

# NONOVERLAPPING DOMAIN DECOMPOSITION IN SPACE AND TIME FOR OPTIMAL CONTROL PROBLEMS ON METRIC GRAPHS BY THE EXAMPLE OF GAS FLOW IN PIPE NETWORKS

GÜNTER LEUGERING

**ABSTRACT.** We consider non-overlapping domain decomposition methods for ordinary and partial differential equations and corresponding optimal control problems on metric graphs. As an exemplary context, we chose a semilinear approximation of the Euler system and a doubly nonlinear parabolic model that has come to be known as friction dominated flow in gas pipe networks. By this choice, we encounter hyperbolic, parabolic and elliptic linear and nonlinear problems in a single highly motivating application. We depart from the classical domain decomposition methods described by P.L. Lions [35] and J.L. Lions and O. Pironneau [32–34] and extend those to problems on metric graphs. The choice of methods is determined by the desire to use a control concept that has come to be known as *virtual controls* which, in turn, possibly lead to a fully parallel decomposition of the corresponding optimality systems. In a second step, we extend the methods to p-Laplace problems on networks. The analysis, due to space limitations, will appear in a forthcoming publication. See however [26]. Furthermore, we then describe methods for space and time domain decomposition or optimal control problems in the spirit of J.E. Lagnese and G. Leugering [22]. We finally provide some comments on PINN-based approximations of the methods described before. We provide numerical evidence for all algorithms discussed.

## 1. INTRODUCTION

**1.1. Modeling of gas flow in a single pipe.** Domain decomposition methods (DDMs) for the simulation of partial differential equations have developed over the last decades and are now very well established. There are numerous textbooks available -cf. Quarteroni and Valli [36] for an early contribution - and the biannual proceedings of the <http://www.DDM.org> conferences can be seen as a regular update of the state of the art. The genuine motivation for parallel domain decomposition methods in space and time is the challenge of very large scale problems involving partial differential equations on complex domains being central in modern applications. Hence, the major focus typically is on efficiency of numerical algorithms. The mathematical interest in PDE related properties of such algorithms - also with respect to mesh independence - focuses on the continuous level and the relations to discretizations. When it comes to the functional analytic treatment of DDMs, i.e. the proof of convergence in function spaces, the literature is much less abundant. Such methods for optimal control problems, and in particular the corresponding analysis in function spaces, are even less developed, but still, the scope would go far beyond the format of this article, see [22] for a reference, where the particular focus has been on problems on 2-d networked domains and 1-d metric graphs. The material available in the literature becomes even much more confined when looking for methods covering nonlinear systems. Therefore, in these these notes, we focus on a particular case of models on networks including nonlinear hyperbolic and parabolic problems. In order to fix ideas, it is convenient to consider an example that is rich enough to demonstrate the range of the methods and simple enough to be described in the given space limits. However, the methods described apply to much more general examples.

---

1991 *Mathematics Subject Classification.* 35J70,49J20,49J45,93C73,65M55,65N55.

*Key words and phrases.* Optimal control, pdes on graphs, p-Laplace problem on a graph, p-parabolic problems, instantaneous control, optimality system, domain decomposition.

The author was supported by the DFG-TRR 154 "Modellierung Simulation und Optimierung am Beispiel von Gasnetzwerken" (A05) .

We begin with the description of the model problem which goes back to early articles by Bamberger [4, 5]. The Euler equations are given by a system of nonlinear hyperbolic partial differential equations (PDEs) which represent the motion of a compressible non-viscous fluid or a gas. They consist of the continuity equation, the balance of moments and the energy equation. The full set of equations is given by (see [8, 28, 29, 40]). We briefly introduce the model. Let  $\rho$  denote the density,  $v$  the velocity of the gas and  $p$  the pressure. We further denote  $g$  the gravitational constant,  $\lambda$  the friction coefficient of the pipe,  $D$  the diameter,  $a$  the area of the cross section. The state variables of the system are  $\rho$ , the flux  $q = \rho v$ . We also denote  $c$  the the speed of sound, i.e.  $c^2 = \frac{\partial p}{\partial \rho}$  (for constant entropy). For natural gas we have  $340 \frac{\text{m}}{\text{s}}$ . In particular, in the subsonic case ( $|v| < c$ ), the one which we consider in the sequel, two boundary conditions have to be imposed on the left end and one at the right end of the pipe. We consider here the isothermal case only. Thus, for horizontal pipes we have

$$(1.1) \quad \begin{aligned} \frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x}(\rho v) &= 0 \\ \frac{\partial}{\partial t}(\rho v) + \frac{\partial}{\partial x}(p + \rho v^2) &= -\frac{\lambda}{2D} \rho v |v|. \end{aligned}$$

In the particular case, where we have a constant speed of sound  $c = \sqrt{\frac{p}{\rho}}$ , for small velocities  $|v| \ll c$  and the flux  $q = \rho v a$ , we arrive at the semi-linear model

$$(1.2) \quad \begin{aligned} \frac{\partial p}{\partial t} + \frac{c^2}{a} \frac{\partial}{\partial x} q &= 0 \\ \frac{\partial}{\partial t} q + a \frac{\partial p}{\partial x} &= -\frac{\lambda c^2}{2aDp} q |q|. \end{aligned}$$

If we further neglect the inertia with respect to the flux, we arrive at

$$(1.3) \quad \begin{aligned} \frac{\partial p}{\partial t} + \frac{c^2}{a} \frac{\partial}{\partial x} q &= 0 \\ \frac{\partial p^2}{\partial x} &= -\frac{\lambda c^2}{Da^2} q |q| =: -\gamma^2 q |q|. \end{aligned}$$

We now set  $y := p^2$  and obtain from the second equation in (1.3)

$$q = -\frac{1}{\gamma} \frac{\frac{\partial y}{\partial x}}{\sqrt{\left| \frac{\partial y}{\partial x} \right|}}.$$

With  $\alpha := \frac{\gamma a}{c}$  we obtain

$$(1.4) \quad \alpha \frac{\partial}{\partial t} \frac{y}{\sqrt{|y|}} - \frac{\partial}{\partial x} \frac{\frac{\partial y}{\partial x}}{\sqrt{\left| \frac{\partial y}{\partial x} \right|}} = 0.$$

We introduce the monotone function  $\beta(s) := \frac{s}{\sqrt{|s|}}$ . With this, (1.4) reads as

$$(1.5) \quad \alpha \frac{\partial}{\partial t} \beta(y) - \frac{\partial}{\partial x} \beta\left(\frac{\partial y}{\partial x}\right) = 0.$$

It is also possible to write this down in the p-Laplace format: (1.4) reads as

$$(1.6) \quad \alpha \frac{\partial}{\partial t} \left( |y|^{p-2} y \right) - \frac{\partial}{\partial x} \left( \left| \frac{\partial y}{\partial x} \right|^{p-2} \frac{\partial y}{\partial x} \right) = 0,$$

where  $p = \frac{3}{2}$ . Equation (1.6) has come to be known as doubly nonlinear parabolic equation of p-Laplace type. See e.g. [37], [7]. Notice that  $p < 2$  and that the system is, therefore, singular for  $\frac{\partial}{\partial x}y(x) = 0$ . For  $p > 2$  such equations exhibit instead degeneration. Equations similar to (1.5) have been considered in the literature, see e.g. [4, 5]. In this contribution, we aim at a discussion of such equations together with optimal control problems on networks. A more recent study of doubly nonlinear parabolic equations in the context of friction dominated flow has been provided in [2]. Equations of the type (1.5) are known to exhibit positive solutions and satisfy a maximum principle. As a matter of fact, to the best knowledge of the authors, there are no studies on optimal control of such systems on general graphs available from the literature besides [26, 27]. We note that the doubly nonlinear parabolic problem associated with the friction-dominated flow on a network has been considered in the thesis [39]. Optimal control problems for the p-Laplace operator have been studied since the 1980ies, see e.g. [9]. Moreover, in [41], an optimal control problem for the p-Laplace equation  $p \geq 2$  has been recently considered. See also [11] for a problem of optimal control in the coefficient for the p-Laplace equation, again for  $p \geq 2$ .

We first recall the network modelling from [27] (also presented in [39]), formulate an exemplary optimal control problem together with its instantaneous (rolling horizon) approximation via a time discretization. The resulting static optimal control for each time step is then based on a static p-Laplace problem on the network. This, in turn, can be handled by a primal or a primal-dual approach. In both cases, a domain decomposition method at the level of static p-Laplace problems is at order. We will work on both approaches. The primal-dual approach relies on a first order optimality condition. We aim to iteratively decompose this coupled system of the state and the adjoint equation. We will focus on a particular feature, namely that, the decomposed optimality system is itself an optimality system for an optimal control problem on the decomposed domain involving so-called *virtual controls*. Thus, in the numerical realization, one may resort to any method to solve these local *virtual constrained control problems* that are iteratively coupled but at each iteration fully parallel. Finally, one closes the cycle in advancing the time step.

The original problem for the time-dependent doubly nonlinear p-parabolic equation can also be handled directly. In this context a time domain decomposition maybe desired to break down the possible long time horizon to smaller time intervals. In fact, on the small but still continuous time intervals one may consider a space domain decomposition of the graph, leading to a nonoverlapping space-time domain decomposition into suitably small sub-graphs or even edges on small time intervals. The corresponding methods share the feature described above.

Finally, there is the desire in complex application, as gas transport in large scale networks, to resort to surrogate models in certain subnetworks. This can be approached using physics informed neural networks (PINN). We dwell on this possibility in the outlook.

We note that such non-overlapping domain decompositions have not been considered in the literature so far with the exception of [12], where, for  $p \geq 2$ , a similar problem without control has been considered, however, with a different updating rule that is more related to a Gauß-Seidel-type iteration based on a Peaceman-Rachford scheme and, hence, is not completely parallel. We refer to [26] and a forthcoming publication [25]. As for time-domain decomposition, we refer to [22], [13], [14], [15], [31].

**1.2. Network modeling.** We start with the notation for the description of problems on metric graphs. To this end, let  $G = (V, E)$  denote the graph of the gas network with vertices (nodes)  $V = \{n_1, n_2, \dots, n_{|V|}\} = \{n_j | j \in \mathcal{J}\}$  and edges  $E = \{e_1, e_2, \dots, e_{|E|}\} = \{e_i | i \in \mathcal{I}\}$ . We associate to each edge a direction.

$$d_{ij} = \begin{cases} -1, & \text{if node } n_j \text{ if the the edge } e_i \text{ starts at node } n_j e_i, \\ +1, & \text{if node } n_j \text{ if the edges } e_i \text{ end at node } n_j e_i, \\ 0, & \text{else.} \end{cases}$$

We provide the network description of both the semilinear model (1.2) and the doubly nonlinear paparabolic model (1.3). Physically, the pressure variables  $p_i(n_j)$  (for (1.2)) and  $y_i(n_j)$  (for (1.3)) coincide for all  $i \in \mathcal{I}_j := \{i \in 1, \dots, E | d_{ij} \neq 0\}$ , while the fluxes at a multiple joint  $n_j$  which are represented by  $d_{ij}q_i(n_j)$  and  $d_{ij}\beta_i(\partial_x y_i(n_j))$ ,

respectively, add up to zero. We introduce the edge degree  $d_j := |\mathcal{I}_j|$ . We use the  $d_j$  in order to decompose the index set for nodes  $\mathcal{J}$  into  $\mathcal{J} = \mathcal{J}^M \cup \mathcal{J}^S$ , where  $\mathcal{J}^M = \{j \in \mathcal{J} | d_j > 1\}$  represents the multiple nodes and  $\mathcal{J}^S = \{j \in \mathcal{J} | d_j = 1\}$  the simple nodes. According to Dirichlet or Neumann boundary conditions a the simple nodes, we further decompose  $\mathcal{J}^S = \mathcal{J}_D^S \cup \mathcal{J}_N^S$ . We summarize the equations as follows: the network of semilinear equations (1.2) is governed by

$$\begin{aligned}
(1.7) \quad & \frac{\partial p_i}{\partial t} + \frac{c^2}{a} \frac{\partial}{\partial x} q_i = 0 \\
& \frac{\partial}{\partial t} q_i + a \frac{\partial p_i}{\partial x} = -\frac{\lambda c^2}{2aD p_i} q_i |q_i|, \quad i \in \mathcal{I}, x \in (0, \ell_i), t \in (0, T) \\
& p_i(n_j, t) = p_k(n_j, t), \quad \forall i, k \in \mathcal{I}_j, j \in \mathcal{J}^M, t \in (0, T) \\
& \sum_{i \in \mathcal{I}_j} d_{ij} q_i(n_j, t) = 0, \quad j \in \mathcal{J}^M, t \in (0, T) \\
& p_i(n_j, t) = 0, i \in \mathcal{I}_j, j \in \mathcal{J}_D^S, t \in (0, T) \\
& d_{ij} q_i(n_j, t) = u_j(t), \quad i \in \mathcal{I}_j, j \in \mathcal{J}_N^S, t \in (0, T) \\
& p_i(x, 0) = p_{i,0}(x), \quad q_i(x, 0) = q_{i0}(x), \quad x \in (0, \ell_i), i \in \mathcal{I},
\end{aligned}$$

while the doubly nonlinear p-parabolic network (for (1.3)) is described as follows

$$\begin{aligned}
(1.8) \quad & \alpha_i \partial_t \beta(y_i(x, t)) - \partial_x (\beta(\partial_x y_i(x, t))) = u_i, \quad i \in \mathcal{I}, x \in (0, \ell_i), t \in (0, T) \\
& y_i(n_j, t) = y_k(n_j, t), \quad \forall i, k \in \mathcal{I}_j, j \in \mathcal{J}^M, t \in (0, T) \\
& \sum_{i \in \mathcal{I}_j} d_{ij} \beta(\partial_x y_i(n_j, t)) = 0, \quad j \in \mathcal{J}^M, t \in (0, T) \\
& y_i(n_j, t) = 0, i \in \mathcal{I}_j, j \in \mathcal{J}_D^S, t \in (0, T) \\
& d_{ij} \beta(\partial_x y_i(n_j, t)) = u_j(t), \quad i \in \mathcal{I}_j, j \in \mathcal{J}_N^S, t \in (0, T) \\
& y_i(x, 0) = y_{i,0}(x), \quad x \in (0, \ell_i),
\end{aligned}$$

where the functions  $u_i, i \in \mathcal{I}, u_j, j \in \mathcal{I}_j, j \in \mathcal{J}_N^S$  serve as controls.

## 2. OPTIMAL CONTROL PROBLEMS

**2.1. Problem formulation.** We are now in the position to formulate optimal control problems on the level of the gas networks. We first describe the general format for an optimal control problem. This involves a cost function that assigns to each admissible pair  $(y, u)$  a 'cost'  $I(y, u)$ , which is represented on each individual edge by a contribution on the state  $I_i(y)$  and the controls acting at simple nodes. The typical example, the one that we will use in the sequel is given by

$$(2.1) \quad I_i(q_i)(x) := \frac{\kappa}{2} |q_i(x) - q_i^d(x)|^2, \quad x \in (0, \ell_i),$$

for the system (1.7) and

$$(2.2) \quad I_i(y_i)(x) := \frac{1}{r} |y_i(x) - y_i^d(x)|^r, \quad x \in (0, \ell_i), r \in [\frac{3}{2}, 2].$$

for system (1.8). For system (1.7), we consider the optimal control problem

$$(2.3) \quad \begin{aligned} \min_{(p,q,u) \in \Xi} I(p,q,u) &:= \sum_{i \in \mathcal{I}} \int_0^T \int_0^{\ell_i} I_i(q_i) dx dt + \frac{\nu}{2} \sum_{j \in \mathcal{J}_N^S} \int_0^T |u_j(t)|^2 dt \\ &s.t. \\ &(p,q,u) \text{ satisfies (1.7),} \end{aligned}$$

$$(2.4) \quad \Xi := \{(p,q,u) : \underline{p}_i \leq p_i \leq \bar{p}_i, i \in \mathcal{I}, \underline{u}_j \leq u_j \leq \bar{u}_j, j \in \mathcal{J}_N^S\}$$

and analogously for (1.8)

$$(2.5) \quad \begin{aligned} \min_{(y,u) \in \Xi} I(y,u) &:= \sum_{i \in \mathcal{I}} \int_0^T \int_0^{\ell_i} I_i(y_i) dx dt + \frac{\nu}{2} \sum_{j \in \mathcal{J}_N^S} \int_0^T |u_j(t)|^2 dt \\ &s.t. \\ &(y,u) \text{ satisfies (1.8),} \end{aligned}$$

$$(2.6) \quad \bar{\Xi} := \{(y,u) : \underline{p}_i \leq y_i \leq \bar{y}_i, i \in \mathcal{I}, \underline{u}_j \leq u_j \leq \bar{u}_j, j \in \mathcal{J}_N^S\}.$$

In (2.4),(2.6), the quantities  $\underline{p}_i, \bar{p}_i, \underline{y}_i, \bar{y}_i$  are given constants that determine the feasible pressures and flows in the pipe  $i$ , while  $\underline{u}_i, \bar{u}_i$  describe control constraints. (2.3), (2.5) are typical examples of optimal control problems for systems of semi-linear hyperbolic and fully nonlinear parabolic equations on metric graphs. For the first type of problems ((2.3)), we refer to [21], [20] and [26], while the second set of problems ((2.5) has been considered in [27] and [26]. To the best knowledge of the author, for problem (2.5) no published result on well-posedness and characterization of optimal controls seems to be available in the literature.

We describe the corresponding optimality systems. As for (1.7) and (2.3), we introduce the adjoint variables  $\phi, \psi$  and denote  $f_i(p_i, q_i) := \frac{\lambda c^2}{2aDp_i} q_i |q_i|$ . For the sake of brevity, we take the case without control constraints.

$$(2.7) \quad \begin{aligned} \frac{\partial p_i}{\partial t} + \frac{c^2}{a} \frac{\partial}{\partial x} q_i &= 0 \\ \frac{\partial}{\partial t} q_i + a \frac{\partial p_i}{\partial x} &= -f_i(p_i, q_i) - \frac{1}{\nu} \phi_i, \quad i \in \mathcal{I}, x \in (0, \ell_i), \\ \frac{\partial \phi_i}{\partial t} + \frac{c^2}{a} \frac{\partial}{\partial x} \psi_i &= \partial_q f_i(p_i, q_i)^* \phi_i \\ \frac{\partial}{\partial t} q_i + a \frac{\partial p_i}{\partial x} &= \partial_p f_i(p_i, q_i)^* \psi_i + \kappa(q_i - q_i^d) \quad i \in \mathcal{I}, x \in (0, \ell_i), t \in (0, T) \\ p_i(n_j, t) &= p_k(n_j, t), \quad \phi_i(n_j, t) = \phi_k(n_j, t) \quad \forall i, k \in \mathcal{I}_j, j \in \mathcal{J}^M, t \in (0, T) \\ \sum_{i \in \mathcal{I}_j} d_{ij} q_i(n_j, t) &= 0, \quad \sum_{i \in \mathcal{I}_j} d_{ij} \psi_i(n_j, t) = 0 \quad j \in \mathcal{J}^M, t \in (0, T) \\ p_i(n_j, t) &= 0, \quad \phi_i(n_j, t) = 0, \quad i \in \mathcal{I}_j, j \in \mathcal{J}_D^S, t \in (0, T) \\ d_{ij} q_i(n_j, t) &= u_j(t), \quad d_{ij} \psi_i(n_j, t) = 0, \quad i \in \mathcal{I}_j, j \in \mathcal{J}_N^S, t \in (0, T) \\ p_i(x, 0) &= p_{i,0}(x), \quad q_i(x, 0) = q_{i0}(x), \quad \phi_i(x, T) = 0, \quad \psi_i(x, T) = 0, \quad x \in (0, \ell_i). \end{aligned}$$

The corresponding optimality system for (1.8) and (2.5) reads as



*Example 2.1.* We consider the second problem (2.9) for a star graph with 3 edges. Edge 1 starts at a Dirichlet node at  $x = 0$  and connects to edges 2 and 3 at  $x = 1$  where the other two edges emerge from  $x = 0$  each. We impose a homogeneous Dirichlet condition for edge 2 at  $x = 1$  and a Neuman-control for edge 3 at  $x = 1$ . We include pointwise box-control constraints on the flux in  $[0.5, 1.5]$ , only. We take initial data  $y_0 = .5\sin(\pi x)^2$  for edge 2 and zero initial data for edges 1 and 3. As a target, we put  $y_d \equiv 1$  for all edges. It is obvious that the set-up is purely academic, but this is due to the space-limitations. Moreover, we use the standard discretization, already introduced in [4]. We take 50 spatial discretization points for each edges defined on  $[0, 1]$  and a 100 time discretization points for the time interval  $[0, 10]$ . Moreover, we use a  $1e. - 5$  regularization of the p-Laplacian in order to avoid large condition numbers of the Jacobians involved. The optimization is done using casADI with the mums-solver for linear systems. We refer to the page <https://web.casadi.org> for details of the optimization procedure. See fig. 1.

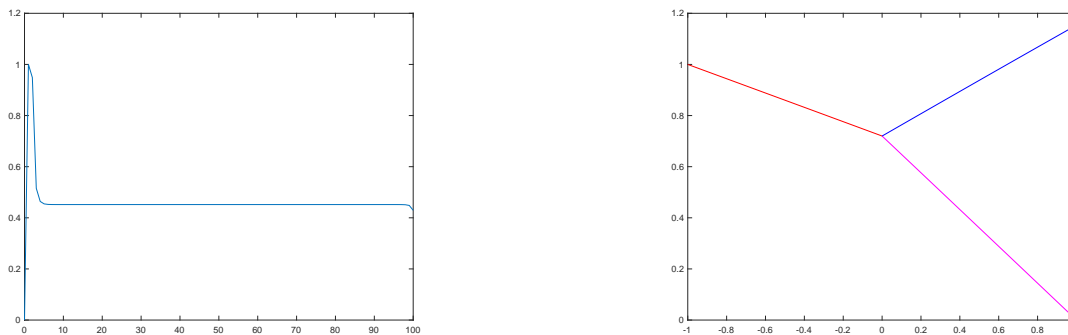


FIGURE 1. Right: control; Left: States

Objective :	1.8998825239667934e + 02	Dual infeasibility :	1.8474111129762605e - 12
Constraint violation :	9.7978365693480448e - 16	Overall NLP error :	7.2287062056791611e - 09

*Example 2.2.* We now consider the semi-linear problem (2.3). As the situation and its representation in figures becomes very complex easily, we just take a single pipe. Indeed, the purpose at this point is just to give an impression on the fully discrete approach, handled with professional software as IPOPT without resorting to DDMs. We take, as an artificial nominal value, constant  $p_0 \equiv .1$  initial data for the pressure and a linearly decreasing flux. We apply a box-constrained control  $u(t) \in [.1, .4]$  at  $x = 0$  to the flux  $q(t, 0)$ . We want to achieve a terminal constant flux  $q(T, x) \equiv \frac{1}{2}$ , however, under the state constraints for the pressure  $p(t, x) \in [0, .5]$ . We use IPOPT as in the last example 2.1. We display the results in fig. 2. In the top, we show on the left the pressure and on the right the flux. On the bottom, we show on the left the control with state constraints. The right figure on the bottom shows the control with constraints in  $[0.1, .4]$  and no state constraints for which we do not show the corresponding pressure/flux due to lacking space. The controls are reminiscent of bang-bang-type controls. Even though it is obvious from our (further) experiments that the switching pattern depends on the discretization, we have no analysis on how! It may well happen that a *chattering control* emerges that would not appear a result of a purely continuous treatment. This is yet another reason for working (also) with the *first-optimize-then-discretize* approach.

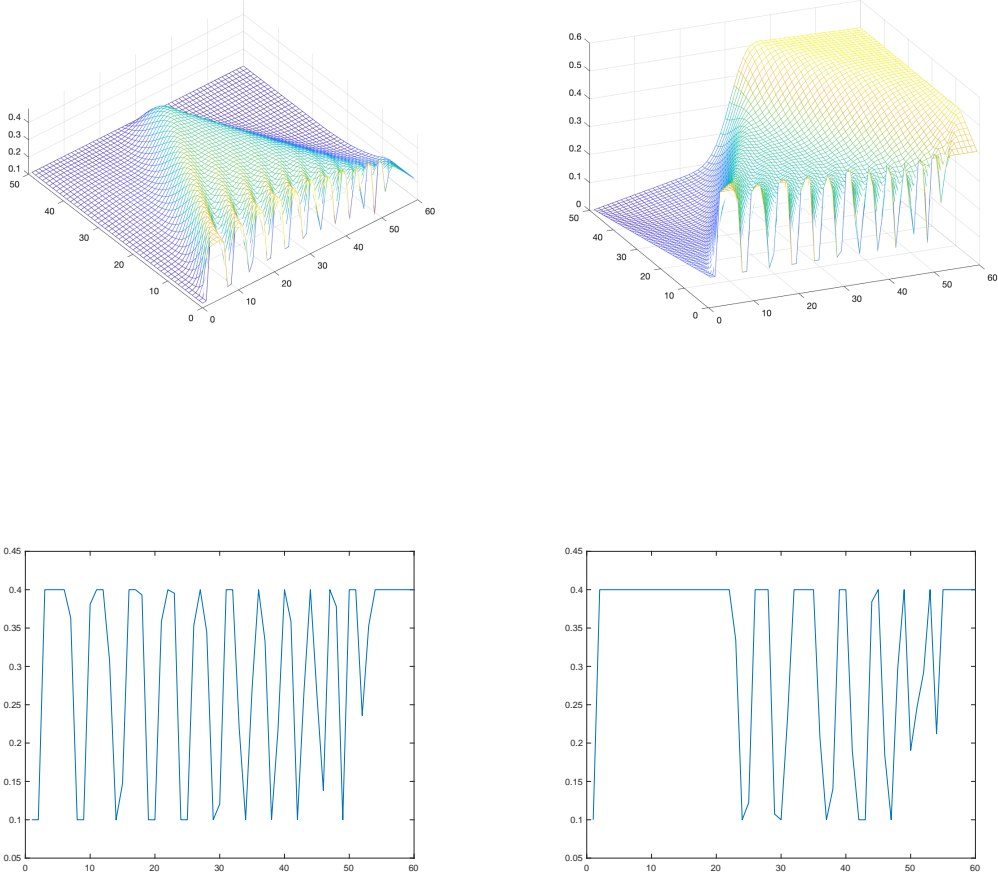


FIGURE 2. Top: left: pressure, right: flux; Bottom: control, left: with state constraints, right: without

**2.3. Instantaneous control.** It is clear that (2.9) involves all time steps in the cost functional. We would like to reduce the complexity of the problem even further. To this aim, we consider the so-called *instantaneous control* regime. This amounts to reducing the sums in the cost function of (2.9) to the time-level  $t_{n+1}$ . This strategy has also come to be known as *rolling horizon* approach, the simplest case of the *moving horizon* paradigm. It is now convenient to discard the actual time level  $n + 1$  and redefine the states at the former time as input data. To this end, we replace  $\alpha_i := \frac{1}{\Delta t}$ ,  $f_i^1 := \alpha_i \beta(y_{i,n})$ . Thus, for each  $n = 1, \dots, N - 1$  and given  $y_{i,n}$ , we consider the problems



$$(2.10) \quad \min_{(y,u)} I(y,u) := \sum_{i \in \mathcal{I}} \int_0^{\ell_i} I_i(y_i) dx + \frac{\nu}{2} \sum_{j \in \mathcal{J}_N^S} |u_j|^2$$

*s.t.*

$$(2.11) \quad \begin{aligned} \alpha_i \beta(y_i)(x) - \partial_x (\beta(\partial_x y_i(x))) &= f_i^1(x), \quad i \in \mathcal{I}, \quad x \in (0, \ell_i) \\ y_i(n_j) &= y_k(n_j), \quad \forall i, k \in \mathcal{I}_j, \quad j \in \mathcal{J}^M \\ \sum_{i \in \mathcal{I}_j} d_{ij} \beta(\partial_x y_i)(n_j) &= 0, \quad j \in \mathcal{J}^M \\ \beta(\partial_x y_i)(n_j) &= u_j, \quad d_j = 1, \quad i \in \mathcal{I}_j, \quad j \in \mathcal{J}_N^S \\ y_i(n_j) &= 0, \quad i \in \mathcal{I}_j, \quad j \in \mathcal{J}_D^S \end{aligned}$$

Wellposedness of (2.11) has been discussed in [27]. We recall the results as follows.

**Theorem 2.3** ([27]). *For  $f \in \Pi_{i \in \mathcal{I}} L^3(0, \ell_i)$ ,  $u \in \mathbb{R}^{|\mathcal{J}_N^S|}$ , problem (2.11) admits a unique weak solution  $y \in V$ .*

We refer to section 3 for the space setting. Even though, the mapping  $\beta(\cdot)$  is differentiable in  $\mathbb{R} \setminus \{0\}$ , the control-to-state-mapping  $u \rightarrow y^u$  is not Gâteaux differentiable for  $p < 2$ . This has already been observed in [10]. However, the control-to-state-map is continuous. By the continuity of  $\beta(\cdot)$  and the strong convergence of  $y^k$  to  $y^0$  in  $V$ , we obtain

**Theorem 2.4** ([27]). *The mapping  $u \rightarrow y^u$ , where  $y^u$  solves (2.11) is continuous between  $\mathbb{R}^{|\mathcal{J}_N^S|}$  and  $V$ .*

As for the existence of optimal pairs  $(y, u)$  for the optimal control problem (2.10), we cite

**Theorem 2.5** ([27]). *The optimal control problem (2.10) admits a unique solution  $(\bar{y}, \bar{u}) \in V \times \mathbb{R}^{|\mathcal{J}_N^S|}$ .*

*Example 2.6.* We now use the same set-up as in Example 2.1, but now we perform the instantaneous control strategy. See fig. 3.

Objective :	7.6923076923120462e - 02	Dual infeasibility :	5.7909232964448165e - 13
Constraint violation :	1.0852975892908631e - 15	Overall NLP error :	2.5059037031159630e - 09

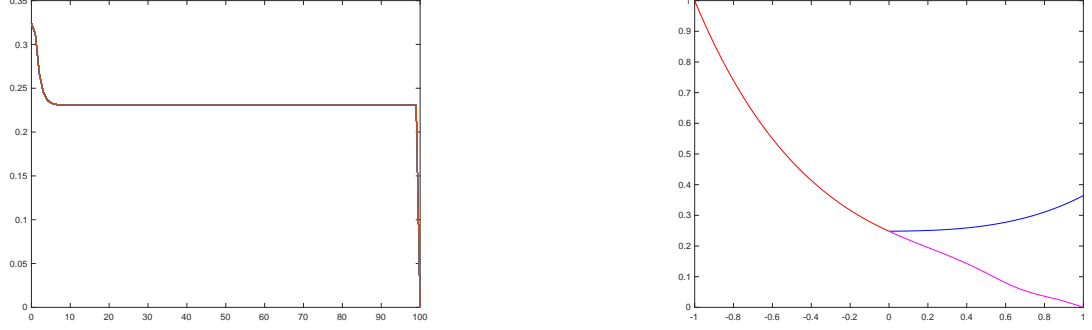


FIGURE 3. Left figure:  $p = 1.5$  control, right figure:  $p = 1.5$ , final states

**Theorem 2.7** ([27]). *There exists  $\bar{p} \in V$  satisfying together with the optimal pair  $(\bar{y}, \bar{u})$  the first order optimality condition.*

$$\begin{aligned}
 & \alpha_i \beta(\bar{y}_i) - \partial_x (\beta(\partial_x \bar{y}_i)) = -\frac{1}{v} p_i, \quad i \in \mathcal{I}, \quad x \in (0, \ell_i) \\
 & \alpha_i \beta'(\bar{y}_i) \bar{p}_i - \partial_x (\beta'(\partial_x \bar{y}_i) \partial_x \bar{p}_i) = \kappa (|\bar{y}_i - y_i^d|^{p-2} (\bar{y}_i - y_i^d)), \quad i \in \mathcal{I}, \quad x \in (0, \ell_i) \\
 & \bar{y}_i(n_j) = \bar{y}_k(n_j), \quad \forall i, k \in \mathcal{I}_j, \quad \bar{p}_i(n_j) = \bar{p}_k(n_j), \quad \forall i, k \in \mathcal{I}_j, \quad j \in \mathcal{J}^M \\
 & \sum_{i \in \mathcal{I}_j} d_{ij} \beta(\partial_x \bar{y}_i)(n_j) = 0, \quad j \in \mathcal{J}^M \\
 & \sum_{i \in \mathcal{I}_j} d_{ij} \beta'(\partial_x \bar{y}_i)(n_j) \partial_x \bar{p}_i(n_j) = 0, \quad j \in \mathcal{J}^M \\
 & \beta(\partial_x \bar{y}_i)(n_j) = \bar{u}_j, \quad d_j = 1, \quad i \in \mathcal{I}_j, \quad j \in \mathcal{J}_N^S \\
 & \beta'(\partial_x \bar{y}_i)(n_j) \partial_x \bar{p}_i(n_j) = 0, \quad d_j = 1, \quad i \in \mathcal{I}_j, \quad j \in \mathcal{J}_N^S \\
 & \bar{y}_i(n_j) = 0, \quad \bar{p}_i(n_j) = 0 \quad i \in \mathcal{I}_j, \quad j \in \mathcal{J}_D^S \\
 & \bar{p}_i^\epsilon(n_j) + v \bar{u}_j = 0, \quad i \in \mathcal{I}_j, \quad j \in \mathcal{J}_N^S.
 \end{aligned}
 \tag{2.12}$$

One way to handle this problem is to use an iterative procedure. To this end, we introduce the common algorithm

- Algorithm 2.8.* (1) Choose  $u^0$   
(2) For  $i = 1$  until satisfied  
(3) (a) Solve for  $y_i$

$$\begin{cases}
 \beta_i(y_i) - \partial_x (\beta_i(\partial_x y_i)) = u_i, \quad i = 1, 2, \quad x \in I_i \\
 y_1(0) = 0, \quad y_2(2) = 0, \\
 y_1(1) = y_2(1), \quad \beta_1(\partial_x y_1)(1) = \beta_2(\partial_x y_2)(1).
 \end{cases}$$

(b) Solve for  $p_i$

$$\begin{cases} \beta'_i(y_i)p_i - \partial_x(\beta'_i(\partial_x y_i)\partial_x p_i) = \kappa(|y_i - y_i^d|^{r-2}(y_i - y_i^d)), i = 1, 2, x \in I_i \\ p_1(0) = 0, p_2(2) = 0, \\ p_1(1) = p_2(1), \beta'_1(\partial_x y_1)(1)\partial_x p_1(1) = \beta'_2(\partial_x y_2(1))\partial_x p_2(1). \end{cases}$$

(c) Set  $u = \frac{1}{\nu}p$

(4) Terminate with the (approximate) fixed point  $u, y, p$ .

*Remark 2.9.* algorithm is classic for linear problems. For the p-Laplace operator, this needs further arguments. For the sake of brevity, we leave the convergence analysis to a forthcoming paper. See, however [41].

(1) We remark that the analogous procedure can be used for the original optimal control problems (2.3), (2.5) on the continuous time level. We omit the details due to space limitations.

### 3. NON-OVERLAPPING DOMAIN DECOMPOSITION

The point of interest here is that, giving up the concept of decomposing the optimality system at once, we can concentrate on the two systems in step 3. iteratively. That is to say, we first apply the domain decomposition procedure to the state system (a) and then to the adjoint system (b). Of course, the adjoint system is linear, given the state  $y_i$ . Hence, the classical P. Lions type non-overlapping domain decompositions applies. See e.g. [22] for the details. Thus, the problem is with the state system. To the best knowledge of the author, not even for this p-Laplace type problem results are available in the literature. The only exception is a very recent article [12] where the p-Laplace problem is treated, however, with a slightly but importantly different iteration which, in turn, is reminiscent of a GaußSeidel type version of the P.L. Lions method, whereas the method of consideration here is more a Jacobi-type iteration and is, therefore, completely parallel.

**3.1. Decomposition of the state equation: the concept of virtual controls.** As a result, we have seen that there are different levels of potential domain decomposition: domain decomposition for the time dependent optimal control problems, domain decomposition for the semi-discrete and finally fully-discrete optimal control problems (in these notes the instantaneous control paradigm) and the method that uses domain decomposition of the system of states and adjoint states independently, essentially just for the purpose of enhancing the simulation, while the previously mentioned methods explore the intrinsic interaction of DDMs and optimal control. Historically, one started with the last option, as the methods were designed for the numerical simulation to large scale applications. This is where we start also this survey. We go back to the articles by J.L. Lions and O. Pironneau [32] and P.L. Lions [35]. The reason for doing so is that these articles paved the way to what has come to be know as the concept of *virtual controls*. The basic idea is to use concept of controllability and optimal tracking control in order to restore the mismatch between the states and the fluxes at artificial interfaces, resulting in a non-overlapping domain decomposition paradigm. For problems on metric graphs, the non-overlapping domain decomposition paradigm is very adequate as it can be interpreted as a *structuring* method. Overlapping domain decomposition at multiple nodes, therefore, doesn't seem to be intuitive. In these notes, we, therefore, stay with non-overlapping methods.

**3.1.1. The Lions-Pironneau method for the p-Laplacian on graphs.** In the sequel, we extend the method of J.L. Lions and O. Pironneau [32] to p-Laplace-type problems on metric graphs. To the best knowledge of the author, there is no literature available for this part. We prefer to work in a variational framework and, for the convenience of reader, we use the standard notation  $u$  for the state, instead of  $y$  as in the context of PDE-control. To this end, we

begin with some notation. We introduce the individual form

$$(3.1) \quad a_i(u_i, y_i) = \int_0^{\ell_i} \beta_i(\partial_x u_i) \partial_x v_i + \beta_i(u_i) v_i dx.$$

In order to consistently refer to nodal values, we denote  $u_i(x_{ij})$  the nodal value of  $u_i$  at the node  $n_j$ .  $x_{ij} = 0$  if  $d_{ij} = -1$  and  $x_{ij} = \ell_i$  if  $d_{ij} = 1$ . We introduce the space

$$(3.2) \quad V_i := \{\phi_i \in L^2(0, \ell_i) := H_i \mid \beta_i(\phi_i), \beta_i(\partial_x \phi_i) \in H_i, \phi(x_{ij}) = 0, \forall j \in \mathcal{J}_D^S, i \in \mathcal{I}\}.$$

On the boundary of the edge  $e_i$ , we introduce the forms

$$(3.3) \quad b_{ij}(\lambda_{ij}, v_i) := \alpha_{ij} \lambda_{ij} v_i(x_{ij}), j \in \mathcal{J}, i \in \mathcal{I}.$$

Here,  $\alpha_{ij} = 1$  if  $d_{ij} \neq 0$  and 0 otherwise. For each edge  $e_i$  there are exactly two nodes  $j, k$  such that  $i \in \mathcal{I}_{j_i} \cap \mathcal{I}_{k_i}$ . To put it the other way round, we introduce  $\mathcal{J}_i := \{j \in \mathcal{J} \mid i \in \mathcal{I}_j\}$ , the set of all edges incident at node  $n_j$ . Now, for each edge and  $f_i \in H_i$  we can write down the variational equation

$$(3.4) \quad a_i(u_i, v_i) = (f_i, v_i)_{H_i} + \sum_{j \in \mathcal{J}_i} b_{ij}(\lambda_{ij}, v_i), \forall v_i \in V_i.$$

As for the entire graph problem, we introduce

$$(3.5) \quad a(u, v) := \sum_{i \in \mathcal{I}} a_i(u_i, v_i)$$

on

$$(3.6) \quad V = \{\phi \in \prod_{i \in \mathcal{I}} V_i \mid \phi_i(x_{ij}) = \phi_k(x_{kj}), \forall i, k \in \mathcal{I}_j, j \in \mathbb{J}^M\}.$$

On the entire set of nodes, we have

$$(3.7) \quad b(\lambda, v) := \sum_{j \in \mathcal{J}} \sum_{i \in \mathcal{I}_j} \lambda_{ij} v_i(x_{ij}).$$

With  $(f, v) := \sum_{i \in \mathcal{I}} (f_i, v_i)$ , we can formulate the general graph problem for the p-Laplace system.

$$(3.8) \quad a(u, v) = (f, v) + b(\lambda, v), \forall v \in V.$$

In order to decompose this system, one needs to relax the continuity conditions  $u_i(x_{ij}) = u_k(x_{ki}), \forall i, k \in \mathcal{I}_i$  and also the Kirchhoff conditions  $\sum_{i \in \mathcal{I}_j} d_{ij} \beta_i(\partial_x u_i)(x_{ij}) = 0, \forall j \in \mathcal{J}^M$ . The framework for this uses the  $\lambda_{ij}$  as *virtual controls* in the following way.

$$(3.9) \quad \begin{aligned} \min_{\lambda, u} J(\lambda) &:= \frac{1}{2} \sum_{j \in \mathcal{J}^M} \sum_{i \in \mathcal{I}_i} |\lambda_{ij}|^2 \\ \text{s. t.} & \\ a_i(u_i, v_i) &= (f_i, v_i) + \sum_{j \in \mathcal{J}_i} b_{ij}(\lambda_{ij}, v_i(x_{ij})), \forall v_i \in V_i \\ u_i(x_{ij}) &= u_k(x_{kj}), \forall i, k \in \mathcal{I}_i \\ \sum_{i \in \mathcal{I}_j} \lambda_{ij} &= 0. \end{aligned}$$

Problem (3.9) is an optimal control problem with  $\lambda$  as control which, in turn, is subject to control constraints (3.9)<sub>5</sub>, namely,  $\sum_{i \in \mathcal{I}_j} \lambda_{ij} = 0$ . We now relax the constraints by replacing  $\lambda_{ij}$  by  $\lambda_{ij} - \frac{1}{d_j} \sum_{l \in \mathcal{I}_i} \lambda_{lj}$  because the sum over  $i \in \mathcal{I}_i$  of these new controls is zero. In order to replace the continuity condition, it is convenient to write them as follows:

$$(3.10) \quad u_i(x_{ij}) = \frac{1}{d_j} \sum_{l \in \mathcal{I}_j} u_l(x_{lj}) = 0, \quad \forall i \in \mathcal{I}_j.$$

We now introduce a Lagrange relaxation of (3.10), using a Lagrange multiplier  $q_{ij}$ . We, therefore, consider the saddle-point problem

$$(3.11) \quad \inf_{\lambda, u} \sup_q \left\{ J(\lambda) + \sum_{j \in \mathcal{J}^M} \sum_{i \in \mathcal{I}_j} q_{ij} \left( u_i(x_{ij}) - \sum_{l \in \mathcal{I}_j} u_l(x_{lj}) \right) \right\}$$

s.t.

$$a_i(u_i, v_i) = (f_i, v_i) + \sum_{j \in \mathcal{J}_i} \left( \lambda_{ij} - \frac{1}{d_j} \sum_{l \in \mathcal{I}_i} \lambda_{lj} \right).$$

Using the Lagrange principle, we deduce the following optimality system with respect to control variable  $\lambda$ :

$$(3.12) \quad a_i(u_i, \hat{u}_i) = (f_i, \hat{u}_i) + \sum_{j \in \mathcal{J}_i} \left( p_i(x_{ij}) - \frac{1}{d_j} \sum_{l \in \mathcal{I}_j} p_j(x_{lj}) \right) \hat{u}_i(x_{ij}), \quad \forall \hat{u}_i \in V_i$$

$$(3.13) \quad a_i^*(p_i, \hat{p}_i) = - \sum_{j \in \mathcal{J}_i} \left( q_{ij} - \frac{1}{d_j} \sum_{l \in \mathcal{I}_i} q_{lj} \right) \hat{p}_i(x_{ij}), \quad \forall \hat{p}_i \in V_i,$$

where

$$a_i^*(p_i, \hat{p}_i) := a'_i(\hat{p}_i, p_i; u_i) := \int_0^{\ell_i} \beta'_i(\partial_x u_i) \partial_x \hat{p}_i \partial_x p_i + \beta'_i(u_i) \hat{p}_i p_i dx.$$

As for the gradient with respect to  $q$ , we introduce

$$(3.14) \quad I(q) := - \inf_{\lambda} \left\{ \frac{1}{2} \sum_{j \in \mathcal{J}^M} \sum_{i \in \mathcal{I}_i} |\lambda_{ij}|^2 + \sum_{j \in \mathcal{J}^M} \sum_{i \in \mathcal{I}_j} q_{ij} \left( u_i(x_{ij}) - \frac{1}{d_j} \sum_{l \in \mathcal{I}_j} u_l(x_{lj}) \right) \right\}.$$

Then, the gradient with respect to  $q$  is

$$(3.15) \quad \partial_q I(q) = u_i(x_{ij}) - \frac{1}{d_j} \sum_{l \in \mathcal{I}_j} u_l(x_{lj}).$$

The details for this derivation are standard and, therefore, left to the reader. We now describe the extension of the Lions-Pironneau algorithm to problems on graphs.

*Algorithm 3.1.* (1) Given  $u^k, p^k, \lambda^k$   
(2) Update  $q^k$  to  $q^{k+1}$ :

$$(3.16) \quad q_{ij}^{k+1} = q_{ij}^k - \rho \left\{ u_i^k(x_{ij}) - \frac{1}{d_j} \sum_{l \in \mathcal{I}_j} u_l^k(x_{lj}) \right\},$$

(3) Solve for  $u^{k+1}, p^{k+1}$

$$(3.17) \quad a_i(u_i^{k+1}, \hat{u}_i) = (f_i, \hat{u}_i) + \sum_{j \in \mathcal{J}_i} \left( p_i^{k+1}(x_{ij}) - \frac{1}{d_j} \sum_{l \in \mathcal{I}_j} p_j^{k+1}(x_{lj}) \right) \hat{u}_i(x_{ij}), \quad \forall \hat{u}_i \in V_i$$

$$(3.18) \quad a_i^*(p_i^{k+1}, \hat{p}_i) = - \sum_{j \in \mathcal{J}_i} \left( q_{ij}^{k+1} - \frac{1}{d_j} \sum_{l \in \mathcal{I}_i} q_{lj}^{k+1} \right) \hat{p}_i(x_{ij}), \quad \forall \hat{p}_i \in V_i,$$

(4) Return to 1.)

*Remark 3.2.* (1) Algorithm 3.1, for two domains and linear state equation ( $p = 2$ ) is precisely the algorithm presented in [32].

(2) In the linear case ( $p = 2$ ), one first solves the adjoint problem for the given  $q^{k+1}$  to obtain  $p^{k+1}$  which is then used in the forward problem in order to solve for  $u^{k+1}$ .

(3) The algorithm does not provide an iterative decomposition of the optimality system (3.12), (3.13).

(4) The author is not aware of any proof of convergence on the continuous level.

*Example 3.3.* We take a star-graph with 3 edges. Edge 1 stretches from  $x = -1$  to  $x = 0$ , where the edges 2 and 3 emanate. We consider the linear problem ( $p = 2$ ), only. Each edge satisfies homogeneous Dirichlet conditions at the simple nodes. We impose a constant load  $f_i \equiv 1, i = 1, 2, 3$ . For the step-size parameter  $\rho$ , we choose  $\rho = 1$ . See fig. 4. The solutions and derivatives for edges 2 and 3 are plotted on top of each other.

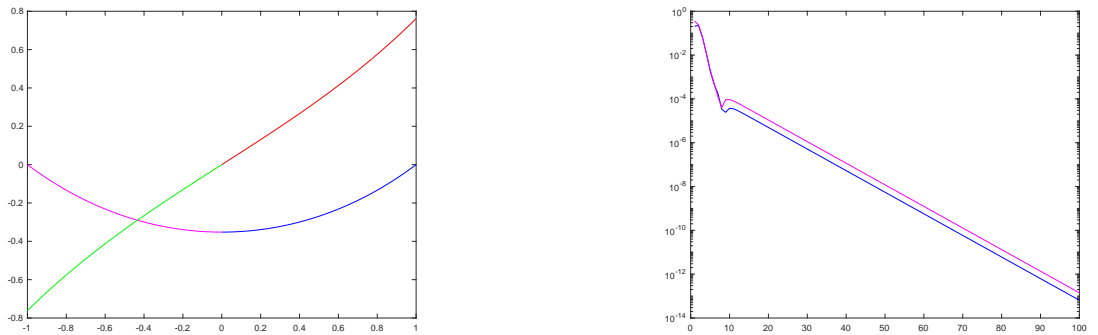


FIGURE 4. Left figure:  $p = 2$  state and derivative, right figure: errors at the multiple node

We now introduce a second Lagrange multiplier for the relaxation of the Kirchhoff condition and obtain the following saddle-point problem.

$$(3.19) \quad \inf_{\lambda, u} \sup_q \left\{ J(\lambda) + \sum_{j \in \mathcal{J}^M} \sum_{i \in \mathcal{I}_j} q_{ij} \left( u_i(x_{ij}) - \frac{1}{d_j} \sum_{l \in \mathcal{I}_j} u_l(x_{lj}) \right) + \sum_{j \in \mathcal{J}^M} \mu_j \sum_{i \in \mathcal{I}_j} \lambda_{ij} \right\}$$

s.t.

$$a_i(u_i, v_i) = (f_i, v_i) + \sum_{j \in \mathcal{J}_i} (\lambda_{ij}).$$

We derive the following algorithm, which is an extension of the corresponding one in [32] extended to nonlinear graph problems.

*Algorithm 3.4.* (1) Given  $q^k, \mu^k$

(2) Solve for  $p^k, u^k$ :

$$(3.20) \quad a_i(u_i^k, \hat{u}_i) = (f_i, \hat{u}_i) + \sum_{j \in \mathcal{J}_i} \left( p_i^k(x_{ij}) - \mu_{ij}^k \right) \hat{u}_i(x_{ij}), \quad \forall \hat{u}_i \in V_i$$

$$(3.21) \quad a_i^*(p_i^k, \hat{p}_i) = - \sum_{j \in \mathcal{J}_i} \left( q_{ij}^k - \frac{1}{d_j} \sum_{l \in \mathcal{I}_i} q_{lj}^k \right) \hat{p}_i(x_{ij}), \quad \forall \hat{p}_i \in V_i,$$

(3) Update  $q^{k+1}, \mu^{k+1}$ :

$$(3.22) \quad q_{ij}^{k+1} = q_{ij}^k + \rho \left\{ u_i^k(x_{ij}) - \frac{1}{d_j} \sum_{l \in \mathcal{I}_j} u_l^k(x_{lj}) \right\}$$

$$\mu_{ij}^{k+1} = \mu_{ij}^k + \rho \left( \sum_{l \in \mathcal{I}_j} p_l(x_{lj}) - \mu_{ij}^k \right), \quad j \in \mathcal{J}^M, i \in \mathcal{I}_j$$

*Remark 3.5.* (1) Notice again that the optimality system is forwardly coupled in the linear case, but two-way coupled in the general case.

(2) Algorithm 3.4 provides a decoupling of the optimality system. However, this is not of interest here, as in the limit, the adjoint problem decouples from the forward problem, due to the fact that  $p_i^k(x_{ij}) - \mu_{ij}^k$  tends to zero. Moreover, also the adjoint variable tends to zero. Due to space limitations, we defer the details to a forthcoming publication.

(3) The algorithm can be used also for optimal control problems, as the *actual* control and the corresponding load in the adjoint system can be read in the optimality system right away. For the sake of brevity, we, therefore, provide an example of this algorithm in the context of optimal control, only.

**3.1.2. The method by P.L. Lions for the  $p$ -Laplacian on a graph.** We now extend the concept of the last sub-section by introducing yet another virtual control. The idea is to relax the continuity conditions at multiple joints ( $n_j$ ) by the constraint  $u_i(x_{ij}) = \eta_j, \forall i \in \mathcal{I}_j$ , where  $\eta_j$  is taken as a virtual control. In contrast to the previous methods, we now use an augmented Lagrangian ansatz for this constraint. In order to describe the method in a more compact

format, we follow a Ritz-type approach. More precisely, we introduce the following augmented Lagrangian

$$(3.23) \quad \begin{aligned} \mathcal{L}(u, \eta; q) := & \sum_{i \in \mathcal{I}} \int_0^{\ell_i} \frac{1}{p} (|\partial_x u_i|^p - f_i u_i) dx \\ & + \sum_{j \in \mathcal{J}^M} \sum_{i \in \mathcal{I}_j} q_{ij} (u_i(x_{ij}) - \eta_j) + \frac{\rho}{2} \sum_{j \in \mathcal{J}^M} \sum_{i \in \mathcal{I}_j} (u_i(x_{ij}) - \eta_j)^2. \end{aligned}$$

We now use a variant of the Uzawa-algorithm for the saddle point problem

$$(3.24) \quad \sup_q \inf_{u, \eta} \mathcal{L}(u, \eta; q).$$

Indeed, we take the fractional step Algorithm 3 of R. Glowinski and P. LeTallec [16].

*Algorithm 3.6.* (1) Given  $\eta^{k-1}, q^k$   
(2) solve for  $u^k$

$$\partial_{u_i} \mathcal{L}(u, \eta; q) = 0.$$

(3) Update  $q^k$  using a fractional step

$$q_{ij}^{k+\frac{1}{2}} = q_{ij}^k + \rho(u_i^k(x_{ij}) - \eta_j^{k-1}).$$

(4) Solve for  $\eta_j^k$

$$\partial_{\eta_j} \mathcal{L}(u^k, \eta^k; q^{k+\frac{1}{2}}) = 0.$$

(5) Update  $q^{k+\frac{1}{2}}$  to the full step

$$q_{ij}^{k+1} = q_{ij}^{k+\frac{1}{2}} + \rho(u_i^k(x_{ij}) - \eta_j^k)$$

It turns out, with a little extra calculation, that this algorithms leads to the following iterative domain decomposition that, in turn, we display in terms of the strong formulation, for better understanding.

$$(3.25) \quad \begin{aligned} -\partial_x \left( |\partial_x u_i^{k+1}|^{p-2} \partial_x u_i^{k+1} \right) &= f_i, \quad \text{in } (0, \ell_i), i \in \mathcal{I} \\ d_{ij} \partial_x |u_i^{k+1}(x_{ij}) + \rho u_i^{k+1}(x_{ij}) &= \rho \left( \frac{2}{d_j} \sum_{l \in \mathcal{I}_j} u_l^k(x_{lj}) - u_i^k(x_{ij}) \right) \\ &\quad - \left( \frac{2}{d_j} \sum_{l \in \mathcal{I}_j} |\partial_x u_l^k(x_{lj})|^{p-2} \partial_x u_l^k(x_{lj}) - |\partial_x u_i^k(x_{ij})|^{p-2} \partial_x u_i^k(x_{ij}) \right), \quad j \in \mathcal{J}^M, i \in \mathcal{I}_j. \end{aligned}$$

For  $p = 2$  and two sub-domains, this is the classical Robin-Robin-type non-overlapping domain decomposition method introduced by P.L. Lions in [35] which is now extended to the p-Laplace problem and metric graphs. A proof of convergence for the decomposition of serial nodes is provided in [26]. Due to space limitations, the general case will be published in a forthcoming article.

*Example 3.7.* We consider now a serial problem, where the original domain stretches from 0 to 2 with constant load  $f \equiv 1$ , Dirichlet condition at  $x = 0$  and Neuman condition at  $x = 2$ . We take  $x = 1$  as interface with two edges and constant load  $f_i \equiv 1$ . We apply a 1.e – 5 regularization for the p-Laplace operator for  $p = 3/2$  and  $p = 1.1$ . See figure 5



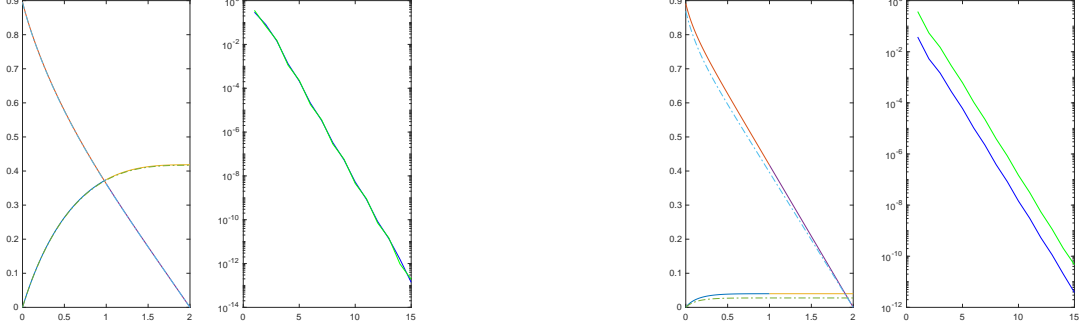


FIGURE 5. State and derivative and errors at multiple node: left figure:  $p = 3/2$ , right figure:  $p = 1.1$

We comment on the domain decomposition of the time-dependent problem. We notice that there is no publication available treating this case with non-overlapping domain decomposition. We take the idea from the Robin-Robin-type approach (3.25). More precisely, we recall system (1.8) and apply the analogous iterative condition as in (3.25).

$$\begin{aligned}
 & \partial_t |y_i^{k+1}(x, t)|^{p-2} y_i^{k+1}(x, t) - \partial_x \left( |\partial_x y_i^{k+1}(x, t)|^{p-2} \partial_x y_i^{k+1}(x, t) \right) = u_i^{k+1}, \quad i \in \mathcal{I}, \quad x \in (0, \ell_i), \quad t \in (0, T) \\
 & y_i^{k+1}(n_j, t) = 0, \quad i \in \mathcal{I}_j, \quad j \in \mathcal{J}_D^S, \quad t \in (0, T) \\
 (3.26) \quad & d_{ij} \beta(\partial_x y_i^{k+1}(n_j, t)) = 0, \quad i \in \mathcal{I}_j, \quad j \in \mathcal{J}_N^S, \quad t \in (0, T) \\
 & d_{ij} \partial_x |y_i^{k+1}|^{p-2}(x_{ij}, t) y_i^{k+1}(x_{ij}, t) + \rho y_i^{k+1}(x_{ij}, t) = \rho \left( \frac{2}{d_j} \sum_{l \in \mathcal{I}_j} y_l^k(x_{lj}, t) - y_i^k(x_{ij}, t) \right) \\
 & \quad - \left( \frac{2}{d_j} \sum_{l \in \mathcal{I}_j} |\partial_x y_l^k(x_{lj}, t)|^{p-2} \partial_x y_l^k(x_{lj}, t) - |\partial_x y_i^k(x_{ij}, t)|^{p-2} \partial_x y_i^k(x_{ij}, t) \right), \quad j \in \mathcal{J}^M, \quad i \in \mathcal{I}_j. \\
 & y_i^{k+1}(x, 0) = y_i(x); \quad x \in (0, \ell_i), \quad i \in \mathcal{I}.
 \end{aligned}$$

As in [22], it is possible to apply a relaxation with a relaxation parameter  $\epsilon \in [0, 1)$ . This amounts to

$$\begin{aligned}
& d_{ij} \partial_x |y_i^{k+1}|^{p-2}(x_{ij}, t) y_i^{k+1}(x_{ij}, t) + \rho y_i^{k+1}(x_{ij}, t) = \lambda_{ij}^k(t), \\
& \lambda_{ij}^k = (1 - \epsilon) \left\{ \rho \left( \frac{2}{d_j} \sum_{l \in I_j} y_l^k(x_{lj}, t) - y_i^k(x_{ij}, t) \right) \right. \\
& \quad \left. - \left( \frac{2}{d_j} \sum_{l \in I_j} |\partial_x y_l^k(x_{lj}, t)|^{p-2} \partial_x y_l^k(x_{lj}, t) - |\partial_x y_i^k(x_{ij}, t)|^{p-2} \partial_x y_i^k(x_{ij}, t) \right) \right\} \\
(3.27) \quad & \in \{d_{ij} \partial_x |y_i^k|^{p-2}(x_{ij}, t) y_i^k(x_{ij}, t) + \rho y_i^k(x_{ij}, t)\}.
\end{aligned}$$

We do not have sufficient space to provide the prove of convergence for this algorithm 3.6. The proof will be published in a forthcoming publication. We also refrain from showing convergence results for this method. The method can be obtain from the corresponding DDM for static optimal control problems by discarding the control functions. The performance is very similar as in example 4.8

*Example 3.8.* We consider a two-link problem. The first interval is  $(-1, 0)$  and the second  $(0, 1)$ . We take  $\sin(\pi x)^2$  as initial data on each link. We have implemented an FEM-code for the spatial part along with an implicit Euler time stepping, but, for the interest of the reader, we also used the freely available Matlab code **pdepe** which can easily be adopted to handle the p-Laplace parabolic problem with Robin-Robin-type boundary conditions. See fig. 6 for the result. We refrain from showing the solutions, but rather show the interface along the time and the corresponding errors in the state and the fluxes there. We used the pdepe code to also compute the problem on the entire domain  $(-1, 1)$  and plotted the solution using ' - '. Of course, due to the precision, this is not visible in the figure. For comparison, we display the results for the case  $p = 2$ , the linear case, and  $p = 3/2$ .

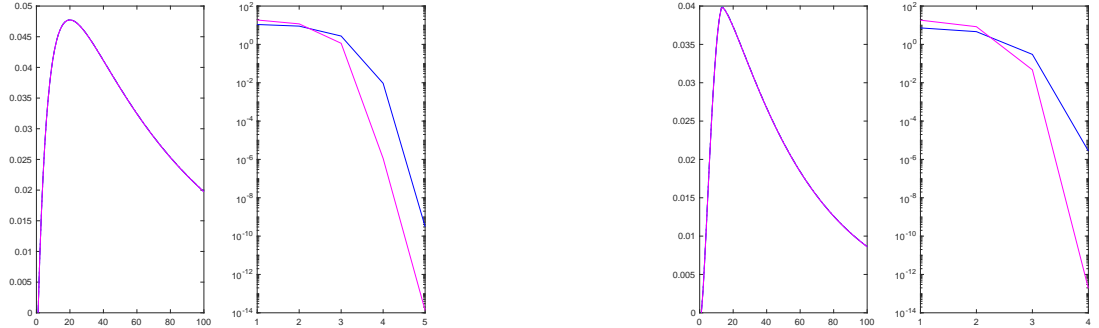


FIGURE 6. Left: DDM results for  $p=2$ , interface/errors; Right: DMM Results for  $p=3/2$ , interface/errors

#### 4. DOMAIN DECOMPOSITION FOR THE STATIC OPTIMAL CONTROL PROBLEM

We now embark on non-overlapping domain decomposition method for static optimal control problems. According to the previous section, we consider two approaches, namely, the one by J.L. Lions and O. Pironneau [32]

extended to non-linear problems on metric graphs in subsection 3.1.1 and the one by P.L. Lion [35] developed in subsection 3.1.2. We have seen that the domain decomposition methods all by themselves can be framed in the context of optimal control, in fact *virtual* problems. They are, therefore, likely to be well suited for *real* or *actual* control problems. In this section, we pursue this approach. Indeed, the *actual* cost-functions and the corresponding *actual* controls together with their control constraints interfere via the penalty parameters, only.

We go back to the optimality condition (2.12). The goal is to decompose this system at given multiple nodes  $n_j$ . Accordingly, we change Algorithm 3.4 as follows. As it is more convenient in the context of optimal control to denote the control variables by  $u$ , we denote the states by  $y$  and keep the notation for the adjoint variable  $p$ .

*Algorithm 4.1.* (1) Given  $q^k, \mu^k$

(2) Solve for  $p^k, p^k$ :

$$(4.1) \quad a_i(y_i^k, \hat{y}_i) = -\frac{1}{\nu}(p_i^k, \hat{y}_i)(f_i, \hat{y}_i) + \sum_{j \in \mathcal{I}_i} (p_i^k(x_{ij}) - \mu_{ij}^k) \hat{u}_i(x_{ij}), \quad \forall \hat{u}_i \in V_i$$

$$(4.2) \quad a_i^*(p_i^k, \hat{p}_i) = -\kappa(y_i^k - y_i^d, \hat{p}_i) - \sum_{j \in \mathcal{I}_i} \left( q_{ij}^k - \frac{1}{d_j} \sum_{l \in \mathcal{I}_j} q_{lj}^k \right) \hat{p}_i(x_{ij}), \quad \forall \hat{p}_i \in V_i,$$

(3) Update  $q^{k+1}, \mu^{k+1}$ :

$$(4.3) \quad q_{ij}^{k+1} = q_{ij}^k + \rho \left\{ u_i^k(x_{ij}) - \frac{1}{d_j} \sum_{l \in \mathcal{I}_j} u_l^k(x_{lj}) \right\}$$

$$\mu_{ij}^{k+1} = \mu_{ij}^k + \rho \left( \sum_{l \in \mathcal{I}_j} p_l(x_{lj}) - \mu_{ij}^k \right), \quad j \in \mathcal{J}^M, i \in \mathcal{I}_j.$$

*Remark 4.2.* (1) Algorithm 4.1 does not seem to have been considered in the literature before. To the best knowledge of the author, there is no convergence proof.

(2) Because of space limitations, we do not write down the optimality system in the strong form.

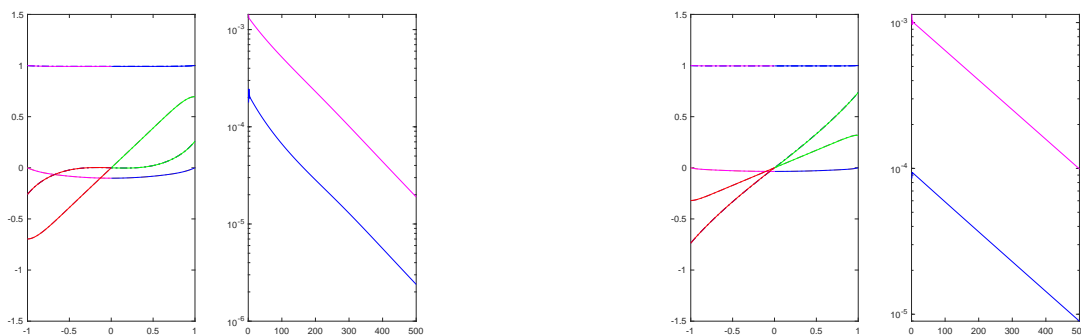


FIGURE 7. State and derivative and errors at multiple node: left figure:  $p = 3/2$ , right figure:  $p = 1.1$

*Example 4.3.*

The extension of Algorithm 3.1 to optimal control problems does not seem to be as straightforward, at least at the first sight. However, at least for linear distributed controls, J. D. Benamou [6] showed that the optimality system can be recast into a complex Helmholtz problem for which, in turn, the P.L.Lions algorithm could be applied. This observation has been the basis for the development of the non-overlapping domain decompositions in space and time in the monograph by J. E. Lagnese and G. Leugering [22].

To fix ideas and for the sake of simplicity, and brevity, we take a serial network consisting of two links. The generalization to arbitrary networks is then a matter of straightforward arguments in the spirit of the previous sections.

*Remark 4.4.* Another reason for the choice of a serial transmission problem is that it directly generalizes to time-domain decomposition for optimality systems, as we will briefly discuss below. The point is that, as the state evolves forwardly in time, the adjoint progresses backwardly. Therefore, the initial data for the state and the final data for the adjoint variable serve as boundary conditions for the time variable. Indeed, in the case of distributed controls, the optimality system can be reduced, either by eliminating the state or the adjoint state, resulting in a space-time 'elliptic' problem which, in turn, can be solved using the DDMs discussed in this article. See [24].

Moreover, also for the sake of simplicity, we discuss in these notes distributed controls only. Boundary controls can easily be introduced and will be covered in a forthcoming publication.

We introduce the two consecutive intervals  $I_1 := (0, 1)$ ,  $I_2 := (1, 2)$  and look into the optimal control problem

$$(4.4) \quad \min_{(u,y)} \frac{\kappa}{r} \sum_{i=1}^2 \int_0^{\ell_i} |y_i - y_i^d|^r dx + \frac{\nu}{2} \sum_{i=1}^2 \int_0^{\ell_i} |u_i|^2 dx$$

s.t.

$$\beta_i(y_i) - \partial_x(\beta_i(\partial_x y_i)) = u_i, i = 1, 2, x \in I_i$$

$$y_1(0) = 0, y_2(2) = 0,$$

$$y_1(1) = y_2(1), \beta_1(\partial_x y_1)(1) = \beta_2(\partial_x y_2)(1).$$

We easily derive the corresponding optimality system.

$$(4.5) \quad \beta_i(y_i) - \partial_x(\beta_i(\partial_x y_i)) = \frac{1}{\nu} p_i, i = 1, 2, x \in I_i$$

$$\beta'_i(y_i) p_i - \partial_x(\beta'_i(\partial_x y_i) \partial_x p_i) = -\kappa(|y_i - y_i^d|^{r-2} (y_i - y_i^d)), i = 1, 2, x \in I_i$$

$$y_1(0) = 0, y_2(2) = 0,$$

$$p_1(0) = 0, p_2(2) = 0,$$

$$y_1(1) = y_2(1), \beta_1(\partial_x y_1)(1) = \beta_2(\partial_x y_2)(1),$$

$$p_1(1) = p_2(1), \beta'_1(\partial_x y_1)(1) \partial_x p_1(1) = \beta'_2(\partial_x y_2(1)) \partial_x p_2(1).$$

We propose the following iterative domain decomposition method.

$$(4.6) \quad \beta_i(y_i^{n+1}) - \partial_x(\beta_i(\partial_x y_i^{n+1})) = \frac{1}{\nu} p_i^{n+1}, i = 1, 2, x \in I_i$$

$$\beta'_i(y_i^{n+1}) p_i^{n+1} - \partial_x(\beta'_i(\partial_x y_i^{n+1}) \partial_x p_i^{n+1}) = -\kappa(|y_i^{n+1} - y_i^d|^{r-2} (y_i^{n+1} - y_i^d)), i = 1, 2, x \in I_i$$

$$y_1^{n+1}(0) = 0, y_2^{n+1}(2) = 0,$$

$$p_1^{n+1}(0) = 0, p_2^{n+1}(2) = 0,$$

$$(4.7) \quad d_{ij}\beta_i(\partial_x y_i^{n+1}(1)) + \sigma y_i^{n+1}(1) + \mu p_i^{n+1}(1) = -d_{ij}\beta_j(\partial_x y_j^n(1)) + \sigma y_j^n(1) + \mu p_j^n(1) := \lambda_i^n$$

$$(4.8) \quad d_{ij}\beta_i'(\partial_x y_i^{n+1}(1))\partial_x p_i^{n+1}(1) + \sigma p_i^{n+1}(1) - \mu y_i^{n+1}(1) = -d_{ij}\beta_j'(\partial_x y_j^n(1))\partial_x p_j^n(1) + \sigma p_j^n(1) - \mu y_j^n(1) =: \rho_i^n$$

with parameters  $\sigma, \mu \geq 0$ . The iteration (4.6), (4.7), (4.8) is started at  $n = 0$ .

We notice that the decomposed optimality system (4.6), (4.7), (4.8) can be seen as the optimality system for the virtual control problem on each edge, with solution  $y, p$  updated at the iteration index  $n + 1$ . We write down the general virtual optimal control problem:

$$(4.9) \quad \min_{u, g, y} \left\{ J_i(u_i, y_i) + \frac{1}{2\mu} \sum_{j \in \mathcal{J}_i} [ |g_j|^2 + |\mu y_i - \rho_{ij}^n|^2 ] \right\}$$

s. t.

$$\beta_i(y_i) - \partial_x(\beta_i(\partial_x y_i)) = u_i, i = 1, 2, x \in I_i$$

$$d_{ij}\beta_i(\partial_x y_i(x_{ij})) + \sigma y_i(x_{ij}) = \lambda_{ij}^n + g_{ij}, j \in \mathcal{J}_i, i \in \mathcal{I}_j, y_i(x_{ij}) = 0, i \in \mathcal{I}_j, j \in \mathcal{J}_D^S.$$

In the case above,  $i = 1, 2$  and only one interface at  $x = 1$  is relevant while at the ends a Dirichlet condition holds. The proof of this very important feature is left to the reader. It is analogous to the linear result in [22].

*Example 4.5.* We consider the common distributed optimal control problem on the interval  $[-1, 1]$  with Dirichlet conditions at both ends. We take the p-Laplace problem with the target  $y \equiv 1$  which, in turn, we enforce with  $\kappa = 100$ , the unconstrained distributed control is penalized by  $\nu = .1$ . We then solve the virtual optimal control problem (4.9) on the segment  $[-1, 0]$  and  $[0, 1]$  with  $\sigma = 1, \mu = 5$ . For the solver, we use IPOPT as in example 2.1. See the figure 8 where on the left side we plotted the state and on the right the errors at the interface at  $x = 0$ . On the left we also plotted (with '·') the solution obtain by IPOPT on the entire interval  $[-1, 1]$  on top of the individual solutions on the sub-domains. We remark that in this special case of a distributed control, we may use the optimality condition  $p = \nu u$  to calculate the updates for  $\lambda^n, \rho^n$ , respectively. In case of boundary controls or localized distributed controls, one has to solve the adjoint equation in order to perform the necessary updates. Nevertheless, the optimality system is fully decoupled and restored in course of the iterations.

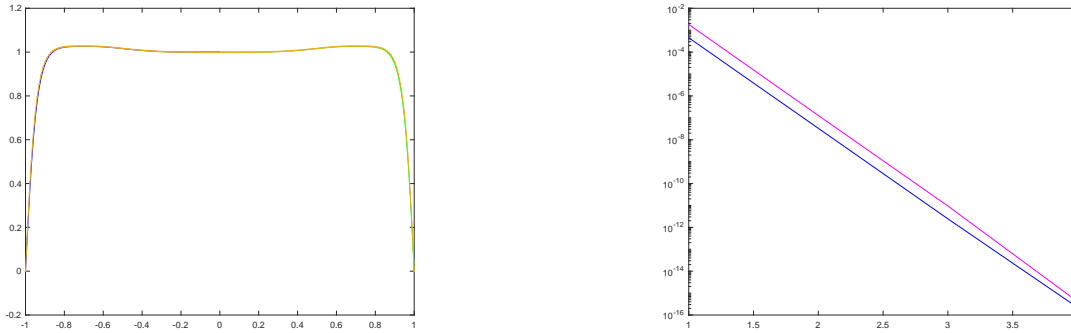


FIGURE 8. State and errors at multiple node for  $p = 3/2$ ;

Objective :	$1.79896988521e + 02$	Dual infeasibility :	$3.365926648513e - 09$
Constraint violation :	$1.129872028846e - 14$	Overall NLP error :	$1.076947125968e - 09$

**Theorem 4.6.** *Let the  $(y_i, p_i) \in H^1(I_i)$  and initial errors  $\mathcal{X}^0$  be sufficiently small. Then there are parameters  $\sigma, \mu, \kappa, \nu$  such that the iteration  $\mathcal{X}^{n+1} = \mathcal{T}\mathcal{X}^n$  converges. Moreover, we have*

$$(4.10) \quad \begin{aligned} \tilde{y}_i^n, \tilde{p}_i^n &\rightarrow 0, \quad n \rightarrow \infty, \quad \text{in } H^1(I_i), \quad i = 1, 2, \\ \tilde{y}_i^n(1), \partial_x \tilde{y}_i^n(1), \tilde{p}_i^n(1), \partial_x \tilde{p}_i^n(1) &\rightarrow 0, \quad n \rightarrow \infty, \quad i = 1, 2. \end{aligned}$$

*Remark 4.7.* We remark that when we take  $p = 2$ , then the iterative domain decomposition procedure above reduces to the well-known iteration from [22].

*Example 4.8.* In this example, we consider the static optimal control problem for a two-link model and the corresponding optimality system. The iterative decomposition is now based on the Robin-Robin DDM. We choose in particular a constant target function  $y_d = 1$ , constant Dirichlet conditions  $y(x, t) = 1$  at the outer ends  $x = 0, x = 2$  and parameters  $\sigma = 10, \mu = .001, \kappa = 100, \nu = .1$ . We perform two experiments, one for  $p = 3/2$ , the case of interest in gas flow, and  $p = 1.1$ , which is important in e.g. imaging, where  $1 < p \leq 2$  and the limiting case  $p = 1$  are considered. We solve the boundary value problems arising on the full domain and the two subdomains using the routine **bvpc4** from Matlab with a resolution of  $1e - 14$ . We have chosen a constant target because it is critical, as vanishing  $y$  and  $\partial_x y$  play a special role for the p-Laplacian. For  $p > 2$ , we, therefore, needed a regularization. On the left of figure 9, where  $p = 3/2$ , we show on the top left the solution of the optimality system, on the top right the result of the iteration on each subdomain stitched together at  $x = 1$ . On the bottom left we display the errors for the primal variables and their derivatives at the interface  $x = 1$  and on the right the corresponding dual variables. The same configuration is used for  $p = 1.1$  in the right part of figure 9. For more experiments and a more detailed description, we refer to a forthcoming publication.

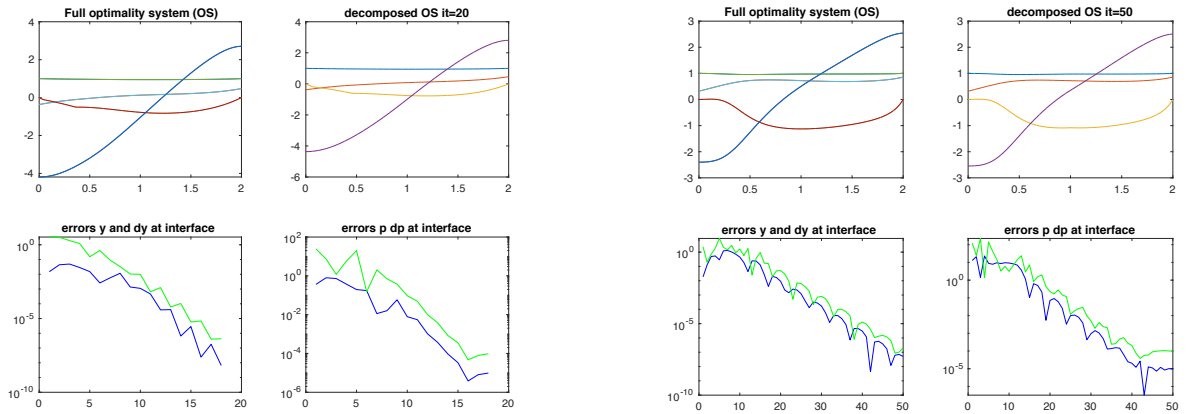


FIGURE 9. Left figure: two-domain matching, right figure: errors of states and fluxes

## 5. TIME-DOMAIN DECOMPOSITION

We now provide some remarks on time-domain decomposition of optimal control problems by decomposing the corresponding optimality systems. Due to space limitations, we do so for the semi-linear problem (1.7) and leave the case of friction dominated flow to a forthcoming publication. We can now directly continue the discussion of the last section, as we pointed out in remark 3.5. The procedure pursued in this article is the same as in [21] and is,

in turn, very much inspired by [23], where the linear wave equation and the Maxwell equations are considered. We introduce a coarse time discretization with

$$0 = T_0 < T_1 < \dots < T_k < T_{k+1} < \dots < T_K < T_{K+1} = T.$$

Instead of using this grid for a direct numerical solve, we introduce the intervals  $I_k := (T_k, T_{k+1})$ . At the time-interfaces  $T_k, T_{k+1}$ , we employ continuity conditions  $(p_{ki}, q_{ki})^*(T_k) = (p_{k-1,i}, q_{k-1,i})^*(T_k)$  for  $i \in \mathcal{I}$   $k = 1, \dots, K+1$ , and similarly for the adjoint variables. Below, we omit the edge index  $i$ . The iterative time domain decomposition procedure is then described as follows.

$$(5.1) \quad \begin{pmatrix} p \\ q \end{pmatrix}_k^{n+1} (T^{k+1}) + \sigma \begin{pmatrix} \phi \\ \psi \end{pmatrix}_k^{n+1} (T_{k+1}) = \Phi_{k,k+1}^n, \quad \begin{pmatrix} p \\ q \end{pmatrix}_k^{n+1} (T^k) - \sigma \begin{pmatrix} \phi \\ \psi \end{pmatrix}_k^{n+1} (T_k) = \Phi_{k,k-1}^k,$$

with

$$(5.2) \quad \Phi_{k,k+1}^n = \begin{pmatrix} p \\ q \end{pmatrix}_{k+1}^n (T^{k+1}) + \sigma \begin{pmatrix} \phi \\ \psi \end{pmatrix}_{k+1}^n (T_{k+1}), \quad \Phi_{k,k-1}^n = \begin{pmatrix} p \\ q \end{pmatrix}_{k-1}^n (T^k) - \sigma \begin{pmatrix} \phi \\ \psi \end{pmatrix}_{k-1}^n (T_k).$$

We note that again a relaxation of the iteration history as in (3.27) is possible and in fact advisable. We, therefore, introduce

$$(5.3) \quad \begin{aligned} \Phi_{k,k+1}^n &= (1 - \varepsilon) \left( \begin{pmatrix} p \\ q \end{pmatrix}_{k+1}^n (T_{k+1}) + \sigma \begin{pmatrix} \phi \\ \psi \end{pmatrix}_{k+1}^n (T_{k+1}) \right) + \varepsilon \left( \begin{pmatrix} p \\ q \end{pmatrix}_{k+1}^n (T_{k+1}) + \sigma \begin{pmatrix} \phi \\ \psi \end{pmatrix}_{k+1}^n (T_{k+1}) \right), \quad k = 0, \dots, K-1, \\ \Phi_{k,k-1}^n &= (1 - \varepsilon) \left( \begin{pmatrix} p \\ q \end{pmatrix}_{k-1}^n (T_k) - \sigma \begin{pmatrix} \phi \\ \psi \end{pmatrix}_{k-1}^n (T_k) \right) + \varepsilon \left( \begin{pmatrix} p \\ q \end{pmatrix}_k^n (T_k) - \sigma \begin{pmatrix} \phi \\ \psi \end{pmatrix}_k^n (T_k) \right), \quad k = 1, \dots, K, \end{aligned}$$

We leave this to the reader, as we do with the proof of convergence, for which refer to [21]. It is also obvious that this TDDM leads to a *virtual control* problem, just as in (4.9). See [21] for details.

*Example 5.1.* We consider 2.7 for a single link, i.e.,  $d = 2$ . We do not consider boundary controls but assume, in fact, homogeneous Dirichlet boundary conditions. We apply a distributed control without constraints and take two sub-intervals  $I_1 = [0, 1)$  and  $I_2 = (1, 2]$ . For the space discretization, we use the standard approximations corresponding to the standard discrete Dirichlet-Laplacian  $A_h$  on the second-order level in space. The corresponding global optimality system on the entire interval  $[0, 2]$  and the local optimality systems on each time interval  $I_1$  and  $I_2$  are treated as a boundary value problems wrt the time variable. Then, we solve these problems using the MATLAB solver `bvp4c` with tolerance  $10^{-8}$ . For any initial data and tracking term, the system governing the errors is homogeneous and, hence, the local optimality systems are homogeneous up to the errors at the transmission boundary  $T = 1$ . In particular, for vanishing initial data and target, the global optimality system has zero as the unique solution. We take  $n = 10$  discretization points wrt the space variable and choose  $\sigma = 10^5$ ,  $\kappa = 10^3$ ,  $\nu = 10^3$ , and  $\varepsilon \in \{0.95, 0.5, 0.005\}$ . See fig.10 for the results, for the three choices of  $\varepsilon$ , left, middle and right, respectively.

## 6. REMARKS ON PINN-BASED DEEP DDM: D<sup>3</sup>M

The methods described in this survey article are based on the paradigm of virtual controls. The corresponding DDMs can, therefore, be seen in the context of PDE-constraint optimization. Physics informed neural networks (PINN) use neural network technology in order to approximate PDEs and the corresponding initial and boundary conditions in the sense of least squares. See e.g. [3]. More recently, coupled problems have been addressed by what is now known as the XPINN-approach, X standing for crossings or interfaces, see e.g. [19]. Clearly, once the model is penalized using least squares, also transmission conditions can be handled using penalization. Even more, it is possible to include optimization variables and cost functions in the context of PDE-constraint optimal control

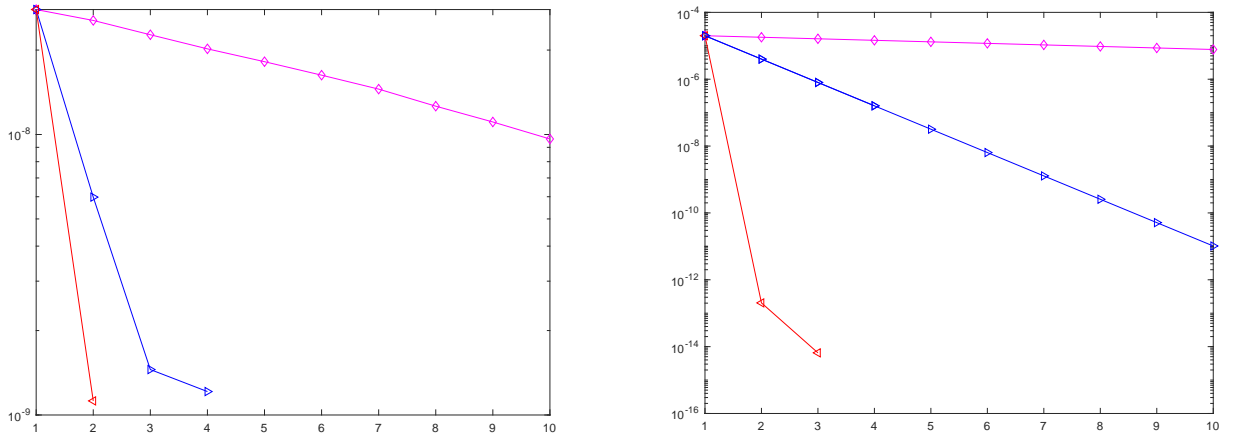


FIGURE 10. Left figure: errors of the state, right figure: errors of fluxes

by this method. See e.g. [38]. It does not come as a surprise that domain decomposition methods have also been investigated in this context. Indeed, a deep-Schwarz overlapping domain decomposition method has been introduced in [30]. There is no space to elaborate at length on this promising emerging subject. In this article, we apply the P.L. Lions Robin-Robin-type approach to a two-link problem. The only difference to the exact Robin-Robin-approach (4.8), (4.7) is that we substitute the model-based numerical solver for the sub-domains (or one of the sub-domains) by a PINN-based solver. This approach has, to the best knowledge of the author, not been published elsewhere. The interest in this non-overlapping PINN-based approach is that in the applications, in particular in the network problems we discuss here, the PINN-approach, as a surrogate, may be used in part of the complex network (say, daughter networks), while classical numerical methods are used in a parent network. This novel paradigm that we can call NETI (instead of FETI) as NETwork Tearing and Integration is the subject of further research in the CRC 154 *Mathematical modelling, simulation and optimization using the example of gas networks*, project A05, (see <https://www.trr154.fau.de/trr-154-en/>).

*Example 6.1.* We consider the same situation as in example 3.3. As for the PINN set-up, we use the following specification of training parameters:  $N_e = 2000$ ; # of Epochs (1 Epoch contains  $T_b$  training batches),  $T_b = 600$ ; # of training batches (or corrections during 1 Epoch),  $l_r = 0.005$ ; Learning rate coefficient (relaxation for the update),  $N_n = 10$ ; Number of nodes in the 1st hidden layer,  $T_t = 1e-30$ ; Training tolerance N.B. See fig.11 and build on a code provided by Almqvist [1].

**6.1. Summary, comments and further research.** We have tried to provide a fairly complete account of iteration methods that can be attributed to the notion of virtual controls in the sense of J.L. Lions and O. Pironneau [32]. We have seen that also the algorithm by P.L. Lions [35] can be interpreted as the result of a virtual control problem. This feature suggests itself to be used in the context of optimal control with PDE-constraints which are subject to non-overlapping domain decomposition in space and time. Due to space limitations, we were not able to provide mathematical analysis, but rather focused on the algorithms and their implementation. In particular with respect to the fully doubly nonlinear p-parabolic problem, a detailed study of well-posedness of the underlying optimal control problems together with proofs of convergence on the continuous level will appear elsewhere. Domain decomposition an *turnpike phenomena* are subject to further studies as well as the algorithmic realization of *reachability* or exact controllability (or even nodal profile) constraints. Moreover, the methods treated allow for control constraints, also.



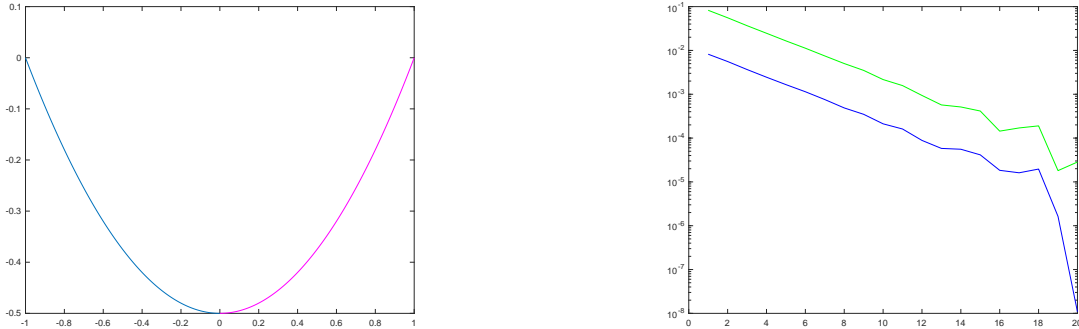


FIGURE 11. Left figure: two-domain matching, right figure: errors of states and fluxes

DDMs for state constraint optimal control problems appear to be open. Obviously, such problems are of importance in particular in gas-pipe networks, where e.g. the pressure is limited in each pipe by box-constraints. With respect to mechanical multi-structures, further research is necessary in order to extend the algorithms provided here to 2-d and 3-d networked systems, see [22] for 2-d networks. The fact that the DDMs can eventually be represented by virtual control problems on sub-domains make them interesting. This applies also to discrete-continuous control problems, see e.g. [18], [17]. We refer to <https://www.trr154.fau.de/trr-154-en/> for further information.

#### REFERENCES

- [1] P Almqvist. Fundamentals of physics-informed neural networks applied to solve the reynolds boundary value problem. *Lubricants*, 9(83), 2021.
- [2] R. Alonso, M. Santillana, and C. Dawson. On the diffusive wave approximation of the shallow water equations. *European J. Appl. Math.*, 19(5):575–606, 2008.
- [3] G. Bai, U. Koley, S. Mishra, and R. Molinaro. Physics informed neural networks (PINNs) for approximating nonlinear dispersive PDEs. *J. Comput. Math.*, 39(6):816–847, 2021.
- [4] A. Bamberger. étude d’une équation doublement non linéaire. *J. Functional Analysis*, 24(2):148–155, 1977.
- [5] A. Bamberger, M. Sorine, and J. P. Yvon. Analyse et contrôle d’un réseau de transport de gaz. In *Computing methods in applied sciences and engineering (Proc. Third Internat. Sympos., Versailles, 1977), II*, volume 91 of *Lecture Notes in Phys.*, pages 347–359. Springer, Berlin-New York, 1979.
- [6] J.-D. Benamou. A domain decomposition method with coupled transmission conditions for the optimal control of systems governed by elliptic partial differential equations. 33(6):2401–2416, 1996.
- [7] V. Bögelein, F. Duzaar, P. Marcellini, and C. Scheven. A variational approach to doubly nonlinear equations. *Atti Accad. Naz. Lincei Rend. Lincei Mat. Appl.*, 29(4):739–772, 2018.
- [8] J. Brouwer, I. Gasser, and M. Herty. Gas pipeline models revisited: Model hierarchies, nonisothermal models, and simulations of networks. 9(2):601–623.
- [9] E. Casas and L. A. Fernández. Optimal control of quasilinear elliptic equations. In *Control of partial differential equations (Santiago de Compostela, 1987)*, volume 114 of *Lect. Notes Control Inf. Sci.*, pages 92–99. Springer, Berlin, 1989.
- [10] E. Casas and L. A. Fernández. Optimal control of quasilinear elliptic equations with nondifferentiable coefficients at the origin. *Rev. Mat. Univ. Complut. Madrid*, 4(2-3):227–250, 1991.
- [11] E. Casas, P. I. Kogut, and G. Leugering. Approximation of optimal control problems in the coefficient for the  $p$ -Laplace equation. I. Convergence result. *SIAM J. Control Optim.*, 54(3):1406–1422, 2016.
- [12] E. Engström and E. Hansen. Convergence analysis of the nonoverlapping robin-robin method for nonlinear elliptic equations, 2021.
- [13] M. J. Gander. Analysis of the parareal algorithm applied to hyperbolic problems using characteristics. (42):21–35, 2008.

- [14] M. J. Gander and F. Kwok. Schwarz methods for the time-parallel solution of parabolic control problems. In *Domain decomposition methods in science and engineering XXII*, volume 104 of *Lecture Notes in Computational Science and Engineering*, pages 207–216. Springer, Cham, 2016.
- [15] M. J. Gander and S. Vandewalle. Analysis of the parareal time-parallel time-integration method. 29(2):556–578, 2007.
- [16] R. Glowinski and P. Le Tallec. Augmented Lagrangian interpretation of the nonoverlapping Schwarz alternating method. In *Third International Symposium on Domain Decomposition Methods for Partial Differential Equations (Houston, TX, 1989)*, pages 224–231. SIAM, Philadelphia, PA, 1990.
- [17] F. Hante, R. Krug, and M. Schmidt. Time-domain decomposition for mixed-integer optimal control problems. 2021.
- [18] F. Hante and M. Schmidt. Convergence of finite-dimensional approximations for mixed-integer optimization with differential equations. *Control Cybernet.*, 48(2):209–229, 2019.
- [19] A. D. Jagtap and G. E. Karniadakis. Extended physics-informed neural networks (XPINNs): a generalized space-time domain decomposition based deep learning framework for nonlinear partial differential equations. *Commun. Comput. Phys.*, 28(5):2002–2041, 2020.
- [20] R. Krug, G. Leugering, A. Martin, M. Schmidt, and D. Weninger. Time-domain decomposition for optimal control problems governed by semilinear hyperbolic systems with mixed two-point boundary conditions ( submitted). 2021.
- [21] R. Krug, G. Leugering, M. Martin, A. and. Schmidt, and D. Weninger. Time-domain decomposition for optimal control problems governed by semilinear hyperbolic systems. *SIAM J. Control Optim.*, 59(6):4339–4372, 2021.
- [22] J. E. Lagnese and G. Leugering. *Domain decomposition methods in optimal control of partial differential equations*, volume 148 of *International Series of Numerical Mathematics*. Birkhäuser Verlag.
- [23] John E Lagnese and Günter Leugering. Time-domain decomposition of optimal control problems for the wave equation. 48(3-4):229–242, 2003.
- [24] G. Leugering. Space-time-domain decompositions for optimal control problems governed by linear hyperbolic systems. *Journal of Optimization, Differential Equations and their Applications (JODEA)*, 29(2):24–47, 2021.
- [25] G. Leugering. Domain decomposition of optimal control problems for the doubly nonlinear p-parabolic equation on metric graphs. Technical report, FAU TRR154, 2022. in preparation.
- [26] G. Leugering. Nonoverlapping domain decomposition for instantaneous optimal control of friction dominated flow in a gas-network. Proceedings of the INDAM Conference 2020, 2022.
- [27] G. Leugering and G. Mophou. Instantaneous optimal control of friction dominated flow in a gas-network. In *Shape optimization, homogenization and optimal control*, volume 169 of *Internat. Ser. Numer. Math.*, pages 75–88. Birkhäuser/Springer, Cham, 2018.
- [28] R. J. Le Veque.
- [29] R. J. Le Veque. *Finite Volume Methods for Hyperbolic Problems*. Cambridge University Press.
- [30] K. Li, K.J. Tang, T.F. Wu, and Q.F. Liao. D3m: A deep domain decomposition method for partial differential equations. *IEEE ACCESS*, 8:5283–5294, 2020.
- [31] J.-L. Lions, Y. Maday, and G. Turinici. Résolution d’EDP par un schéma en temps “pararéel”. 332(7):661–668, 2001.
- [32] J.-L. Lions and O. Pironneau. Algorithmes parallèles pour la solution de problèmes aux limites. *C. R. Acad. Sci. Paris Sér. I Math.*, 327(11):947–952, 1998.
- [33] J.-L. Lions and O. Pironneau. Domain decomposition methods for CAD. *C. R. Acad. Sci. Paris Sér. I Math.*, 328(1):73–80, 1999.
- [34] J.-L. Lions and O. Pironneau. Virtual control, replicas and decomposition of operators. *C. R. Acad. Sci. Paris Sér. I Math.*, 330(1):47–54, 2000.
- [35] P.-L. Lions. On the Schwarz alternating method. III. A variant for nonoverlapping subdomains. In *Third International Symposium on Domain Decomposition Methods for Partial Differential Equations (Houston, TX, 1989)*, pages 202–223. SIAM, Philadelphia, PA, 1990.
- [36] A. Quarteroni and A. Valli. *Domain decomposition methods for partial differential equations*. Numerical Mathematics and Scientific Computation. The Clarendon Press, Oxford University Press, New York, 1999. Oxford Science Publications.
- [37] T. Roubíček. *Nonlinear partial differential equations with applications*, volume 153 of *International Series of Numerical Mathematics*. Birkhäuser/Springer Basel AG, Basel, second edition, 2013.
- [38] M. Saviz and S. Nabi. Optimal control of pdes using physics-informed neural networks. Technical report, arXiv:2111.09880, 2021.
- [39] L. Schöbel-Kröhn. *Analysis and Numerical Approximation of Nonlinear Evolution Equations on Network Structures*. PhD thesis, TU-Darmstadt, 2020.
- [40] J. Smoller. *Shock Waves and Reaction-Diffusion Equations*, volume 258 of *Grundlehren der mathematischen Wissenschaften*. Springer Verlag.
- [41] W. Wollner. An optimal control problem for equations with p-structure and its finite element discretization, 2021.

DEPARTMENT OF DATA SCIENCE, FRIEDRICH-ALEXANDER-UNIVERSITÄT ERLANGEN-NÜRNBERG (FAU) CAUERSTR. 11, D-91058 ERLANGEN, GERMANY,

Email address: guenter.leugering@fau.de