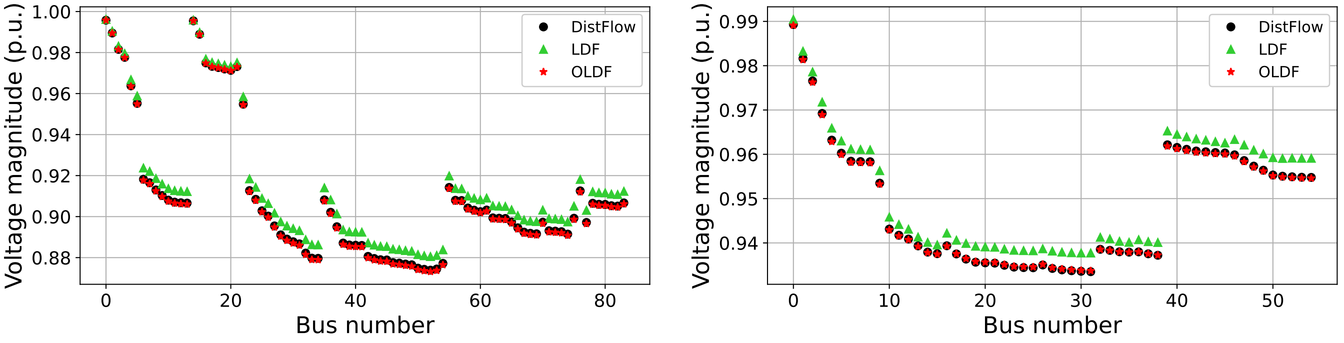
We next characterize the optimized parameter values' performance relative to alternative LinDistFlow formulations under three different types of load scenarios: *base load*, *high load*, and *random load*.



(a)

(b)

Fig. 5. Boxplots showing the distributions of the bias parameters for multiple test cases. Each test case is represented by three boxplots indicating the $\boldsymbol{\gamma}^{opt}$, $\boldsymbol{\rho}^{opt}$, and $\boldsymbol{\varrho}^{opt}$ parameter values. (b) Plotting only $\boldsymbol{\rho}^{opt}$ and $\boldsymbol{\varrho}^{opt}$ parameters for better comparison.



(a) `85-bus`

(b) Modified `123-bus`

Fig. 6. Voltage profiles for the base case using initial LinDistFlow parameters vs. the optimized parameters.

TABLE I
MODEL EVALUATION - BASE LOAD

Test case	ε_{avg}^{LDF}	ε_{avg}^{PLPF}	ε_{avg}^{LoDF}	ε_{avg}^{DLPF}	ε_{avg}^{OLDF}	$\varepsilon_{\max}^{LDF}$	$\varepsilon_{\max}^{PLPF}$	$\varepsilon_{\max}^{LoDF}$	$\varepsilon_{\max}^{DLPF}$	$\varepsilon_{\max}^{OLDF}$
22-bus	0.00023	0.00014	0.00236	0.00040	**0.00001**	0.00030	0.00025	0.00314	0.00066	**0.00001**
33-bus	0.00198	0.00080	0.00288	0.00368	**0.00015**	0.00284	0.00125	0.00402	0.00638	**0.00025**
69-bus	0.00119	0.00075	0.00112	0.00186	**0.00023**	0.00388	0.00290	0.00327	0.00766	**0.00094**
85-bus	0.00531	0.00180	0.00206	0.00942	**0.00056**	0.00663	0.00221	0.00261	0.01377	**0.00075**
123-bus	0.00218	0.00160	0.00494	0.00348	**0.00018**	0.00255	0.00186	0.00579	0.00460	**0.00034**
141-bus	0.00152	0.00071	0.00326	0.00280	**0.00003**	0.00207	0.00099	0.00409	0.00453	**0.00004**

The best performing method (smallest loss function) is bolded for each test case. All values are in per unit.

1) Base Load Evaluation: We first assess the LinDistFlow approximation accuracy when using the optimized parameter values on the base loading scenarios in the test cases. Table I shows the OLDF performance metrics, notably the maximum and average voltage magnitude estimation errors ($\varepsilon_{\max}$ and ε_{avg}), compared to the alternative models PLPF, LoDF, LDF, and DLPF. As shown in this table, the OLDF model consistently outperforms its counterparts for all the test cases.

Detailing two examples, Fig. 6 shows the voltage profiles for the `85-bus` and `123-bus` test cases with the base case loading. While the voltages from the traditional LinDistFlow overestimate the true values, the optimized parameters result in a close alignment with the true DistFlow solution.

2) High Load Evaluation: Following the methodology described in [22], we generated high-load scenarios by scaling the base loads with a factor that ranges from $[-2,-1]\cup[1,2]$ at a granularity of $\frac{1}{14}$, yielding 30 distinct test scenarios. Table II presents a comparative performance analysis of various algorithms under these loading conditions. Our proposed OLDF algorithm consistently surpasses the others in reducing the average voltage estimation error (ε_{avg}) across nearly all test scenarios with the exception of the `22-bus` case. For this case, our OLDF results were better than all, but the PLPF approximation where the average error was still quite close (0.00028 for PLPF versus 0.00020 per unit for OLDF). Regarding the maximum errors in the high-load scenarios, no individual approximation consistently dominated the others across all test cases. However, we note that summing the maximum errors across all test cases reveals that the OLDF parameters lead to the best performance in aggregate for this metric. These results show that OLDF parameters trained with scenarios around base-load conditions nevertheless perform well for high-load conditions. Furthermore, an advantage that the OLDF has as an adaptive power flow approximation is the ability to tailor the parameters to perform even better for these conditions by including more training scenarios associated with high loading.

3) Random Load Evaluation: The OLDF approximation's accuracy is further analyzed for random loading conditions using $10,000$ scenarios generated from a uniform distribution within the range $(0, 1.5\cdot \mathrm{s}_{\mathrm{ref}})$, where $\mathrm{s}_{\mathrm{ref}}$ signifies the base load. Similar to the base load and high load conditions, the

TABLE II
MODEL EVALUATION - HIGH LOAD

Test case	ε_{avg}^{LDF}	ε_{avg}^{PLPF}	ε_{avg}^{LoDF}	ε_{avg}^{DLPF}	ε_{avg}^{OLDF}	$\varepsilon_{\max}^{LDF}$	$\varepsilon_{\max}^{PLPF}$	$\varepsilon_{\max}^{LoDF}$	$\varepsilon_{\max}^{DLPF}$	$\varepsilon_{\max}^{OLDF}$
22-bus	0.00050	**0.00020**	0.00376	0.00091	0.00028	0.00132	**0.00080**	0.00733	0.00280	0.00097
33-bus	0.00418	0.00239	0.00556	0.00795	**0.00224**	0.01573	**0.01018**	0.01713	0.03133	0.01254
69-bus	0.00254	0.00199	0.00278	0.00403	**0.00129**	0.02253	0.01975	0.01926	0.03929	**0.01572**
85-bus	0.01156	0.00817	0.01037	0.02055	**0.00719**	0.04700	0.03471	**0.02487**	0.08027	0.03373
123-bus	0.00464	0.00301	0.00900	0.00555	**0.00258**	0.01294	0.00800	0.02017	0.02173	**0.00766**
141-bus	0.00323	0.00234	0.00601	0.00608	**0.00182**	0.01070	**0.00648**	0.01501	0.02134	0.00802

The best performing method (smallest loss function) is bolded for each test case. All values are in per unit.

performance metrics for the random load conditions detailed in Table III also show the OLDF approximation's dominance, with this approximation having the smallest maximum and average voltage magnitude estimation errors ($\varepsilon_{\max}$ and ε_{avg}). Across all test cases, the OLDF accuracy improvement over traditional LDF ranges from 40% to 91.67% for average error (ε_{avg}) and from 53.67% to 87.78% for maximum error (ε_{max}). Compared to the best of PLPF and LoDF, OLDF's accuracy improvement ranges from 10% to 80% for average error and from 5.56% to 80.70% for maximum error.

E. Computational Efficiency

As shown in Fig. 2 and discussed in Section III, our algorithm computes LinDistFlow parameters during an offline phase where ample computing time is available. These parameters are then used in online applications where computing time may be limited. Thus, the training process for our proposed algorithm requires computational tractability consistent with an offline context. As shown by the training times in Table IV that range from 0.645 seconds for the 22-bus case to 2.694 seconds for the 141-bus case, leveraging mature optimization methods like TNC enables acceptable scalability for the training phase. Moreover, given the algorithm's similarity to training machine learning models, we anticipate that future implementations of our method will be able to exploit the rapid developments in both hardware and software acceleration for machine learning.

The computation times for online uses of the optimized LinDistFlow parameters depend on the particular application for which they are employed. However, since the only changes are to the parameter values and not the mathematical form of the LinDistFlow expressions, online computation times with our optimized parameters should be comparable to existing LinDistFlow approximations. To illustrate this, at 0.0004 to 0.0014 seconds, the average calculation times for the $10,000$ random load conditions for the experiment in Section IV-D3 were within 2% for the LinDistFlow with traditional parameters versus our optimized parameters.

F. Topology Analysis

Engineers may seek to optimize the topology of distribution systems for a variety of purposes including voltage management, loss minimization, and outage restoration [47]. Network reconfiguration problems inherently involve discrete decisions regarding the status of various switches. Linear power flow approximations like LinDistFlow are essential for obtaining tractable mixed-integer linear programming formulations for these and other problems. Furthermore, the adaptability of LinDistFlow parameters to topology changes is crucial. This ensures their effectiveness across various configurations without the need for retraining. Accordingly, this section next explores the performance of our proposed OLDF parameters with varying network topologies.

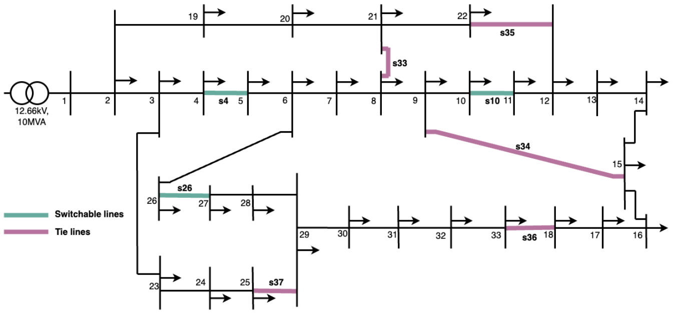


Fig. 7. The IEEE 33-bus distribution network with three switchable lines and five tie-lines.

Addressing topology changes with the LinDistFlow approximation can be approached in various ways. For example, the traditional LinDistFlow approximation allows for straightforward topology adjustments by updating the $\mathbf{D}_r$ and $\mathbf{D}_x$ matrices with new values corresponding to the altered topology to calculate voltages. Similarly, in our optimization-based algorithm, adjustments can be made by excluding optimized parameters for removed lines and incorporating original resistance and reactance values for new lines in $\mathbf{D}_r^{opt}$ and $\mathbf{D}_x^{opt}$, without altering the bias parameters (γ^{opt}, $\boldsymbol{\rho}^{opt}$, and $\boldsymbol{\varrho}^{opt}$). Successfully maintaining performance with this strategy indicates that our algorithm adapts well to different network topologies, avoiding overfitting to a specific configuration. Alternatively, optimizing parameters specifically for each topology through dedicated optimization could enhance accuracy but at the cost of increased computational time and storage for the additional parameters needed. Nevertheless, such calculations could be efficiently executed in parallel for a specified set of topologies, making this process suitable for high-performance computing environments, as each topology's optimization process operates independently from others.

To explore these different approaches, we next describe a small-scale experiment. Fig. 7 depicts the IEEE 33-bus distribution system, as described in [16], which has 33 nodes and 37 lines. As shown in Fig. 7, we consider a version of this system with eight switchable lines (i.e., lines 4, 10, and 26 as normally closed switches, NCS, and lines 33 to 37 as tie lines or normally open switches, NOS). With these switchable lines,

TABLE III
MODEL EVALUATION - RANDOM LOAD

Test case	ε_{avg}^{LDF}	ε_{avg}^{PLPF}	ε_{avg}^{LoDF}	ε_{avg}^{OLDF}	$\varepsilon_{\max}^{LDF}$	$\varepsilon_{\max}^{PLPF}$	$\varepsilon_{\max}^{LoDF}$	$\varepsilon_{\max}^{OLDF}$
22-bus	0.00024	0.00010	0.00220	**0.00002**	0.00090	0.00057	0.00477	**0.00011**
33-bus	0.00114	0.00051	0.00262	**0.00019**	0.00443	0.00312	0.00662	**0.00124**
69-bus	0.00077	0.00051	0.00101	**0.00017**	0.00918	0.00816	0.00476	**0.00347**
85-bus	0.00278	0.00056	0.00261	**0.00031**	0.00735	0.00321	0.00347	**0.00257**
123-bus	0.00120	0.00080	0.00392	**0.00072**	0.00300	0.00227	0.00624	**0.00139**
141-bus	0.00084	0.00035	0.00273	**0.00013**	0.00241	0.00108	0.00516	**0.00102**

The best performing method (smallest loss function) is bolded for each test case. All values are in per unit.

TABLE IV
COMPUTATION TIMES IN SECONDS

Test case	22-bus	33-bus	69-bus	85-bus	123-bus	141-bus
t_{train}	0.6451	0.7852	8.1701	2.7182	2.3093	2.6942
t_{base}	0.0071	0.0092	0.0161	0.0174	0.0112	0.0114
t_{10000}	0.0004	0.0005	0.0008	0.0009	0.0006	0.0014

TABLE V
TOPOLOGY CHANGES: OPENED AND CLOSED LINES

Opened	**Closed**	**Opened**	**Closed**	**Opened**	**Closed**
-	-	(4, 10)	(34, 35)	(10, 26)	(35, 36)
(4)	(33)	(4, 10)	(34, 37)	(10, 26)	(35, 37)
(4)	(35)	(4, 10)	(35, 36)	(10, 26)	(36, 37)
(4)	(37)	(4, 10)	(35, 37)	(4, 10, 26)	(33, 34, 35)
(10)	(34)	(4, 10)	(36, 37)	(4, 10, 26)	(33, 34, 36)
(10)	(35)	(4, 26)	(33, 36)	(4, 10, 26)	(33, 35, 36)
(10)	(36)	(4, 26)	(33, 37)	(4, 10, 26)	(33, 35, 37)
(26)	(36)	(4, 26)	(35, 36)	(4, 10, 26)	(33, 36, 37)
(26)	(37)	(4, 26)	(35, 37)	(4, 10, 26)	(34, 35, 36)
(4, 10)	(33, 34)	(4, 26)	(36, 37)	(4, 10, 26)	(34, 35, 37)
(4, 10)	(33, 35)	(10, 26)	(34, 35)	(4, 10, 26)	(34, 36, 37)
(4, 10)	(33, 36)	(10, 26)	(34, 36)		

These are all the valid switching combinations that lead to connected radial configurations for the 33-bus test case shown in Fig. 7. Note that the table indicates the changes from the configuration shown in Fig. 7.

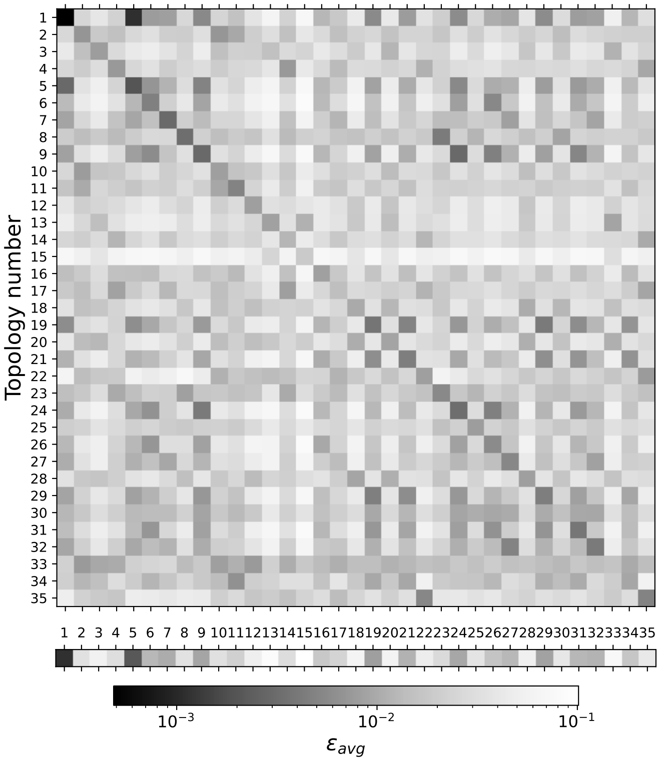


Fig. 8. Heatmap visualization of average error metric (ε_{avg}) across 35 topologies for the IEEE 33-bus test case, with a log-scaled color representation to highlight performance variations. The accompanying vector plot below the heatmap quantitatively assesses the LinDistFlow approximation's baseline performance across the same topologies, facilitating a direct comparison of adaptability and optimization effectiveness.

we can create 35 distinct and valid (i.e., radial and connected) topologies out of 56 possible topologies. Table V lists these valid topologies for the IEEE 33-bus network.

We next evaluate the adaptability of optimized parameters, i.e., assess how well parameters optimized for one topology perform in others. To accomplish this, we performed the offline training phase of our proposed algorithm using the same setup as before across the 35 network topologies to obtain a dataset with 35 sets of optimized coefficients and bias parameters. We tested the performance of these optimized parameters using the same $10,000$ test samples as before on each topology.

Illustrating this cross-topology assessment, Fig. 8 shows the performance of optimized parameters considering all 35 topologies. Specifically, the heatmap at the top of this figure shows the average error metric, ε_{avg}, of the optimized parameters for a given topology (rows) when applied to different topologies (columns). The horizontal vector plot at the bottom of the figure shows the performance of the traditional LinDistFlow approximation on the same 35 topologies. The heatmap employs a logarithmic scale for color representation to enable comparisons across a broad range of error magnitudes. The results underscore the variability in the algorithm's adaptability, with darker shades indicating lower errors (better performance) and lighter shades denoting higher errors. The vector plot beneath the heatmap contrasts the overall baseline performance of the traditional LinDistFlow approximation.

The matrix shown in Fig. 9 further illustrates the performance of our optimized parameters relative to the traditional LinDistFlow approximation (i.e., comparing each row from heatmap in Fig. 8 to the horizontal vector plot at the bottom of the figure). This matrix employs a binary color coding—green for topologies where the optimized parameters outperform the traditional LinDistFlow approximation and white where they do not. For example, row 1 in this matrix shows that training the parameters using the base topology results in the optimized parameters outperforming the traditional LinDistFlow approximation on topologies 1, 5, 6, 9, 19, 21, 24, 26, 29, and 31. Furthermore, if we train the parameters using topology 2, it will outperform the traditional LinDistFlow approximation on

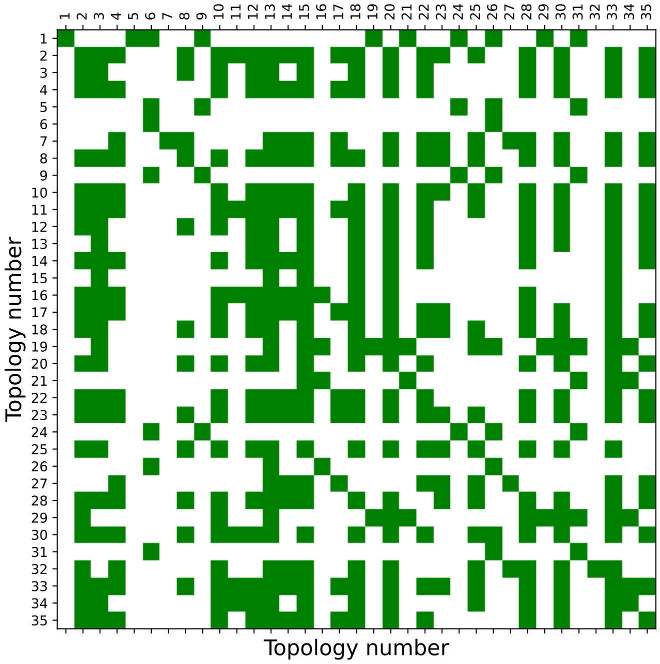


Fig. 9. Binary comparison matrix between the traditional LinDistFlow approximation's performance and optimized parameters from our proposed algorithm across 35 topologies in the `IEEE 33-bus` test case. Each column represents a topology, with green indicating that the optimized parameters for the topologay associated with the corresponding row outperform the traditional LinDistFLow approximation and white indicating that the traditional LinDistFlow approximation performance is better.

topologies $\{[2,4]\cup 8\cup[10,15]\cup[17,18]\cup 20,\cup[22,23]\cup 25\cup 28\cup 30\cup 33\cup 35\}$.

These visualizations collectively shed light on the nuanced performance of the optimized parameters vis-à-vis the traditional LinDistFlow approximation across diverse topologies. Our future work aims to characterize clusters of topologies for which jointly optimized parameter values can provide accurate LinDistFlow approximations.

G. Application to Hosting Capacity

As an illustrative example application, this section demonstrates the use of the proposed optimized LinDistFlow model to determine the hosting capacity of inverter-based generation units. Following [48], the hosting capacity problem is:

$$\min_{p_n, q_n} \quad \sum_{n\in\mathcal{N}'}\left(\frac{(\overline{p_n}-p_n)^2}{\overline{p_n}}+\xi\frac{q_n^2}{\overline{s_n}}\right) \tag{11a}$$

$$\text{s.t.} \quad 0\leq p_n\leq\overline{p_n}, \quad |q_n|\leq\sqrt{\overline{s_n}^2-p_n^2} \tag{11b}$$

$$v_0=1, \quad \underline{v_n}\leq v_n\leq\overline{v_n} \tag{11c}$$

$$\sqrt{P_n^2+Q_n^2}\leq S_n \tag{11d}$$

$$\sqrt{P_T^2+Q_T^2}\leq S_T \tag{11e}$$

$$P_T=\sum_{n:(0,n)\in\mathcal{E}}P_{0n}, \quad Q_T=\sum_{n:(0,n)\in\mathcal{E}}Q_{0n} \tag{11f}$$

$$\text{LDF (3) or OLDF (4)} \tag{11g}$$

where $\mathcal{N}'$ denotes the set of buses with inverter-based generators, p_n and q_n are the active and reactive power generation, $\overline{p_n}$, $\overline{s_n}$ are their maximum capacities, and ξ is a weighting factor controlling the tradeoff between active power and reactive power utilization. The constraints ensure adherence to power generation limits (11b) and voltage regulation requirements (11c) along with line (11d) and transformer capacities (11e)–(11f).

An evaluation using the `IEEE 33-bus` test system illustrates the OLDF model's effectiveness. We set $\overline{s_n}=0.6$ MVA with a 0.98 power factor, $\xi=0.02$, and voltage limits between 1.05 and 0.95 per unit, with a substation capacity of 10 MVA.

Upon solving (11) with both LDF and OLDF models, we obtain the optimal active and reactive power settings for the inverter-based generation units. By assessing these optimal settings using the original nonlinear DistFlow model, we compare the performance of the OLDF and LDF approximations. Fig. 10 shows that while the traditional LinDistFlow approximation leads to voltage violations at certain buses within the hosting capacity problem (11), the application of the proposed OLDF model avoids such violations.

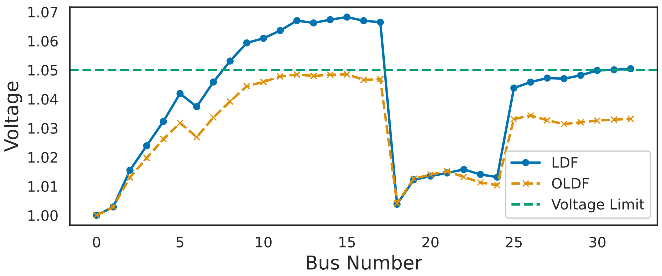


Fig. 10. Voltage profile of the `IEEE 33-bus` test case after feeding the active/reactive power injections obtained from solving (11) with LDF (blue curve) and OLDF (orange curve) into the DistFlow equations.

V. Conclusion

The LinDistFlow approximation is commonly used to improve the computational tractability of optimization problems for distribution systems. This paper presents a new algorithm that significantly improves the accuracy of the LinDistFlow approximation. Inspired by methods for training machine learning models, the offline phase of the proposed algorithm optimizes the LinDistFlow approximation's coefficient and bias parameters using analytically derived sensitivities within the TNC optimization method. These optimized parameters can then provide increased accuracy for many applications. The results from numerical testing empirically confirm the algorithm's effectiveness, demonstrating substantially better alignment with solutions to the nonlinear DistFlow equations compared to both the traditional LinDistFlow approximation as well as several recently proposed alternatives across a range of test cases and operating conditions. Application to a hosting capacity problem also illustrates the advantages of the proposed optimized parameters.

Since our optimized formulation has the same linear mathematical form as the traditional LinDistFlow approximation and performs well for varying topologies, the accuracy advantages of the proposed approach can be directly exploited in a wide range of other applications such as those in [14]–[16], [18], [19], [23]–[27]. Our future work aims to explore such applications. To enable broader applicability, our future work will also build on publications such as [17], [24] to adapt our algorithm to handle unbalanced three-phase distribution networks. While this will increase problem size and complexity, we do not anticipate any major conceptual challenges in performing this extension to unbalanced three-phase networks.

References

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

[2] D. Bienstock and A. Verma, "Strong NP-hardness of AC power flows feasibility," *Oper. Res. Lett.*, vol. 47, no. 6, pp. 494–501, 2019.

[3] I. A. Hiskens and R. J. Davy, "Exploring the power flow solution space boundary," *IEEE Trans. Power Syst.*, vol. 16, no. 3, pp. 389–395, Aug. 2001.

[4] D. K. Molzahn, "Computing the feasible spaces of optimal power flow problems," *IEEE Trans. Power Syst.*, vol. 32, no. 6, pp. 4752–4763, November 2017.

[5] K. Lehmann, A. Grastien, and P. Van Hentenryck, "AC-feasibility on tree networks is NP-hard," *IEEE Trans. Power Syst.*, vol. 31, no. 1, pp. 798–801, January 2016.

[6] T. J. Overbye, X. Cheng, and Y. Sun, "A comparison of the AC and DC power flow models for LMP calculations," in *37th Hawaii Int. Conf. Syst. Sci. (HICSS)*, 2004.

[7] S. Misra, L. A. Roald, M. Vuffray, and M. Chertkov, "Fast and robust determination of power system emergency control actions," in *IREP Symp. Bulk Power Syst. Dynamics Control–X*, August 2017.

[8] L. A. Roald, D. Pozo, A. Papavasiliou, D. K. Molzahn, J. Kazempour, and A. Conejo, "Power Systems Optimization under Uncertainty: A Review of Methods and Applications," *Electric Power Syst. Res.*, vol. 214, no. 108725, January 2023, presented at the *22nd Power Syst. Comput. Conf. (PSCC 2022).*

[9] C. Barrows, S. Blumsack, and P. Hines, "Correcting optimal transmission switching for AC power flows," in *47th Hawaii Int. Conf. Syst. Sci. (HICSS)*, Jan. 2014, pp. 2374–2379.

[10] A. Castillo, C. Laird, C. A. Silva-Monroy, J.-P. Watson, and R. P. O'Neill, "The unit commitment problem with AC optimal power flow constraints," *IEEE Trans. Power Syst.*, vol. 31, no. 6, pp. 4853–4866, 2016.

[11] B. Austgen, E. Kutanoglu, J. J. Hasenbein, and S. Santoso, "Comparisons of two-stage models for flood mitigation of electrical substations," *arXiv:2302.12872*, 2023.

[12] N. Rhodes, E. Haag, and L. Roald, "Long solution times or low solution quality: On trade-offs in choosing a power flow formulation for the optimal power shut-off problem," *arXiv:2310.13843*, 2023.

[13] D. K. Molzahn and I. A. Hiskens, "A survey of relaxations and approximations of the power flow equations," *Found. Trends Electr. Energy Syst.*, vol. 4, no. 1-2, pp. 1–221, 2019.

[14] M. Baran and F. Wu, "Optimal capacitor placement on radial distribution systems," *IEEE Trans. Power Del.*, vol. 4, no. 1, pp. 725–734, Jan 1989.

[15] ——, "Optimal sizing of capacitors placed on a radial distribution system," *IEEE Trans. Power Del.*, vol. 4, no. 1, pp. 735–743, Jan 1989.

[16] ——, "Network reconfiguration in distribution systems for loss reduction and load balancing," *IEEE Trans. Power Del.*, vol. 4, no. 2, pp. 1401–1407, April 1989.

[17] L. Gan and S. H. Low, "Convex relaxations and linear approximation for optimal power flow in multiphase radial networks," in *18th Power Syst. Comput. Conf. (PSCC)*, August 2014.

[18] V. Kekatos, L. Zhang, G. B. Giannakis, and R. Baldick, "Voltage regulation algorithms for multiphase power distribution grids," *IEEE Trans. Power Syst.*, vol. 31, no. 5, pp. 3913–3923, Sep 2016.

[19] B. A. Robbins and A. D. Domínguez-García, "Optimal reactive power dispatch for voltage regulation in unbalanced distribution systems," *IEEE Trans. Power Syst.*, vol. 31, no. 4, pp. 2903–2913, Jul 2016.

[20] Z. Yang, H. Zhong, A. Bose, T. Zheng, Q. Xia, and C. Kang, "A linearized OPF model with reactive power and voltage magnitude: A pathway to improve the MW-only DC OPF," *IEEE Trans. Power Syst.*, vol. 33, no. 2, pp. 1734–1745, 2017.

[21] E. Schweitzer, S. Saha, A. Scaglione, N. G. Johnson, and D. Arnold, "Lossy DistFlow formulation for single and multiphase radial feeders," *IEEE Trans. Power Syst.*, vol. 35, no. 3, pp. 1758–1768, May 2020.

[22] M. Marković and B.-M. Hodge, "Parameterized linear power flow for high fidelity voltage solutions in distribution systems," *IEEE Trans. Power Syst.*, vol. 38, no. 5, pp. 4391–4403, September 2023.

[23] K. Baker, A. Bernstein, E. Dall'Anese, and C. Zhao, "Network-cognizant voltage droop control for distribution grids," *IEEE Trans. Power Syst.*, vol. 33, no. 2, pp. 2098–2108, 2017.

[24] D. B. Arnold, M. Sankur, R. Dobbe, K. Brady, D. S. Callaway, and A. Von Meier, "Optimal dispatch of reactive power for voltage regulation and balancing in unbalanced distribution systems," in *IEEE Power Energy Soc. Gen. Meeting*, 2016.

[25] R. Mieth and Y. Dvorkin, "Data-driven distributionally robust optimal power flow for distribution systems," *IEEE Control Syst. Lett.*, vol. 2, no. 3, pp. 363–368, 2018.

[26] R. K. Gupta and D. K. Molzahn, "Improving fairness in photovoltaic curtailments via daily topology reconfiguration for voltage control in power distribution networks," *arXiv:2403.07853*, 2024.

[27] S. Bose, K. Chen, and Y. Zhang, "On LinDistFlow model congestion pricing: Bounding the changes in power tariffs," *arXiv:2305.00400*, 2023.

[28] X. Chen, W. Wu, and B. Zhang, "Robust capacity assessment of distributed generation in unbalanced distribution networks incorporating anm techniques," *IEEE Trans. Sustain. Energy*, vol. 9, no. 2, pp. 651–663, Apr 2018.

[29] S. Misra, D. K. Molzahn, and K. Dvijotham, "Optimal adaptive linearizations of the AC power flow equations," in *20th Power Syst. Comput. Conf. (PSCC)*, June 2018.

[30] T. Mühlpfordt, V. Hagenmeyer, D. K. Molzahn, and S. Misra, "Optimal adaptive power flow linearizations: Expected error minimization using polynomial chaos expansion," in *IEEE Milan PowerTech*, 2019.

[31] Y. Liu, N. Zhang, Y. Wang, J. Yang, and C. Kang, "Data-driven power flow linearization: A regression approach," *IEEE Trans. Smart Grid*, vol. 10, no. 3, pp. 2569–2580, May 2019.

[32] Y. Liu, Z. Li, and Y. Zhou, "Data-driven-aided linear three-phase power flow model for distribution power systems," *IEEE Trans. Power Syst.*, vol. 37, no. 4, pp. 2783–2795, Jul 2022.

[33] J. Chen and L. A. Roald, "Topology-adaptive piecewise linearization for three-phase power flow calculations in distribution grids," in *55th North American Power Symp. (NAPS)*, 2023.

[34] P. Buason, S. Misra, and D. K. Molzahn, "A Sample-Based Approach for Computing Conservative Linear Power Flow Approximations," *Electric Power Syst. Res.*, vol. 212, p. 108579, 2022, presented at the *22nd Power Syst. Comput. Conf. (PSCC 2022).*

[35] M. Jia and G. Hug, "Overview of data-driven power flow linearization," in *IEEE Belgrade PowerTech*, 2023.

[36] M. Jia, G. Hug, N. Zhang, Z. Wang, and Y. Wang, "Tutorial on data-driven power flow linearization–Part I: Challenges and training algorithms," preprint available at https://doi.org/10.3929/ethz-b-000606654, 2023.

[37] ——, "Tutorial on data-driven power flow linearization–Part II: Supportive techniques and experiments," preprint available at https://doi.org/10.3929/ethz-b-000606656, 2023.

[38] B. Stott, J. Jardim, and O. Alsaç, "DC power flow revisited," *IEEE Trans. Power Syst.*, vol. 24, no. 3, pp. 1290–1300, 2009.

[39] K. Dvijotham and D. K. Molzahn, "Error bounds on the DC power flow approximations: A convex relaxation approach," *IEEE 55th Conf. Decis. Control (CDC)*, December 2016.

[40] B. Taheri and D. K. Molzahn, "Optimizing parameters of the DC power flow," to appear in *Electric Power Syst. Res.,* to be presented at the *23rd Power Syst. Comput. Conf. (PSCC),* arXiv:2310.00447, 2024.

[41] S. H. Low, "Convex relaxation of optimal power flow–Part I: Formulations and equivalence," *IEEE Trans. Control Netw. Syst.*, vol. 1, no. 1, pp. 15–27, March 2014.

[42] J. Nocedal and S. Wright, *Numerical Optimization*. New York, NY: Springer Science & Business Media, 2006.

[43] S. G. Nash, "Newton-type minimization via the Lanczos method," *SIAM J. Numerical Analy.*, vol. 21, no. 4, pp. 770–788, 1984.

[44] S. Bolognani and S. Zampieri, "On the existence and linear approximation of the power flow solution in power distribution networks," *IEEE Trans. Power Syst.*, vol. 31, no. 1, pp. 163–172, 2015.

[45] 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 Trans. Power Syst.*, vol. 26, no. 1, pp. 12–19, 2011.

[46] C. Coffrin, R. Bent, K. Sundar, Y. Ng, and M. Lubin, "PowerModels.jl: An open-source framework for exploring power flow formulations," in *20th Power Syst. Comput. Conf. (PSCC)*, 2018.

[47] M. Mahdavi, H. H. Alhelou, N. D. Hatziargyriou, and F. Jurado, "Reconfiguration of electric power distribution systems: Comprehensive review and classification," *IEEE Access*, vol. 9, pp. 118 502–118 527, 2021.

[48] S. Zhan, J. Morren, W. van den Akker, A. van der Molen, N. G. Paterakis, and J. Slootweg, "Fairness-incorporated online feedback optimization for real-time distribution grid management," *IEEE Trans. Smart Grid*, vol. 15, no. 2, pp. 1792–1806, 2024.