



ENGINEERING MATHEMATICS
AND COMPUTING LAB



UNIVERSITÄT
HEIDELBERG
ZUKUNFT
SEIT 1386

A Domain Decomposition Approach for Solving Dynamic Optimal Power Flow Problems in Parallel with Application to the German Transmission Grid

Philipp Gerstner, Michael Schick, Vincent Heuveline, Nico Meyer-Hübner, Michael Suriyah, Thomas Leibfried, Viktor Slednev, Wolf Fichtner, Valentin Bertsch

Preprint No. 2016-01

Preprint Series of the Engineering Mathematics and Computing Lab (EMCL)





Preprint Series of the Engineering Mathematics and Computing Lab (EMCL)

ISSN 2191-0693

Preprint No. 2016-01

The EMCL Preprint Series contains publications that were accepted for the Preprint Series of the EMCL. Until April 30, 2013, it was published under the roof of the Karlsruhe Institute of Technology (KIT). As from May 01, 2013, it is published under the roof of Heidelberg University.

A list of all EMCL Preprints is available via Open Journal System (OJS) on <http://archiv.ub.uni-heidelberg.de/ojs/index.php/emcl-pp/>

For questions, please email to

info.at.emcl-preprint@uni-heidelberg.de

or directly apply to the below-listed corresponding author.

Affiliation of the Authors

Philipp Gerstner^{a,b,1}, Michael Schick^{a,b}, Vincent Heuveline^{a,b}, Nico Meyer-Hübner^c, Michael Suriyah^c, Thomas Leibfried^c, Viktor Slednev^d, Wolf Fichtner^d, Valentin Bertsch^e

^a*Engineering Mathematics and Computing Lab (EMCL), Interdisciplinary Center for Scientific Computing (IWR), Heidelberg University, Germany*

^b*Data Mining and Uncertainty Quantification Group (DMQ), Heidelberg Institute for Theoretical Studies (HITS), Germany*

^c*Institute of Electric Energy Systems and High-Voltage Technology, Karlsruhe Institute of Technology (KIT), Germany*

^d*Institute for Industrial Production, Karlsruhe Institute of Technology (KIT), Germany*

^e*Energy Policy Research Centre, Economic and Social Research Institute, Dublin, Ireland*

¹*Corresponding Author: Philipp Gerstner, philipp.gerstner@uni-heidelberg.de*

Impressum

Heidelberg University

Interdisciplinary Center for Scientific Computing (IWR)

Engineering Mathematics and Computing Lab (EMCL)

Im Neuenheimer Feld 205,

69120 Heidelberg

Germany

Published on the Internet under the following Creative Commons License:

<http://creativecommons.org/licenses/by-nc-nd/3.0/de> .



<http://emcl.iwr.uni-heidelberg.de>

A Domain Decomposition Approach for Solving Dynamic Optimal Power Flow Problems in Parallel with Application to the German Transmission Grid

Philipp Gerstner, Michael Schick, Vincent Heuveline, Nico Meyer-Hübner, Michael Suriyah,
Thomas Leibfried, Viktor Slednev, Wolf Fichtner, Valentin Bertsch

November 11, 2016

Abstract

We propose a parallel solver for linear systems of equations arising from the application of Primal Dual Interior Point methods to Dynamic Optimal Power Flow problems. Our solver is based on the Generalized Minimal Residual method in combination with an additive Schwarz domain decomposition method as preconditioner. This preconditioner exploits the structure of Dynamic Optimal Power Flow problems which, after linearization, is given as block-tridiagonal matrix with large diagonal blocks and only few off-diagonal entries. These entries correspond to intertemporal couplings due to ramping and energy storage constraints and are partially neglected in order to induce parallelism. We test our method on a large-scale optimization problem based on data of the German transmission grid and show that a significant parallel speedup can be obtained.

1 Introduction

In many countries, the energy sector continues to undergo substantial changes. The expansion of renewable energy sources (RES) necessitates an extensive structural rearrangement of the power system with the power grid taking center stage. While today's power grid infrastructure has been designed for centralized and controllable power generation in conventional power plants, the RES expansion leads to an increasingly uncertain, volatile and decentralized generation. In order to ensure a dependable grid operation and to maintain today's security of supply in the light of these developments, especially in times of peak generation or demand, methods are needed which are able to consider a high resolution of regional and temporal input data. Inevitably, this requirement leads to a target conflict between model complexity and computational intensity on the one hand and model accuracy on the other hand [18]. It is therefore a central challenge to provide efficient methods for power grid optimization, including an accurate consideration of non-linear and non-convex AC power flow constraints.

The problem of finding the optimal operating state of a given power grid, also known as *Optimal Power Flow* (OPF), is stated as the minimization of a cost function with respect to a vector of continuous optimization variables like node voltages, generated active power and generated reactive power, see e.g., [22], [14]. A feasible vector has to satisfy the AC power flow equations based on Kirchhoff's circuit law in order to guarantee that the power demand is covered by the generated power. In addition, a set of inequalities has to be fulfilled to ensure that no technical restrictions, like transmission line limits, are violated.

For the problem of *Dynamic Optimal Power Flow* (DOPF), one considers several OPF problems, each corresponding to one specific point in time. Here, the power demand is time-dependent and one has to take into account additional constraints that couple optimization variables corresponding to different time steps. Among others, these couplings are introduced by energy storage facilities and ramping constraints

for conventional power plants [23]. Since DOPF is a large-scale non-linear optimization problem, solving it on a parallel computer architecture is of crucial importance.

In this context, *Primal Dual Interior Point Methods* (PDIPM) have proven to be among the most powerful optimization algorithms because their number of iterations to obtain convergence is rather insensitive to the problem size. Moreover, the main computational effort when applying PDIPM lies in the solution of linear systems of equations which is a suitable task to be carried out in parallel on multi-core CPUs.

There exist a few works on parallel solution of linear systems that arise from PDIPM applied to DOPF. One such approach is based on parallel matrix factorization by means of Schur complement techniques [7], [23]. Other strategies use Benders Decomposition to decompose the complete optimization problem into smaller ones [1].

Our contribution in this paper is the use of *Schwarz Domain Decomposition Methods* as preconditioner for Krylov-type iterative methods for solving linear systems of equations in parallel. The original Schwarz method was formulated in 1870 as theoretical tool for proving existence of elliptic PDEs on complicated domains [17]. Later on, modifications of it have been used as stand-alone iterative methods for solving PDEs and have become a standard technique for preconditioning Krylov-type methods in context of PDEs [19].

We apply these techniques for DOPF by decomposing the time period of interest into several smaller subdomains, which allows the use of multiple cores for computation. All modifications are done on an algebraic level and allow the use of *Inexact Interior Point Methods*, which may further reduce the overall computational effort.

We apply our method to a large scale DOPF problem that is based on data of the German transmission grid and show that we can achieve a significant speedup compared to the sequential and parallel state of the art direct solver Mumps [2].

This paper is structured in the following way: In Section 2, we describe the DOPF problem, followed by a brief review on the application of PDIPM in Section 3. In Section 4, we formulate an additive Schwarz method for solving linear systems that arise in the context of DOPF. In Section 5, we investigate the numerical behavior of our method based on two test cases, followed by a critical review of our approach in Section 6. We conclude this work in Section 7.

2 Dynamic Optimal Power Flow

In this section, we formulate the previously described DOPF problem in a mathematically rigorous way. This formulation has already been stated in more detail by several authors, e.g. in [23], [11] and [16]. We remark that we consider a continuous formulation of the DOPF problem without any discrete decision variables. A nonlinear mixed-integer formulation of OPF is considered, e.g., in [20].

2.1 Formulation of DOPF

In order to formulate the DOPF problem for a given time period of interest $[0, T]$, we consider a uniform partition $0 = T_1 < T_2 < \dots < T_{N_T} = T$ with constant step size $\tau = T_t - T_{t-1}$ for all $t \in \mathcal{T} \setminus \{1\}$, where $\mathcal{T} := \{1, \dots, N_T\}$.

The power grid, consisting of $N_{\mathcal{B}}$ nodes denoted by $\mathcal{B} := \{1, \dots, N_{\mathcal{B}}\}$ and $N_{\mathcal{E}}$ transmission lines denoted by $\mathcal{E} \subset \mathcal{B} \times \mathcal{B}$, is described by an admittance matrix

$$Y = G + jB \in \mathbb{C}^{N_{\mathcal{B}} \times N_{\mathcal{B}}} \text{ with } G, B \in \mathbb{R}^{N_{\mathcal{B}} \times N_{\mathcal{B}}} \text{ and } j = \sqrt{-1}. \quad (2.1)$$

It holds $Y = Y^T$ and $Y_{lk} = Y_{kl} \neq 0$ if and only if there is a branch connecting node k and l , i.e., $kl \in \mathcal{E}$ and $lk \in \mathcal{E}$. Therefore, Y is a sparse matrix for most real world power grids.

The complex voltage at node $k \in \mathcal{B}$ for time step $t \in \mathcal{T}$ is given by

$$V_k^t = E_k^t + jF_k^t \quad (2.2)$$

with real part $E_k^t \in \mathbb{R}$ and imaginary part $F_k^t \in \mathbb{R}$. We write $E^t := [E_k^t]_{k \in \mathcal{B}}$ and $F^t := [F_k^t]_{k \in \mathcal{B}}$.

Furthermore, we define by $\mathcal{C} := \{1, \dots, N_{\mathcal{C}}\}$ the set of conventional power plants, by $\mathcal{R} := \{1, \dots, N_{\mathcal{R}}\}$ the set of renewable energy sources, by $\mathcal{S} := \{1, \dots, N_{\mathcal{S}}\}$ the set of storage facilities, by $\mathcal{LS} := \{1, \dots, N_{\mathcal{LS}}\}$

the set of load shedding possibilities, by $\mathcal{DC} := \{1, \dots, N_{\mathcal{DC}}\}$ the set of virtual generators for modeling DC transmission lines and by $\mathcal{V} := \{1, \dots, N_{\mathcal{V}}\}$ the set of virtual generators for modeling the import / export of power from / to neighboring countries.

The resulting vector of variables corresponding to active power and energy for time step t is given by

$$P^t = \begin{pmatrix} [P_{g,i}^t]_{i \in \mathcal{C}} & [P_{r,i}^t]_{i \in \mathcal{R}} & [P_{sg,i}^t]_{i \in \mathcal{S}} & [P_{sl,i}^t]_{i \in \mathcal{S}} & [SOC_i^t]_{i \in \mathcal{S}} & \dots \\ \dots & [P_{l,i}^t]_{i \in \mathcal{LS}} & [P_{dc+,i}^t]_{i \in \mathcal{DC}} & [P_{dc-,i}^t]_{i \in \mathcal{DC}} & [P_{ex,i}^t]_{i \in \mathcal{V}} & [P_{im,i}^t]_{i \in \mathcal{V}} \end{pmatrix} \in \mathbb{R}^{N_P}. \quad (2.3)$$

The definitions of the individual components of P^t are stated in Appendix A.

In a similar way, the vector of reactive power injections is defined by

$$Q^t = ([Q_{g,i}^t]_{i \in \mathcal{C}} \quad [Q_{r,i}^t]_{i \in \mathcal{R}} \quad [Q_{sg,i}^t]_{i \in \mathcal{S}} \quad [Q_{l,i}^t]_{i \in \mathcal{LS}}) \in \mathbb{R}^{N_Q}. \quad (2.4)$$

Every variable related to active and reactive power can be assigned to a specific grid node by means of power injection / extraction matrices $C_P \in \{-1, 0, 1\}^{N_{\mathcal{B}} \times N_P}$ and $C_Q \in \{-1, 1\}^{N_{\mathcal{B}} \times N_Q}$. Since the states of charge SOC of storage facilities do not directly influence the power flow equations, the corresponding rows in C_P are equal to the zero vector.

2.1.1 Constraints

Denoting the active and reactive power load at node k for time step t by $L_{p,k}^t$ and $L_{q,k}^t$, respectively, the AC power flow equations are given by [21]

$$C_P P^t - L_p^t - Pf_p(E^t, F^t) = 0 \in \mathbb{R}^{N_{\mathcal{B}}}, \quad (2.5)$$

$$C_Q Q^t - L_q^t - Pf_q(E^t, F^t) = 0 \in \mathbb{R}^{N_{\mathcal{B}}}, \quad (2.6)$$

with

$$Pf_{p,k}(E, F) = \sum_{l=1}^{N_{\mathcal{B}}} G_{kl}(E_k E_l + F_k F_l) + B_{kl}(F_k F_l - E_k E_l), \quad k \in \mathcal{B}, \quad (2.7)$$

$$Pf_{q,k}(E, F) = \sum_{l=1}^{N_{\mathcal{B}}} G_{kl}(F_k E_l - E_k E_l) - B_{kl}(E_k E_l + F_k F_l), \quad k \in \mathcal{B}, \quad (2.8)$$

where we skipped the index t . $Pf_{p,k}$ and $Pf_{q,k}$ can also be written in terms of active and reactive power flow p_{kl}, q_{kl} over all lines incident to node k [22]:

$$Pf_{p,k}(E, F) = \sum_{l \neq k, Y_{kl} \neq 0} p_{kl}(E_k, E_l, F_k, F_l), \quad (2.9)$$

$$Pf_{q,k}(E, F) = \sum_{l \neq k, Y_{kl} \neq 0} q_{kl}(E_k, E_l, F_k, F_l). \quad (2.10)$$

In the following, we denote the equations (2.5), (2.6) by

$$AC(E^t, F^t, P^t, Q^t) = 0. \quad (2.11)$$

Due to technical restrictions arising from power plants, renewable energy sources, storage facilities and DC transmission lines, one has to impose lower and upper bounds for active and reactive power injections, given by

$$P_{min}^t \leq P^t \leq P_{max}^t, \quad \text{and} \quad Q_{min}^t \leq Q^t \leq Q_{max}^t. \quad (2.12)$$

For our test cases, only bounds corresponding to power injected by renewable energy sources, P_r^t , and load shedding, P_l^t , depend on the time step t . The node voltages and the active power flow over transmission lines have to be bounded by

$$U_{min}^2 \leq (E^t)^2 + (F^t)^2 \leq U_{max}^2 \quad (2.13)$$

and

$$p_{kl}(E_k^t, E_l^t, F_k^t, F_l^t) \leq Pf_{max,kl} \quad \text{for all } kl \in \mathcal{E}, \quad (2.14)$$

respectively. Additionally, we impose ramping constraints for conventional power plants to bound the rate of change for injected power between consecutive time steps by

$$P_g^t - P_g^{t+1} \leq \tau Pr_{max} \text{ and } P_g^{t+1} - P_g^t \leq \tau Pr_{max} \text{ for } t = 1, \dots, N_T - 1. \quad (2.15)$$

Besides the power flow equations (2.11), one has to take into account equality constraints for the slack bus, given by $F_1^t = 0$ for all t , and for virtual generators $k, l \in \mathcal{DC}$ modeling a DC line \overline{kl} :

$$P_{dc+,k}^t = -\nu_{DC} P_{dc-,l}^t \text{ and } P_{dc+,l}^t = -\nu_{DC} P_{dc-,k}^t. \quad (2.16)$$

Here, line losses are taken into account by means of the factor $\nu_{DC} \in (0, 1)$.

The extracted reactive power due to load shedding is modeled by the equality constraint $Q_{l,i}^t = \frac{L_{p,k}^t}{L_{p,k}^t} P_{l,i}^t$ for $i \in \mathcal{LS}$ and $k \in \mathcal{B}$ is the node, to which the load shedding process is assigned.

Finally, an additional set of equality constraints has to be imposed for modeling the state of charge, SOC_i^t , of storage facilities. These constraints are given by

$$SOC_i^{t+1} = SOC_i^t + \tau(\nu_l P_{sl,i}^t - \nu_g^{-1} P_{sg,i}^t), \quad i \in \mathcal{S} \quad (2.17)$$

with factors $\nu_l, \nu_g \in (0, 1)$ describing the efficiency of the loading process $P_{sl,i}^t$ and generation process $P_{sg,i}^t$, respectively.

2.1.2 Cost Function

The cost function f accounts for costs related to active power processes for the complete time interval $[0, T]$ and is composed by contributions from each time step $t \in \mathcal{T}$ [6]:

$$f(P) = \sum_{t=1}^{N_T} \tau f_t(P^t) \quad (2.18)$$

with

$$\begin{aligned} f_t(P^t) = & \sum_{i \in \mathcal{C}} (a_{\mathcal{C},i,2} (P_{g,i}^t)^2 + a_{\mathcal{C},i,1} P_{g,i}^t) + \sum_{i \in \mathcal{R}} (a_{\mathcal{R},i,2} (P_{r,i}^t)^2 + a_{\mathcal{R},i,1} P_{r,i}^t) \\ & + \sum_{i \in \mathcal{V}} (a_{ex,i,2} (P_{ex,i}^t)^2 + a_{ex,i,1} P_{ex,i}^t) + \sum_{i \in \mathcal{V}} (a_{im,i,2} (P_{im,i}^t)^2 + a_{im,i,1} P_{im,i}^t) \\ & + \sum_{i \in \mathcal{LS}} (a_{\mathcal{LS},i,2} (P_{l,i}^t)^2 + a_{\mathcal{LS},i,1} P_{l,i}^t) \end{aligned} \quad (2.19)$$

and coefficients $a_{*,*,*} \in \mathbb{R}$.

2.1.3 Optimization Problem

With the definitions of the previous sections at hand, the DOPF problem can be stated in an abstract way as

$$\begin{aligned} \min_x \quad & f(x) \text{ s.t.} \\ & g_I^t(x^t) = 0, \quad t = 1, \dots, N_T \\ & g_S^t(x^{t+1}, x^t) = 0, \quad t = 1, \dots, N_T - 1 \\ & h_I^t(x^t) \leq 0, \quad t = 1, \dots, N_T \\ & h_R^t(x^{t+1}, x^t) \leq 0, \quad t = 1, \dots, N_T - 1, \end{aligned} \quad (2.20)$$

where the vector of optimization variables is given by

$$x = [x^t]_{t \in \mathcal{T}} \in \mathbb{R}^{n^x} \text{ and } x^t = (E^t \quad F^t \quad P^t \quad Q^t) \in \mathbb{R}^{n^{x,t}}. \quad (2.21)$$

It holds $n^{x,t} = 2N_B + N_P + N_Q$ and $n^x = N_T n^{x,t}$.

Note that the ramping constraints h_R induced by (2.15) and storage constraints g_S induced by (2.17) establish the only couplings between variables of different time steps. Without them, a solution to (2.20) could be computed by solving N_T independent OPF problems. The optimization problem (2.20) is non-linear due to the AC equations and due to the voltage and line flow inequality constraints. Further, the latter ones and the AC equations are the sources of non-convexity in (2.20).

To simplify the notation in the following sections, we rewrite the optimization problem (2.20) as

$$\min_x f(x) \text{ s.t. } g(x) = 0, h(x) \leq 0 \quad (2.22)$$

with quadratic functions

$$f: \mathbb{R}^{n^x} \rightarrow \mathbb{R}, g: \mathbb{R}^{n^x} \rightarrow \mathbb{R}^{n^\lambda}, h: \mathbb{R}^{n^x} \rightarrow \mathbb{R}^{n^\mu}. \quad (2.23)$$

3 Primal Dual Interior Point Method for DOPF

In this section, we briefly describe the application of a *Primal Dual Interior Point Method* (PDIPM) for DOPF, as it is done in the package MATPOWER [24] and in [6].

3.1 Formulation of PDIPM

For the general optimization problem (2.22), the corresponding Lagrangian function is given by

$$L: \mathbb{R}^{n^x} \times \mathbb{R}^{n^\lambda} \times \mathbb{R}^{n^\mu} \rightarrow \mathbb{R}, (x, \lambda, \mu) \mapsto f(x) + \lambda^T g(x) + \mu^T h(x). \quad (3.1)$$

Assuming additional constraint qualifications like (LICQ), for every local minimum x^* of (2.22), there exist corresponding Lagrangian multipliers λ^*, μ^* such that (x^*, λ^*, μ^*) solves the following KKT-conditions [6]:

$$\mathcal{F}_0(x, s, \lambda, \mu) = \begin{pmatrix} \nabla_x L(x, \lambda, \mu) \\ g(x) \\ h(x) + s \\ S\mu \end{pmatrix} = 0, s, \mu \geq 0, \quad (3.2)$$

with a vector of slack variables $s \in \mathbb{R}^{n^\mu}$ and $S \in \mathbb{R}^{n^\mu \times n^\mu}$ denoting a diagonal matrix with diagonal elements $S_{kk} = s_k$.

We use the PDIPM for solving (3.2), which requires computing solutions $(x^{(k)}, \lambda^{(k)}, \mu^{(k)})$ of

$$\mathcal{F}_{\gamma}(x, s, \lambda, \mu) = \mathcal{F}_0(x, s, \lambda, \mu) - \begin{pmatrix} 0 & 0 & 0 & \gamma e \end{pmatrix}^T = 0, s, \mu \geq 0, \quad (3.3)$$

for a sequence of *barrier parameters* $\gamma_k = \sigma \frac{(\mu^{(k-1)})^T s^{(k-1)}}{n^\mu}$ with *centering parameter* $\sigma = 0.1$ and *complementary gap* $(\mu^{(k-1)})^T s^{(k-1)}$, see [6], [24] and [9]. Here, $e = (1, \dots, 1) \in \mathbb{R}^{n^\mu}$.

Remark We use a very basic PDIPM implementation without any globalization strategy and with a step size control that only ensures $(s^{(k)}, \mu^{(k)}) > 0$ for all k . Since we focus on parallel linear algebra techniques for DOPF problems, a locally convergent algorithm is sufficient for our purpose.

3.2 KKT Matrix in PDIPM

In every iteration of PDIPM, one has to solve the linear system

$$\nabla \mathcal{F}_{\gamma_k}(x^{(k)}, s^{(k)}, \lambda^{(k)}, \mu^{(k)}) \Delta^{(k)} = -\mathcal{F}_{\gamma_k}(x^{(k)}, s^{(k)}, \lambda^{(k)}, \mu^{(k)}). \quad (3.4)$$

In the following, we omit the iteration index k . Assuming that $s, \mu > 0$, the Newton system (3.4) can be transformed into a reduced, symmetric saddle point form

$$\underbrace{\begin{pmatrix} \nabla_{xx}^2 \mathcal{L}(x, \lambda, \mu) + (\nabla h(x))^T \Sigma (\nabla h(x)) & (\nabla g(x))^T \\ \nabla g(x) & 0 \end{pmatrix}}_{=: A(x, s, \lambda, \mu) =: A} \begin{pmatrix} \Delta x \\ \Delta \lambda \end{pmatrix} = \begin{pmatrix} r_x \\ r_\lambda \end{pmatrix} \quad (3.5)$$

with diagonal matrix $\Sigma = \text{diag}(\frac{\mu_1}{s_1}, \dots, \frac{\mu_n}{s_n})$.

When dealing with linear systems arising from PDIPM, one generally has to face the problem of ill-conditioning induced by the sequence of scaling matrices $\Sigma^{(k)}$. If strict complementary holds at the KKT point x^* and if $x^{(k)} \rightarrow x^*$, one can show that $\Sigma_{ii}^{(k)} \rightarrow \infty$ for increasing PDIPM iteration index k if h_i is an active inequality. On the other hand, $\Sigma_{ii}^{(k)} \rightarrow 0$ if h_i is inactive [12]. For this reason, many IPM software packages (like IPOPT [8]) use direct methods such as LDL^T -factorizations to solve the arising linear systems.

In contrast, iterative linear solvers like GMRES [15] are very sensitive to ill-conditioned matrices, unless a good preconditioner is used. In exchange, they offer a higher potential of parallelization and allow the use of inexact Interior Point methods, see Section 3.3.

The distribution of the spectrum $\sigma(K) = \{\lambda \in \mathbb{C}, \lambda \text{ is eigenvalue of } K\}$ of a matrix K is an indicator for the convergence behavior of GMRES applied to $Kv = b$. By rule of thumb, a spectrum which is clustered away from 0 is beneficial for the speed of convergence of GMRES.

For general non-linear optimization problems with corresponding KKT matrix

$$K = \begin{pmatrix} H & B^T \\ B & 0 \end{pmatrix}, \quad H \in \mathbb{R}^{n^x \times n^x}, \quad B \in \mathbb{R}^{n^\lambda \times n^x}, \quad n^\lambda \leq n^x, \quad (3.6)$$

the so called *constraint preconditioner* is an example of a widely used preconditioner. It is given by

$$\tilde{K} = \begin{pmatrix} \tilde{H} & B^T \\ B & 0 \end{pmatrix}, \quad (3.7)$$

with \tilde{H} being an approximation to H that is easy to factorize, e.g., $\tilde{H} = \text{diag}(H)$ [12].

One can show that $\sigma(\tilde{K}^{-1}K) = \{1\} \cup \tilde{\sigma}$ with $|\tilde{\sigma}| = n^x - n^\lambda$ [10]. Therefore, at most $n^x - n^\lambda$ eigenvalues of $\tilde{K}^{-1}K$ are not equal to 1. The distribution of these remaining eigenvalues depends on how well \tilde{H} approximates H on the nullspace of B .

In Section 4, we describe how to exploit the special structure given by (2.20) to construct a parallel preconditioner. Furthermore, we will see that there is a close relation between our proposed preconditioner and the constraint preconditioner defined above, see Section 4.4.

3.3 Inexact PDIPM

In contrast to direct methods, iterative solvers allow solving linear systems with prescribed accuracy. One can exploit this fact by means of inexact Interior Point methods. Within these methods, the linear systems corresponding to the first PDIPM iterations are solved with a rather low accuracy and therefore a low number of iterations. As the PDIPM iterate approaches the exact solution, the accuracy of the linear solver is increased. We use the approach from [4] for determining the accuracy in each PDIPM iteration. Here, (3.4) has to be solved with residual $\tilde{r}^{(k)}$, i.e.,

$$\nabla \mathcal{F}_{\gamma_k}(x^{(k)}, s^{(k)}, \lambda^{(k)}, \mu^{(k)}) \Delta^{(k)} = -\mathcal{F}_{\gamma_k}(x^{(k)}, s^{(k)}, \lambda^{(k)}, \mu^{(k)}) + \tilde{r}^{(k)}, \quad (3.8)$$

that satisfies $\|\tilde{r}^{(k)}\|_2 \leq \tilde{\eta}_k \frac{(\mu^{(k)})^T s^{(k)}}{n^\mu}$ with forcing sequence $\tilde{\eta}_k \in (0, 1)$. With this choice, the update step $\Delta^{(k)}$ satisfies

$$\nabla \mathcal{F}_0(x^{(k)}, s^{(k)}, \lambda^{(k)}, \mu^{(k)}) \Delta^{(k)} = -\mathcal{F}_0(x^{(k)}, s^{(k)}, \lambda^{(k)}, \mu^{(k)}) + r^{(k)} \quad (3.9)$$

with $\|r^{(k)}\|_2 \leq (\gamma_k + \tilde{\eta}_k) \|\mathcal{F}_0(x^{(k)}, s^{(k)}, \lambda^{(k)}, \mu^{(k)})\|_2$. Therefore, " $\gamma_k + \tilde{\eta}_k$ can be viewed as forcing sequence of inexact Newton methods" [4]. We use $\tilde{\eta}_k = 0.1$ and stop the linear solver additionally if it reaches a relative residual norm of 10^{-10} . When using a direct solver, we observed a relative residual norm of similar magnitude.

Remark Since we solve the reduced linear system (3.5), the obtained residual is not equal to the prescribed residual $\tilde{r}^{(k)}$ of the unreduced system (3.4). However, one can show that the l^2 -norms of both residuals coincide, see Appendix C.

3.4 Termination criteria for PDIPM

We use the following criteria for monitoring the progress of PDIPM [24]:

$$c_{feas}(x, s, \lambda, \mu) := \frac{\max\{\|g(x)\|_\infty, \max_i\{h_i(x)\}\}}{1 + \max\{\|x\|_\infty, \|s\|_\infty\}} \quad (3.10)$$

$$c_{grad}(x, s, \lambda, \mu) := \frac{\|\nabla_x L(x, \lambda, \mu)\|_\infty}{1 + \max\{\|\lambda\|_\infty, \|\mu\|_\infty\}} \quad (3.11)$$

$$c_{comp}(x, s, \lambda, \mu) := \frac{(\mu^T s)}{1 + \|x\|_\infty} \quad (3.12)$$

$$c_{cost}(x, s, \lambda, \mu) := \frac{|f(x) - f(x^-)|}{1 + |f(x^-)|} \quad (3.13)$$

with x^- denoting the previous iterate.

For our test cases, numerical experiments show that in each IPM step the complementary condition c_{comp} is several orders of magnitude higher than the other criteria. In order to keep the number of IPM iterations rather short, we choose a termination tolerance ϵ_{comp} for c_{comp} that is higher than the tolerances for the other conditions, e.g. $10^{-2}\epsilon_{comp} = \epsilon_{feas} = \epsilon_{grad} = \epsilon_{cost}$. Of course, this approach may lead to suboptimal approximate solutions. One can get an indicator for the resulting error in the cost function by means of the following result:

Lemma 3.1. *Let $z^* = (x^*, s^*, \lambda^*, \mu^*)$ solve (3.2) and let $\tilde{z} = (\tilde{x}, \tilde{s}, \tilde{\lambda}, \tilde{\mu})$ be such that*

$$\mathcal{F}_0(\tilde{x}, \tilde{s}, \tilde{\lambda}, \tilde{\mu}) = \begin{pmatrix} 0 & 0 & 0 & * \end{pmatrix}^T \quad (3.14)$$

with $\tilde{s}, \tilde{\mu} \geq 0$. Then it holds

$$f(\tilde{x}) - f(x^*) = \frac{1}{2}(\tilde{s}^T \tilde{\mu} + \tilde{s}^T \mu^* - \tilde{\mu}^T s^*) + \Delta \quad (3.15)$$

with $\Delta = \frac{1}{12} \max_{t \in [0,1]} l'''(t)$ and

$$l(t) = L((x^*, \lambda^*, \mu^* + te), e = (\tilde{x} - x^*, \tilde{\lambda} - \lambda^*, \tilde{\mu} - \mu^*)). \quad (3.16)$$

Proof See Appendix B.

Assuming that the k -th PDIPM iterate $z^{(k)}$ is a good approximation to z^* with $c_{feas}(z^{(k)})$ and $c_{grad}(z^{(k)})$ being small, one might apply Lemma 3.1, neglect the third-order term $\Delta = O(\|e\|^3)$ and approximate $(s^{(k)})^T \mu^* - (\mu^{(k)})^T s^* \approx (s^{(k)})^T \mu^{(k+1)} - (\mu^{(k)})^T s^{(k+1)}$. This leads to an error indicator for the cost function at iteration k given by

$$f(x^{(k)}) - f(x^*) \approx \frac{1}{2}((s^{(k)})^T \mu^{(k)} + (s^{(k)})^T \mu^{(k+1)} - (\mu^{(k)})^T s^{(k+1)}) =: E_k. \quad (3.17)$$

E_k can be used to decide when to terminate the PDIPM algorithm in order to compute the optimal value with some prescribed accuracy, under the assumption that $c_{feas}, c_{grad}, c_{cost}$ already satisfy their respective tolerances.

4 Schwarz Domain Decomposition Method for DOPF

The key idea behind Schwarz domain decomposition methods for DOPF problems is the decomposition of the entire set of time steps \mathcal{T} into several smaller subsets. Then, one can define corresponding submatrices of the global matrix A defined in (3.5) that have smaller dimension and therefore are faster to factorize. Furthermore, the factorization of these submatrices can be done in parallel. After computing subsolutions corresponding to these submatrices, one can reconstruct an approximate solution for the global system (3.5), which is used as preconditioner within a Krylov-type iterative solver.

In Section 4.1, we present the formulation of the *additive Schwarz Method* (ASM) in the context of DOPF problems. In Section 4.2, we briefly describe some issues related to its implementation. Section 4.3 is intended to provide a deeper insight into the subproblems that have to be solved when applying ASM. Finally, we describe the relation between ASM and the above mentioned constraint preconditioner in Section 4.4.

4.1 Mathematical Formulation of Additive Schwarz Method

To apply ASM as preconditioner for the KKT matrix $A = A(x, s, \lambda, \mu) \in \mathbb{R}^{n \times n}$ defined by (3.5), we decompose the set of time steps $\mathcal{T} = \{1, \dots, N_T\}$ into q non-overlapping subdomains:

$$\mathcal{T} = \bigcup_{l=1}^q \tilde{\mathcal{T}}_l, \quad \tilde{\mathcal{T}}_l \cap \tilde{\mathcal{T}}_k = \emptyset \text{ for } k \neq l, \quad \tilde{\mathcal{T}}_l = \{\tilde{t}_l^-, \tilde{t}_l^- + 1, \dots, \tilde{t}_l^+\}. \quad (4.1)$$

Afterward, each subdomain $\tilde{\mathcal{T}}_l$ is augmented by additional s_{ol} time steps on both ends, yielding an overlapping decomposition of \mathcal{T} (see Figure 1):

$$\mathcal{T} = \bigcup_{l=1}^q \mathcal{T}_l, \quad \mathcal{T}_l := \{t_l^-, t_l^- + 1, \dots, t_l^+\}, \quad (4.2)$$

with

$$t_l^- = \begin{cases} \tilde{t}_l^- - s_{ol}, & l > 1 \\ \tilde{t}_l^-, & l = 1 \end{cases}, \quad t_l^+ = \begin{cases} \tilde{t}_l^+ + s_{ol}, & l < q \\ \tilde{t}_l^+, & l = q \end{cases}. \quad (4.3)$$

Typically, $s_{ol} \in \{1, 2\}$.

In the following, let $n^{\lambda,t}$ be the number of equality constraints corresponding to time step t , i.e. $\lambda^t \in \mathbb{R}^{n^{\lambda,t}}$ with $\lambda^t = (\lambda_I^t, \lambda_S^t)$ denoting the Lagrangian multipliers corresponding to g_I^t and g_S^t , respectively. Then, we define $n_l^x := \sum_{t \in \mathcal{T}_l} n^{x,t}$, $n_l^\lambda := \sum_{t \in \mathcal{T}_l} n^{\lambda,t}$ and $n_l = n_l^x + n_l^\lambda$, $n = n^x + n^\lambda$. Analogously, $\mu^t = (\mu_I^t, \mu_R^t) \in \mathbb{R}^{n^{\mu,t}}$ denotes the Lagrangian multipliers for the inequality constraints h_I^t and h_R^t . $s^t = (s_I^t, s_R^t) \in \mathbb{R}^{n^{\mu,t}}$ denotes the corresponding slack variables. In the following, for all expressions involving $\tilde{\cdot}$, \mathcal{T}_l is replaced by $\tilde{\mathcal{T}}_l$.

When restricting optimization variables to their respective components contained in \mathcal{T}_l , we write

$$x_{[l]} = [x^t]_{t \in \mathcal{T}_l}, \quad (4.4)$$

$$\lambda_{[l]} = \begin{bmatrix} (\lambda_I^t) \\ (\lambda_S^t) \end{bmatrix}_{t \in \mathcal{T}_l}, \quad (4.5)$$

$$\mu_{I,[l]} = [\mu_I^t]_{t \in \mathcal{T}_l}, \quad \mu_{R,[l]} = [\mu_R^t]_{t \in \mathcal{T}_l \cup \{t_l^- - 1\}}, \quad (4.6)$$

$$s_{I,[l]} = [s_I^t]_{t \in \mathcal{T}_l}, \quad s_{R,[l]} = [s_R^t]_{t \in \mathcal{T}_l \cup \{t_l^- - 1\}}. \quad (4.7)$$

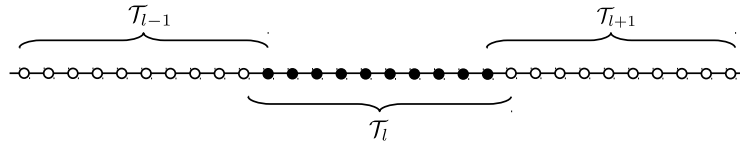


Figure 1: Decomposition of time steps with overlap $s_{ol} = 1$.

The constraint functions g and h are restricted accordingly by

$$g_{[l]}(x_{[l]}, x^{t_i^+ + 1}) = \left[\left(\begin{array}{c} g_I^t(x^t) \\ g_S^t(x^{t+1}, x^t) \end{array} \right)_{t \in \mathcal{T}_l} \right] \quad (4.8)$$

$$h_{I,[l]}(x_{[l]}) = [h_I^t(x^t)]_{t \in \mathcal{T}_l}, \quad (4.9)$$

$$h_{R,[l]}(x_{[l]}, x^{t_i^- - 1}, x^{t_i^+ + 1}) = [h_R^t(x^{t+1}, x^t)]_{t \in \mathcal{T}_l \cup \{t_i^- - 1\}}, \quad (4.10)$$

$$h_{[l]}(x_{[l]}, x^{t_i^- - 1}, x^{t_i^+ + 1}) = \left(\begin{array}{c} h_{I,[l]}(x_{[l]}) \\ h_{R,[l]}(x_{[l]}, x^{t_i^- - 1}, x^{t_i^+ + 1}) \end{array} \right) \quad (4.11)$$

and the local scaling matrices are defined by

$$\Sigma_{I,[l]} = \text{diag} \left(\frac{\mu_{I,[l],1}}{s_{I,[l],1}}, \dots, \frac{\mu_{I,[l],|\mu_{I,[l]}|}}{s_{I,[l],|\mu_{I,[l]}|}} \right), \quad (4.12)$$

$$\Sigma_{R,[l]} = \text{diag} \left(\frac{\mu_{R,[l],1}}{s_{R,[l],1}}, \dots, \frac{\mu_{R,[l],|\mu_{R,[l]}|}}{s_{R,[l],|\mu_{R,[l]}|}} \right), \quad (4.13)$$

$$\Sigma_{[l]} = \begin{pmatrix} \Sigma_{I,[l]} & 0 \\ 0 & \Sigma_{R,[l]} \end{pmatrix}. \quad (4.14)$$

For $l = 1, \dots, q$ we define by

$$\begin{pmatrix} \Delta x \\ \Delta \lambda \end{pmatrix}_{[l]} := \begin{pmatrix} \Delta x_{[l]} \\ \Delta \lambda_{[l]} \end{pmatrix} \in \mathbb{R}^{n_l} \quad (4.15)$$

the components of the solution vector of (3.5) that can be assigned to time steps in \mathcal{T}_l . Let further $R_l \in \{0, 1\}^{n_l \times n}$ be the restriction matrix corresponding to subset \mathcal{T}_l such that

$$R_l \begin{pmatrix} \Delta x \\ \Delta \lambda \end{pmatrix} = \begin{pmatrix} \Delta x \\ \Delta \lambda \end{pmatrix}_{[l]}, \quad R_l = \begin{pmatrix} R_l^x & 0 \\ 0 & R_l^\lambda \end{pmatrix} \quad (4.16)$$

holds with local restriction matrices of the form

$$R_l^x = \begin{pmatrix} 0 & I_{n_l^x} & 0 \end{pmatrix} \in \{0, 1\}^{n_l^x \times n^x}, \quad R_l^\lambda = \begin{pmatrix} 0 & I_{n_l^\lambda} & 0 \end{pmatrix} \in \{0, 1\}^{n_l^\lambda \times n^\lambda}. \quad (4.17)$$

Here, I_m denotes the identity matrix in \mathbb{R}^m and the size of the zero matrices may vary for each restriction matrix and each l . Analogously, \tilde{R}_l is defined as restriction operator with respect to the non-overlapping subdomain $\tilde{\mathcal{T}}_l$.

With these definitions at hand, one can define local submatrices of A by

$$A_l := R_l A R_l^T. \quad (4.18)$$

In Section 4.3, we analyze the structure of these submatrices.

In the following, we assume that all submatrices A_l are non-singular. Then the *multiplicative Schwarz Method* (MSM) for approximately solving $Av = b$ is given by [19]:

Algorithm 1. *Multiplicative Schwarz Method* $v = \text{MSM}(b)$

Set $v_0 = 0$

For $l = 1, \dots, q$

$$v_l = v_{l-1} + R_l^T A_l^{-1} R_l (b - A v_{l-1})$$

Return $v = v_q$

In every iteration l , the current error with respect to the exact solution, $e_{l-1} = v^* - v_{l-1}$, satisfies $A e_{l-1} = r_{l-1}$ with residual $r_{l-1} = b - A v_{l-1}$. By restricting r_{l-1} to subdomain \mathcal{T}_l and solving with A_l^{-1} , one obtains an approximation $\tilde{e}_l := A_l^{-1} R_l r_{l-1}$ to the exact error $R_l e_{l-1}$ on subdomain \mathcal{T}_l . Prolongation with R_l^T yields a correction $d_l = R_l^T \tilde{e}_l$ to the current approximation v_{l-1} . Thus, the multiplicative Schwarz method can be seen as defect correction algorithm.

Unfortunately, the MSM is a sequential algorithm. In order to parallelize it, one omits residual updates in each iteration, yielding the ASM [19]:

Algorithm 2. *Additive Schwarz Method* $v = \text{ASM}(b)$

Set $v_0 = 0$

For $l = 1, \dots, q$

$$v_l = v_{l-1} + R_l^T A_l^{-1} R_l b$$

Return $v = v_q$

which can be written as

$$v = M_{ASM} b \quad \text{with} \quad M_{ASM} := \sum_{l=1}^q R_l^T A_l^{-1} R_l.$$

The right preconditioned linear system for solving $Av = b$ is then given by

$$AM_{ASM}u = b, \quad v = M_{ASM}u. \quad (4.19)$$

Since M_{ASM} is symmetric but in general not positive definite, we use the Generalized Minimal Residual (GMRES) method for solving the non-symmetric system (4.19). In general, ASM computes a less accurate approximation to the solution of $Av = b$ and therefore requires an increased number of GMRES iterations compared to MSM given by algorithm 1. However, ASM offers a higher potential of parallelization, which results usually in better parallel scaling.

Remark By means of coloring techniques, it is also possible to parallelize the multiplicative Schwarz method up to a certain degree, see [19].

4.2 Implementation

In our implementation, we use one core per subdomain \mathcal{T}_l and distribute A such that every core stores $\tilde{R}_l A$ in its local memory. $\tilde{R}_l A$ contains the non-overlapping portion of rows of $A_l = R_l A R_l^T$.

To set up M_{ASM} once, every process first has to form its local submatrix A_l , i.e., in this step the overlapping part of A_l has to be communicated to process l by process $l - 1$ and $l + 1$. Afterward, each process computes an LU-factorization of A_l , i.e., $A_l = L_l U_l$. This step doesn't involve any inter-process communication and can be done in parallel.

Applying ASM as preconditioner of an iterative method requires computation of $v_k = M_{ASM} b_k$ in each iteration k . For this step, each process l first restricts b to its overlapping part $b_l := R_l b$, which requires communication with process $l - 1$ and $l + 1$. The computation of $A_l^{-1} b_l$ is done by one forward and backward solve with L_l and U_l , respectively. This step does not involve any communication. As final step, the local solution $v_l := A_l^{-1} b_l$ is prolonged back to update the global solution vector v . This step again requires communicating the overlapping part of v_l to process $l - 1$ and $l + 1$.

One can further improve the performance of ASM by using the so called *restricted* version of ASM [5], which is given by

$$M_{rASM} := \sum_{l=1}^p \tilde{R}_l^T A_l^{-1} R_l. \quad (4.20)$$

For this preconditioner, just the non-overlapping part of the local solution v_l is prolonged instead of the entire (overlapping) vector. Experiments show a beneficial behavior in terms of GMRES iterations compared to standard ASM [5]. Furthermore, prolongation by \tilde{R}_l doesn't involve any communication.

So far, we have assumed that the local submatrices A_l are non-singular. If the LU-factorization of one of these submatrices fails, one can transform it to $A_l + \epsilon I_{n_l}$ with a small stabilization parameter $\epsilon > 0$. Since this modification is done inside the preconditioner, it only affects the number of required GMRES iterations but not the obtained approximate solution to (3.5).

4.3 Structure of Local KKT Matrices

In this section, we investigate the structure of the local submatrices A_l . For convenience, let $1 < l < q$. For $l = 1$ and $l = q$ the same results hold with minor modifications. By definition,

$$\begin{aligned} & A_l(x, s, \lambda, \mu) \\ &= R_l A(x, s, \lambda, \mu) R_l^T \\ &= \begin{pmatrix} R_l^x (\nabla_{xx}^2 \mathcal{L}(x, \lambda, \mu) + (\nabla_x h(x))^T \Sigma (\nabla_x h(x))) (R_l^x)^T & R_l^x (\nabla_x g(x))^T (R_l^\lambda)^T \\ R_l^\lambda (\nabla_x g(x)) (R_l^x)^T & 0 \end{pmatrix}. \end{aligned} \quad (4.21)$$

The off-diagonal blocks are given by

$$R_l^\lambda (\nabla_x g(x)) (R_l^x)^T = \nabla_{x_{[l]}} g_{[l]}(x_{[l]}, x^{t_l^+ + 1}). \quad (4.22)$$

After some algebra, we further obtain

$$\begin{aligned} & R_l^x (\nabla_x h(x))^T \Sigma (\nabla_x h(x)) (R_l^x)^T \\ &= \sum_{t \in \mathcal{T}_l} R_l^x (\nabla h_l^t(x^t))^T \Sigma_l^t (\nabla h_l^t(x^t)) (R_l^x)^T \\ &+ \sum_{t=t_l^- - 1}^{t_l^+} R_l^x (\nabla h_R^t(x^{t+1}, x^t))^T \Sigma_R^t (\nabla h_R^t(x^{t+1}, x^t)) (R_l^x)^T \\ &= (\nabla_{x_{[l]}} h_{[l]}(x_{[l]}, x^{t_l^- - 1}, x^{t_l^+ + 1}))^T \Sigma_{[l]} (\nabla_{x_{[l]}} h_{[l]}(x_{[l]}, x^{t_l^- - 1}, x^{t_l^+ + 1})) \end{aligned} \quad (4.23)$$

Note that $h_{[l]}$ is a function of x^t for $t \in \mathcal{T}_l \cup \{t_l^- - 1\} \cup \{t_l^+ + 1\}$, while $g_{[l]}$ is a function of x^t for $t \in \mathcal{T}_l \cup \{t_l^+ + 1\}$. In the following, we consider them as functions of $x_{[l]}$ with some fixed vectors v^-, v^+ in place of $x^{t_l^- - 1}$ and $x^{t_l^+ + 1}$.

We define the local Lagrangian function by

$$\mathcal{L}_{[l]}(x_{[l]}, \lambda_{[l]}, \mu_{[l]}) = \sum_{t \in \mathcal{T}_l} \tau f_t(x^t) + \lambda_{[l]}^T g_{[l]}(x_{[l]}) + \mu_{[l]}^T h_{[l]}(x_{[l]}). \quad (4.24)$$

Since all temporal couplings are linear, $\nabla_{xx}^2 \mathcal{L}$ is block diagonal and it holds

$$R_l^x \nabla_{xx}^2 \mathcal{L}(x, \lambda, \mu) (R_l^x)^T = \nabla_{x_{[l]} x_{[l]}}^2 \mathcal{L}(x, \lambda, \mu) = \nabla_{x_{[l]} x_{[l]}}^2 \mathcal{L}_{[l]}(x_{[l]}, \lambda_{[l]}, \mu_{[l]}). \quad (4.25)$$

Thus, A_l can be written as

$$A_l(x, s, \lambda, \mu) = \begin{pmatrix} H_{[l]}(x_{[l]}, \lambda_{[l]}, \mu_{[l]}) & (\nabla g_{[l]}(x_{[l]}))^T \\ \nabla g_{[l]}(x_{[l]}) & 0 \end{pmatrix} \quad (4.26)$$

with $H_{[l]}(x_{[l]}, \lambda_{[l]}, \mu_{[l]}) = \nabla_{x_{[l]} x_{[l]}}^2 \mathcal{L}_{[l]}(x_{[l]}, \lambda_{[l]}, \mu_{[l]}) + (\nabla h_{[l]}(x_{[l]}))^T \Sigma_{[l]} (\nabla h_{[l]}(x_{[l]}))$.

Therefore, A_l is the KKT matrix obtained when PDIPM is applied to the optimization problem corresponding to the Lagrangian function $\mathcal{L}_{[l]}$. This problem has the form of (2.20) with \mathcal{T} being replaced by \mathcal{T}_l and additional "boundary conditions" v^-, v^+ being imposed:

$$\begin{aligned} & \min_{(x^t)_{t \in \mathcal{T}_l}} \sum_{t \in \mathcal{T}_l} \tau f(x^t) \text{ s.t.} \\ & \quad g_l^t(x^t) = 0, \quad t \in \mathcal{T}_l \\ & \quad g_S^t(x^{t+1}, x^t) = 0, \quad t \in \mathcal{T}_l \setminus \{t_l^+\} \\ & \quad g_S^{t_l^+}(v^+, x^{t_l^+}) = 0 \\ & \quad h_l^t(x^t) \leq 0, \quad t \in \mathcal{T}_l \\ & \quad h_R^t(x^{t+1}, x^t) \leq 0, \quad t \in \mathcal{T}_l \setminus \{t_l^+\} \\ & \quad h_R^{t_l^- - 1}(x^{t_l^-}, v^-) \leq 0 \\ & \quad h_R^{t_l^+}(v^+, x^{t_l^+}) \leq 0. \end{aligned} \quad (4.27)$$

4.4 Relationship between ASM and Constraint Preconditioner

We consider the DOPF problem (2.20) without storage facilities, i.e. without equality constraints g_S^t . Let

$$\tilde{A}_l := \tilde{R}_l A \tilde{R}_l^T \in \mathbb{R}^{\tilde{n}_l \times \tilde{n}_l} \quad (4.28)$$

be the restriction of A to the non-overlapping subdomain $\tilde{\mathcal{T}}_l = \{\tilde{t}_l^-, \dots, \tilde{t}_l^+\}$, i.e., $s_{ol} = 0$. In this case, the corresponding ASM preconditioner,

$$\tilde{M}_{ASM} = \sum_{l=1}^q \tilde{R}_l^T \tilde{A}_l^{-1} \tilde{R}_l, \quad (4.29)$$

reduces to a block Jacobi method with omitted couplings between variables assigned to \tilde{t}_l^+ and \tilde{t}_{l+1}^- for each subdomain l . Since these couplings only arise in the $(1,1)$ -block of A , namely in $(\nabla h)^T \Sigma (\nabla h)$, \tilde{M}_{ASM} has the form of a constraint preconditioner for A :

$$\tilde{M}_{ASM} = \begin{pmatrix} \tilde{H} & (\nabla g)^T \\ \nabla g & 0 \end{pmatrix}^{-1}. \quad (4.30)$$

As pointed out in Section 3.2, 1 is an eigenvalue of $\tilde{M}_{ASM} A$ of multiplicity $2n^\lambda$ and the remaining $n^x - n^\lambda$ eigenvalues are solutions of a generalized eigenvalue problem of the following form [10]:

$$Z^T \underbrace{(\nabla_{xx}^2 \mathcal{L} + (\nabla h)^T \Sigma (\nabla h))}_{=: H} Z v = \lambda Z^T \tilde{H} Z v. \quad (4.31)$$

For a low number of subdomains q , one might expect \tilde{H} to be a good approximation to H , leading to a clustered spectrum $\mathcal{S}(\tilde{M}_{ASM} A)$ close to one. However, the more subdomains, the more neglected couplings and the less accurate does \tilde{H} approximate H . This results in a more scattered eigenvalue distribution of $\tilde{M}_{ASM} A$, but still a large number of eigenvalues are equal to 1. The corresponding behavior of GMRES preconditioned by \tilde{M}_{ASM} compared to M_{ASM} is illustrated by the numerical example in Section 5.1.

5 Numerical Experiments

In this section, we present some results for our previously proposed method applied to two different DOPF problems. The considered power grid topology and generation system represent the German transmission grid including 380kV, 220kV and some 110kV transmission lines. The data is based on scenario B for the year 2023, provided by the German Network Regulatory Body ("Bundesnetzagentur"). The whole grid consists of 1215 nodes, 1292 AC lines, 8 DC lines, 406 conventional power plants, 1405 renewable energy sources and 57 storage facilities.

We consider a time period of 32 days with a temporal resolution of 1h, i.e. $N_T = 768$, leading to a linear system of dimension $n^x + n^\lambda \approx 6.7 \cdot 10^6$. We use a 24h profile for power load and renewable feed-in that is concatenated to obtain a profile for the complete time interval. The ramping constraints bound the change in power generation of conventional plants by 40% of their respective maximum power generation per hour, i.e. $Pr_{max,i} = 0.4P_{max,i}$. More details about our data and parameter values can be found in [11].

For solving (2.20), we use the PDIPM algorithm `mips` which is written in Matlab code and part of MATPOWER [24]. In this algorithm, we replace the standard Matlab backslash operator `\` for solving linear systems by our own linear solver. This solver consists of GMRES with a right preconditioner given by the restricted version of ASM. For computing LU-factorizations of the local systems A_l , we use the software package Mumps [2]. Our solver is written in C++ and makes use of the KSPFGMRES and PCASM methods provided by PETSc [3], which is compiled in Release mode with the gcc compiler of version 4.9.1. Inter-process communication is realized by means of the Message Passing Interface (MPI). All tests are performed on a Linux computing node with 8 Intel(R) Xeon(R) CPU E7-8880 v2 @ 2.50GHz CPUs, each consisting of 15 cores. The complete node has 4TB of RAM.

We set the PDIPM termination criteria $\epsilon_{feas} = \epsilon_{grad} = \epsilon_{cost} = 10^{-5}$, $\epsilon_{comp} = 10^{-3}$ and solve the arising linear systems with residual tolerance as described in Section 3.3. GMRES is used without restarting and the maximum number of iterations is set to 400. The overlap s_{ol} is set to 1 for all tests.

q		1	2	4	8	16	32	64	100
ASM	[#]		2.8	2.7	3.2	3.0	3.1	3.1	3.3
CP	[#]		49.0	57.3	64.1	66.8	68.5		
ASM	[sec]		153.2	74.5	38.1	18.8	9.0	4.9	4.0
CP	[sec]		470.9	286.2	187.4	118.4	67.7		
Mumps	[sec]	208.1	155.3	128.7	118.7	113.8	109.7		

Table 1: Case 1: Average number of GMRES iterations and solution time for GMRES+ASM, GMRES+CP and Mumps

5.1 Test Case 1 - Grid without Storage Facilities

For the first test case, storage facilities are not taken into account. Therefore, ramping constraints establish the only couplings between variables of different time steps. We apply ASM and compare its performance with a block Jacobi preconditioner and the parallel direct solver given by Mumps. As shown in Section 3.3, the block Jacobi preconditioner has the form of a constraint preconditioner (CP) if no intertemporal couplings are induced by equality constraints.

`mips` needs 43 iterations to satisfy the criteria defined above and the initial value $3.03 \cdot 10^5$ of the cost function is decreased to $3.72 \cdot 10^4$. Due to the very ill-conditioned matrix in the final steps, it was not possible to reach the accuracy in solving the linear system that is necessary to run `mips` until c_{comp} is arbitrarily small. When using a direct solver, we obtained a minimal value of $c_{comp} = 1.34 \cdot 10^{-5}$ after 45 PDIPM steps. Taking this as a reference, the predicted difference in the cost function according to Section 3.2, $E_{43} = 5.6 \cdot 10^{-3}$, is of the same order of magnitude as the "exact" difference $f(x_{43}) - f(x_{45}) = 1.0 \cdot 10^{-3}$. Therefore, the relative error in the cost function is of order 10^{-8} .

In Table 1, the average number of GMRES iterations per PDIPM step is shown for ASM and CP being used as preconditioner, respectively, for a varying number of subdomains q . One can observe that the number of GMRES+ASM iterations is almost insensitive to q . This behavior is quite uncommon for domain decomposition methods without coarse grid correction [19].

On the other hand, CP is not able to maintain a stable number of iterations for increasing q . Since in test case 1, CP is formally equivalent to ASM with zero overlap (see Section 3.3), it seems that adding a minimal amount of overlap completely changes the quality of the preconditioner in terms of approximating A^{-1} . The better this approximation, the fewer iterations are needed for GMRES to obtain a prescribed accuracy. We remark that increasing the overlap to moderate sizes $s_{ol} = 2, 3, 4$ did not lead to a further reduction of required GMRES iterations.

Table 1 additionally lists the average solution times T_q for different solvers and different numbers of subdomains q . Figure 2 shows the parallel speedup, $s_q := \frac{T_1}{T_q}$, of parallel Mumps, GMRES+CP and GMRES+ASM with respect to sequential Mumps for solving the linear systems. Due to the stable number of iterations, GMRES+ASM scales quite well for an increasing number of subdomains q and achieves a maximum speedup of 50 for $q = 100$. In contrast, parallel Mumps attains a maximal speedup of 2 for $q = 32$ and CP performs hardly better.

Remark We note that we observed a linear complexity $O(N_T)$ with respect to the problem size for the sequential Mumps solver. Therefore, it was not possible to obtain a parallel efficiency $e_q = \frac{s_q}{q}$, for GMRES+ASM greater than 0.5, although the number of iterations was independent of q . This is due to the additional computational overhead for applying a Krylov solver and a reduction of complexity for computing LU-factorizations that scales only linear with the matrix size.

Further, CP with $q = 32$ does not achieve the desired tolerance in the final IPM steps within 400 GMRES iterations and $c_{comp} = 5.6 \cdot 10^{-3}$ can be obtained as minimal complementary criterion. We therefore did not test CP for $q > 32$.

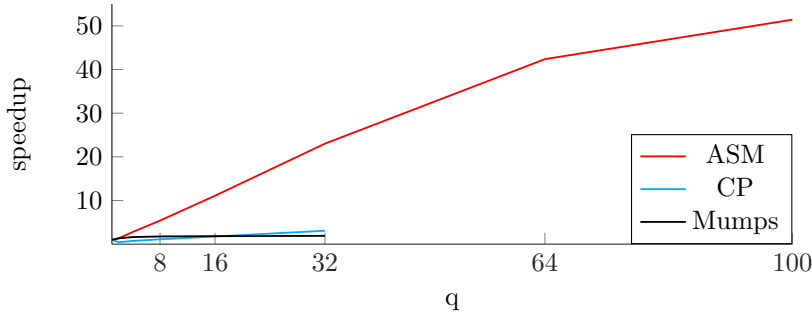


Figure 2: Case 1: Parallel Speedup for GMRES+ASM, GMRES+CP and Mumps

k	1	10	20	30	40	50
I_{eff}	61.5	7.3	5.0	2.8	2.0	1.4

Table 2: Case 2: Effectiveness of error estimator at different PDIPM iterations

5.2 Test Case 2 - Grid with Storage Facilities

We now consider the full model as described in Section 2.1 and compare GMRES+ASM with the parallel Mumps solver. In this case, `mips` needs 50 iterations to satisfy the criteria defined above and the initial value $2.92 \cdot 10^5$ of the cost function is decreased to $3.25 \cdot 10^4$, which is around 13% lower than in case 1. As in case 1, it was not possible to run `mips` in combination with GMRES+ASM until c_{comp} is arbitrarily small. When using a direct solver, we obtained a minimal value of $c_{comp} = 2.67 \cdot 10^{-6}$ after 54 PDIPM steps.

In Figure 3, the error in the cost function, $f(x_k) - f(x_{54})$, is plotted for each PDIPM step k (black, solid). One can see that the error in the cost function obtained for $c_{comp} \approx 10^{-3}$ at $k = 50$ is again quite small, namely of order 10^{-2} , yielding a relative error around 10^{-8} . Figure 3 and Table 2 further show the estimated difference E_k (red, solid) and the effectiveness $I_{eff,k} = \frac{E_k}{f(x_k) - f(x_{54})}$ of the error estimator (red, dashed). Here, a value of 1 for I_{eff} would be optimal. In accordance with Lemma 3.1, $I_{eff,k}$ approaches 1 for an increasing iteration index k .

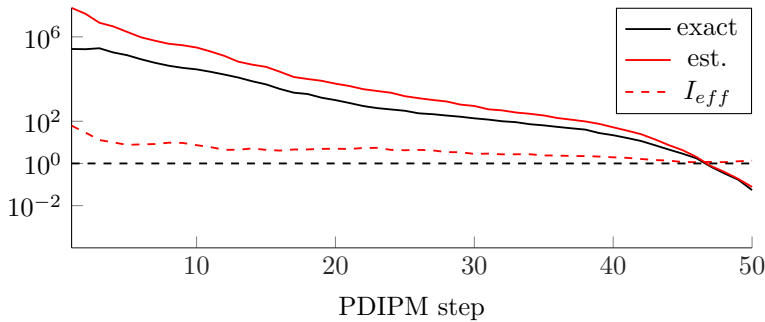


Figure 3: Case 2: Exact and estimated difference in cost function per PDIPM step

Table 3 lists the average number of GMRES+ASM iterations and solution times compared to Mumps. In contrast to case 1, the number of iterations grows with increasing q , which results in a lower parallel speedup, see Figure 4. For $q = 64$, the additional overhead due to GMRES exceeds the time gained by factorizing smaller submatrices and the speedup decreases. However, the maximal speedup obtained by GMRES+ASM for $q = 32$ is still about a factor 4 higher than the one obtained by parallel Mumps.

To get further insight, Figure 5 and 6 show the number of GMRES iterations and speedup s_q for each linear system that has to be solved in the course of PDIPM. It is interesting to note that number of iterations during the first, say, 20 PDIPM iterations is still rather insensitive to q . Therefore, these linear systems can be solved with a maximum speedup between 25 and 40 for $q = 64$. However, for

q		1	2	4	8	16	32	64
ASM	[#]		8.3	10.0	13.1	19.5	32.4	77.0
ASM	[sec]		213.8	103.2	61.8	37.5	28.9	46.4
Mumps	[sec]	246.5	176.2	143.5	127.3	119.2	115.2	

Table 3: Case 2: Average number of GMRES iterations and solution time for GMRES+ASM and Mumps

$k > 20$ the approximation quality of ASM seems to decrease for increasing q and for increasing k , which has a negative effect on the speedup. In the next section, we give a possible explanation for the different behavior of ASM in case 1 and 2.

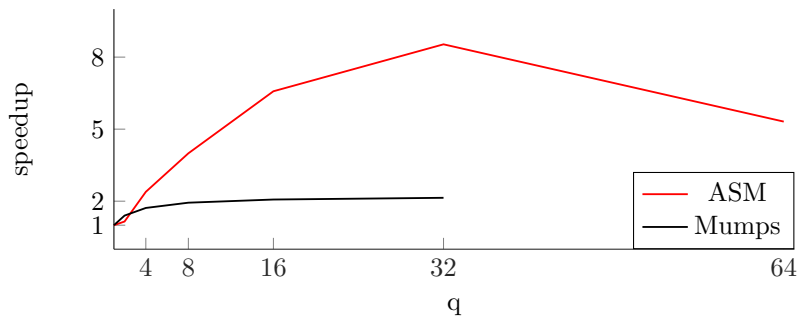


Figure 4: Case 2: Parallel speedup for GMRES+ASM and Mumps

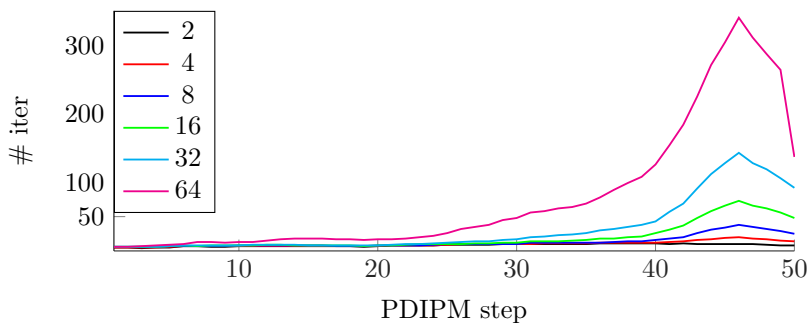


Figure 5: Case 2: GMRES iterations per PDIPM step for different q

5.3 Comparison of Case 1 and Case 2

Neglecting a certain amount of intertemporal couplings (matrix entries) in the global system matrix A , is the crucial point for deriving the local submatrices that are used in ASM. On the one hand, the more couplings are neglected (larger q), the worse should be the approximation quality of ASM. On the other hand, neglecting matrix entries of large magnitude should also decrease the approximation quality more strongly, than do matrix entries of low magnitude.

In the following, we consider the intertemporal couplings induced by ramping constraints $h_R^t(x^{t+1}, x^t) \leq 0$. These couplings enter the global system matrix A via $(\nabla h(x))^T \Sigma (\nabla h(x))$, i.e., their magnitude is proportional to the corresponding entries $\sigma_R^t = [\frac{\mu_{R,i}^t}{s_{R,i}^t}]_{i=1}^{N_R}$, $N_R = 2|\mathcal{C}|$ in Σ . As mentioned in Section 3.2, these entries converge to ∞ if the corresponding inequalities are active at the exact KKT point. We define a

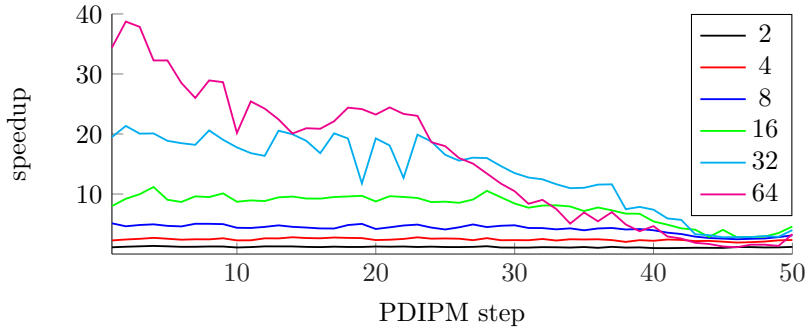


Figure 6: Case 2: Parallel speedup per PDIPM step for different q

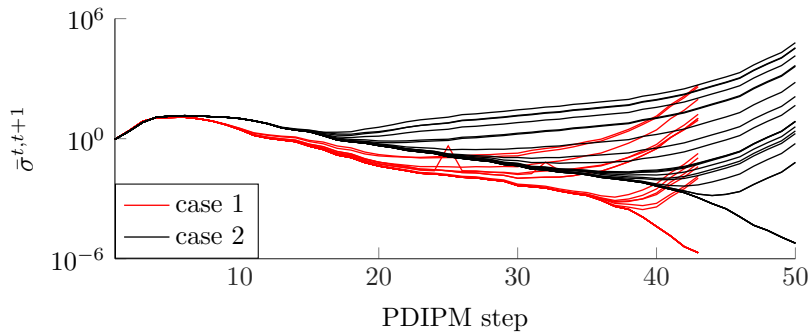


Figure 7: Strength of intertemporal couplings per PDIPM step

measure for the strength of the intertemporal couplings between time steps t and $t + 1$ by

$$\bar{\sigma}^{t,t+1} := \frac{1}{N_R} \sum_{i=1}^{N_R} \sigma_{R,i}^t > 0. \quad (5.1)$$

In Figure 7, the values of $\bar{\sigma}^{t,t+1}$ for case 1 (red) and case 2 (black) are plotted for $t = 1, \dots, 24$ at each PDIPM iteration k . In case 2, these values are several orders of magnitude higher than in case 1, especially in the latter phase of PDIPM. This indicates a stronger coupling between different time steps and might be an explanation for the different behavior of ASM in both cases.

We note that there are still several pairs of consecutive time steps with $\bar{\sigma}^{t,t+1}$ being smaller than 1. In our future work, we want to exploit this fact by means of a decomposition of \mathcal{T} according to points with low values of $\bar{\sigma}^{t,t+1}$. This decomposition might have to be updated in each PDIPM iteration.

6 Critical Review

In the last section, we described how the performance of our solver depends on the strength of intertemporal ramping constraints. For simplicity, we assumed a maximal change in active power generation per hour of 40% of the respective maximal possible power injection for each conventional power plant. In practice, these factors depend on the specific type of the given power plants and more realistic values most certainly have an influence on our solver. Furthermore, we concatenated 24h load and renewable profiles in order to get the corresponding profiles for 32 days. Using more realistic profiles with potentially higher variances can also affect the strength of ramping constraints. Therefore, making our methodology more robust against diversifying problem characteristics will be one of our main goals in future work.

7 Conclusion

In this work we proposed a way of solving linear systems arising from Dynamic Optimal Power Flow (DOPF) problems in parallel by means of overlapping Schwarz domain decomposition methods. It was shown how to apply these methods in the context of DOPF and that the obtained submatrices correspond to localized formulations of DOPF with additional boundary conditions. Numerical tests on two large-scale, real-world DOPF problems showed that the combination of the Generalized Minimal Residual method (GMRES) and the Additive Schwarz Method (ASM) outperforms parallel Mumps and leads to a maximal obtained parallel speedup factor of 50 and 8, respectively, compared to the sequential Mumps solver. Such efficient optimization methods, including an accurate consideration of non-linear and non-convex AC power flow constraints are highly important to ensure a dependable power grid operation in the face of an increasingly uncertain, volatile and decentralized power generation. Furthermore, we outlined how the performance of our solver depends on problem characteristics such as the strength of intertemporal couplings.

Future enhancements of the proposed algorithm will include its extension to power generation and transmission expansion planning problems since particularly grid expansion is key for maintaining today's level of supply security in the long term [13]. However, determining an optimal grid expansion leads to large-scale mixed-integer nonlinear optimization problems (MINLP) which result in a yet higher complexity than the problems in focus of this paper. Solving such problems will require the attention and joint efforts of different disciplines, including mathematics, economics and electrical engineering.

Acknowledgement

This work was carried out with the financial support of the German Research Foundation (DFG) within the project HE 4760 / 8-1. Further, the authors gratefully acknowledge the support of the Klaus Tschira Foundation.

A List of Variables

$$P_g^t : \text{vector of injected active power by conventional plants} \quad (\text{A.1})$$

$$P_r^t : \text{vector of injected active power by renewable energy sources} \quad (\text{A.2})$$

$$P_{sg}^t : \text{vector of injected active power by storage facilities} \quad (\text{A.3})$$

$$P_{sl}^t : \text{vector of extracted active power by storage facilities} \quad (\text{A.4})$$

$$SOC^t : \text{vector of stored energy in storage facilities} \quad (\text{A.5})$$

$$P_l^t : \text{vector of active load sheds} \quad (\text{A.6})$$

$$P_{dc+}^t : \text{vector of injected active power by dc lines} \quad (\text{A.7})$$

$$P_{dc-}^t : \text{vector of extracted active power by dc lines} \quad (\text{A.8})$$

$$P_{ex}^t : \text{vector of extracted active power by export} \quad (\text{A.9})$$

$$P_{im}^t : \text{vector of injected active power by import} \quad (\text{A.10})$$

$$Q_g^t : \text{vector of injected reactive power by conventional plants} \quad (\text{A.11})$$

$$Q_r^t : \text{vector of injected reactive power by renewable energy sources} \quad (\text{A.12})$$

$$Q_{sg}^t : \text{vector of injected reactive power by storage facilities} \quad (\text{A.13})$$

$$Q_l^t : \text{vector of reactive load sheds} \quad (\text{A.14})$$

B Proof of Lemma 3.1

Lemma 3.1 Let $z^* = (x^*, s^*, \lambda^*, \mu^*)$ be a solution of (3.2) and let $\tilde{z} = (\tilde{x}, \tilde{s}, \tilde{\lambda}, \tilde{\mu})$ solve (3.2) with some given residual, i.e.

$$\mathcal{F}_0(\tilde{x}, \tilde{s}, \tilde{\lambda}, \tilde{\mu}) = (r_x \quad r_\lambda \quad r_\mu \quad r_{comp})^T$$

and $\tilde{s}, \tilde{\mu} \geq 0$. Then it holds

$$f(\tilde{x}) - f(x^*) = \Delta_{comp} + \Delta_{res} + \Delta_{lin}$$

with

$$\Delta_{comp} = \frac{1}{2}(\tilde{s}^T \tilde{\mu} + \tilde{s}^T \mu^* - \tilde{\mu}^T s^*) \quad (\text{B.1})$$

$$\Delta_{res} = \frac{1}{2}(r_x^T(\tilde{x} - x^*) - r_\lambda^T(\tilde{\lambda} + \lambda^*) - r_\mu^T(\tilde{\mu} + \mu^*)) \quad (\text{B.2})$$

$$\Delta_{lin} = \frac{1}{12} \max_{t \in [0,1]} l'''(t) \quad (\text{B.3})$$

$$l(t) = L(x^* + t(\tilde{x} - x^*), \lambda^* + t(\tilde{\lambda} - \lambda^*), \mu^* + t(\tilde{\mu} - \mu^*)). \quad (\text{B.4})$$

Proof It holds

$$\begin{aligned} f(\tilde{x}) - f(x^*) &= L(\tilde{z}) - \tilde{\lambda}^T g(\tilde{x}) - \tilde{\mu}^T h(\tilde{x}) - L(z^*) \\ &= l(1) - l(0) - \underbrace{(\tilde{\lambda}^T r_\lambda + \tilde{\mu}^T (r_\mu - \tilde{s}))}_{=:r} \\ &= \int_0^1 l'(t) dt - r \\ &= \frac{1}{2}[l'(1) + l'(0)] + \Delta_{lin} - r, \end{aligned} \quad (\text{B.5})$$

where we applied trapezoidal rule for approximating the integral. Δ_{lin} denotes the corresponding error. The last line can further be written as

$$\begin{aligned} &\frac{1}{2}(l'(1) + l'(0)) + \Delta_{lin} - r \\ &= \frac{1}{2} \left(\underbrace{(\nabla_x L(\tilde{z}))}_{=:r_x^T}, \underbrace{(\nabla_\lambda L(\tilde{z}))}_{=:r_\lambda^T}, \underbrace{(\nabla_\mu L(\tilde{z}))}_{=:r_\mu^T - \tilde{s}^T} + \underbrace{(\nabla_x L(z^*))}_{=0}, \underbrace{(\nabla_\lambda L(z^*))}_{=0}, \underbrace{(\nabla_\mu L(z^*))}_{=:-(s^*)^T} \right) \begin{pmatrix} \tilde{x} - x^* \\ \tilde{\lambda} - \lambda^* \\ \tilde{\mu} - \mu^* \end{pmatrix} + \Delta_{lin} - r \\ &= \frac{1}{2}(\tilde{s}^T \tilde{\mu} + \tilde{s}^T \mu^* - \tilde{\mu}^T s^*) + \frac{1}{2}(r_x^T(\tilde{x} - x^*) - r_\lambda^T(\tilde{\lambda} + \lambda^*) - r_\mu^T(\tilde{\mu} + \mu^*)) + \Delta_{lin} \\ &= \Delta_{comp} + \Delta_{res} + \Delta_{lin}, \end{aligned} \quad (\text{B.6})$$

where $(s^*)^T \mu^* = 0$ was used. \square

C Reduction of KKT Matrix

The linear system (3.4) arising in each PDIPM step is of the form

$$\begin{pmatrix} H & 0 & J_g^T & J_h^T \\ 0 & M & 0 & S \\ J_g & 0 & 0 & 0 \\ J_h & I & 0 & 0 \end{pmatrix} \begin{pmatrix} x \\ s \\ \lambda \\ \mu \end{pmatrix} = \begin{pmatrix} b_x \\ b_s \\ b_\lambda \\ b_\mu \end{pmatrix} \Leftrightarrow \bar{A}q = \bar{b} \quad (\text{C.1})$$

with positive definite diagonal matrices M, S and identity matrix $I \in \mathbb{R}^{n^\mu \times n^\mu}$.

Eliminating s and μ by $s = -\Sigma^{-1}\mu + M^{-1}b_s$ and $\mu = \Sigma J_h x - \Sigma b_\mu + S^{-1}b_s$ with $\Sigma = S^{-1}M$ yields the reduced system (3.5)

$$\begin{pmatrix} H + J_h^T \Sigma J_h & J_g^T \\ J_g & 0 \end{pmatrix} \begin{pmatrix} x \\ \lambda \end{pmatrix} = \begin{pmatrix} b_x + J_h^T \Sigma (b_\mu + M^{-1}b_s) \\ b_\lambda \end{pmatrix} : \Leftrightarrow Ap = b. \quad (\text{C.2})$$

Assume that this system is approximately solved with residual $r = (r_x, r_\lambda)$, i.e., $A\tilde{p} = b + r$ with $\tilde{p} = (\tilde{x}, \tilde{\lambda})$. Setting $\tilde{q} = (\tilde{x}, \tilde{s}, \tilde{\lambda}, \tilde{\mu})$ with $\tilde{s} = -\Sigma^{-1}\tilde{\mu} + M^{-1}b_s$ and $\tilde{\mu} = \Sigma J_h \tilde{x} - \Sigma b_\mu + S^{-1}b_s$ yields a residual \bar{r} for the unreduced system

$$\bar{r} = \bar{A}\tilde{q} - \bar{b} = \begin{pmatrix} r_x \\ 0 \\ r_\lambda \\ 0 \end{pmatrix} \quad (\text{C.3})$$

and therefore $\|\bar{r}\|_2 = \|r\|_2$.

References

- [1] N. Alguacil and A.J. Conejo. Multiperiod optimal power flow using benders decomposition. *IEEE Transactions on Power Systems*, 15(1):196–201, Feb 2000.
- [2] P.R. Amestoy, I.S. Duff, and J.-Y. L’Excellent. Multifrontal parallel distributed symmetric and unsymmetric solvers. *Computer Methods in Applied Mechanics and Engineering*, 184(24):501 – 520, 2000.
- [3] Satish Balay, William D. Gropp, Lois Curfman McInnes, and Barry F. Smith. Efficient management of parallelism in object oriented numerical software libraries. In E. Arge, A. M. Bruaset, and H. P. Langtangen, editors, *Modern Software Tools in Scientific Computing*, pages 163–202. Birkhäuser Press, 1997.
- [4] S. Bellavia. Inexact interior-point method. *Journal of Optimization Theory and Applications*, 96(1):109–121, 1998.
- [5] Xiao-Chuan Cai and Marcus Sarkis. A restricted additive schwarz preconditioner for general sparse linear systems. *SIAM Journal on Scientific Computing*, 21(2):792–797, 1999.
- [6] Florin Capitanescu, Mevludin Glavic, Damien Ernst, and Louis Wehenkel. Interior-point based algorithms for the solution of optimal power flow problems. *Electric Power Systems Research*, 77(56):508 – 517, 2007.
- [7] C.Y. Chung, Wei Yan, and Fang Liu. Decomposed predictor-corrector interior point method for dynamic optimal power flow. *IEEE Transactions on Power Systems*, 26(3):1030–1039, Aug 2011.
- [8] Frank E. Curtis, Johannes Huber, Olaf Schenk, and Andreas Wächter. A note on the implementation of an interior-point algorithm for nonlinear optimization with inexact step computations. *Mathematical Programming*, 136(1):209–227, 2012.
- [9] P. Gerstner, V. Heuveline, and M. Schick. A multilevel domain decomposition approach for solving time constrained optimal power flow problems. *Preprint Series of the Engineering Mathematics and Computing Lab (EMCL)*, 2015 / 4:1–30, 9 2015.
- [10] Carsten Keller, Nicholas I. M. Gould, and Andrew J. Wathen. Constraint preconditioning for indefinite linear systems. *SIAM Journal on Matrix Analysis and Applications*, 21(4):1300–1317, 2000.
- [11] N. Meyer-Hübner, M. Suriyah, T. Leibfried, V. Slednev, V. Bertsch, W. Fichtner, P. Gerstner, M. Schick, and V. Heuveline. Time constrained optimal power flow calculations on the german power grid. In *accepted at International ETG Congress, Bonn*, pages 1–7, 2015.
- [12] Jorge Nocedal and Steve J. Wright. *Numerical optimization*. Springer Series in Operations Research and Financial Engineering. Springer, Berlin, 2006.
- [13] Christoph Nolden, Martin Schönfelder, Anke Eßer-Frey, Valentin Bertsch, and Wolf Fichtner. Network constraints in techno-economic energy system models: towards more accurate modeling of power flows in long-term energy system models. *Energy Systems*, 4(3):267–287, 2013.
- [14] William Rosehart, Codruta Roman, and Laleh Behjat. Interior point models for power system stability problems. *European Journal of Operational Research*, 171(3):1127 – 1138, 2006. Feature Cluster: Heuristic and Stochastic Methods in Optimization Feature Cluster: New Opportunities for Operations Research.
- [15] Youcef Saad and Martin H Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.*, 7(3):856–869, July 1986.
- [16] Martin Schönfelder, Anke Eßer-Frey, Michael Schick, Wolf Fichtner, Vincent Heuveline, and Thomas Leibfried. New developments in modeling network constraints in techno-economic energy system expansion planning models. *Zeitschrift für Energiewirtschaft*, 36(1):27–35, 2012.
- [17] H.A. Schwarz. *Ueber einen Grenzübergang durch alternirendes Verfahren*. Vierteljahrsschrift der Naturforschenden Gesellschaft in Zürich. Zürcher u. Furrer, 1870.
- [18] V. Slednev, V. Bertsch, and W. Fichtner. A multi-objective time segmentation approach for power generation and transmission models. In Gernot Tragler Ivana Ljubic, Georg Pflug and Karl Doerner, editors, *Operations Research Proceedings*. Springer, 2015.

- [19] Barry Francis Smith, Petter E. Bjrstad, and William D. Gropp. *Domain decomposition : parallel multilevel methods for elliptic partial differential equations*. Cambridge University Press, Cambridge, 1996.
- [20] Edilaine Martins Soler, Vanusa Alves de Sousa, and Geraldo R.M. da Costa. A modified primaldual logarithmic-barrier method for solving the optimal power flow problem with discrete and continuous control variables. *European Journal of Operational Research*, 222(3):616 – 622, 2012.
- [21] Andréa A. Sousa, Geraldo L. Torres, and Claudio A. Canizares. Robust optimal power flow solution using trust region and interior-point methods. *IEEE Transactions on Power Systems*, PP(99), 2010.
- [22] G.L. Torres and V.H. Quintana. An interior-point method for nonlinear optimal power flow using voltage rectangular coordinates. *IEEE Transactions on Power Systems*, 13(4):1211–1218, Nov 1998.
- [23] K. Xie and Y.H. Song. Dynamic optimal power flow by interior point methods. *IEE Proceedings-Generation, Transmission and Distribution*, 148(1):76–84, Jan 2001.
- [24] 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*, 26(1):12–19, Feb 2011.

Preprint Series of the Engineering Mathematics and Computing Lab

recent issues

- No. 2015-04 Philipp Gerstner, Vincent Heuveline, Michael Schick : A Multilevel Domain Decomposition approach for solving time constrained Optimal Power Flow problems
- No. 2015-03 Martin Wlotzka, Vincent Heuveline: Block-asynchronous and Jacobi smoothers for a multigrid solver on GPU-accelerated HPC clusters
- No. 2015-02 Nicolai Schoch, Fabian Kießler, Markus Stoll, Sandy Engelhardt, Raffaele de Simone, Ivo Wolf, Rolf Bendl, Vincent Heuveline: Comprehensive Pre- & Post-Processing for Numerical Simulations in Cardiac Surgery Assistance
- No. 2015-01 Teresa Beck, Martin Baumann, Leonhard Scheck, Vincent Heuveline, Sarah Jones: Comparison of mesh-adaptation criteria for an idealized tropical cyclone problem
- No. 2014-02 Christoph Paulus, Stefan Suwelack, Nicolai Schoch, Stefanie Speidel, Rüdiger Dillmann, Vincent Heuveline: Simulation of Complex Cuts in Soft Tissue with the Extended Finite Element Method (X-FEM)
- No. 2014-01 Martin Wlotzka, Vincent Heuveline: A parallel solution scheme for multiphysics evolution problems using OpenPALM
- No. 2013-02 Nicolai Schoch, Stefan Suwelack, Rüdiger Dillmann, Vincent Heuveline: Simulation of Surgical Cutting in Soft Tissue using the Extended Finite Element Method (X-FEM)
- No. 2013-01 Martin Schindewolf, Björn Rocker, Wolfgang Karl, Vincent Heuveline: Evaluation of two Formulations of the Conjugate Gradients Method with Transactional Memory
- No. 2012-07 Andreas Helfrich-Schkarbanenko, Vincent Heuveline, Roman Reiner, Sebastian Ritterbusch: Bandwidth-Efficient Parallel Visualization for Mobile Devices
- No. 2012-06 Thomas Henn, Vincent Heuveline, Mathias J. Krause, Sebastian Ritterbusch: Aortic Coarctation simulation based on the Lattice Boltzmann method: benchmark results
- No. 2012-05 Vincent Heuveline, Eva Ketelaer, Staffan Ronnas, Mareike Schmidtbreich, Martin Wlotzka: Scalability Study of HiFlow³ based on a Fluid Flow Channel Benchmark
- No. 2012-04 Hartwig Anzt, Armen Beglarian, Suren Chilingaryan, Andrew Ferrone, Vincent Heuveline, Andreas Kopmann: A unified Energy Footprint for Simulation Software
- No. 2012-03 Vincent Heuveline, Chandramowli Subramanian: The Coffee-table Book of Pseudospectra
- No. 2012-02 Dominik P.J. Barz, Hendryk Bockelmann, Vincent Heuveline: Electrokinetic optimization of a micromixer for lab-on-chip applications
- No. 2012-01 Sven Janko, Björn Rocker, Martin Schindewolf, Vincent Heuveline, Wolfgang Karl: Software Transactional Memory, OpenMP and Pthread implementations of the Conjugate Gradients Method - a Preliminary Evaluation
- No. 2011-17 Hartwig Anzt, Jack Dongarra, Vincent Heuveline, Piotr Luszczek: GPU-Accelerated Asynchronous Error Correction for Mixed Precision Iterative Refinement

Preprint Series of the Engineering Mathematics and Computing Lab (EMCL)

