On the Construction of Linear Approximations of Line Flow Constraints for AC Optimal Power Flow

Dmitry Shchetinin, *Student Member,*, Tomas Tinoco De Rubira, *Member,* Gabriela Hug, *Senior Member*

***Abstract*—The AC optimal power flow problem is a non-convex optimization problem that is difficult to solve quickly and reliably for large-scale grids. While various approximations have been proposed, they may lead to physically meaningless solutions. This paper presents a computationally efficient algorithm for constructing accurate linear approximations of line flow constraints. These approximations reduce the complexity of the optimization problem while ensuring that the solution is physically meaningful and has a high quality. The algorithm is based on an in-depth analysis of the feasible set of the line flow constraint. Numerical experiments are performed on ten large-scale systems using three nonlinear programming solvers. Obtained results indicate that the proposed formulation helps improve the solvers' reliability and reduce the computation time for hard large-scale problems. At the same time, the developed algorithm provides high quality approximations of the line flow constraints.**

***Index Terms*—convex approximation, linearization, optimal power flow, thermal limits**

I. Introduction

A notoriously difficult example of optimization problems in power system operation is the AC Optimal Power Flow (OPF) problem, which was first formulated in [1]. This problem is nonlinear, non-convex and has a large scale for realistic grids. Over the years, a number of studies have been dedicated to it, an overview of which can be found in [2], [3]. Despite advances in Nonlinear Programming (NLP) techniques, modern NLP solvers cannot always find a local optimum of the AC OPF problem in reasonable time [4], [5]. To enable the application of reliable solution algorithms, a number of approximations of this problem have been proposed.

One widespread approach is to use the so-called DC approximation [6]–[8]. It is based on a simplification of the steady-state system model and results in a convex optimization problem. However, the obtained solution is typically AC-infeasible due to inaccuracies introduced by linearization, which can be significant [9], [10]. While several studies have demonstrated that the solution quality can be improved by iterative linearization of the system model [11], [12], the accuracy of the DC OPF model remains inherently limited.

Another approach is based on convex relaxations [13]. Its main idea is to replace non-convex constraints with their relaxed convex approximations. The proposed types of relaxations include semi-definite programming [14], second-order cone [15] and quadratic convex relaxations [16]. This is a promising research direction because tight relaxations, which exist for certain test systems and grid topologies, enable the recovery of a globally optimal solution of the original problem [17], [18]. However, existing relaxations are not guaranteed to produce physically meaningful solutions for arbitrary power systems, although the relaxation quality can be improved by enforcing tighter bounds on variables [19], [20].

This paper employs a hybrid formulation of the AC OPF problem, wherein only the constraints on the maximum line flows are linearized. As the power balance constraints are kept in their original form, the solution is guaranteed to be physically meaningful. Since nonlinear line flow constraints can increase the computation time of NLP solvers [21], replacing them with linear approximations can potentially improve the solvers' performance. Existing endeavors to approximate line flow constraints have been largely focused on the DC formulation. While some of them model the impact of reactive power flows [22]–[25], their quality suffers from the assumption of constant bus voltage magnitudes. In [26], we proposed an algorithm for constructing high-quality conservative linear approximations to the line flow constraints. The results indicated that using them instead of the original constraints lead to a faster solution of the AC OPF problem.

The main contribution of this paper is threefold. First, it presents an in-depth analysis of the feasible set of a line flow constraint in polar coordinates. A closed-form expression of the set's boundary and the properties of its curvature are derived, which provides a theoretical foundation for the proposed approximation algorithm. Second, the paper extends our previous work by presenting an efficient algorithm for constructing both relaxed and conservative approximations of the line flow constraints. In addition, the approximation quality for a given number of linear constraints is improved by iterative refinement. The algorithm is made available in the form of an open-source library. Lastly, the paper presents an extensive analysis of the accuracy of the approximations and their impact on the solution of the AC OPF problem. The obtained results show that for hard large-scale test cases the linearization of line flow constraints improves the performance and robustness of the NLP solvers IPOPT, SNOPT, and KNITRO.

The rest of the paper is structured as follows. Section II presents a formulation of the AC OPF problem. Section III analyzes the feasible set of the line flow constraint and Section IV describes the proposed algorithm for constructing its linear approximation. Section V covers the details of the algorithm's implementation. Section VI presents the results of numerical experiments and Section VII provides the summary.

II. AC OPF Formulation

The goal of the AC OPF problem is to find an operating point that satisfies physical and operational constraints and is optimal with respect to a given objective. This study considers the minimization of the total supply cost in the system. The supply cost of the i-th generator is modeled as a quadratic function of its active power output P_{G_i}:

$$C_i(P_{G_i}) := c_{2_i} P_{G_i}^2 + c_{1_i} P_{G_i} + c_{0_i}, \tag{1}$$

where $c_{2_i} \geq 0, c_{1_i}, c_{0_i}$ are machine-specific cost coefficients.

In this paper, the considered decision variables for the AC OPF problem include only active P_G and reactive Q_G power outputs of generators. The dependent variables are the complex bus voltages, which are represented in a polar form by their magnitudes V and phase angles θ.

In order to ensure that the AC OPF solution satisfies Kirchhoff's laws, nodal power balance equations are included in the problem in the form of equality constraints. For bus i, active and reactive power mismatches W_{P_i} and W_{Q_i}, which should be zero at the solution, are given by:

$$W_{P_i} = P_{G_i} - P_{D_i} - \sum_{k \in \Omega_i} V_i V_k (G_{ik} \cos\theta_{ik} + B_{ik} \sin\theta_{ik}) \quad (2)$$

$$W_{Q_i} = Q_{G_i} - Q_{D_i} - \sum_{k \in \Omega_i} V_i V_k (G_{ik} \sin\theta_{ik} - B_{ik} \cos\theta_{ik}) \quad (3)$$

where set Ω_i includes bus i and all buses connected to it, P_{D_i} and Q_{D_i} are active and reactive power demands at bus i, G_{ik} and B_{ik} are the real and imaginary parts of the corresponding element of the admittance matrix, and θ_{ik} is the phase angle difference between the voltages at buses i and k.

The optimization problem also contains several types of inequality constraints. Bus voltage magnitudes and generator outputs have upper and lower bounds dictated by physical properties of the equipment and stability issues. Line flows have so-called thermal limits based on the maximum allowed temperature of the wires. This study models these limits by imposing bounds on the magnitudes I of line currents, which is what is typically done by system operators.

The considered AC OPF problem is formulated as follows:

$$\begin{aligned}
&\underset{P_G, Q_G, V, \theta}{\text{minimize}} && \sum_{i \in \mathcal{G}} C_i(P_{G_i}) && && (4a)\\
&\text{subject to} && W_{P_i} = 0, && i \in \mathcal{N} && (4b)\\
& && W_{Q_i} = 0, && i \in \mathcal{N} && (4c)\\
& && P_{G_i}^{\min} \leq P_{G_i} \leq P_{G_i}^{\max}, && i \in \mathcal{G} && (4d)\\
& && Q_{G_i}^{\min} \leq Q_{G_i} \leq Q_{G_i}^{\max}, && i \in \mathcal{G} && (4e)\\
& && V_i^{\min} \leq V_i \leq V_i^{\max}, && i \in \mathcal{N} && (4f)\\
& && I_i \leq I_i^{\max}, && i \in \mathcal{L}, && (4g)
\end{aligned}$$

where $\mathcal{N}$, $\mathcal{L}$, and $\mathcal{G}$ are the sets of all buses, lines, and generators, respectively. Problem (4a)-(4g) is non-convex because of the nonlinear equality constraints and non-convex constraints on line flows. To ensure that the OPF solution is AC-feasible, the power balance equations have to be kept in their original form. On the other hand, since thermal limits are represented by inequality constraints, one can construct their inner (conservative) and outer (relaxed) convex approximations. In both cases, the approximation error is bounded away from zero because of the non-convexity of the constraint's feasible set. If this non-convexity is relatively small, it is possible to produce a high-quality approximation. Replacing (4g) with its approximation can reduce the complexity of the optimization problem, which can in turn lead to a faster solution of the optimization problem. For the inner approximation, the solution is guaranteed to be physically and operationally feasible. If the solution algorithm fails to find a feasible point, the problem can be solved again using the outer approximation. This can help establish whether the original problem is infeasible or the infeasibility occurred due to the approximation error.

III. Feasible Set of a Line Flow Constraint

To construct a linear approximation of the line flow constraint, it is essential to investigate the properties of its feasible set. Consider a branch between buses i and j represented by a π-model with admittance $G + jB$ and shunt susceptance B_s. For simplicity, all formulations will be given for transmission lines but they can be easily extended to transformers. The line current is a function of the complex voltages at both ends of the line. The current magnitudes at the beginning and end of the line, denoted by I_{ij} and I_{ji}, are given by:

$$\begin{aligned} I_{ij}(V_i, V_i, \theta_{ij}) = \big(&a_1 V_i^2 + a_2 V_j^2 + \\ &2 V_i V_j \left(a_3 \sin\theta_{ij} - a_4 \cos\theta_{ij}\right)\big)^{1/2}, \end{aligned} \quad (5)$$

$$\begin{aligned} I_{ji}(V_i, V_i, \theta_{ij}) = \big(&a_2 V_i^2 + a_1 V_j^2 - \\ &2 V_i V_j \left(a_3 \sin\theta_{ij} + a_4 \cos\theta_{ij}\right)\big)^{1/2}, \end{aligned} \quad (6)$$

where $a_1 := G^2 + (B + B_s)^2$, $a_2 := G^2 + B^2$, $a_3 := B_s G$, and $a_4 := G^2 + B^2 + B_s B$. To ensure secure operation of the grid, both values are constrained to be below a given limit:

$$I_{ij}(V_i, V_j, \theta_{ij}) \leq I_{ij}^{\max}, \quad (7)$$

$$I_{ji}(V_i, V_j, \theta_{ij}) \leq I_{ji}^{\max}, \quad (8)$$

where $I_{ij}^{\max}$ is equal to $I_{ji}^{\max}$. These two inequalities correspond to constraint (4g) in the AC OPF problem formulation. For the sake of simplicity, this paper only presents the analysis of (7).

The feasible set of constraint (7), denoted by $\mathcal{I}$, is defined as $\mathcal{I} := \left\{(V_i, V_j, \theta_{ij}) \in \mathbb{R}_+^2 \times \mathbb{R} \mid I_{ij}(V_i, V_j, \theta_{ij}) \leq I_{ij}^{\max}\right\}$. The set's boundary is a surface that consists of all points for which the constraint becomes binding. Therefore, the boundary surface is implicitly given by

$$I_{ij}(V_i, V_i, \theta_{ij}) = I_{ij}^{\max}. \quad (9)$$

Figure 1 shows an example of this boundary surface. Let $\bar{\theta}_{ij}$ denote the value of the phase angle difference for a point at the boundary. From (9), $\bar{\theta}_{ij}$ can be expressed using V_i and V_j:

$$\bar{\theta}_{ij}(V_i, V_j) = \arcsin\left(\frac{-a_3 t \pm |a_4| \sqrt{4 a_1 a_2 - t^2}}{2 a_1 a_2}\right), \quad (10)$$

where $t = a_1 V_i / V_j + a_2 V_j / V_i - (I_{ij}^{\max})^2 / (V_i V_j)$. Note that since the range of arcsin is $[-\pi/2, \pi/2]$, (9) and (10) are equivalent only for $|\bar{\theta}_{ij}| \leq \pi/2$. Given that in steady-state operation θ_{ij} cannot exceed $\pi/2$, the boundary surface of the feasible set of constraint (7) can be represented as $\left\{(V_i, V_j, \bar{\theta}_{ij}(V_i, V_j)) \mid (V_i, V_j) \in \mathbb{R}_+^2\right\}$. The properties of the feasible set can be studied by analyzing (10) together with (7).

Equation (10) has (a) real-valued solution(s) if the argument of arcsin belongs to the domain of reals and its magnitude does not exceed one. Since $a_3^2 + a_4^2 = a_1 a_2$, it can be shown that the magnitude of the arcsin argument in (10) cannot exceed one. Therefore, a real-valued solution of (10) exists if $4 a_1 a_2 - t^2 \geq 0$, which yields the following system of inequalities:

$$V_j \leq \alpha V_i + \beta, \quad (11)$$

$$V_j \geq \alpha V_i - \beta, \quad (12)$$

$$\alpha V_i + V_j \geq \beta, \quad (13)$$

where $\alpha := \sqrt{a_1 / a_2}$, $\beta := I_{ij}^{\max} / \sqrt{a_2}$. The relation between (11)-(13) and the solutions of (10) is discussed below.

1) Equation (10) *has two solutions:* This happens when $4 a_1 a_2 - t^2 > 0$, which means that inequalities (11)-(13) are strictly satisfied. Then for a given pair of (V_i, V_j), equation

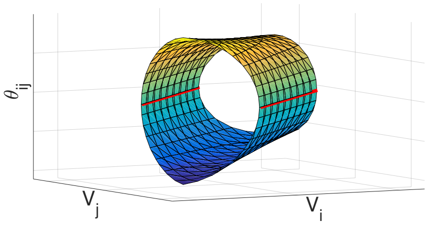


Fig. 1. Boundary of feasible set of line flow constraint

(10) has two solutions denoted by $\bar{\theta}_{ij}^-$ and $\bar{\theta}_{ij}^+$. According to the orientation of the axes in Figure 1, $\bar{\theta}_{ij}^-$ and $\bar{\theta}_{ij}^+$ correspond to the lower and upper parts of the boundary surface, respectively. This convention is used in the rest of the paper. Since for realistic transmission lines and transformers a_3 is small, $\bar{\theta}_{ij}^- \approx -\bar{\theta}_{ij}^+$. To establish whether the feasible set of (7) lies inside or outside of the surface depicted in Figure 1, one can examine $\partial I_{ij}/\partial\theta_{ij}$ and $\partial^2 I_{ij}/\partial\theta_{ij}^2$. The analysis reveals that for any (V_i, V_j) the value of I_{ij} reaches its minimum for $\theta_{ij}^{\min} = -\arcsin\left(a_3/\sqrt{a_1 a_2}\right)$ and its maximum for $\theta_{ij}^L = -\pi + \theta_{ij}^{\min}$ and $\theta_{ij}^U = \pi - \theta_{ij}^{\min}$. Moreover, for any (V_i, V_j) the value of I_{ij} increases monotonically as θ_{ij} moves from $\theta_{ij}^{\min}$ towards either θ_{ij}^L or θ_{ij}^U and the following holds:

$$\theta_{ij}^L \le \bar{\theta}_{ij}^- \le \theta_{ij}^{\min} \le \bar{\theta}_{ij}^+ \le \theta_{ij}^U. \tag{14}$$

This means that for a given pair of (V_i, V_j) constraint (7) is satisfied for all values of θ_{ij} in $\bar{\theta}_{ij}^- \le \theta_{ij} \le \bar{\theta}_{ij}^+$. Hence, the feasible set of (7) lies inside the surface.

2) Equation (10) *has one solution:* This happens when $4a_1a_2 - t^2 = 0$, which means that one of inequalities (11)-(13) becomes binding. If either (11) or (12) is binding, solving (10) yields $\bar{\theta}_{ij}^- = \bar{\theta}_{ij}^+ = \theta_{ij}^{\min}$. Therefore, the upper and lower parts of the boundary surface intersect along two parallel horizontal lines, which are shown in Figure 1 in red and are referred to as the *boundary lines*. If (13) is binding, equations (9) and (10) are not equivalent because (9) is satisfied by $\theta_{ij} = \theta_{ij}^L$ or $\theta_{ij} = \theta_{ij}^U$, whose magnitudes are greater than $\pi/2$. Since in practice $|\theta_{ij}| \le \pi/2$, inequality (13) cannot become binding.

3) Equation (10) *has no solution:* This happens when $4a_1a_2 - t^2 < 0$, which corresponds to the violation of at least one of inequalities (11)-(13). If (11) or (12) is violated for a given pair of (V_i, V_j), $I_{ij}(V_i, V_j, \theta_{ij})$ exceeds $I_{ij}^{\max}$ for any θ_{ij}. Therefore, this pair of (V_i, V_j) does not belong to the feasible set of constraint (7). If a given pair of (V_i, V_j) violates (13), constraint (7) is strictly satisfied for any value of θ_{ij}.

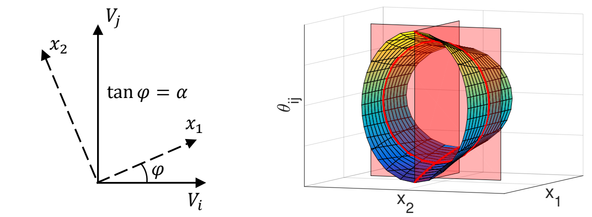
So far it was shown that the feasible set of (7) is contained between (11) and (12). Next, the curvature of its boundary surface is investigated. Consider a new coordinate system wherein the axes in the (V_i, V_j) plane are rotated counterclockwise by $\varphi = \arctan\alpha$. Let x_1 and x_2 denote the new axes (see Figure 2a). Since $\alpha := \sqrt{a_1/a_2}$, the old and new axes are related through:

$$\begin{aligned} \sqrt{a_1 + a_2}V_i &= \sqrt{a_2}x_1 - \sqrt{a_1}x_2, \\ \sqrt{a_1 + a_2}V_j &= \sqrt{a_1}x_1 + \sqrt{a_2}x_2. \end{aligned} \tag{15}$$

In the new coordinate system, (7) can be expressed as:

$$I_{ij}(x_1, x_2, \theta_{ij}) \le I_{ij}^{\max}. \tag{16}$$

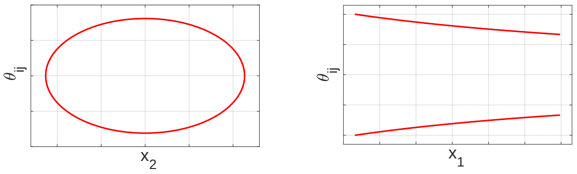
Figure 2b shows the boundary surface of the feasible set of (16), which is simply a rotated boundary surface of (7). As follows from (15) and (11)-(12), the boundary lines of the feasible set of (16) are parallel to the x_1 axis. Moreover, the projection of the feasible set of (16) onto the (x_1, x_2) plane, which is denoted by $\mathcal{I}_x$, is bounded by $|x_2| \le \beta/\sqrt{2}$. Consider the cross-sections of the boundary surface along the x_1 and x_2 axes as shown in Figure 2b. The resulting curves are visualized in Figure 3, which shows that for the chosen cross-sections the curvature of the boundary surface is such that the feasible set of (16) is convex along the x_2 axis and non-convex along the x_1 axis. To prove that the boundary has this curvature for all cross-sections along x_1 and x_2 axis, one has to show that:



(a) Rotation of axes

(b) Boundary surface

Fig. 2. Boundary surface in rotated coordinate system



(a) Along x_2 axis

(b) Along x_1 axis

Fig. 3. Cross-section of boundary surface along a given axis

$$\frac{\partial^2 \bar{\theta}_{ij}^+}{\partial x_1^2} \ge 0, \quad \frac{\partial^2 \bar{\theta}_{ij}^-}{\partial x_1^2} \le 0, \quad \forall (x_1, x_2) \in \mathcal{I}_x, \tag{17}$$

$$\frac{\partial^2 \bar{\theta}_{ij}^+}{\partial x_2^2} \le 0, \quad \frac{\partial^2 \bar{\theta}_{ij}^-}{\partial x_2^2} \ge 0, \quad \forall (x_1, x_2) \in \mathcal{I}_x. \tag{18}$$

For branches with $B_s = 0$, it is trivial to formally prove (17) and (18). In case of $B_s \ne 0$, the formal proof is difficult to obtain due to the complexity of the resulting expressions. However, extensive numerical experiments showed that (17) and (18) hold for all realistic lines and operating conditions.

The obtained properties of the feasible set of (7) will be used for constructing its inner and outer linear approximations. Since the feasible set is non-convex, it is impossible to approximate it arbitrarily well. However, it is clear that the tighter the bounds on V_i and V_j are, the better approximation quality can be achieved. While in the analysis above no bounds on V_i and V_j were considered, in reality the bounds on bus voltage magnitudes always exist and are enforced by system operators. In the AC OPF formulation, these bounds are represented by constraint (4f). Therefore, one should consider the intersection of $\mathcal{I}$ with the feasible set of (4f) denoted by $\mathcal{V}$ when constructing approximations of the line flow constraints.

IV. Proposed Approximation Algorithm

A. Algorithm Overview

The goal of the proposed algorithm is to construct a linear approximation of the line flow constraint (7) efficiently with the desired accuracy using as few linear constraints as possible. Its main inputs are the approximation type (inner or outer), maximum approximation error in the current magnitude $\varepsilon^{\max}$, and maximum number of linear constraints $N^{\max}$. Let $\mathcal{S} := \mathcal{I} \cap \mathcal{V}$ denote the intersection of the feasible sets of (7) with (4f), and $x := [V_i, V_j, \theta_{ij}]^T$. The algorithm constructs

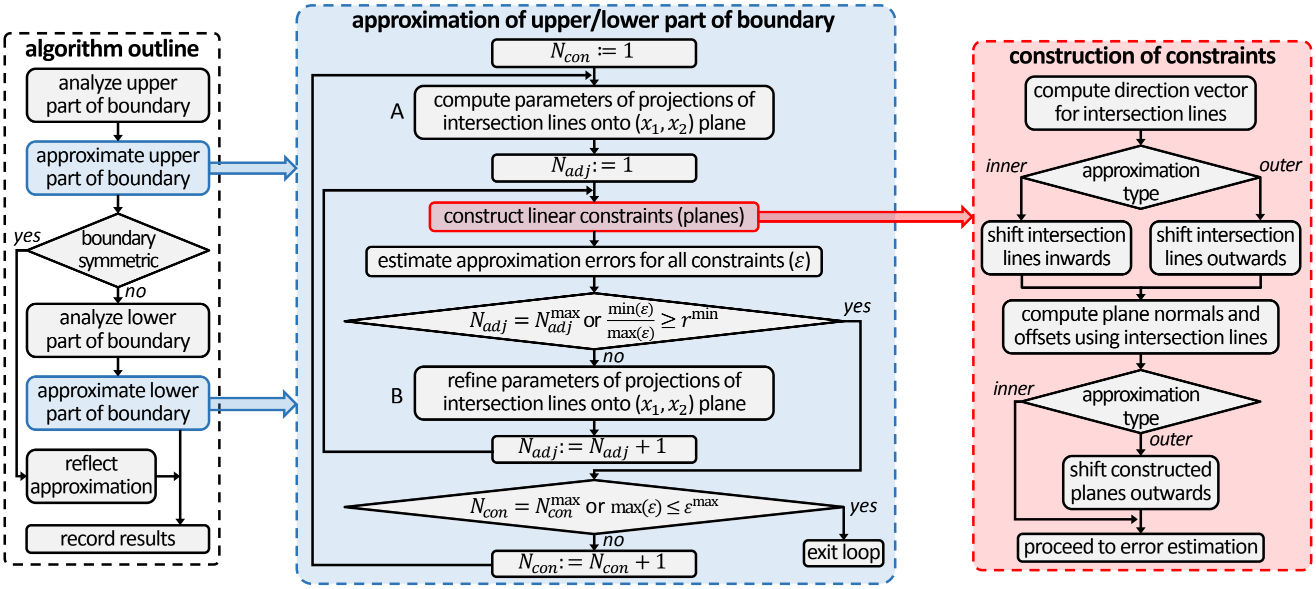

algorithm outline
analyze upper part of boundary
approximate upper part of boundary
boundary symmetric
yes
no
analyze lower part of boundary
approximate lower part of boundary
reflect approximation
record results
approximation of upper/lower part of boundary
$N_{con} := 1$
A
compute parameters of projections of intersection lines onto (x_1, x_2) plane
$N_{adj} := 1$
construct linear constraints (planes)
estimate approximation errors for all constraints (ε)
$N_{adj} = N_{adj}^{\max}$ or $\frac{\min(\varepsilon)}{\max(\varepsilon)} \geq r^{\min}$
yes
no
B
refine parameters of projections of intersection lines onto (x_1, x_2) plane
$N_{adj} := N_{adj} + 1$
$N_{con} = N_{con}^{\max}$ or $\max(\varepsilon) \leq \varepsilon^{\max}$
yes
no
exit loop
$N_{con} := N_{con} + 1$
construction of constraints
compute direction vector for intersection lines
inner
approximation type
outer
shift intersection lines inwards
shift intersection lines outwards
compute plane normals and offsets using intersection lines
inner
approximation type
outer
shift constructed planes outwards
proceed to error estimation

Fig. 4. Outline of the proposed approximation algorithm

a polyhedron $\mathcal{P} := \{x \,|\, Ax \leq b\}$, where $A \in \mathbb{R}^{m\times 3}$ and $b \in \mathbb{R}^m$, such that $m \leq N^{\max}$, $(\mathcal{P} \cap \mathcal{V}) \subseteq \mathcal{S}$ for the inner and $(\mathcal{P} \cap \mathcal{V}) \supseteq \mathcal{S}$ for the outer approximation. The maximum relative approximation error ρ in the current magnitude is:

$$\rho := \max_{x\in\text{boundary}(\mathcal{P})\cap\mathcal{V}} \left|I_{ij}\left(x\right) - I_{ij}^{\max}\right| / I_{ij}^{\max} \qquad (19)$$

Ideally, the number of constructed linear constraints m should be the minimum needed to ensure that $\rho \leq \varepsilon^{\max}$. However, computing ρ as well as obtaining the minimum required m are difficult tasks. Since this study focuses on the efficiency of the approximation algorithm, a computationally cheap estimate of ρ is used and the obtained m may not be the minimum possible although the algorithm attempts to minimize it.

Figure 4 shows the outline of the proposed algorithm, which relies on the properties of the feasible set of (7) to ensure that the approximation is conservative/relaxed. Its main idea is to first construct the edges of $\mathcal{P}$, which represent the intersections of planes that form $\mathcal{P}$ and are referred to as *intersection lines*, and then compute the values of A and b. The projections of these edges onto the (x_1, x_2) plane are parallel to the x_1 axis. The rows of matrix A represent the normals of the approximation planes, and the entries of vector b are proportional to the signed distances from the planes to the origin. For brevity, the entries of b are called plane offsets.

The algorithm first analyzes the upper part of the boundary surface and constructs its approximation. As can be seen from (10), if the value of $a_3/(2a_1a_2)$ is small, $\bar{\theta}_{ij}^{+} \approx -\bar{\theta}_{ij}^{-}$. Hence, the surface is almost symmetric about the θ_{ij} axis, in which case the approximation of its lower part can be obtained by reflecting the approximation of its upper part. Otherwise, the lower part of the surface is approximated using the same algorithm as the one used for approximating its upper part.

The algorithm for approximating the upper/lower part of the boundary surface consists of two nested loops. The outer loop iterates over the number of linear constraints N_{con} starting from $N_{con} = 1$. At each iteration, the approximation with N_{con} linear constraints is constructed and the maximum approximation error in terms of the current magnitude is estimated for each constraint. Let $\varepsilon \in \mathbb{R}^{N_{con}}$ denote the vector of the obtained approximation errors associated with individual linear constraints at a given iteration. The loop continues until either $\max_i \varepsilon_i \leq \varepsilon^{\max}$ or $N_{con} = N_{con}^{\max}$, where $N_{con}^{\max} = \lfloor N^{\max}/2 \rfloor$. In addition, the loop terminates when the change in $\max_i \varepsilon_i$ at two consecutive iterations is smaller than a predefined value. Thus, the goal of the outer loop is to achieve the desired approximation accuracy with the fewest possible number of linear constraints.

The goal of the inner loop is to improve the approximation accuracy for a given value of N_{con} by iteratively reducing $||\varepsilon||_\infty$, which defines the overall approximation error. This is done by reducing the standard deviation of entries of ε, i.e. making them more uniform. To achieve this, at each iteration the projections of the intersection lines onto the (x_1, x_2) plane are adjusted based on the value of ε at the previous iteration. The loop continues until the number of adjustments N_{adj} reaches a given maximum $N_{adj}^{\max}$, or $\min_i \varepsilon_i / \max_i \varepsilon_i$ exceeds $r^{\min}$, which is an algorithm parameter.

This algorithm can be applied for approximating the line flow constraints at both ends of the branch. Having obtained these two approximations, one can retain only those linear constraints that are relevant for the intersection of feasible sets of (7) and (8). This helps reduce the number of required linear constraints per nonlinear constraint by half without sacrificing the approximation accuracy. A more detailed description of the algorithm's main steps is given below.

B. Analysis of Boundary

Prior to constructing the upper/lower part of the approximation, set $\mathcal{S}$ and its boundary are analyzed. The approximation should only be constructed if $\mathcal{S} \neq \emptyset$ and (7) can be violated for at least one point (V_i, V_j, θ_{ij}) that satisfies (4f) and $|\theta_{ij}| \leq \pi/2$. To check whether $\mathcal{S}$ is empty, it suffices to check if there exists a point $(V_i, V_j) \in \mathcal{V}$ that satisfies (11)-(12). If it does not exist, $\mathcal{S} = \emptyset$ and problem (4) is infeasible. Otherwise, the algorithm checks whether (7) can be violated. Due to the shape of $\mathcal{S}$, it is enough to check this for only four points that correspond to the vertices of $\mathcal{V}$ and have $\theta_{ij} = \pi/2$ (for the upper part of the approximation) or $\theta_{ij} = -\pi/2$ (for the lower part). If (7) is satisfied for these four points, it cannot become binding and the approximation is not constructed. Otherwise, the algorithm determines the minimum $x_2^{\min}$ and maximum $x_2^{\max}$ values of x_2 for the points that belong to the projection of $\mathcal{S}$ onto the (x_1, x_2) plane. These values, shown in Figure 5b, are used for computing the intersection lines' projections.

C. Calculation of Projections of Intersection Lines

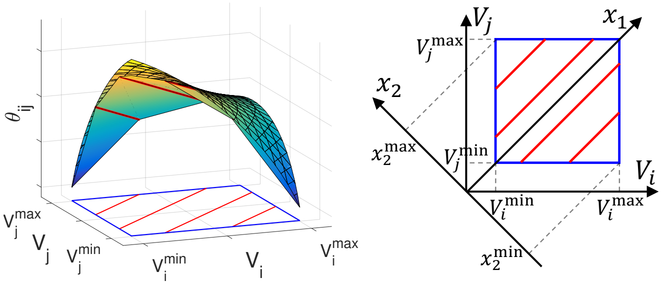
The *intersection lines* represent the edges of polyhedron $\mathcal{P}$, which approximates the nonlinear set $\mathcal{S}$. Figure 5 shows in red an example of the intersection lines for the upper part of the inner approximation and their projections onto the (x_1, x_2) plane. The projections are chosen to be parallel to the x_1 axis, which is done for two reasons. First, as will be explained later it is used to ensure that the constructed approximation is indeed conservative/relaxed. Second, it helps to estimate and reduce the approximation error efficiently. Recall that $\mathcal{S}$ is convex along the x_2 axis and non-convex along the x_1 axis, as shown in Figure 3. Because of the non-convexity, it cannot be approximated arbitrarily well with $\mathcal{P}$, regardless of the value of N_{con}. However, one can eliminate the part of the approximation error caused by the nonlinearity of the boundary of $\mathcal{S}$ along the x_2 axis. This can be achieved by making the projections of the intersection lines onto the (x_1, x_2) plane parallel to the x_1 axis and setting $N_{con} \to \infty$. Then the resulting approximation error is the minimum possible because it is caused purely by the non-convexity of $\mathcal{S}$.

In practice, it is important to achieve the desired approximation accuracy with the minimum value of N_{con} to reduce the size of the resulting AC OPF problem. The proposed algorithm attempts to accomplish this by iteratively increasing the value of N_{con} while making the errors associated with the individual constraints close to each other for each N_{con}. The errors are equalized by iteratively adjusting the intersection lines' projections onto the (x_1, x_2) plane. Initially, these projections are spaced evenly from $x_2^{\min}$ to $x_2^{\max}$ as shown in Figure 5. After each adjustment, the approximation is constructed and the vector ε of the error estimates associated with individual constraints is obtained. Let $d_i^{(k)}$ denote the distance between the x_2 coordinates of the projections of two adjacent intersection lines belonging to the i-th constraint after k adjustments. The value of $d_i^{(k+1)}$ is obtained using the following update:

$$d_i^{(k+1)} := d_i^{(k)} \frac{\left(x_2^{\max} - x_2^{\min}\right)\sqrt{\bar{\varepsilon}/\varepsilon_i}}{\sum_{j=1}^{N_{con}} d_j^{(k)} \sqrt{\bar{\varepsilon}/\varepsilon_j}}, \tag{20}$$

where $\bar{\varepsilon}$ is the mean value of the elements of ε. Such an update helps redistribute the errors more evenly by penalizing deviations from $\bar{\varepsilon}$. If $\varepsilon_i < \bar{\varepsilon}$, $d_i^{(k+1)} > d_i^{(k)}$ and vice versa. This procedure corresponds to block B in Figure 4.

The addition of a new constraint to the approximation corresponds to increasing the number of intersection lines by one. Therefore, the projections of the intersection lines have to be adjusted again to equalize the elements of ε. To reduce the number of such adjustments, the algorithm leverages information about the projections computed for $N_{con} - 1$ instead of spacing the new projections evenly from $x_2^{\min}$ to $x_2^{\max}$. The detailed description of this procedure, which corresponds to block A in Figure 4, is omitted due to space limitations.



(a) Intersection lines (b) Projection on (x_1, x_2) plane

Fig. 5. Intersection lines and their projections onto (x_1, x_2) plane

D. Construction of Linear Constraints

To construct linear constraints, first the parameters of the intersection lines are obtained. Since the projections of these lines onto the (x_1, x_2) plane are parallel to the x_1 axis and any two adjacent lines lie in the same plane, all intersection lines have the same direction vector. To obtain this vector, one has to select the slope of its projection onto the (x_1, θ_{ij}) plane because its projection onto the (x_1, x_2) plane is known. The slope is selected such that the approximation error is reduced.

Consider an individual intersection line and the curve obtained by intersecting the upper/lower part of the boundary surface with a vertical plane containing this line. An example of such a curve is shown in Figure 3b. To reduce the approximation error, the line has to serve as a good linear approximation of the curve. This can be achieved by making the intersection line parallel to the line connecting the end points of the curve, which lie on the edges of $\mathcal{V}$. For different intersection lines, the slopes of the lines connecting the ends of the corresponding curves are different. Therefore, the slope of the direction vector's projection onto the (x_1, θ_{ij}) plane is selected to be the average of these slopes.

To uniquely define an intersection line, one has to determine a point on it. Clearly, the segments of these lines that belong to $\mathcal{V}$ must lie inside $\mathcal{S}$ in the case of inner approximation and outside it for the outer approximation. To ensure that the approximation is as tight as possible, the line must touch the surface at a point in $\mathcal{V}$. This point is obtained by solving a one-variable equation with the Newton-Raphson method. If it fails, the bisection method is used. Both methods are computationally efficient since the problem has a single variable.

The algorithm then computes the plane normals and offsets from the intersection lines. For the inner approximation, the planes are guaranteed to be inside $\mathcal{S}$ as long as all intersection lines are inside it. For the outer approximation, it does not suffice to ensure that the intersection lines are outside of $\mathcal{S}$. Therefore, the planes are shifted outwards without changing their normals. Due to the properties of $\mathcal{S}$, the planes are guaranteed to be outside $\mathcal{S}$ if they are outside it at the edges of $\mathcal{V}$, i.e. when (4f) is binding. To reduce the approximation error, each plane must touch the boundary of $\mathcal{S}$ at one point, which is obtained by solving a one-variable problem with the Newton-Raphson or bisection methods. These points are then used to recompute the plane offsets.

In certain cases, extra-steps are added to this algorithm in order to increase the approximation accuracy. For the sake of saving space, these steps are not described here.

E. Error Estimation

The approximation error is used as the stopping criterion in the algorithm. To increase the computational efficiency, the true error defined in (19) is replaced with an estimate. The algorithm first estimates the errors ε_i associated with the individual planes and then computes the overall approximation error as $\max_i \varepsilon_i$. In the inner approximation, the points on the intersection lines have smaller errors than the points between them. Therefore, for each plane the algorithm constructs the line segment that lies in the middle of the plane, is parallel to the intersection lines and belongs to $\mathcal{V}$. The value of ε_i is estimated as the maximum of the errors at the end points of the constructed line segment. In the outer approximation, the points on the intersection lines have higher errors than the points between them. Therefore, the algorithm first estimates the errors associated with the intersection lines, with each estimate being the maximum of the errors at the end points of the line segment lying in $\mathcal{V}$. The value of ε_i is then computed as the maximum of the errors for the two intersection lines that belong to the i-th plane.

V. Implementation

The proposed algorithm was implemented in the C programming language in the form of a standalone library. Special care was taken to increase the efficiency and robustness of the algorithm. Since the algorithm does not rely on any external packages, the library does not have dependencies and can be built on any machine that has a C compiler. The library was made open-source and is available on Github[1]. It can be used in PFNET[2], which is a library for modeling and analyzing power networks, and constructing related optimization problems. In addition, two Matlab wrapper functions were developed. The first function constructs the linear approximation of a single line flow constraint, which can be used for a detailed analysis of the approximation accuracy. The second function constructs a sparse matrix and a vector that represent the linear approximation of all line flow constraints in the system. This approximation can be used in the AC OPF problem instead of the original nonlinear line flow constraints, for instance by customizing the problem formulation in MATPOWER [27].

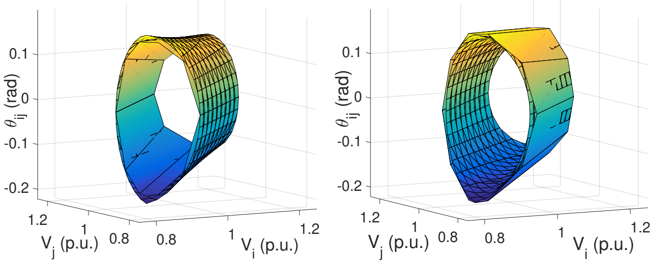
VI. Numerical Experiments

A. Experimental Setup

Numerical experiments were carried out using the MATPOWER test cases. To analyze the impact of the linearization of line flow constraints on the solution of the AC OPF problem, ten test systems of various sizes were considered. Table I presents the overview of these systems, each of which has binding line flow constraints at the obtained locally optimal point. Since the 13659-bus system originally does not have thermal limits, they were artificially generated for all branches.

TABLE I
Overview of Test Systems

Case	Name	Buses	Branches	
			Total	With thermal limits
1	case1951rte	1951	2596	2099
2	case2383wp	2383	2896	2896
3	case2868rte	2868	3808	2281
4	case3120sp	3120	3693	3693
5	case3375wp	3374	4161	4161
6	case6470rte	6470	9005	3110
7	case6495rte	6495	9019	3109
8	case6515rte	6515	9037	3131
9	case9241pegase	9241	16049	6295
10	case13659pegase	13659	20467	20467



(a) Inner approximation (b) Outer approximation

Fig. 6. Linear approximations of line flow constraint[3]

The AC OPF problem was solved in MATPOWER using the state-of-the-art NLP solvers SNOPT, KNITRO, and IPOPT with the MA57 package. Since the standard MATPOWER distribution does not have an interface to SNOPT, it was developed to allow a fair comparison of the results. All solvers were provided with the first and, if required, second derivatives of the objective function and constraints. Note that tuning the solver parameters to achieve the best possible performance is outside of scope of this paper. Therefore, the default options were used for each solver. The results were obtained using a 2.8GHz four-core PC with 32GB of RAM.

B. Analysis of Linear Approximation of Line Flow Constraint

The linear approximation of a line flow constraint with $N_{con} = 10$ is visualized in Figure 6, which shows that the constructed planes closely match the shape of the boundary surface. To assess the approximation accuracy, all branches with thermal limits were extracted from all MATPOWER test cases. In total, there were 59766 such branches. For each of them, the limits on the bus voltage magnitudes were extracted from the bus parameters. The results presented below were obtained for the approximation of (7). The approximation of (8) produced similar results, which are therefore omitted.

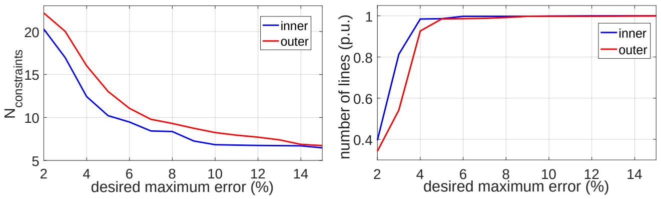
Figure 7 shows the impact of the desired approximation accuracy on the constructed approximation. As the maximum desired error was decreased, the average number of linear constraints needed to achieve it increased. Due to the nonconvexity of the line flow constraint, it was not always possible to approximate it with the desired accuracy. As can be seen from Figure 7b, while the 2% desired error was achieved for less than half of the lines, the 5% error was achieved for 98% of the lines. For this reason, the approximation with $\varepsilon^{\max} = 5\%$ was used in the AC OPF. It is worth noting that the approximation was always strictly conservative or relaxed, i.e. the planes did not intersect with the boundary surface.

More detailed statistics on the parameters of the linear approximation for $\varepsilon^{\max} = 5\%$ for different lines is presented

[1] https://github.com/dmitry-shchetinin/LineFlow

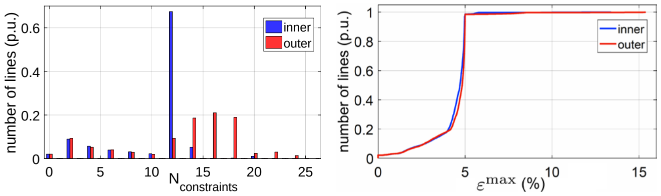
[2] https://github.com/ttinoco/PFNET

[2] Perceived intersections of linear constraints with boundary surface are due to limitations of MATLAB visualization capabilities



(a) Average N_{con} per line (b) Lines with $\varepsilon \leq \varepsilon^{\max}_{\text{desired}}$

Fig. 7. Impact of the value of desired error on linear approximation



(a) Lines with different N_{con} (b) Lines with $\varepsilon \leq \varepsilon^{\max}$

Fig. 8. Parameters of linear approximation for 5% desired error

in Figure 8. As can be seen, most lines required 12 constraints for the inner approximation and 14-18 constraints for the outer approximation. While for almost all lines the desired accuracy was achieved, for certain lines it was impossible to bring the error below 15%. The high approximation error was mainly caused by the loose limits on V_i and V_j since for some lines they were allowed to vary $\pm 30\%$ around their nominal values.

Since the developed algorithm only estimates the true approximation error, the quality of this estimate was assessed. This was done for each branch by computing the errors in one million uniformly distributed points in $\mathcal{V}$. Then the maximum among these errors was calculated and compared to the estimate obtained by the algorithm. The average difference for all tested branches was 0.035% for the inner and 0.139% for the outer approximation. Hence, the algorithm was able to produce a high quality error estimate. In addition, it was observed that 3-4 adjustments of the intersection lines' projections sufficed to almost equalize the errors associated with individual planes.

C. Impact of Linearizing Line Flow Constraints on AC OPF

To analyze the impact of linearizing line flow constraints on the solution of the AC OPF problem, their inner and outer approximations were constructed for all test cases. The desired approximation error of 5% was used for each system and approximation type. For each line, the approximations of the line flow constraints at both ends were constructed and then only the linear constraints relevant for the intersection of the two feasible sets were retained. Depending on the system, the number of linear constraints was 4-8 times the number of original nonlinear constraints. One can see from Table II that despite a high number of constructed constraints, the construction time was low for all systems.

Table II also shows the impact of the inner and outer approximation on the solution quality. Since the inner approximation reduced the feasible set, it led to an increase in the cost function value compared to the case with the exact nonlinear constraints. Naturally, no thermal limits were violated at the solution of the problem with inner approximations of the line flow constraints. On the other hand, the outer approximation increased the feasible set, which resulted in a lower objective function value. This came at a cost of violating some thermal limits at the obtained solution. For several systems, the maximum violation was higher than 5% because the lines with binding thermal limits were also the ones for which it was impossible to achieve the desired approximation accuracy. However, the approximation generally only had a marginal influence on the objective function value.

The impact of the linearization of line flow constraints on the reliability and computation time of the NLP solvers was analyzed using SNOPT, IPOPT, and KNITRO. Each test case was solved by all solvers using the formulations with exact constraints and their inner and outer approximations. Two initial points were considered for each optimization problem: flat start and warm start. The summary of the results is presented in Table III, where the computation times for the problems with linearized line flow constraints include the time spent on the linearization. One can see that for the AC OPF with exact line flow constraints, SNOPT was in general the slowest and least reliable solver. IPOPT and KNITRO showed similar results in terms of the computation time. However, IPOPT was the most reliable solver because it found the solution for all test cases and problem formulations. Test cases 1, 3, 6, 8, and 10 were hard for all solvers, especially when the flat start was used as the initial point.

The linearization of line flow constraints significantly increased the reliability of all three solvers. Each solver converged to a locally optimal point for all test cases regardless of whether the flat or warm start was used. Moreover, the convergence became less affected by the starting point. For the biggest test case, the linearization helped avoid numerical difficulties that IPOPT and KNITRO encountered when using the formulation with the exact constraints.

The impact of the linearization on the computation time depended on the test case and the starting point. For the simple test cases, the computation time increased 1.5-2 times because of the larger problem size. However, for the hard test cases, the linearization helped significantly reduce the computation time, particularly for the flat start. The most significant improvements were observed for IPOPT and SNOPT. Overall, the most robust performance was achieved by IPOPT when using the formulation with the linearized line flow constraints.

VII. Conclusion

This paper presents a computationally efficient algorithm for constructing inner and outer linear approximations of the line flow constraints. The algorithm was implemented as an open-source library and is based on the properties of the feasible set of a line flow constraint. This set is non-convex but for realistic lines and operating limits it can be closely approximated by several linear constraints. Incorporating the linear constraints into the AC optimal power flow problem instead of the original nonlinear constraints helps reduce the problem's complexity without compromising the quality of the solution.

Numerical experiments demonstrated that hard large-scale optimization problems with the linear line flow constraints can be solved more reliably by all considered state-of-the-art NLP solvers. Despite an increase in the problem size, the computation time for these problems was reduced, particularly when using the flat start as the initial point. In addition, an extensive analysis of the approximation accuracy showed that

TABLE II

CONSTRUCTION OF LINEAR LINE FLOW CONSTRAINTS AND THEIR IMPACT ON AC OPF SOLUTION

Case		**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Inner	△Cost, %	0.001	0.521	0.000	0.012	0.002	0.001	0.008	0.004	0.000	0.035
	Constructed constraints	21662	39225	23535	49530	48215	27835	27799	28159	66718	138790
	Construction time, s	0.08	0.17	0.08	0.20	0.20	0.14	0.14	0.17	0.24	0.31
Outer	△Cost, %	-0.001	-0.124	-0.001	-0.015	-0.005	-0.011	-0.083	-0.055	-0.002	-0.054
	Constructed constraints	23417	49646	25808	65515	62452	31310	31238	31720	90066	148692
	Construction time, s	0.18	0.45	0.19	0.56	0.53	0.26	0.26	0.29	0.61	0.42
	Violated thermal limits	6	3	9	11	2	12	22	25	22	21
	Maximum violation, %	4.0	3.6	4.2	3.6	1.3	8.2	8.1	8.1	4.7	5.4

TABLE III

COMPUTATION TIMES FOR AC OPF IN SECONDS

Case	SNOPT						IPOPT						KNITRO					
	Warm start			Flat start			Warm start			Flat start			Warm start			Flat start		
	Exact	Inner	Outer	Exact	Inner	Outer	Exact	Inner	Outer	Exact	Inner	Outer	Exact	Inner	Outer	Exact	Inner	Outer
1	19.6	35.8	35.9	37.8	51.9	48.9	2.1	2.7	3.2	1899	**9.0**	**10.2**	—	**6.0**	**7.7**	9.5	34.0	22.1
2	8.6	12.9	17.3	11.5	15.1	25.7	2.4	4.2	4.3	2.9	5.0	6.0	1.5	2.9	3.0	1.3	2.8	3.0
3	103	154	177	—	**206**	**276**	4.1	4.7	6.0	365	**15.0**	**16.8**	11.8	**6.2**	**6.3**	20.2	31.2	32.3
4	31.4	51.0	60.8	23.1	47.1	53.6	3.5	6.7	8.6	3.3	6.1	7.8	1.5	3.2	4.1	1.4	2.9	3.9
5	52.7	**27.4**	59.3	49.4	**48.1**	88.5	3.0	4.6	5.4	3.9	6.3	7.9	3.2	4.4	5.7	1.9	3.1	3.9
6	181	203	229	—	**309**	**354**	63.6†	69.1†	**53.2**†	2310†	**113**†	**121**†	4.2	5.1	5.4	—	**86.9**	**90.6**
7	158	227	166	737	**249**	**314**	6.2	8.0	16.6	34.3	**24.8**	39.2	3.4	4.9	6.0	60.4	152	106
8	163	197	217	—	**257**	**313**	5.2	7.2	7.2	524	**23.7**	**26.2**	4.5	5.9	6.3	—	**81.3**	**92.9**
9	1914	**838**	**1047**	2449	**1322**	**1400**	11.0	13.6	12.6	39.2	**15.3**	**15.0**	7.1	10.2	14.6	8.1	78.5	80.4
10	—	**4907**	**2.2e5**	—	**1.8e5**	**1.2e5**	142⋆	**52.5**	**59.3**	151⋆	**59.5**	**54.5**	135⋆	558	**108**	614⋆	781	**101**

bold - problem with linear line flow constraints was solved faster than with original nonlinear constraints, — - solver failed to converge, † - solver failed to achieve required accuracy, ⋆ - solution violates line flow constraints

it is possible to achieve a 5% approximation error for almost all realistic lines and operating limits.

Possible future work includes the application of the linear approximations to convex relaxations and bound tightenings for the AC optimal power flow problem.

REFERENCES

[1] J. Carpentier, "Contribution á l'étude du dispatching économique", *Bulletin de la Société Française des Électriciens*, pp. 431–447, 1962.

[2] Florin Capitanescu et al., "State-of-the-art, challenges, and future trends in security constrained optimal power flow", *Electric Power Systems Research*, vol. 81, no. 8, pp. 1731–1741, 2011.

[3] Florin Capitanescu, "Critical review of recent advances and further developments needed in AC optimal power flow", *Electric Power Systems Research*, vol. 136, pp. 57–68, July 2016.

[4] Florin Capitanescu and Louis Wehenkel, "Experiments with the interior-point method for solving large scale Optimal Power Flow problems", *Electric Power Systems Research*, vol. 95, pp. 276–283, Feb. 2013.

[5] Anya Castillo and R. O'Neill, "Computational Performance of Solution Techniques Applied to the ACOPF", Tech. Rep., 2013.

[6] B. Stott, J. Jardim, and O. Alsac, "DC Power Flow Revisited", *IEEE Transactions on Power Systems*, vol. 24, pp. 1290–1300, Aug. 2009.

[7] J. A. Momoh and J. Z. Zhu, "Improved interior point method for OPF problems", *IEEE Transactions on Power Systems*, vol. 14, no. 3, pp. 1114–1120, Aug. 1999.

[8] A. L. Motto, F. D. Galiana, A. J. Conejo, and J. M. Arroyo, "Network-constrained multiperiod auction for a pool-based electricity market", *IEEE Transactions on Power Systems*, vol. 17, pp. 646–653, Aug. 2002.

[9] T.J. Overbye, Xu Cheng, and Yan Sun, "A comparison of the AC and DC power flow models for LMP calculations", in *Proceedings of the 37th Annual Hawaii International Conference on System Sciences*, 2004.

[10] K. Purchala, L. Meeus, D. Van Dommelen, and R. Belmans, "Usefulness of DC power flow for active power flow analysis", in *IEEE Power Engineering Society General Meeting, 2005*, June 2005, pp. 454–459.

[11] A. Castillo, P. Lipka, J. P. Watson, S. S. Oren, and R. P. O'Neill, "A successive linear programming approach to solving the iv-acopf", *IEEE Transactions on Power Systems*, vol. 31, pp. 2752–2763, July 2016.

[12] 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 Transactions on Power Systems*, vol. PP, no. 99, pp. 1–1, 2017.

[13] Subhonmesh Bose, Steven H. Low, Thanchanok Teeraratkul, and Babak Hassibi, "Equivalent Relaxations of Optimal Power Flow", *IEEE Transactions on Automatic Control*, vol. 60, pp. 729–742, Mar. 2015.

[14] R. Madani, S. Sojoudi, and J. Lavaei, "Convex Relaxation for Optimal Power Flow Problem: Mesh Networks", *IEEE Transactions on Power Systems*, vol. 30, no. 1, pp. 199–211, Jan. 2015.

[15] Masoud Farivar and Steven H. Low, "Branch Flow Model: Relaxations and Convexification—Part I", *IEEE Transactions on Power Systems*, vol. 28, no. 3, pp. 2554–2564, Aug. 2013.

[16] C. Coffrin, H. L. Hijazi, and P. Van Hentenryck, "The QC Relaxation: A Theoretical and Computational Study on Optimal Power Flow", *IEEE Transactions on Power Systems*, vol. 31, pp. 3008–3018, July 2016.

[17] Javad Lavaei and Steven H. Low, "Zero Duality Gap in Optimal Power Flow Problem", *IEEE Transactions on Power Systems*, vol. 27, no. 1, pp. 92–107, Feb. 2012.

[18] M. Nick, R. Cherkaoui, J. Y. LeBoudec, and M. Paolone, "An Exact Convex Formulation of the Optimal Power Flow in Radial Distribution Networks Including Transverse Components", *IEEE Transactions on Automatic Control*, vol. PP, no. 99, pp. 1–1, 2017.

[19] C. Chen, A. Atamtürk, and S. S. Oren, "Bound Tightening for the Alternating Current Optimal Power Flow Problem", *IEEE Transactions on Power Systems*, vol. 31, no. 5, pp. 3729–3736, Sept. 2016.

[20] C. Coffrin, H. L. Hijazi, and P. Van Hentenryck, "Strengthening Convex Relaxations with Bound Tightening for Power Network Optimization", in *Principles and Practice of Constraint Programming*. Aug. 2015, Lecture Notes in Computer Science, pp. 39–57, Springer, Cham.

[21] P. Lipka, R. O'Neill, and S. Oren, "Developing line current magnitude constraints for IEEE test problems", Tech. Rep., FERC, 2013.

[22] S. Grijalva, P. W. Sauer, and J. D. Weber, "Enhancement of linear ATC calculations by the incorporation of reactive power flows", *IEEE Transactions on Power Systems*, vol. 18, no. 2, pp. 619–624, May 2003.

[23] Joshua A. Taylor and Franz S. Hover, "Linear Relaxations for Transmission System Planning", *IEEE Transactions on Power Systems*, vol. 26, no. 4, pp. 2533–2538, Nov. 2011.

[24] Carleton Coffrin and Pascal Van Hentenryck, "A Linear-Programming Approximation of AC Power Flows", *INFORMS Journal on Computing*, vol. 26, no. 4, pp. 718–734, May 2014.

[25] C. Coffrin, P. Van Hentenryck, and R. Bent, "Approximating line losses and apparent power in AC power flow linearizations", in *2012 IEEE Power and Energy Society General Meeting*, July 2012, pp. 1–8.

[26] D. Shchetinin, T. T. De Rubira, and G. Hug, "Conservative linear line flow constraints for AC optimal power flow", in *2017 IEEE Manchester PowerTech*, June 2017, pp. 1–6.

[27] R. D. Zimmerman, C. E. Murillo-Sanchez, 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.