Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 55-13 (2022) 145–150

Rapid Scalable Distributed Power Flow with Open-Source Implementation ⋆

Xinliang Dai * **Yichen Cai** * **Yuning Jiang** ** **Veit Hagenmeyer** *

* *Institute for Automation and Applied Informatics, Karlsruhe Institute of Technology, Karlsruhe, Germany(e-mail: xinliang.dai, veit.hagenmeyer@kit.edu, yichen.cai@kit.student.edu).*

** *Laboratoire d'Automatique, École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland. (e-mail: yuning.jiang@epfl.ch)*

Abstract: This paper introduces a new method for solving the distributed AC power flow (PF) problem by further exploiting the problem formulation. We propose a new variant of the ALADIN algorithm devised specifically for this type of problem. This new variant is characterized by using a reduced modelling method of the distributed AC PF problem, which is reformulated as a zero-residual least-squares problem with consensus constraints. This PF is then solved by a Gauss-Newton based inexact ALADIN algorithm presented in the paper. An open-source implementation of this algorithm, called rapidPF+, is provided. Simulation results, for which the power system's dimension varies from 53 to 10224 buses, show great potential of this combination in the aspects of both the computing time and scalability.

Keywords: Power Flow, Large-scale, ALADIN, Distributed Optimization

1. INTRODUCTION

The ongoing implementation of the energy transition leads to heterogeneous energy networks with numerous energy producers, energy consumers, transport, conversion and storage systems. Due to strongly varying renewable-based energy feed-ins and demands of the power system, new challenges arise in the aspect of power flow analysis, including power flow (PF) problems and optimal power flow (OPF) problems.

The conventional PF problem is modeled as a system of nonlinear equations. Usually, it is solved by centralized methods, i.e., Gauss-Seidel or Newton-Raphson (Grainger, 1999). However, the centralized approach requires one central entity, where all generation information and network topology data are collected. Sharing such data is unsatisfactory for system operators. In contrast to the centralized approach, the distributed approach first solves each decoupled sub-problem in its own local agent respectively, and then deals with a coupled problem in a central coordinator, in which some, but not all grid data is collected. As a result, the distributed approach not only preserves the information privacy and decision independence, but also decreases the vulnerability due to single-point-of failure (Mühlpfordt et al., 2021).

The idea to solve a global PF problem by breaking the original problem into several smaller power flow problems is proposed by Sun and Zhang (2008). However, there are no convergence guarantees and the size of the test cases is limited to 200. In their following research (Sun et al., 2014), they provided a simple convergence analysis, but didn't provide actual convergence behavior of the proposed method, and its scalability still remains unknown. Besides, a popular distributed algorithm, Alternating Direction Method of Multipliers (ADMM), has attracted interest from researchers in terms of steady-state analysis of power system (Erseghe, 2014). Nonetheless, its convergence cannot be guaranteed for the nonconvex AC PF problem, and is highly dependent on initial guess and tuning parameters, as pointed out in our previous research (Mühlpfordt et al., 2021).

In addition, Houska et al. (2016) proposed the Augmented Lagrangian based Alternating Direction Inexact Newton method (ALADIN) that is devised for non-convex problems with local convergence guarantee. It has found widespread application for analysis of power systems (Engelmann et al., 2017, 2018; Meyer-Huebner et al., 2019; Du et al., 2019; Jiang et al., 2021a,b). ALADIN shares the same idea with ADMM—update primal variables in an alternating fashion. However ALADIN requires sensitivities information of sub-problems to build a second-order approximation in the coordinator. When using suitable Hessian approximation, ALADIN can achieve locally quadratic convergence. In our previous work (Mühlpfordt et al., 2021), an open-source MATLAB code for rapid prototyping for distributed power flow (rapidPF) [1] is provided, in which the AC PF problem is reformulated as a zero-residual least-squares problem tailored for the ALADIN to speed up the convergence—all the example cases can converge within half-dozen iterates. Nevertheless, the total computing time is not acceptable for large-scale problems due to the rela-

⋆ The authors acknowledge funding from the German Federal Ministry of Education and Research within the project *MOReNet – Modellierung, Optimierung und Regelung von Netzwerken heterogener Energiesysteme mit volatiler erneuerbarer Energieerzeugung.*

[1] The code is available on https://github.com/KIT-IAI/rapidPF

10.1016/j.ifacol.2022.07.250

tive large dimension of the decoupled NLP problem and the problematic code efficiency of the ALADIN-α toolbox (Engelmann et al., 2020).

The contribution of the present paper is two-fold. We propose a Gauss-Newton based ALADIN algorithm for solving the zero-residual least-squares problem and a reduced modelling method for distributed AC PF. Based on them, we upgrade the open-source code of rapidPF. The remainder of this paper is organized as follows: Section 2 formulates the distributed AC PF as a zero-residual least-squares problem. Section 3 presents both the standard ALADIN and the Gauss-Newton based ALADIN algorithms. The upgrade of rapidPF, called rapidPF+, is described in Section 4. The simulation results are compared and discussed in Section 5.

2. PROBLEM FORMULATION

This section introduces the distributed AC PF problem of polar voltage coordination and its zero-residual least-squares formulation. Before further discussion, we first introduce some nomenclature. For a power system, $\mathcal{R}$ represents the set of regions, n^{reg} is the number of regions and n^{conn} is the number of all the connecting tie lines between regions. In a specific region ℓ, $\mathcal{N}_\ell$ is the set of all buses, whereas $\mathcal{N}_\ell^{core}$ and $\mathcal{N}_\ell^{copy}$ are the set of core and copy buses in this region ℓ, respectively.

2.1 Distributed Power Flow

The conventional AC PF problem seeks a deterministic solution to the steady-state operation of an AC electrical power system by applying numerical analysis techniques (Frank and Rebennack, 2016). Each bus in the system has four variables, i.e., voltage angle θ, voltage magnitude v, active power injection p, and reactive power injection q.

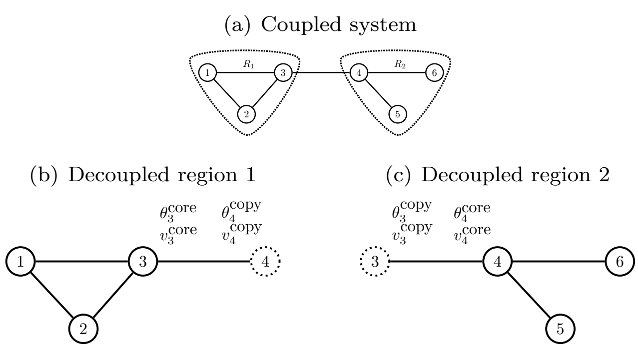

Fig. 1. Decomposition by sharing components for a two-region system

Fact 1. Genetically, there are multiple mathematically valid solutions to a power flow problem, but only one solution has physical meaning (Frank and Rebennack, 2016).This results, e.g., from the periodic voltage angle θ, and the respective trigonometric functions.

In order to apply a distributed algorithm, reformulation of the AC PF problem is necessary. In terms of partitioning the power system, we share the components between neighboring regions to ensure physical consistency. As an example, we take the 6-bus system with 2 regions, shown in Figure 1. The coupled system, shown in Figure 1(a), has been partitioned into 2 local regions. To solve the AC PF problem in region R_1, besides its own buses {1,2,3} called *core buses*, the complex voltage of bus {4} from neighboring region R_2 is required. Hence, for the sub-problem of region R_1, we create an auxiliary bus {4} called *copy bus*, along with its own *core bus*, to formulate a self-contained AC PF problem.

Then, affine consensus constraints of the connecting tie line are added to ensure consistency of the copy bus with its original core bus in the neighboring region. The consensus constraints of the example case in Figure 1 can be written as

$$\theta_3^{\text{core}} = \theta_3^{\text{copy}},\, \theta_4^{\text{core}} = \theta_4^{\text{copy}} \tag{1a}$$
$$v_3^{\text{core}} = v_3^{\text{copy}},\, v_4^{\text{core}} = v_4^{\text{copy}} \tag{1b}$$

In a specific region ℓ, the power flow equations be represented as

$$p_i^g - p_i^l = v_i \sum_{k\in\mathcal{N}_\ell} v_k \left(G_{ik}\cos\theta_{ik} + B_{ik}\sin\theta_{ik}\right) \tag{2a}$$
$$q_i^g - q_i^l = v_i \sum_{k\in\mathcal{N}_\ell} v_k \left(G_{ik}\sin\theta_{ik} - B_{ij}\cos\theta_{ik}\right) \tag{2b}$$

for all *core bus* $i \in \mathcal{N}_\ell^{\text{core}}$ with the angle difference between buses $\theta_{ik} = \theta_i - \theta_k$, complex generation $s^g = p^g + \mathrm{j}q^g$, complex load $s^l = p^l + \mathrm{j}q^l$, complex components of the bus admittance matrix entries $Y_{ik} = G_{ik} + \mathrm{j}B_{ik}$. These equations can also be written as residual function. These power flow equations can be written as

$$r_\ell(\chi_\ell) = 0 \tag{3}$$

where $r_\ell : \mathbb{R}^{2n_\ell^{\text{core}}+2n_\ell^{\text{copy}}} \to \mathbb{R}^{n_\ell^{\text{pf}}}$ is a residual function with its components $r_{\ell,m}$, i.e., the m-th power flow residual in the region ℓ, and χ_ℓ is the state of the region ℓ. Note that the number of power flow equations $n_\ell^{\text{pf}} = 2n_\ell^{\text{core}}$ in all local region.

Hence, the distributed AC PF problem can be represented as a system of nonlinear equations and affinely coupled consensus equations as follows

$$r_\ell(\chi_\ell) = 0,\ \forall \ell \in \mathcal{R} \tag{4a}$$
$$\sum_{\ell\in\mathcal{R}} A_\ell \chi_\ell = A\chi = b. \tag{4b}$$

with the state $\chi = (\chi_1^\top, \chi_2^\top, \cdots, \chi_{n^{\text{reg}}}^\top)^\top$

2.2 Zero-Residual Least-Squares Formulation

Following Mühlpfordt et al. (2021), we reformulate the distributed AC PF problem (4) in a standard least-squares formulation with affine consensus constraint

$$\min_{\chi} \quad f(\chi) := \sum_{\ell\in\mathcal{R}} f_\ell(\chi_\ell) = \frac{1}{2}\sum_{\ell\in\mathcal{R}} \|r_\ell(\chi_\ell)\|_2^2 \tag{5a}$$
$$\text{s.t.} \quad A\chi = b \quad |\ \lambda \tag{5b}$$

with the consensus matrix $A = (A_1, A_2, \cdots, A_{n^{\text{reg}}})$ and the state $\chi = (\chi_1^\top, \chi_2^\top, \cdots, \chi_{n^{\text{reg}}}^\top)^\top$. The problem (5) can be classified as a zero-residual least-squares problem, since all the power flow residuals are equal to zero at the PF solution χ^*.

Proposition 2. Let the power flow problem (2) be feasible, i.e., a primal solution χ^* to the problem (5) exists such that the power flow residual $r_\ell(\chi_\ell^*) = 0$ for all $\ell \in \mathcal{R}$ bounded by consensus constraint (5b), and let linear independence constraint qualification (LICQ) holds at χ^*. Then the dual variable $\lambda^* = 0$ with the primal solution χ^* satisfies the KKT conditions, i.e., $(\chi^*, \lambda^* = 0)$ is a KKT point.

2.3 Sensitivities

The derivatives of the objective $f_\ell(\chi_\ell)$ can be expressed as

$$\nabla f_\ell(\chi_\ell) = J_\ell(\chi_\ell)^\top r_\ell(\chi_\ell) \tag{6a}$$

$$\nabla^2 f_\ell(\chi_\ell) = J_\ell(\chi_\ell)^\top J_\ell(\chi_\ell) + Q_\ell(\chi_\ell) \tag{6b}$$

with

$$J_\ell(\chi_\ell) = \left[\nabla r_{\ell,1}, \nabla r_{\ell,2}, \cdots, \nabla r_{\ell,n^{\text{pf}}}\right]^\top \tag{7a}$$

$$Q_\ell(\chi_\ell) = \sum_{m=1}^{n^{\text{pf}}} r_{\ell,m}(\chi_\ell)\nabla^2 r_{\ell,m}(\chi_\ell). \tag{7b}$$

The Gauss-Newton approximation can be written as

$$\nabla^2 f_\ell(\chi_\ell) \approx J_\ell(\chi_\ell)^\top J_\ell(\chi_\ell) \tag{8}$$

In practice, the first term (7a) dominates the second one (7b), either because the residuals $r_{\ell,m}$ are close to affine near the solution, i.e., $\nabla^2 r_{\ell,m}$ are relatively small, or because of small residuals, i.e.,$r_{\ell,m}$ are relatively small (Nocedal and Wright, 2006).

Remark 3. For solving the zero-residual least-squares problem, the Gauss-Newton approximation is exact at the solution, and can converge to the exact Hessian during iterations rapidly.

3. ALGORITHM

This section presents the standard ALADIN algorithm and its new variant for zero-residual least-squares problems.

3.1 Standard ALADIN

Houska et al. (2016) introduced a novel algorithm, i.e., ALADIN, to handle distributed nonlinear programming. ALADIN for problem (5) is outlined in Algorithm 1. The algorithm has two main steps, i.e., a decoupled step (i) and a consensus step (iii). Pursuing the idea of augmented Lagrangian, the local problem is formulated as (10) in step (i), where ρ is the penalty parameter and Σ_ℓ is the positive definite scaling matrix for the region ℓ. Based on the result from local NLPs (10), the ALADIN algorithm terminates if both the primal and the dual residuals are smaller than tolerance ϵ

$$\left\|\sum_{\ell\in\mathcal{R}} A_\ell x_\ell - b\right\|_\infty \leq \epsilon \text{ and } \max_\ell \|\Sigma_\ell(x_\ell - z_\ell)\|_\infty \leq \epsilon \tag{9}$$

Compared with a simple averaging step of ADMM in the coordinator, ALADIN based on curvature information (11) builds a coupled Quadratic Programing (QP) (12) to coordinate the results of the decoupled step from all regions. Additionally, a slack variable s and a corresponding penalty parameter μ is introduced in the consensus step to ensure feasibility of the coupled QP. Consequently, ALADIN achieves fast and guaranteed convergence. A detailed proof of local convergence can be found in Houska et al. (2016).

Algorithm 1 ALADIN(standard)

Initialization: λ, ρ, μ, z_ℓ, $\Sigma_\ell \succ 0$ **for all** $\ell \in \mathcal{R}$,
Repeat:

(i) Solve decoupled NLPs

$$\min_{x_\ell} \; f_\ell(x_\ell) + \lambda^\top A_\ell x_\ell + \frac{\rho}{2}\|x_\ell - z_\ell\|^2_{\Sigma_\ell} \tag{10}$$

and compute local sensitivities for all $\ell \in \mathcal{R}$

$$g_\ell = \nabla f_\ell(x_\ell) \text{ and } H_\ell \approx \nabla^2 f_\ell(x_\ell) \tag{11}$$

(ii) Check termination condition (9)

(iii) Solve coupled QP

$$\min_{\Delta x, s} \; \frac{1}{2}\Delta x^\top H \Delta x + g^\top \Delta x + \lambda^\top s + \frac{\mu}{2}\|s\|_2^2 \tag{12a}$$

$$\text{s.t. } A(x + \Delta x) = b + s \tag{12b}$$

where Hessian $H = \text{diag}\{H_\ell\}_{\ell\in\mathcal{R}}$ and gradient g with components g_ℓ

(iv) Update primal and dual variables with full-step

$$z^+ = x + \Delta x, \tag{13a}$$

$$\lambda^+ = \lambda^{QP}. \tag{13b}$$

3.2 Gauss-Newton based inexact ALADIN

Based on the framework of standard ALADIN, we propose a tailored version specific for solving zeros-residual least-squares problem in the present paper, see Algorithm 2. Since optimal values of Lagrangian multipliers are equal to zero $\lambda^* = 0$ according to *Proposition 2*, the Lagrangian terms in (10)(12) can be neglected by fixing dual iterates $\lambda = 0$ at the cost of convergence rate. In this way, both coupled and decoupled steps can be viewed as adding a residual to the original problems respectively, and can be solved by equivalent linear systems efficiently.

Algorithm 2 inexact ALADIN(Gauss-Newton)

Initialization: λ, ρ, μ, z_ℓ, $\Sigma_\ell \succ 0$ **for all** $\ell \in \mathcal{R}$,
Repeat:

(i) Solve decoupled linear systems and update primal variables x_ℓ

$$\left(J_\ell^{z\top} J_\ell^z + \rho I\right) p_\ell = -J_\ell^{z\top} r_\ell^z \tag{14}$$

with Gauss-Newton step $p_\ell = x_\ell - z_\ell$, as well as compute local sensitivities for all $\ell \in \mathcal{R}$

$$g_\ell = J_\ell(\hat{x}_\ell)^\top r_\ell(\hat{x}_\ell) \text{ and } H_\ell = J_\ell(\hat{x}_\ell)^\top J_\ell(\hat{x}_\ell) \tag{15}$$

(ii) Check termination condition (9)

(iii) Solve the linear system of coupled QP

$$\left(H + \mu A^\top A\right)\Delta x = -\mu A^\top (A\hat{x} - b) - g \tag{16}$$

where Hessian $H = \text{diag}\{H_\ell\}_{\ell\in\mathcal{R}}$ and gradient g with components g_ℓ

(iv) Update primal variables with full step

$$z^+ = \hat{x} + \Delta x. \tag{17}$$

For the decoupled step (i), the objective function (10) can be approximated by a quadratic model by applying the Gauss-Newton method

$$M_\ell(p_\ell) = \frac{1}{2}p_\ell^\top\left(J_\ell^{z\top} J_\ell^z + \rho I\right)p_\ell + J_\ell^{z\top} r_\ell^z p_\ell + f_\ell(z_\ell) \tag{18}$$

with Gauss-Newton step $p_\ell = x_\ell - z_\ell$, Jacobian matrix $J_\ell^z = J_\ell(z_\ell)$ and residual vector $r_\ell^z = r_\ell(z_\ell)$ at the initial point z_ℓ in every iterate. Accordingly, the decoupled NLP (10) is solved by a linear system (14), where x_ℓ is an inexact solution to this problem.

For the coupled step (iii), the objective function can be rewritten as

$$\min_{\Delta x} \; \frac{1}{2}\Delta x^\top H \Delta x + g^\top \Delta x + \frac{\mu}{2}\left\| A\left(\hat{x}+\Delta x\right) - b\right\|_2^2 \quad (19)$$

In the corresponding linear system (16), Δx in coupled step (iii) is locally equivalent to a standard Gauss-Newton step of the original coupled problem (5), where the slack variable $s = A\left(\hat{x}+\Delta x\right) - b$ can be viewed as an additional weighted residual.

In the present paper, we focus on the local convergence due to *Fact 1* and good initial guess provided by MATPOWER. The local convergence indicates that the starting point and the iterates are all located in a small neighborhood of the optimizer, within which the solution has physical meaning. The convex set Ω concludes all the points in the bounded neighborhood. Besides, the objective f of the original coupled problem (5) is second order continuously differentiable according to Section 2.3, and $\left\|\nabla^2 f(x)\right\|$ is bounded for all $x \in \Omega$. Then, there exists a constant $L > 0$

$$\left\|\nabla f(x) - \nabla f(z^*)\right\| = \left\|\nabla^2 f(\tilde{x})\right\| \left\|x - z^*\right\| \leq L \left\|x - z^*\right\| \quad (20)$$

with $\tilde{x} = x - t(x - z^*) \in \Omega$ for some $t \in (0,1)$. Hence, the function f is twice Lipschitz-continuously differentiable in the neighborhood Ω.

Before discussing further about the convergence property, we introduce a regularity and some nomenclature first: A KKT point is called *regular* if linear independence constraint qualification (LICQ), strict complementarity conditions (SCC) and second order sufficient condition (SOSC) are satisfied. For the analysis of local decoupled step (i), we introduce $\bar{x}$ as the exact solution and $\hat{x}$ as the inexact solution of the decoupled NLPs (10), whereas $x^* = z^*$ is the primal optimizer of the original coupled problem (5).

Next, let's turn to the local convergence property of Algorithm 2.

Theorem 4. Let the minimizer $(x^* = z^*, \lambda^* = 0)$ be a regular KKT point of problem (5), let the initial guess located in the small neighborhood of the optimizer Ω, and let μ sufficient large such that $\frac{1}{\mu} \leq O(\|\hat{x} - z^*\|)$, then the iterates $\hat{x}$ of Algorithm 2 converge quadratically to a local solution.

Proof of *Theorem 4* can be established by three steps, following the analysis in *Appendix* by Engelmann et al. (2018). First, due to the fact that the local inexact solution $\hat{x}_\ell$ is obtained by Gauss-Newton method, the $\hat{x}$ is a linear contraction to the exact solution $\bar{x}$, i.e., there exists a constant $\eta_1 > 0$ such that

$$\left\|\hat{x} - \bar{x}\right\| \leq \eta_1 \left\|z - \bar{x}\right\|. \quad (21)$$

Second, from *Lemma 3* of Houska et al. (2016), we have

$$\left\|\bar{x} - z^*\right\| \leq \eta_2 \left\|z - z^*\right\|, \; \exists \eta_2 > 0 \quad (22)$$

This differs from standard ALADIN by a fixed dual variable $\lambda = 0$.

Third, because the coupled step of Algorithm 2 is a standard Gauss-Newton step of the original coupled problem (5), as well as the Lipschitz continuity of f and sufficient large μ such that $\frac{1}{\mu} \leq O(\|\hat{x} - z^*\|)$, we obtain the following inequality according to the convergence analysis of the standard Gauss-Newton method (Nocedal and Wright, 2006, Section 10.3)

$$\left\|z^+ - z^*\right\| \leq \left\|H(z^*)^{-1}Q(z^*)\right\| \left\|x - z^*\right\| + O(\|x - z^*\|^2) \quad (23)$$

with $Q = \text{diag}\{Q_\ell\}_{\ell\in\mathcal{R}}$. For problem (5), all the optimal residuals are equal to zero, then we have $Q_\ell(z_\ell^*) = 0$ for all $\ell \in \mathcal{R}$. As a result,

$$\left\|z^+ - z^*\right\| \leq O(\|\hat{x} - z^*\|^2) \quad (24)$$

The statement of *Theorem 4* follows by combining of(21), (22) and (24).

4. OPEN-SOURCE IMPLEMENTATION

Based on the Algorithm 2, we improve the existing toolkit rapidPF. To this end, in this section, we introduce a reduced modelling method and describe the structural upgrade of rapidPF+ compared with rapidPF.

4.1 Reduced modelling method

Table 1 summarizes the known and unknown variables of a AC PF problem according to different bus-types in the power system. In the original distributed AC PF model proposed by Mühlpfordt et al. (2021), the known variables are constrained by *bus specification*, which is added as residuals in least-squares formulation. This results in the unnecessary growth of the problem dimension and slows down the run time. To overcome the issue, the present paper distinguishes the known and the unknown variables, and uses a so-called reduced modelling method to reduce the dimension of the distributed AC PF problem.

Table 1. Known and Unknown variables for AC PF problem regarding the bus-type

	REF	PQ	PV
Known variables	θ, v	p, q	v, p
Unknown variables	p, q	θ, v	θ, q

For a specific region $\ell \in \mathcal{N}^{\text{reg}}$, the state consists of variables from both core buses and copy buses. The state of the core bus i is defined according to its own bus-type:

$$\zeta_i^{\text{core}} = \begin{cases} (p_i^{\text{core}}, q_i^{\text{core}}) & (\text{REF}) \\ (\theta_i^{\text{core}}, v_i^{\text{core}}) & (\text{PQ}) \\ (\theta_i^{\text{core}}, q_i^{\text{core}}) & (\text{PV}) \end{cases}, \; \forall i \in \mathcal{N}_\ell^{\text{core}}, \quad (25)$$

whereas the state of the *copy bus j* contains voltage angle and magnitude

$$\zeta_j^{\text{copy}} = (\theta_j^{\text{copy}}, v_j^{\text{copy}}), \; \forall j \in \mathcal{N}_\ell^{\text{copy}}, \quad (26)$$

The state of this specific region $\chi_\ell \in \mathbb{R}^{2n_\ell^{\text{core}}+2n_\ell^{\text{copy}}}$ is composed by all the core and the copy buses in the regions.

Typically, n^{core} dominates n^{copy} in a sub-system of a power grid. Therefore, the dimension by using the reduced modelling method, i.e., $\sum_\ell 2n_\ell^{\text{core}} + 2n_\ell^{\text{copy}}$, is almost reduced by half, compared with the original model— $\sum_\ell 4n_\ell^{\text{core}} + 2n_\ell^{\text{copy}}$—proposed by Mühlpfordt et al. (2021).

4.2 rapidPF vs. rapidPF+

As shown in Figure 2, the rapidPF builds a distributed AC PF problem based on MATPOWER case files and solves it by interfacing with an external ALADIN-α toolbox. Nevertheless, due to the problematic code efficiency of the ALADIN-α toolbox, computing for a large-scale problem is not acceptable—for a 4662-Bus system, it takes 90.1 seconds to converge by using `fmincon`, whereas the initial time by using CasADi is intolerant.

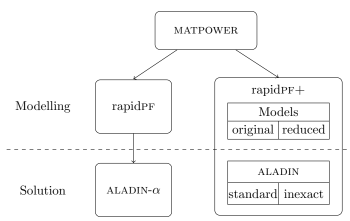

Fig. 2. Flow charts for solving distributed AC PF by the rapidPF and the rapidPF+ toolbox

In contrast, rapidPF+ doesn't rely on the external ALADIN toolbox. The user can switch between two models and two ALADIN algorithms. Comparison of these combinations is carried out in the following section.

5. SIMULATION RESULTS

In this section, we illustrate the performance of several combinations of the two distributed AC PF models and the two variants of ALADIN algorithm. We use the suggested combination by Mühlpfordt et al. (2021) as a benchmark, i.e., the original distributed power flow model with standard ALADIN (Algorithm 1). Towards practical implementation, several test cases by Mühlpfordt et al. (2021) are also modified—multiple connecting tie lines are added and the graph of regions is transferred from radial to meshed topology. Besides, we introduce a 10224-bus test case to exhibit the performance for a large-scale implementation.

The framework [2] is built on MATLAB-R2021a and the case studies are carried out on a standard desktop computer with `Intel® i5-6600K CPU @ 3.50GHz` and 16.0 GB installed RAM. To run the benchmark, adding MATPOWER to MATLAB search path is necessary. The CasADi toolbox (Andersson et al., 2019) is used in MATLAB, and IPOPT (Wächter and Biegler, 2006) is used as the solver for decoupled NLPs. To solve the linear system, a conjugate-gradient technique (Nocedal and Wright, 2006, Algorithm 7.2) is implemented in order to avoid matrix-matrix multiplications, i.e., $J^\top J$.

Following Engelmann et al. (2018), the quantities in the following are used to illustrate the convergence behavior

(1) The deviation of optimization variables from the optimal value $\|x - x^*\|_\infty$.
(2) The primal residual, i.e., the violation of consensus constraint $\|Ax - b\|_\infty = \left\|\sum_{\ell\in\mathcal{R}} A_\ell x_\ell - b\right\|_\infty$.
(3) The dual residual $\gamma = \max_{\ell\in\mathcal{R}} \|\Sigma_\ell(x_\ell - z_\ell)\|_\infty$.
(4) The solution gap calculated as $|f(x) - f(x^*)|$, where $f(x^*)$ is provided by the centralized approach.

The user-defined tolerance ϵ is set to 10^{-8}, while the penalty parameters ρ and μ are set to 10^2.

5.1 Comparison of different combinations

For fair comparison, the primal variables x are initialized with the initial guess provided by MATPOWER (Zimmerman et al., 2010), while the dual variable λ is set to zero. RUNPF from MATPOWER is used to represent a centralized approach.

Table 2 displays the computing time of different combinations. The computing time of both algorithms also benefit from the dimensional reduction—compared with the original distributed AC PF model, the dimension by applying reduced modelling method is decreased almost by half.

What else stands out in this table is the fast computing time of the Gauss-Newton based inexact ALADIN (Algorithm 2). In contrast to solving NLP in a decoupled step of Algorithm 1, Algorithm 2 solves the equivalent linear systems of a quadratic approximation in both decoupled and coupled steps by exploiting the structure of the problem formulation. Consequently, the computation effort has been reduced dramatically. As a result, the computing time of solving the reduced distributed PF model by using Algorithm 2 is in the same order of magnitude with the centralized approach, and can be further improved by implementing parallel computing.

Note that all the test cases can converge within half a dozen iterations, whatever model or algorithm is applied. In summary, the dimensional reduction and the inexact approach has little impact on convergence rate, but can reduce the computational effort remarkably.

5.2 Convergence behavior of 10224-Bus system

Next, we study the convergence behavior of the largest test case, i.e., 10224-bus system. The test case is composed of six 1354-bus MATPOWER test cases, and seven 300-bus MATPOWER test cases. Thereby, there are 13 regions and 242 connecting-tie lines between neighboring regions, as presented in Table 2. Its connection graph of regions are shown in Figure 3. To solve the AC PF problem of the

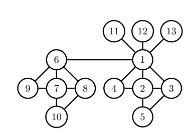

Fig. 3. Connection graph of 10224-bus test case

10224-Bus system, we use the reduced modelling method with the Gauss-Newton based inexact ALADIN algorithm. Figure 4 shows the four quantities in every iterate, i.e., the deviation of current variables from the optimal value, the primal residual, the dual residual and the solution gap. Within half a dozen iterates, the new ALADIN algorithm converges to the optimal solution with high accuracy, as presented in Table 3. At the same time, a locally quadratic convergence rate can be observed from Figure 4.

[2] The code is available on https://github.com/xinliang-dai/rapidPF.

Table 2. Computing time for solving power flow problem with different combinations

Buses	n^{reg}	n^{conn}	Original model			Reduced model			centralized
			Dimension	standard[s]	inexact[s]	Dimension	standard[s]	inexact[s]	
53	3	5	232	0.143	0.027	126	0.114	0.011	0.004
418	2	8	1704	0.485	0.068	868	0.315	0.028	0.014
2708	2	30	10952	3.913	0.236	5536	2.149	0.109	0.051
4662	5	130	19168	10.442	0.451	9844	5.694	0.228	0.129
10224	13	242	41864	25.909	0.996	21416	14.392	0.591	0.257

Table 3. The deviation of the 10224-bus system from the optimizer by applying reduced modeling method with the inexact ALADIN

	θ [rad]	v [p.u.]	p [p.u.]	q [p.u.]
$\|\cdot\|_\infty$	1.7×10^{-8}	7.5×10^{-9}	5.7×10^{-7}	3.2×10^{-6}

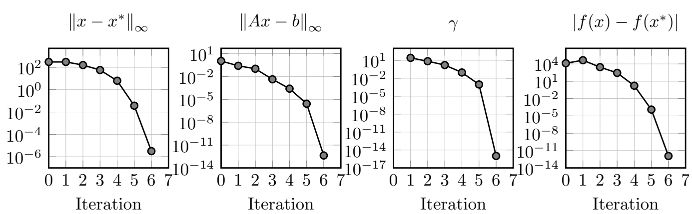

Fig. 4. Convergence behavior of 10224-bus system by applying reduced modeling method with the inexact ALADIN

6. CONCLUSIONS

The present paper investigates the application of a new tailored version of ALADIN for solving the AC power flow (PF) problem. Compared with the previous work by Mühlpfordt et al. (2021), the dimension can be reduced by half by applying reduced modelling method. Theoretically, the convergence rate would be traded off for improvement on computing time by applying the Gauss-Newton based inexact ALADIN. However, the simulation results show no difference between exact and inexact ALADIN in terms of the number of iterations. Besides, no external NLP solver is needed. In general, this new combination is of great potential for handling large-scale systems, and turns out to be as efficient as a centralized approach. For future work, efforts toward parallel computing will be made to reduce the computing time even further, and an extension to the AC OPF problem will be investigated.

REFERENCES

Andersson, J.A., Gillis, J., Horn, G., Rawlings, J.B., and Diehl, M. (2019). Casadi: a software framework for nonlinear optimization and optimal control. *Math. Program. Comput.",* 11(1), 1–36.

Du, X., Engelmann, A., Jiang, Y., Faulwasser, T., and Houska, B. (2019). Distributed state estimation for AC power systems using Gauss-Newton ALADIN. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, 1919–1924. IEEE.

Engelmann, A., M"uhlpfordt, T., Jiang, Y., Houska, B., and Faulwasser, T. (2017). Distributed AC optimal power flow using ALADIN. In *In Proceedings of the 20th IFAC World Congress, Toulouse, France*, 5701–5706.

Engelmann, A., Jiang, Y., Benner, H., Ou, R., Houska, B., and Faulwasser, T. (2020). ALADIN-α—an open-source matlab toolbox for distributed non-convex optimization. *Optimal Control Applications and Methods*.

Engelmann, A., Jiang, Y., Mühlpfordt, T., Houska, B., and Faulwasser, T. (2018). Toward distributed OPF using ALADIN. *IEEE Transactions on Power Systems*, 34(1), 584–594.

Erseghe, T. (2014). Distributed optimal power flow using ADMM. *IEEE transactions on power systems*, 29(5), 2370–2380.

Frank, S. and Rebennack, S. (2016). An introduction to optimal power flow: Theory, formulation, and examples. *IIE transactions*, 48(12), 1172–1197.

Grainger, J.J. (1999). *Power system analysis.* McGraw-Hill.

Houska, B., Frasch, J., and Diehl, M. (2016). An augmented Lagrangian based algorithm for distributed nonconvex optimization. *SIAM Journal on Optimization*, 26(2), 1101–1127.

Jiang, Y., Sauerteig, P., Houska, B., and Worthmann, K. (2021a). Distributed optimization using ALADIN for MPC in smart grids. *IEEE Transactions on Control Systems Technology*, 29(5), 2142–2152.

Jiang, Y., Kouzoupis, D., Yin, H., Diehl, M., and Houska, B. (2021b). Decentralized optimization over tree graphs. *Journal of Optimization Theory and Applications*, 189(2), 384–407.

Meyer-Huebner, N., Suriyah, M., and Leibfried, T. (2019). Distributed optimal power flow in hybrid AC–DC grids. *IEEE Transactions on Power Systems*, 34(4), 2937–2946.

Mühlpfordt, T., Dai, X., Engelmann, A., and Hagenmeyer, V. (2021). Distributed power flow and distributed optimization—formulation, solution, and open source implementation. *Sustainable Energy, Grids and Networks*, 26, 100471.

Nocedal, J. and Wright, S. (2006). *Numerical optimization.* Springer Science & Business Media.

Sun, H., Guo, Q., Zhang, B., Guo, Y., Li, Z., and Wang, J. (2014). Master–slave-splitting based distributed global power flow method for integrated transmission and distribution analysis. *IEEE Transactions on Smart Grid*, 6(3), 1484–1492.

Sun, H. and Zhang, B. (2008). Distributed power flow calculation for whole networks including transmission and distribution. In *2008 IEEE/PES Transmission and Distribution Conference and Exposition*, 1–6. IEEE.

Wächter, A. and Biegler, L.T. (2006). On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Math. Program.*, 106(1), 25–57.

Zimmerman, R.D., Murillo-Sánchez, C.E., and Thomas, R.J. (2010). Matpower: Steady-state operations, planning, and analysis tools for power systems research and education. *IEEE Trans. Power Syst.*, 26(1), 12–19.