Nonoverlapping domain decomposition for optimal control problems on metric graphs

by the example of gas networks

Workshop on Control, Inversion and Numerics for PDEs
CIN-PDE 2023
Erlangen-Shanghai
August 1-4, 2023

Günter Leugering, FAU-Erlangen-Nürnberg

FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG

TRR
154

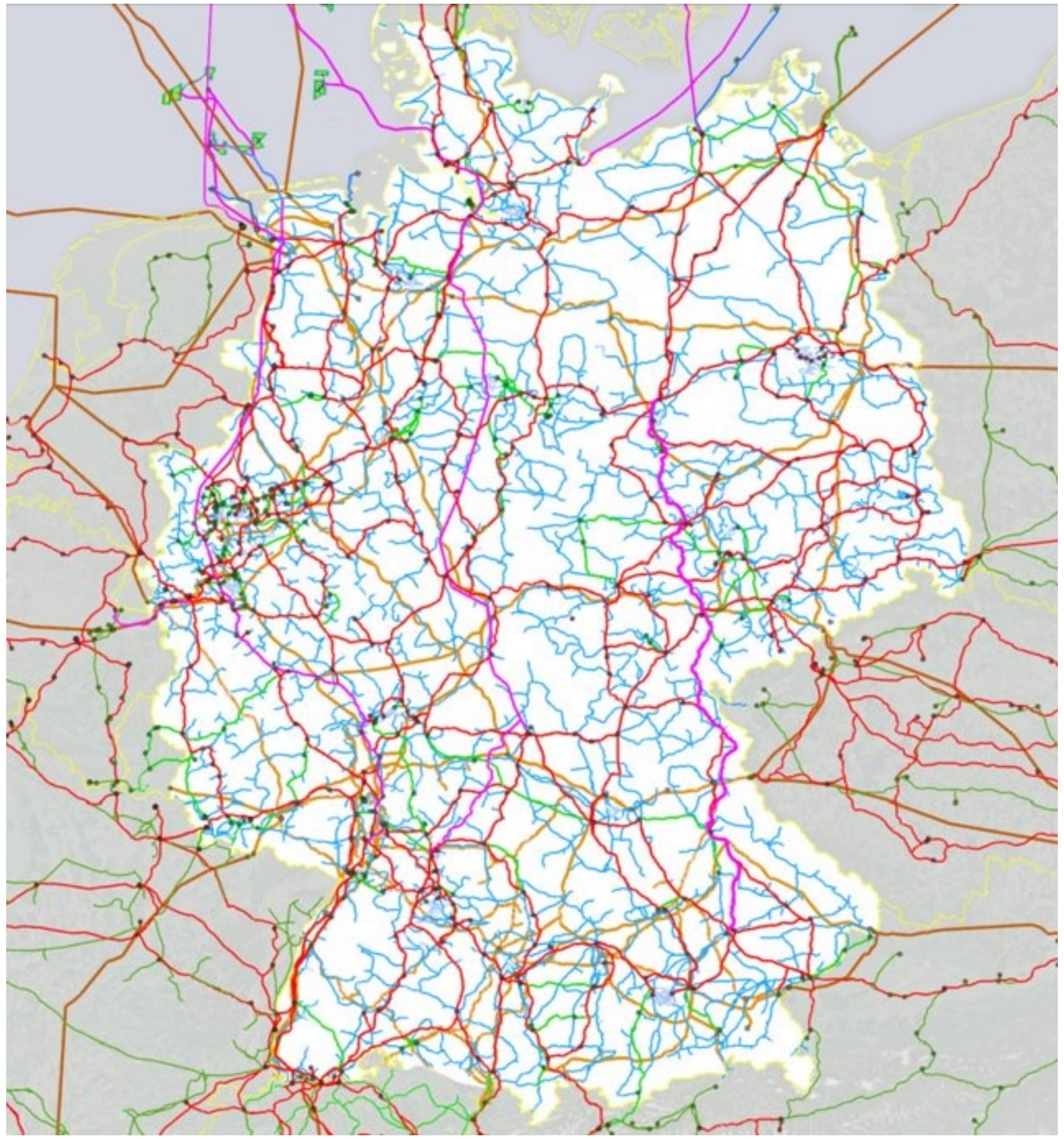
Domain decomposition of flow problems on metric graphs

Why?

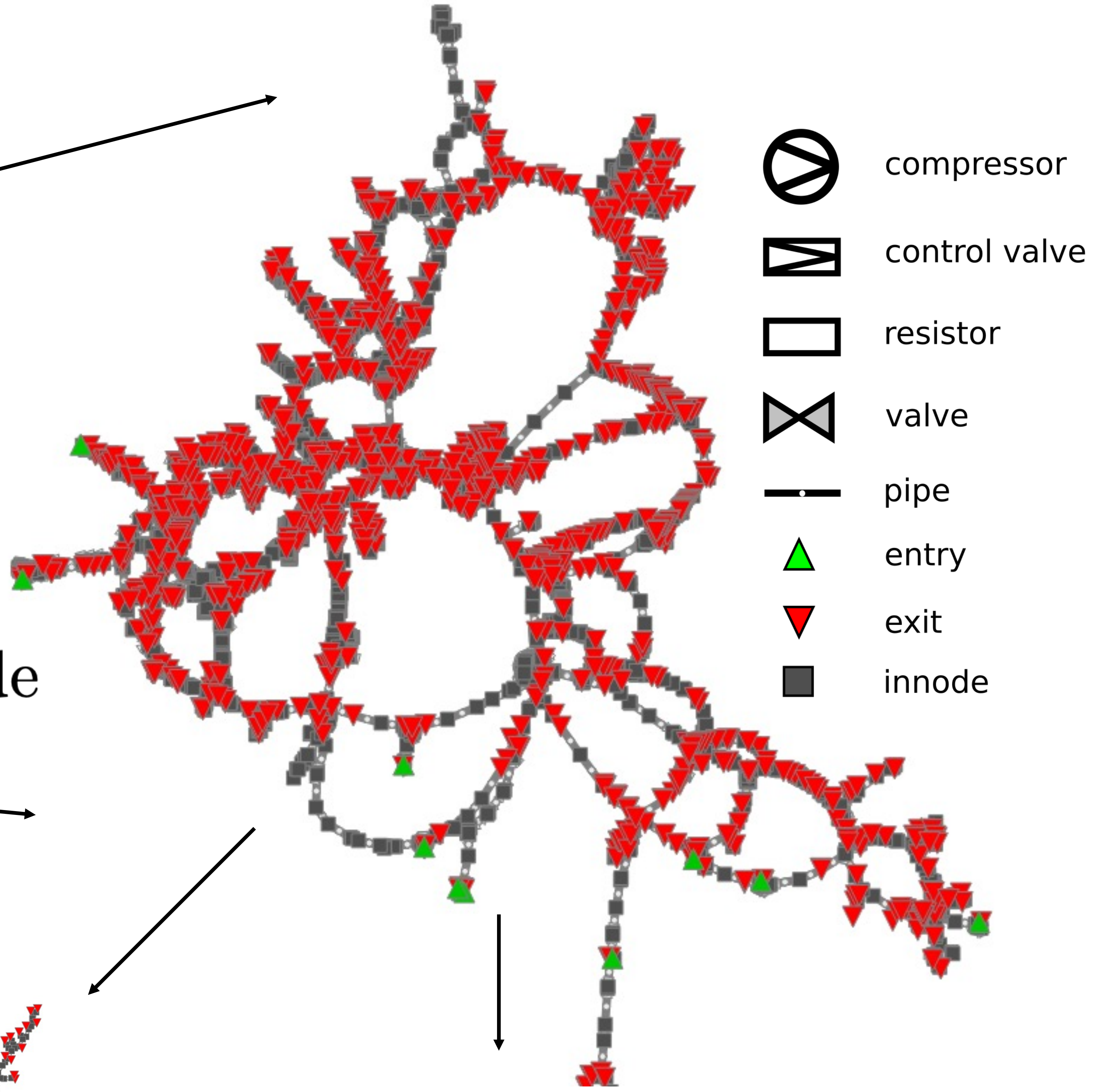
- Large scale networks may contain more than 20K major pipes and many nonlinear elements as compressors, valves etc. See e.g. the German gas network
- For each pipe, one needs space-time discretization for the nonlinear PDEs (e.g. Euler system, shallow water or water-hammer system) and discrete as well as continuous control variables leading to large-scale optimality systems
- In order to incorporate randomness (of the system data), we need to solve optimality systems repeatedly Moreover, in the control of gas networks one faces realtime constraints
- Real-time capable optimal control on large scale flow networks is beyond the current scope of numerical reallizations
- Hence, decomposition is at order at almost every turn (i.e. the optimization level, the network and the time).

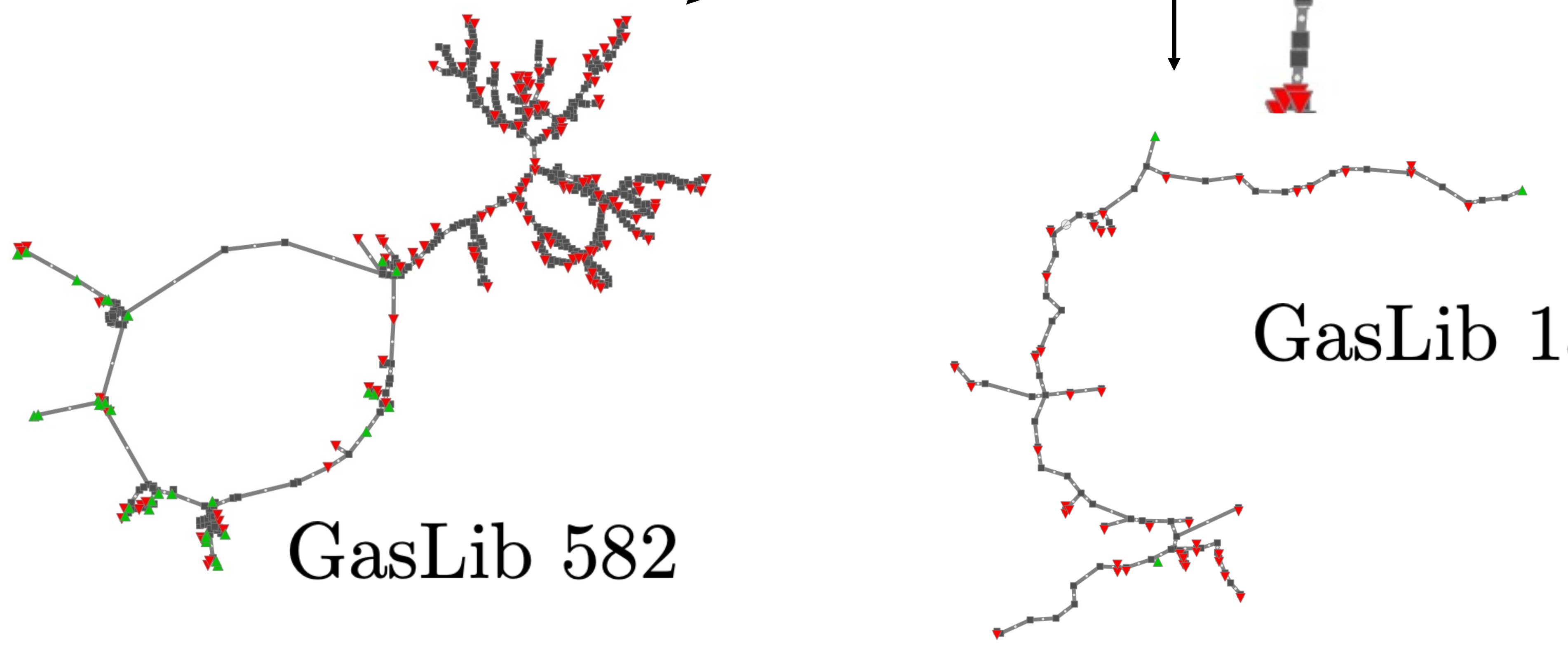
Domain decomposition of optimal control problems on metric graphs

The scope



technical zoom
out of a subnetwork
GasLib 4197
see: http://gaslib.zib.de





Gas flow in pipe networks

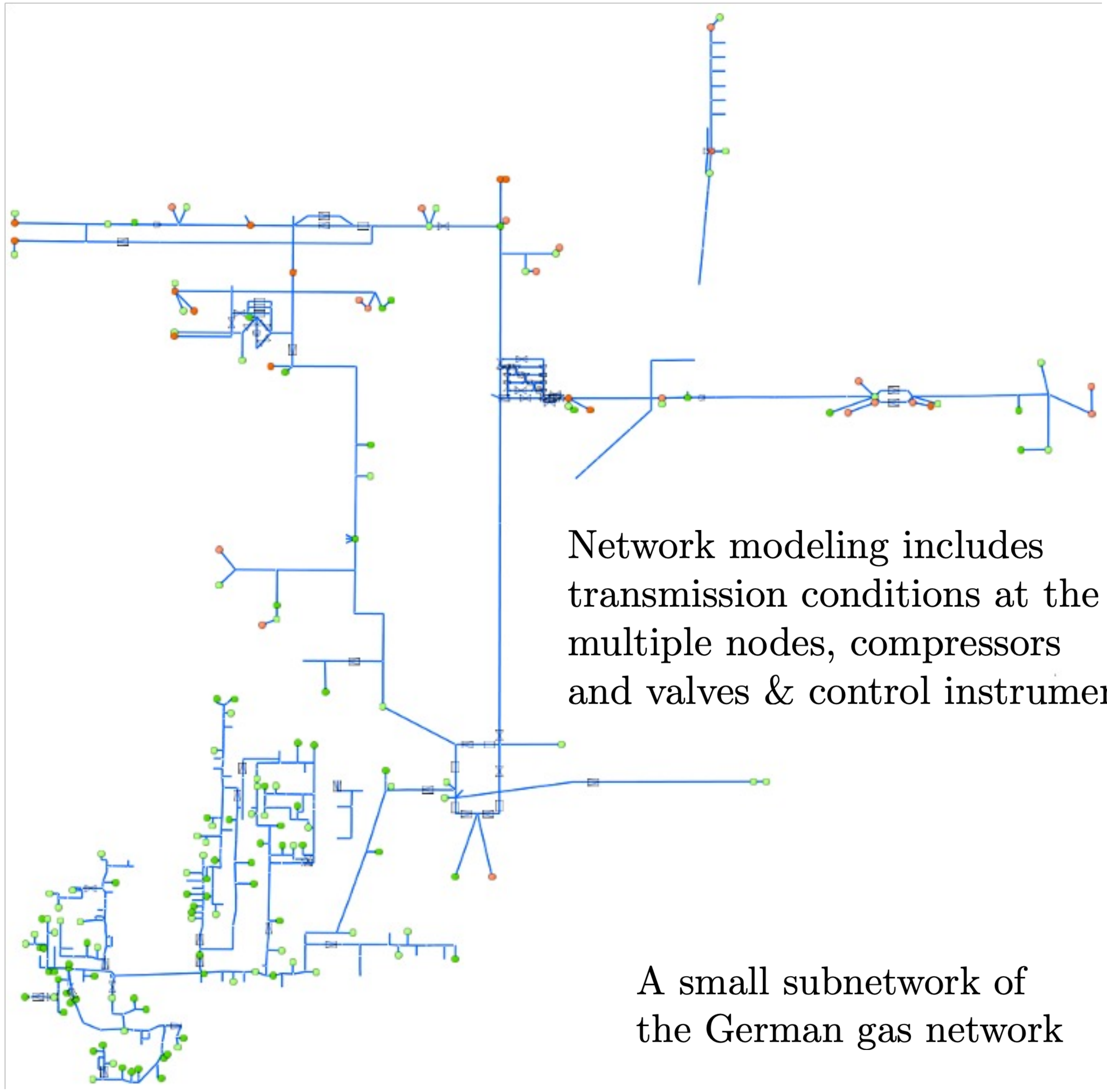
Derivation of the model equations

We start with the Euler system

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

We reduce this to a semi-linear form

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



Network modeling includes transmission conditions at the multiple nodes, compressors and valves & control instruments

A small subnetwork of the German gas network

Gas flow in pipe networks

Model hierarchy

and together with $c\sqrt{q/\rho}$ and $q = a\rho v$

non-smooth and singular

$$\frac{\partial p}{\partial t} + \frac{c^2}{a}\frac{\partial q}{\partial x} = 0$$
$$\frac{\partial q}{\partial t} + \frac{\partial p}{\partial x} = -\frac{\lambda c^2}{2Da^2 p} q|q| \qquad \text{(SL)}$$

If we then neglect the inertia in the second equation we arrive at

$$\frac{\partial p}{\partial t} + \frac{c^2}{a}\frac{\partial}{\partial x} q = 0,$$
$$\frac{\partial p^2}{\partial x} = -\frac{\lambda c^2}{Da^2} q|q| =: -\gamma^2 q|q|.$$

Network modeling for friction dominated flow

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

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

With $\alpha_0 := \frac{\gamma a}{c}$, we obtain

$$\alpha_0 \frac{\partial}{\partial t}\frac{y}{\sqrt{|y|}} - \frac{\partial}{\partial x}\frac{\frac{\partial y}{\partial x}}{\sqrt{|\frac{\partial y}{\partial x}|}} = 0.$$

Thus,

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

where $p = \frac{3}{2}$. More generally, with $\beta_q(s) := \alpha s|s|^{p-2}$ and $1 < q = p, \alpha < \infty$, we obtain

$$\frac{\partial}{\partial t}\beta_\alpha(y) - \frac{\partial}{\partial x}\beta_p(\frac{\partial y}{\partial x}) = 0.$$

M. A. Stoner 1969
P.J. Wong, R.E. Larson 1968

A.Bamberger, M. Sorin, J.P. Yvon'79

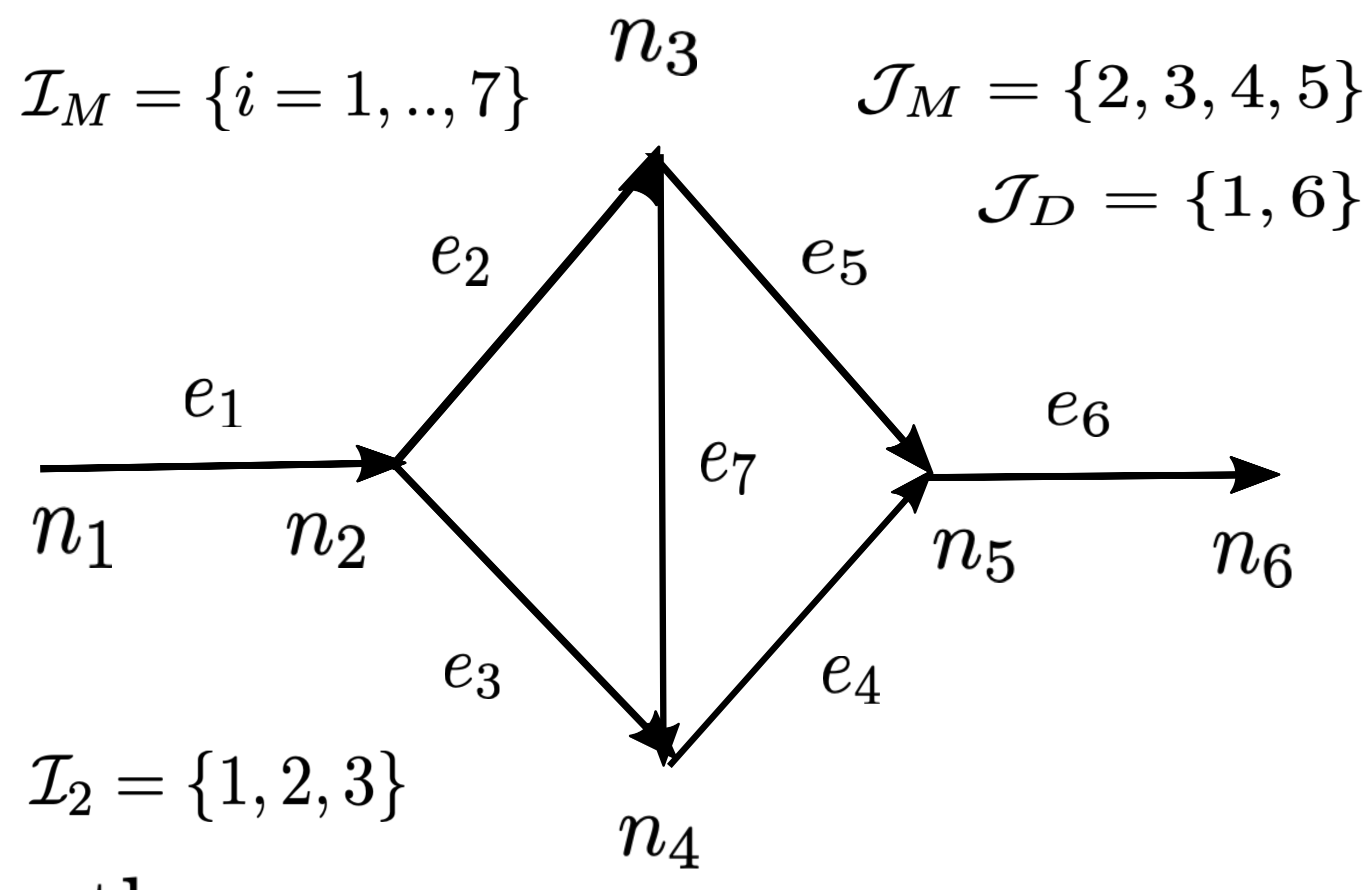
This problem is also singular

P. A. Raviart'70

Graph notation

• Graph $G = (V, E)$, with vertices $V = \{n_1, n_2, \ldots, n_{|V|}\} = \{n_j | j \in \mathcal{J}\}$ and edges $E = \{e_1, e_2, \ldots, e_{|E|}\} = \{e_i | i \in \mathcal{I}\}$.
• Edge-to-node incidence matrix

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



• Each edge e_i is given in general by a line segment $[0, \ell_i]$
• $e_i = [n_j, n_k]$ such that $d_{ij} = -1, d_{ik} = 1$, then $x = 0$, $x = \ell_i$ correspond to the nodes n_j, n_k respectively.
• More precisely, we introduce the notion x_{ij}, where $x_{ij} = 0$ if $d_{ij} = -1$, $x_{ij} = \ell_i$ if $d_{ij} = 1$.

• The edge degree is $d_j := |\mathcal{I}_j|$.

• $\mathcal{J} = \mathcal{J}^M \cup \mathcal{J}^S$, 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 $\mathcal{J}^S = \mathcal{J}^S_D \cup \mathcal{J}^S_N$.

The network model with controls

$$
\begin{aligned}
&\alpha_i \partial_t \beta(y_i(x,t)) - \partial_x \left(\beta(\partial_x y_i(x,t))\right) = u_i(x,t), && i \in \mathcal{I},\ x \in (0,\ell_i),\ t \in (0,T),\\
&\quad y_i(n_j,t) = y_k(n_j,t), && \forall i,k \in \mathcal{I}_j,\ j \in \mathcal{J}^M,\ t \in (0,T),\\
&\quad \sum_{i \in \mathcal{I}_j} d_{ij} \beta(\partial_x y_i(n_j,t)) = 0, && j \in \mathcal{J}^M,\ t \in (0,T)\\
&\quad y_i(n_j,t) = 0, && i \in \mathcal{I}_j, j \in \mathcal{J}^S_D,\ t \in (0,T),\\
&\quad d_{ij} \beta(\partial_x y_i)(n_j,t) = u_j(t), && i \in \mathcal{I}_j, j \in \mathcal{J}^S_N,\ t \in (0,T),\\
&\quad y_i(x,0) = y_i^0(x), && x \in (0,\ell_i),
\end{aligned}
\tag{Net}
$$

where the functions $u_i, i \in \mathcal{I}$, $u_j, j \in \mathcal{I}_j, j \in \mathcal{J}^S_N$ serve as distributed and boundary controls, respectively.

The optimal control problem

$$I_y(y) := \sum_{i\in\mathcal{I}} \int_0^T \int_0^{\ell_i} \frac{\kappa_i}{2} |y_i(x,t) - y_i^d(x,t)|^2 dx dt, \; I_T(y(x,T)) := \sum_{i\in\mathcal{I}} \int_0^{\ell_i} \frac{\kappa_{i,T}}{2} |y_i(x,T) - y_{i,T}|^2 dx$$

for the state, while the norms of the controls are penalized as follows

$$I_u(u) := \sum_{i\in\mathcal{I}} \frac{\nu_{i,d}}{2} \int_0^T \int_0^{\ell_i} |u_i(x,t)|^2 dx dt + \sum_{j\in\mathcal{J}_N^S} \frac{\nu_{i,b}}{2} \int_0^T |u_j(t)|^2 dt,$$

where $\kappa_i, \kappa_{i,T} \geq 0, \nu_{i,d}, \nu_{i,b} \geq 0$ serve as penalty parameters. We pose the following optimal control problem for (1)

$$\begin{gathered} \min_{(y,u)} I(y,u) := I_y(y) + I_T(y(\cdot,T)) + I_u(u) \\ s.t. \\ (y,u) \text{ satisfies } (NET). \end{gathered} \tag{OCP}$$

The corresponding optimality system

$$
\begin{aligned}
&\alpha_i \partial_t \beta(y_i(x,t)) - \partial_x\left(\beta(\partial_x y_i(x,t))\right) = \frac{1}{\nu_{i,d}} p_i(x,t), \\
&\alpha_i \beta'(y_i(x,t))\partial_t p_i(x,t) + \partial_x\left(\beta'(\partial_x y_i(x,t))\partial_x p_i(x,t)\right) = \kappa_i(y_i - y_i^d), && i \in \mathcal{I},\ x \in (0,\ell_i),\ t \in (0,T), \\
&\quad y_i(n_j,t) = y_k(n_j,t),\ p_i(n_j,t) = p_k(n_j,t), && \forall i,k \in \mathcal{I}_j,\ j \in \mathcal{J}^M,\ t \in (0,T), \\
&\quad \sum_{i \in \mathcal{I}_j} d_{ij}\beta(\partial_x y_i(n_j,t)) = 0,\ \sum_{i \in \mathcal{I}_j} d_{ij}\beta'(\partial_x y_i(n_j,t))\partial_x p_i(n_j,t) = 0, && j \in \mathcal{J}^M,\ t \in (0,T), \\
&\quad y_i(n_j,t) = 0,\ p_i(n_j,t) = 0, && i \in \mathcal{I}_j, j \in \mathcal{J}_D^S,\ t \in (0,T), \\
&\quad d_{ij}\beta(\partial_x y_i(n_j,t)) = \frac{1}{\nu_{i,b}} p_j(n_j,t),\ d_{ij}\beta'(\partial_x y_i(n_j,t))\partial_x p_i(n_j,t) = 0, && i \in \mathcal{I}_j, j \in \mathcal{J}_N^S, t \in (0,T), \\
&\quad y_i(x,0) = y_{i,0}(x),\ p_i(x,T) = -\kappa_{i,T}(y_i(x,T) - y_{iT}^d(x)), && x \in (0,\ell_i),
\end{aligned}
\tag{GOS}
$$

where p denotes the adjoint variable (Lagrange multiplier).

We need to be careful with possibly 'flat regions'

Decomposition

Principal remarks

- We want to iteratively decompose the optimality system (GOS) on the ‚global‘ network G into subnetworks (*Network tearing and Interconnection* NETI), in fact here, to each individual edge. Analysis in the continuous setting!

- The decomposed optimality system (DOS) should itself be an optimality system for an optimal control problem on the subnetwork (i.e. edge) including *virtual controls* at the multiple nodes (interfaces), in the sense of J.L. Lions and O. Pironneau 1999.

- The decomposition should be *non-overlapping* (in the sense of P.L. Lions 1989) overlapping domains are not intuitive at multiple nodes. Overlapping Schwarz-type methods at serial connections (‚cutting out stars’) are also under consideration (not here, however), see Gon, Kwok, Tan 2022

- Space-time domain decomposition

Previous work

- **General domains (manifolds, continuous level, no controls; very selective list)**: Early work by P. L. Lions'1989 and O. Pironneau & J.L. Lions'1999 pursued later by J.-D. Benamou'1992-99 for elliptic and parabolic problems, A. Quarteroni'1988-16, F. Nataf' 91-', M. Gander'00-, G. Ciaramella'17-,L. Halpern'00-, J. Haslinger'00-14, J. Kucera,T. Sassi (Signorini-type contact problems), E. Engström, E. Hansen'22 (Robin-type p-Laplace)…M. Dryia, W. Hackbusch'97 (general finite dimensional(!) nonlinear problems)

- **Time domain decomposition (continuous level; again very selective list):** J.L. Lions, Y. Maday, G. Turinici'01, J. Salomon'07-, M. Gander'07-, F. Kwok'18-,G. Ciaramella'21 (semi-linear elliptic) ….(parareal/multiple shooting)…space-time…

- **Optimal control problems:** M. Heinkenschloss'00-11, M. Herty'07, S. Ulbrich'07, M. Gander,'00- F. Kwok'17-, V. Agoshkov'85-, P. Gervasio'04-16, A. Quarteroni'05/06, B. Delourme, L. Halpern, B. Nguyen'06, W. Gong, F. Kwok, Z. Tan'22 (overlapping domains) many others, for linear elliptic and parabolic problems (in almost all cases).

- **Networked domains and optimal control** (non-manifolds; multiple nodes in 1-D and interfaces in 2- or 3-D): J. E. Lagnese & G.L. 2003, G.L. (et al.) 2018-2022.

Example: diamond graph

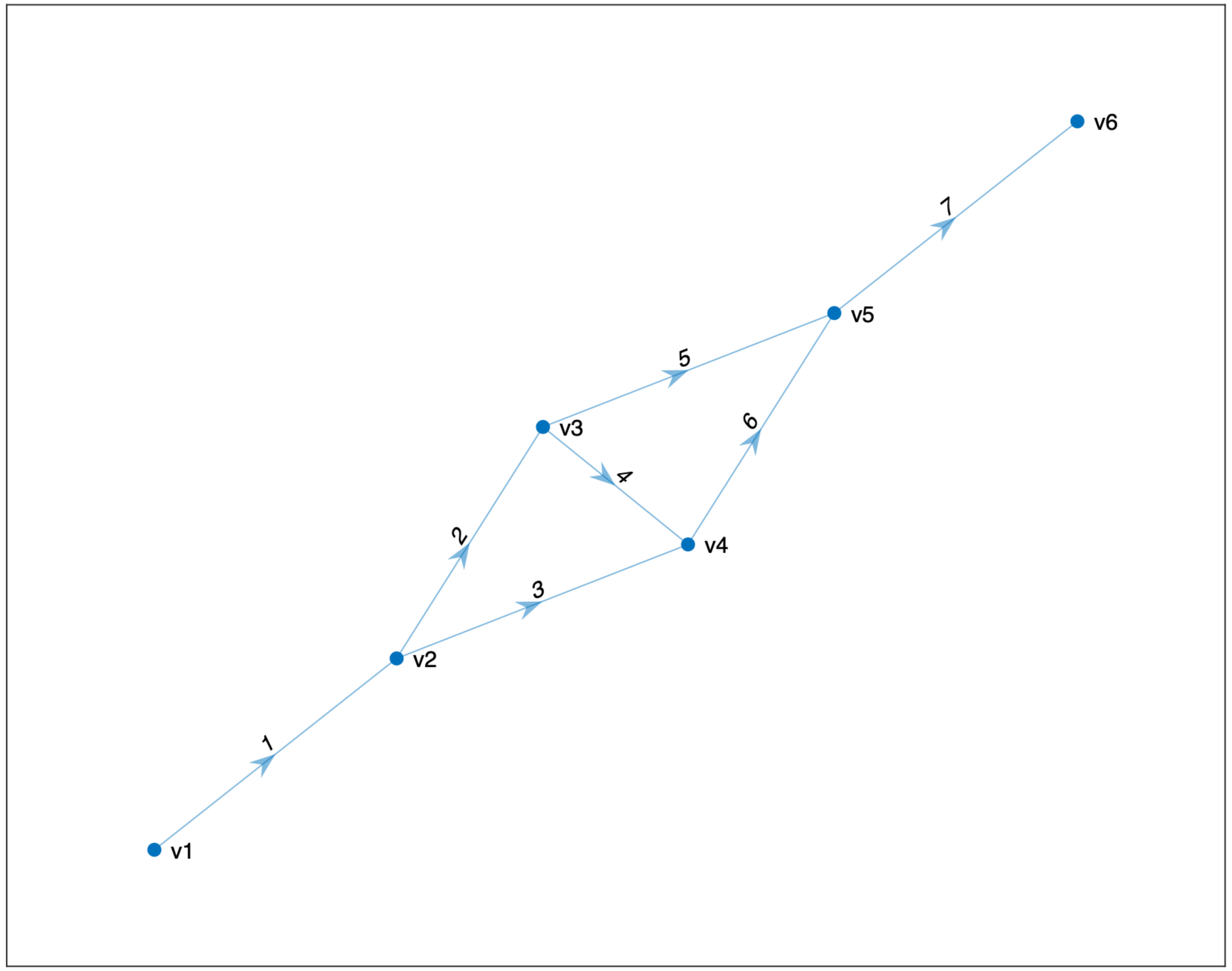
- We consider the so-called diamond graph,

- We apply a Neumann condition at n_6 and
a boundary control at n_1.

- We want to steer y_4 to the constant value 1,
applying running costs and terminal costs, individually.

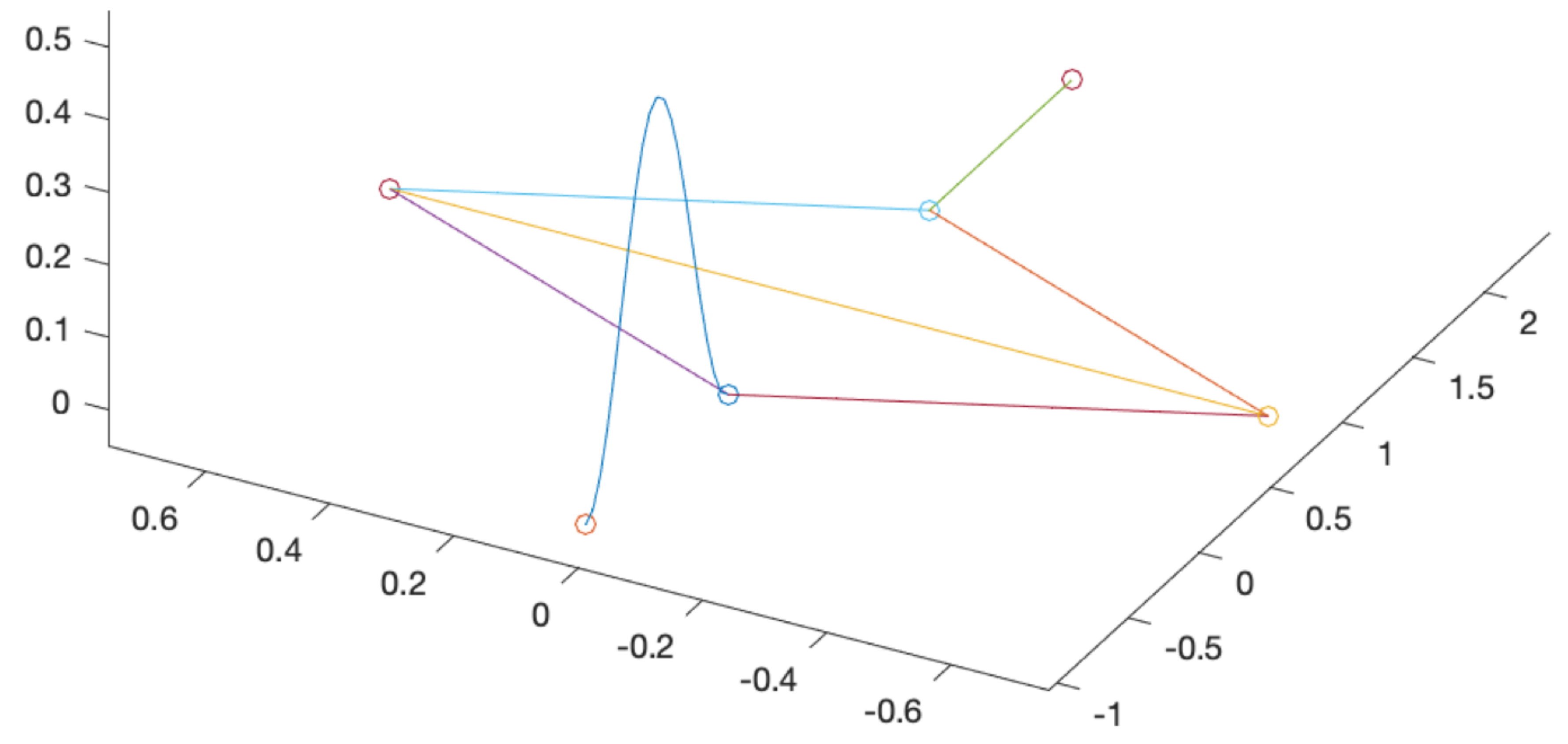
- For the penalty data, we take $\kappa = 1.e4, \nu = 1$

- We use standard discretization in space and time,
as already proposed by Bamberger'77 and Raviart'70

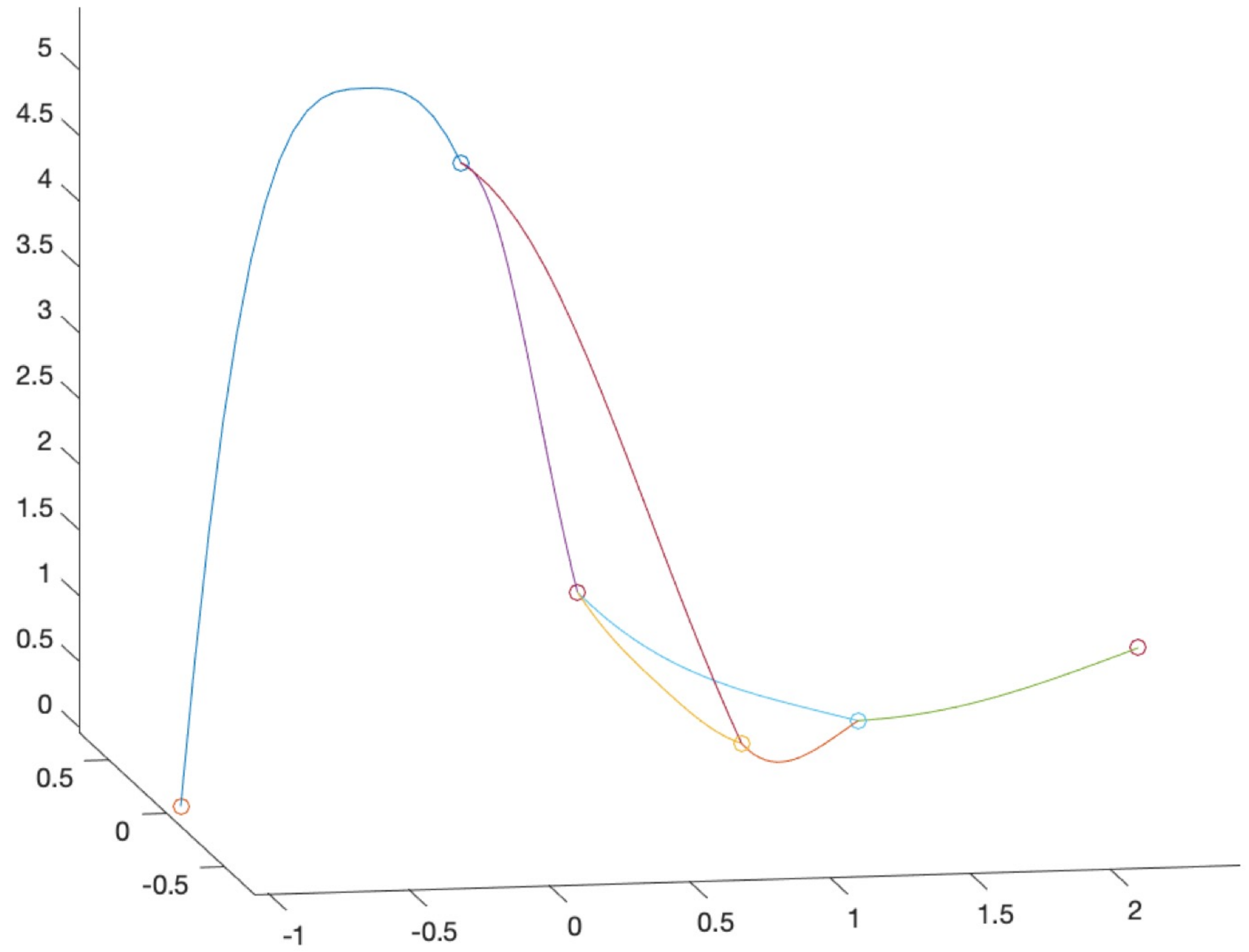


The diamond graph

Example full network

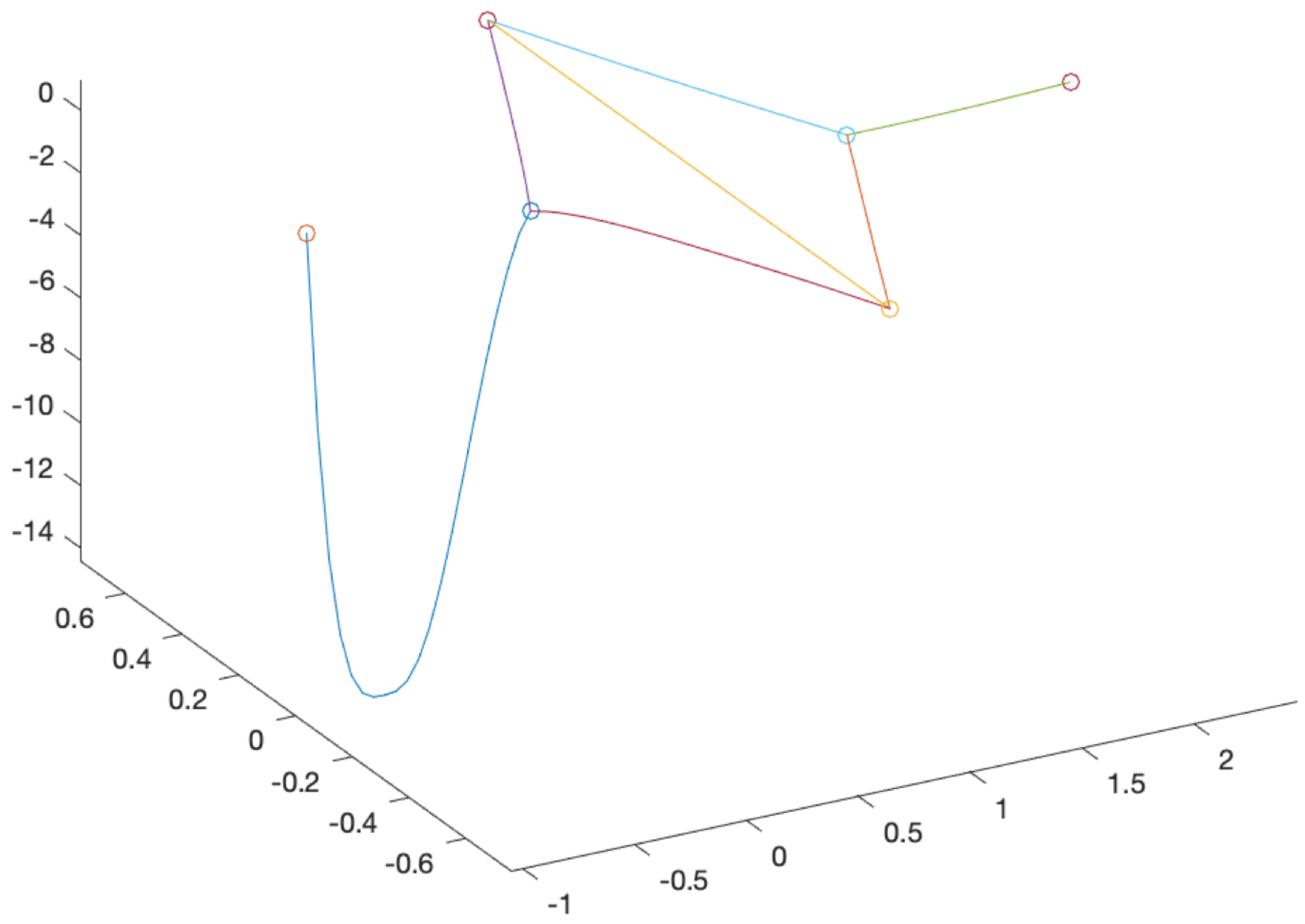


Initial condition

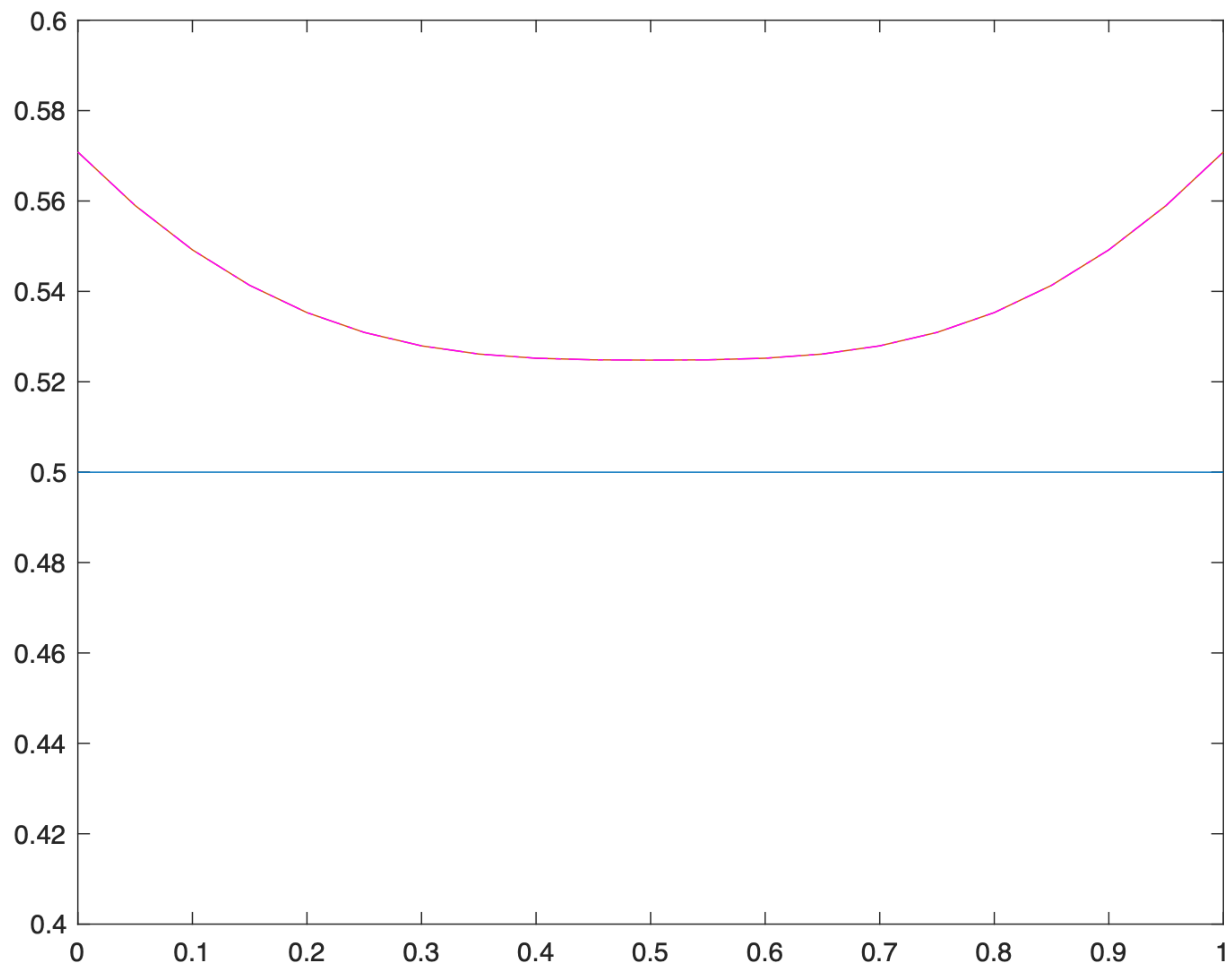


At final time with running costs

Example



Final value control



Comparison of final states

Domain decomposition in space

The P.L. Lions algorithm extended to p-parabolic equations

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

Notice: for serial connections $d_j = 2$, thus the classical P.L. Lions algorithm obtains

DDM á la Lions for two domains revisited

We look at the two link problem on the interval $[-11]$, where we decompose at $x = 0$. We introduce the Steklov-Poincaré mappings

$$S_i(\eta_i) := (-1)^{i+1}\beta(\partial_x y_i(t,0),\ i = 1,2,$$

where y_i are the solutions of the corresponding initial boundary value problems on $(-1,0)$, $(0,1)$ respectively with boundary data η_i. Then the transmission conditions

$$\beta(\partial y_1(t,0) = \beta(y_2(t,0)),\ y_1(t,0) = y_2(t,0)$$

are equivalent to

$$S_1(\eta_1) + S_2(\eta_2) = 0,\ \eta_1 = \eta_2.$$

which, in turn is equivalent to

$$(\sigma I + S_1)\eta_1 = (\sigma I - S_2)\eta_2,\ (\sigma I + S_2)\eta_2 = (\sigma I - S_1)\eta_1$$

DDM á la Lions for two domains revisited

There are now two ways to solve this system iteratively.

- Jacobi-type method

$$
\begin{aligned}
(\sigma I + S_1)\eta_i^{k+1} &= (\sigma I - S_2)\eta_2^k \\
(\sigma I + S_2)\eta_2^{k+1} &= (\sigma I - S1)\eta_1^k
\end{aligned}
$$

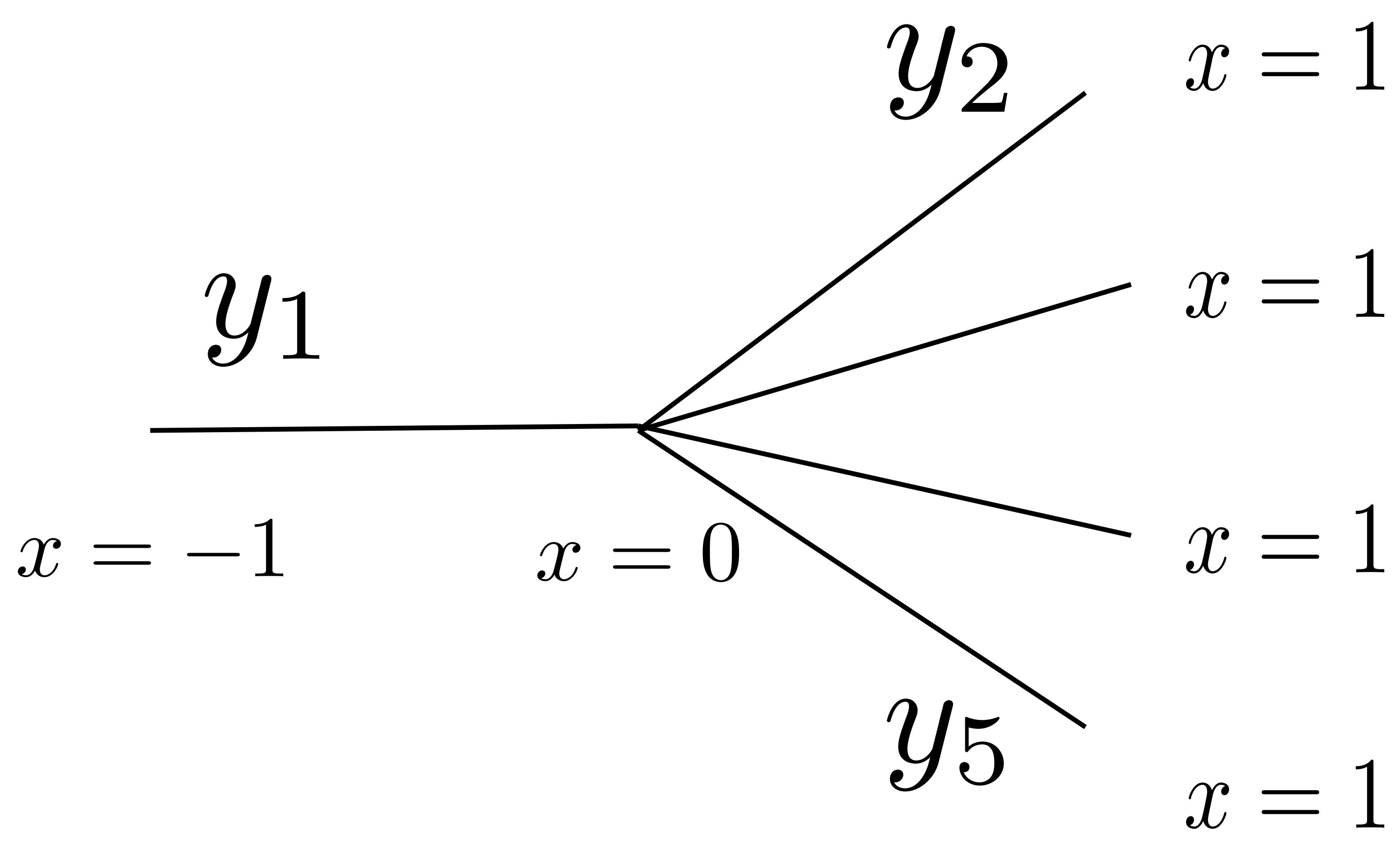
- GaußSeidel type

$$
\begin{aligned}
(\sigma I + S_1)\eta_i^{k+1} &= (\sigma I - S_2)\eta_2^k \\
(\sigma I + S_2)\eta_2^{k+1} &= (\sigma I - S1)\eta_1^{k+1}
\end{aligned}
$$

Notice that the first method is completely parallel, while the second is not. The first iteration is the one, we propose for networks. See Engström, Hansen 2022 for the p-Laplace.

See Jan Sokolowski's talk on Thursday for the use of the Steklov-Poincaré map for (non-iterative) domain decomposition

Splitting of a multiple node



$$S_1(\bar{y})(t) := \beta(\partial_x y_1(t, 0; \bar{y}))$$

$$S_2(\bar{y})(t) := -\sum_{i=2}^{5} \beta(\partial_x y_i(t, 0; \bar{y}))$$

Steklov-Poincaré equation at the multiple node

$$S_1(\bar{y}) + S_2(\bar{y}) = 0.$$

Idea of proof: p-Laplace

To fix ideas, we consider just the p-Laplace problem:

$$
\begin{aligned}
&-\partial_x \beta(\partial_x y_i) = f_i, \; x \in I_i \\
&y_1(-1) = 0, \, y_i(1) = 0, i = 2, 3, 4, 5, \\
&y_i(0) = \bar{y}, i = 1, \ldots, 5, \; \sum_{i=1}^{5} \beta(\partial_x y_i(0)) = 0,
\end{aligned}
$$

$I_1 := (-1, 0), I_i = (0, 1), i = 2, 3, 4, 5$

Realizing the Steklov-Poincaré setting

We first proceed formally (and then reflect on the Robin-trace operators). We have

$$\eta_1^{k+1} = (\sigma I + S_1)^{-1}(\sigma I - S_2)\eta_2^k$$
$$\eta_2^{k+1} = (\sigma I + S_2)^{-1}(\sigma I - S_1)\eta_1^{k+1}$$

and introduce

$$\mu^k = (\sigma I + S_2)\eta_2^k,\ \mu := (\sigma I + S_2)\eta_2$$
$$\lambda^k = (\sigma I - S_2)\eta_2^k,\ \lambda := (\sigma I - S_2)\eta_2$$

Thus

$$\frac{1}{2\sigma}(\mu^k + \lambda^k) = \eta_2^k,\ \frac{1}{2\sigma}(\mu + \lambda) = \eta_2,\ \frac{1}{2\sigma}(\mu^{k+1} + \lambda^k) = \eta_1^{k+1}$$
$$\frac{1}{2}(\mu^k - \lambda^k) = S_2\eta_2^k,\ \frac{1}{2}(\mu - \lambda) = \eta_2,\ \frac{1}{2}(\lambda^k - \mu^{k+1}) = S_1\eta_1^{k+1}.$$

Convergence

This implies

$$(S_2\eta_2^k - S_2\eta)(\eta_2^k - \eta) = \frac{1}{4\sigma}\left((\mu^k - \mu)^2 - (\lambda^k - \lambda)^2\right) \geq 0$$

$$(S_1\eta_1^{k+1} - S_1\eta)(\eta_1^{k+1} - \eta) = \frac{1}{4\sigma}\left((\lambda^k - \lambda)^2 - (\mu^{k+1} - \mu)^2\right) \geq 0,$$

where the inequalities follow from the monotonicity of $\beta(\cdot)$ (see below). This implies

$$|\mu^{k+1} - \mu|^2 \leq |\lambda^k - \lambda|^2, \ |\lambda^k - \lambda|^2 \leq |\mu^k - \mu|^2.$$

and, hence

$$0 \leq \sum_{k=0}^{K} \left(|\mu^k - \mu|^2 - |\mu^{k+1} - \mu|^2\right) \leq |\mu^0 - \mu|^2, \ \forall K$$

Thus $|\mu^k - \mu|^2 - |\mu^{k+1} - \mu|^2 \to 0$ as $k \to \infty$, and, therefore,

$$(S_2\eta_2^k - S_2\eta)(\eta_2^k - \eta) \to 0, \ (S_1\eta_1^{k+1} - S_1\eta)(\eta_1^{k+1} - \eta) \to 0.$$

Monotonicity of the p-Laplacian

The crucial property is the monotonicity of the p-Laplacian also for $1 < p < 2$:

$$-\partial_x \beta(\partial_x y_i) = f_i, \; x \in I_i$$
$$y_1(-1) = 0, \, y_i(1) = 0, i = 2, 3, 4, 5,$$
$$y_i(0) = \bar{y}, i = 1, \ldots, 5, \; \sum_{i=1}^{5} \beta(\partial_x y_i(0)) = 0,$$

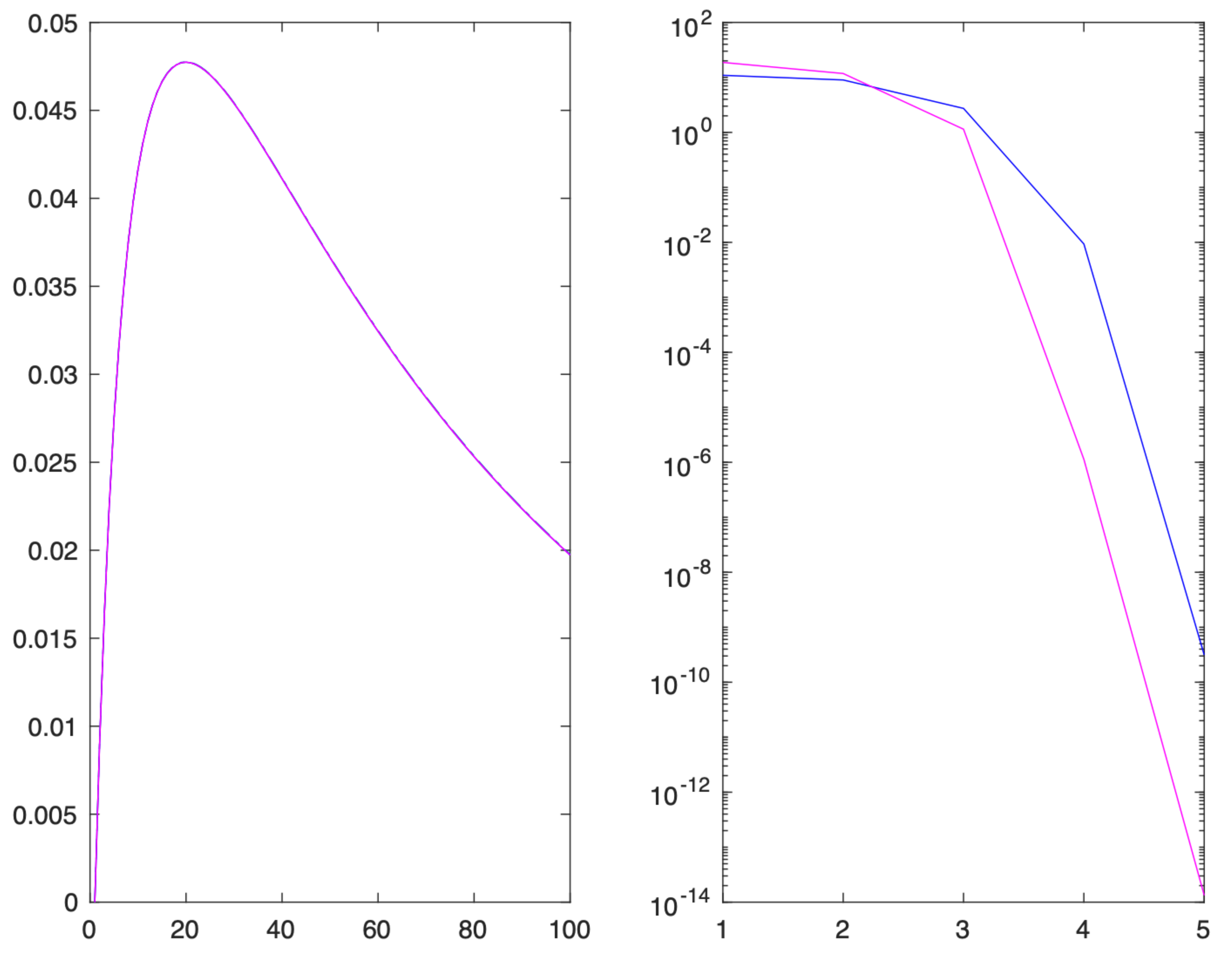
$I_1 := (-1, 0), I_i = (0, 1), i = 2, 3, 4, 5.$ Thus, because of $(\beta(a) - \beta(b))(a - b) \geq (|a| + |b|)^{p-2}|a - b|^2$ for $1 < p < 2$, we have

$$(S_1(a) - S_1(b))(a - b) \geq \int_{-1}^{0} (|\partial_x y_{1,a}| + |\partial_x y_{1,b}|)^{p-2} |\partial_x y_{1,a} - \partial_x y_{1,b}|^2 dx$$

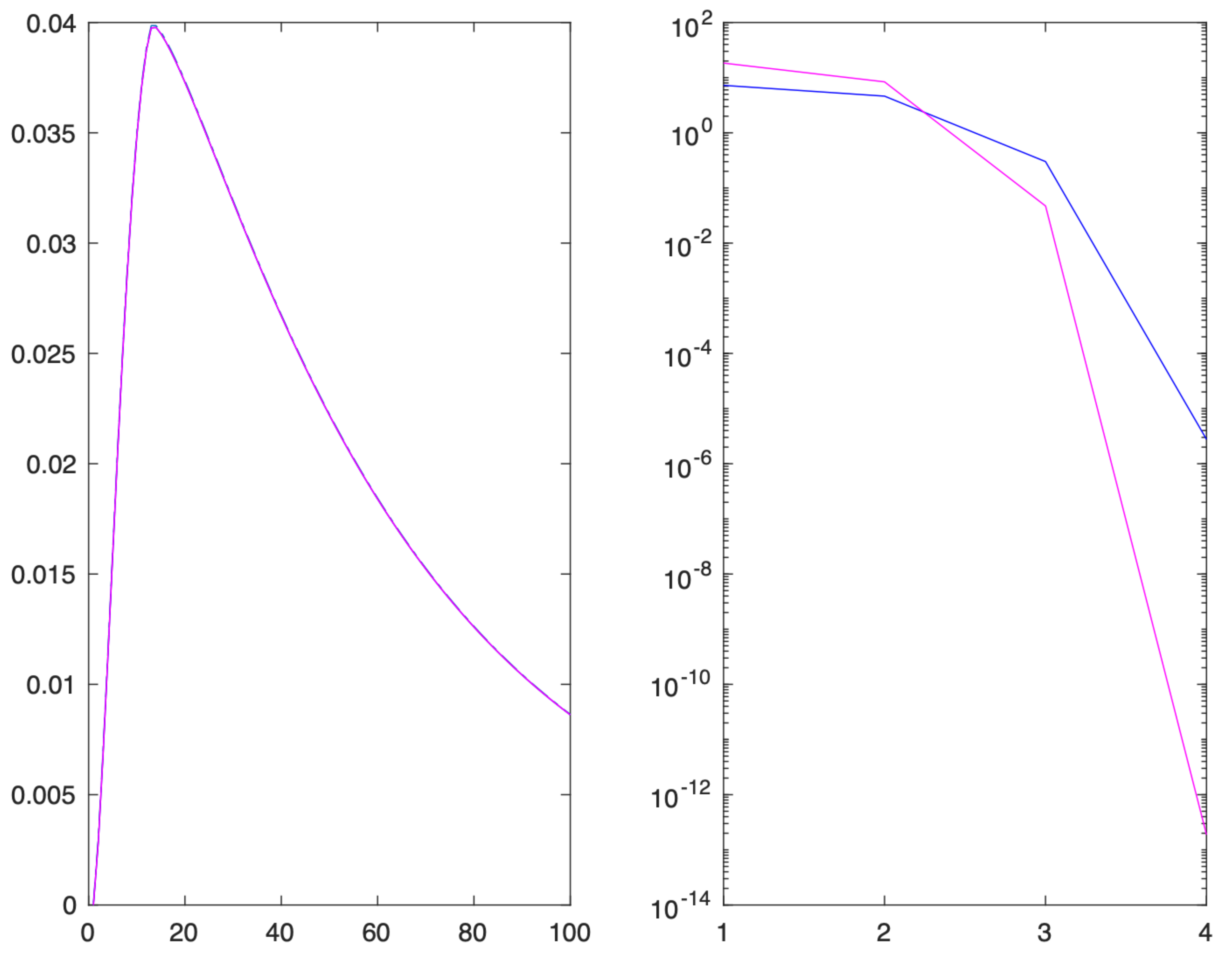
which is clearly positive, but for bounded $\partial_x y_{1,a}, \partial_x y_{1,b}$ this dominates the H^1-norm

Example: two-link p-parabolic problem

We take the interval $(0,2)$ and introduce the interface at $x = 1$. At $x = 0$, we have Dirichlet boundary conditions and at $x = 2$ Neumann conditions, as well as initial conditions $\sin(\pi x)^2$ in each domain. The load is equal to 1 everywhere. We apply the algorithm above with $\rho = .5$ and use the **pdpe** code from Matlab.



p=2; Left: plot of the solutions in domains 1 and 2 on top of ‚true‘ solution
Right: the errors of state and fluxed at the interface



Same as on left figure, but now for p=3/2

Relevance for optimal control problems

We can approach the decomposition of the optimality system by the following fixed point procedure:

1. Choose controls (distributed and boundary controls)
2. Solve the state equation in parallel using the DDM above
3. Input the state in the (linear!) adjoint equation and solve using the classical (still extended) DDM (se e.g. Benamou)
4. Retrieve the controls using the optimality condition and go back to the first step until done.

Notice, however, that this procedure is not a DDM for the optimality system as a whole and, consequently, does not lead to a substitute optimal control problem on the subnetworks.

DDM algorithm for the optimality system: Algorithm

1. Given $\lambda_{ij}^n, \rho_{ij}^n$,
2. solve for y_i^{n+1}, p_i^{n+1}

$$\partial_t \beta_i(y_i^{n+1}) - \partial_x(\beta_i(\partial_x y_i^{n+1})) = \frac{1}{\nu_{i,d}} p_i^{n+1},$$
$$\beta_i'(y_i^{n+1})\partial_t p_i^{n+1} + \partial_x(\beta_i'(\partial_x y_i^{n+1})\partial_x p_i^{n+1}) = \kappa_i(y_i^{n+1} - y_i^d),$$
$$y_i^{n+1}(x_{ij}, t) = 0,\ p_i^{k+1}(x_{ij}, t) = 0,$$
$$d_{ij}\beta_i(\partial_x y_i^{n+1})(x_{ij}, t) = \frac{1}{\nu_{i,b}} p_i(x_{ij}, t),\ d_{ij}\beta_i'(\partial_x y_i^{n+1})\partial_x p_i(x_{ij}^{n+1})(x_{ij}, t) = 0,$$

$$d_{ij}\beta_i(\partial_x y_i^{n+1})(x_{ij}) + \sigma y_i^{n+1}(x_{ij}) + \mu p_i^{n+1}(x_{ij}) = -\left(\frac{2}{d_j}\sum_{l\in\mathcal{I}_j} d_{lj}\beta_l(\partial_x y_l^n)(x_{lj}) - d_{ij}\beta_i(\partial_x y_i^n)(x_{ij})\right)$$
$$+\sigma\left(\frac{2}{d_j}\sum_{l\in\mathcal{I}_j} y_l^n(x_{lj}, t) - y_i(x_{ij}, t)\right) + \mu\left(\frac{2}{d_j}\sum_{l\in\mathcal{I}_j} p_l^n(x_{lj}, t) - p_i(x_{ij}, t)\right) =: \lambda_{ij}(t)^n,$$

Algorithm cont.

$$d_{ij}\beta_i'(\partial_x y_i^{n+1}(x_{ij},t))\partial_x p_i^{n+1}(x_{ij},t) + \sigma p_i^{n+1}(x_{ij},t) - \mu y_i^{n+1}(x_{ij},t)$$
$$= -\left(\frac{2}{d_j}\sum_{l\in\mathcal{I}_j} d_{lj}\beta_l'(\partial_x y_l^n(x_{lj},t))\beta_l(\partial_x p_l^n)(x_{lj},t) - d_{ij}\beta_i'(\partial_x y_i^n(x_{ij},t))(\beta_i(\partial_x p_i^n)(x_{ij},t)\right)$$
$$+\sigma\left(\frac{2}{d_j}\sum_{l\in\mathcal{I}_j} p_l^n(x_{lj},t) - p_i(x_{ij},t)\right) - \mu\left(\frac{2}{d_j}\sum_{l\in\mathcal{I}_j} y_l^n(x_{lj},t) - y_i(x_{ij})\right) =: \rho_{ij}^(t).$$

3. Update $\lambda_{ij}^{n+1}, \rho_{ij}^{n+1}$ for $n \to n+1$.

Notice: for serial connections $d_j = 2$, the algorithm of J.D. Benamou obtains in case $p = 2$ for the parabolic problem

Equivalent virtual control problem

1. Given $\lambda_{ij}^n, \rho_{ij}^n$,

2. solve for $y_i^{n+1}, u_i^{n+1}, u_j^{n+1}, j \in \mathcal{J}_i$

virtual controls

$$\min_{u,g,y} \left\{ J_i(y_i, u_i) + \frac{1}{2\mu} \sum_{j \in \mathcal{J}_i} \int_0^T \left[|g_i|^2 + |\mu y_i - \rho_{ij}^n|^2 \right] dt \right\}$$

s. t.

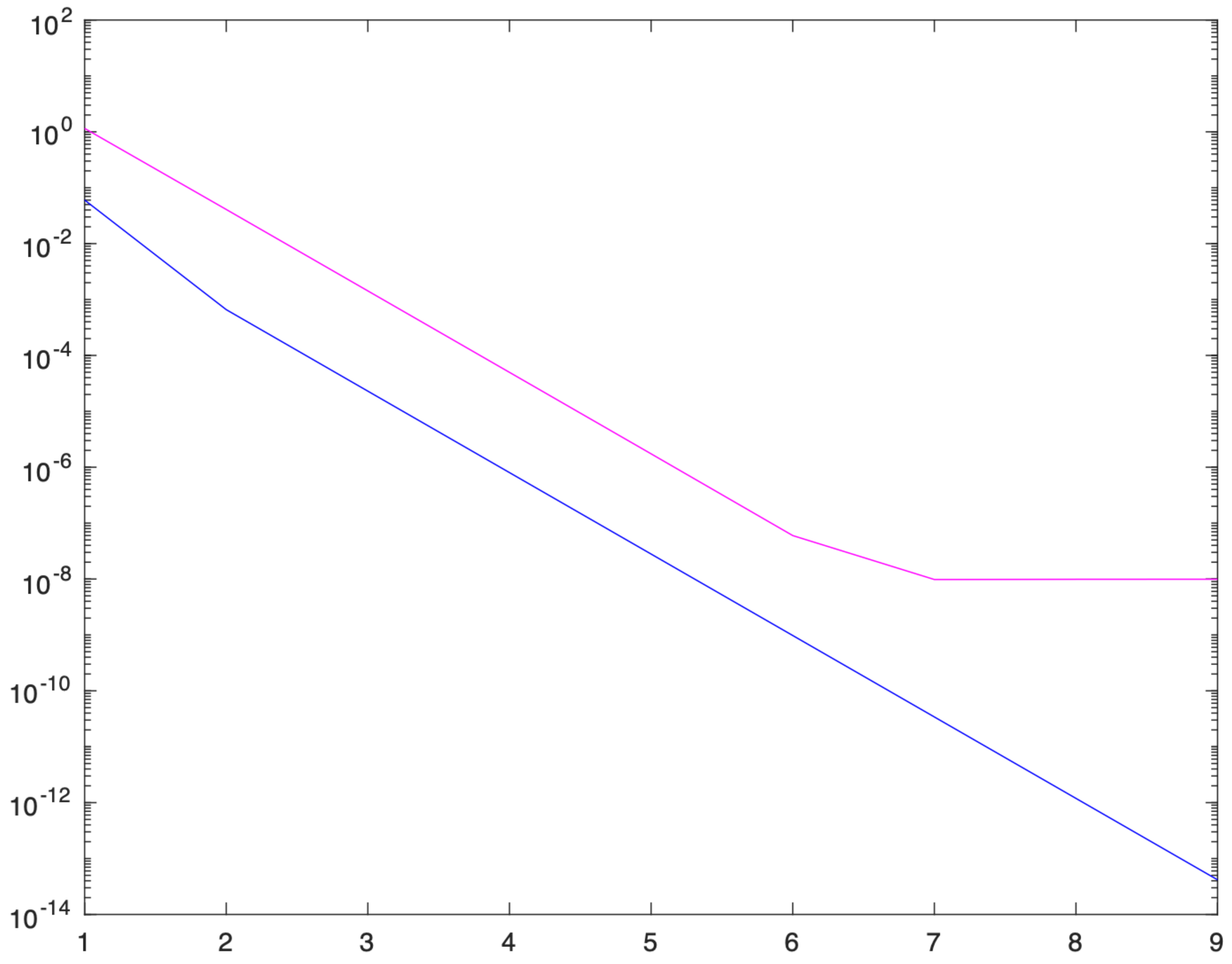
$$\begin{aligned}
&\partial_t \beta_i(y_i) - \partial_x(\beta_i(\partial_x y_i)) = u_i, && i \in \mathcal{I}, x \in I_i, t \in (0,T) \\
&d_{ij}\beta_i(\partial_x y_i(x_{ij},t)) + \sigma y_i(x_{ij},t) = \lambda_{ij}(t)^n + g_{ij}(t), && j \in \mathcal{J}_i, i \in \mathcal{I}_j, t \in (0,T) \\
&y_i, t = 0, && i \in \mathcal{I}_j, j \in \mathcal{J}_D^S, t \in (0,T) \\
&d_{ij}\beta_i(\partial_x y_i(x_{ij},t)) = u_j(t), && i \in \mathcal{I}_j, j \in \mathcal{J}_N^S, t \in (0,T).
\end{aligned}$$

'real' controls

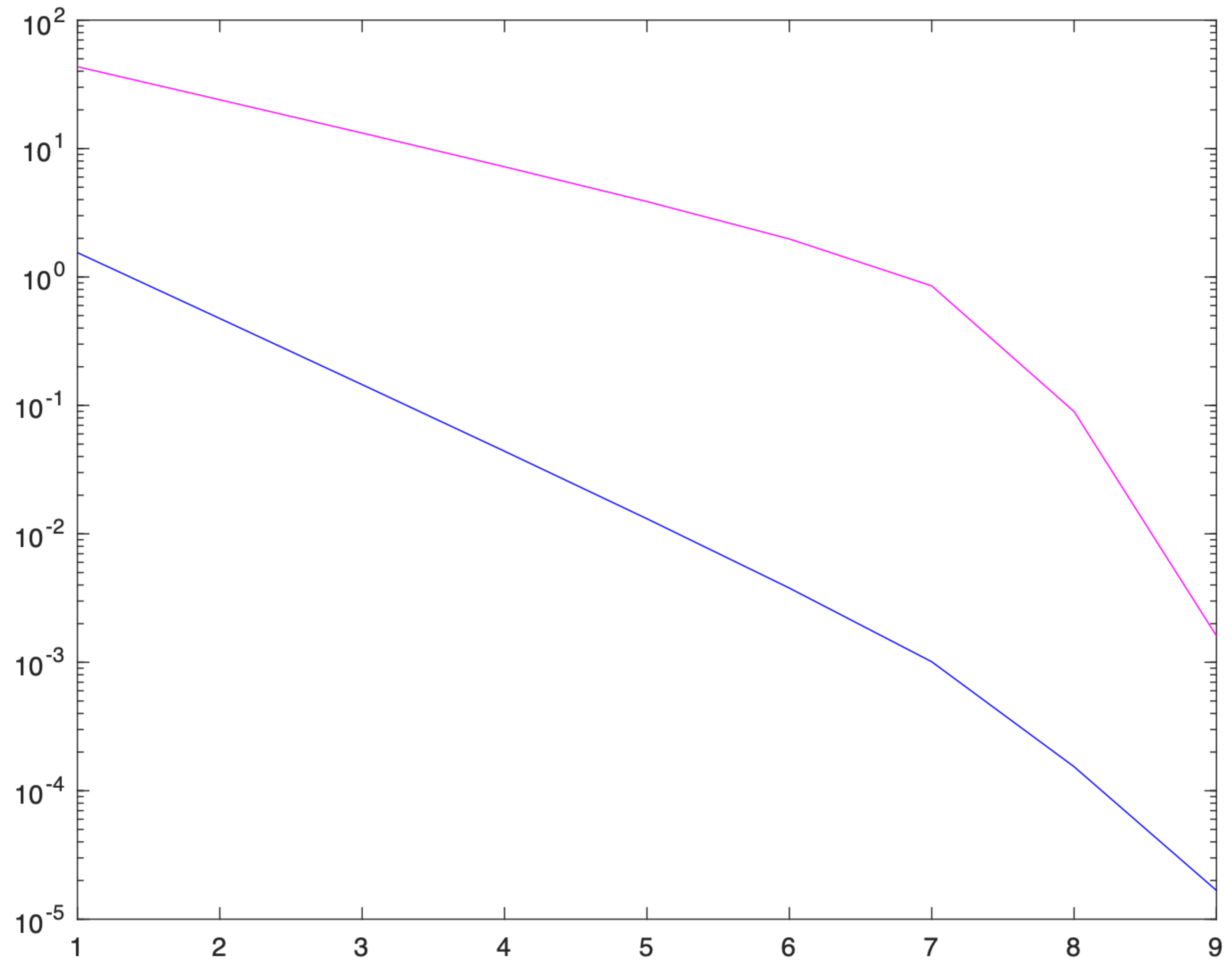
3. Update $\lambda_{ij}^{n+1}, \rho_{ij}^{n+1}$ for $n \to n+1$.

Example

A two-link problem



p=2, sigma=0, mu=10, nu=0, kappa=1000



p=3/2, sigma=50, mu=100, nu=.001, kappa=1000

Time-domain decomposition

- We introduce a coarse time discretization with

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

- We introduce the intervals $I_k := (T_k, T_{k+1})$.
- We take the optimality system and restrict the to time interval I_k.
- At the time-interfaces T_k, T_{k+1}, we employ continuity conditions $(y_k)(T_k) = (y_{k-1})(T_k)$ $k = 1, \ldots, K+1$, and similarly for the adjoint variables.

The time-domain-decomposition algorithm

Algorithm

This is equivalent to Lagnese/Leugering2002

1. Given $\mu^n_{k,k-1}, \mu^n_{k,k+1}$,

2. solve the restricted $\mathrm{OS}|_{I_k}$ for y_k^{n+1}, p_k^{n+1}

$$(y_k^{n+1})(T_{k+1}) + \sigma\beta'(y_k^{n+1})(p_k^{n+1})(T_{k+1}) = \mu^n_{k,k+1}, \quad \beta(y_k^{n+1})(T_k) - \sigma p_k^{n+1}(T_k) = \mu^n_{k,k-1}, \tag{1}$$

with

$$\begin{aligned} \mu^n_{k,k+1} &= (1-\varepsilon)\left(\beta(y^n_{k+1})(T_{k+1}) + \sigma p^n_{k+1}(T_{k+1})\right) + \varepsilon\left(y_k^{n+1}(T_{k+1}) + \sigma\beta'(y_k^{n+1})(p_k^{n+1})(T_{k+1})\right), \ k = 0, \dots, \\ \mu^n_{k,k-1} &= (1-\varepsilon)\left(y^n_{k-1}(T_k) - \sigma\beta'(y^n_{k-1})p^n_{k-1}(T_k)\right) + \varepsilon\left(\beta(y_k^{n+1})(T_k) - \sigma p_k^{n+1}(T_k)\right), \ k = 1, \dots, K. \end{aligned} \tag{2}$$

3. Update $\mu^{n+1}_{k,k-1}, \mu^{n+1}_{k,k+1}$ for $n \to n+1$.

$\epsilon \in [0, 1)$ is a relaxation parameter

Virtual control problem

The corresponding virtual optimal control problem for the generic interval I_k reads as follows. With

$$J_k^n(u_k, y_k, h_{k,k-1}) := \frac{\kappa}{2}\int_{T_k}^{T_{k+1}}\int_0^\ell (y_k - y_k^d)^2 dx dx dt + \frac{\nu}{2}\int_{T_k}^{T_{k+1}}\int_0^\ell {u_k}^2 dx dt + \frac{1}{2\sigma}\int_0^\ell \left((y_k(T_{k+1}) - \mu_{k,k+1})^2 + (h_{k,k-1})^2\right) dx$$

we have

$$\begin{aligned} &\min_{u_k, y_k, h_{k,k-1}} \quad J_k^n(u_k, y_k, h_{k,k-1}) \\ &\qquad\qquad \text{s. t.} \\ &\partial_t \beta_k(y_k) - \partial_x(\beta_k(\partial_x(y_k))) = u_k, \quad \text{in } (T_k, T_{k+1}) \times (0,\ell) \\ &\qquad \beta_k(y_k)(T_k) = h_{k,k-1} + \mu_{k,k-1}^n, \quad \text{in } (0,\ell), \end{aligned}$$

where $h_{k,k-1}$ serves as the *virtual control.*

Virtual control problem: first interval

This system has to be complemented by the problems on the first and the last interval.

$$\min_{u_0, y_0} J_0^n(u_0, y_0) := \frac{\kappa}{2} \int_{T_0}^{T_1} \int_0^{\ell} (y_0 - y_0^d)^2 dx dt + \frac{\nu}{2} \int_{T_0}^{T_1} \int_0^{\ell} u_0^2 dx dt$$

$$+ \frac{1}{2\sigma} \int_0^{\ell} (y_0(T_1) - \mu_{0,1})^2 dx$$

s. t.

$$\partial_t \beta_0(y_0) - \partial_x(\beta_0(\partial_x(y_0))) = u_0, \quad \text{in } (T_0, T_1) \times (0, \ell)$$

$$\beta_k(y_0)(T_0) = y_0, \quad \text{in } (0, \ell),$$

Virtual control problem: last interval

$$\min_{u_K, y_K, h_{K,K-1}} J_K^n(u_K, y_K) := \frac{\kappa}{2} \int_{T_K}^{T_{K+1}} \int_0^{\ell} (y_K - y_K^d)^2 dx dt + \frac{\kappa}{2} \int_0^{\ell} (y_K(T_{K+1}) - y_T^d)^2 dx$$

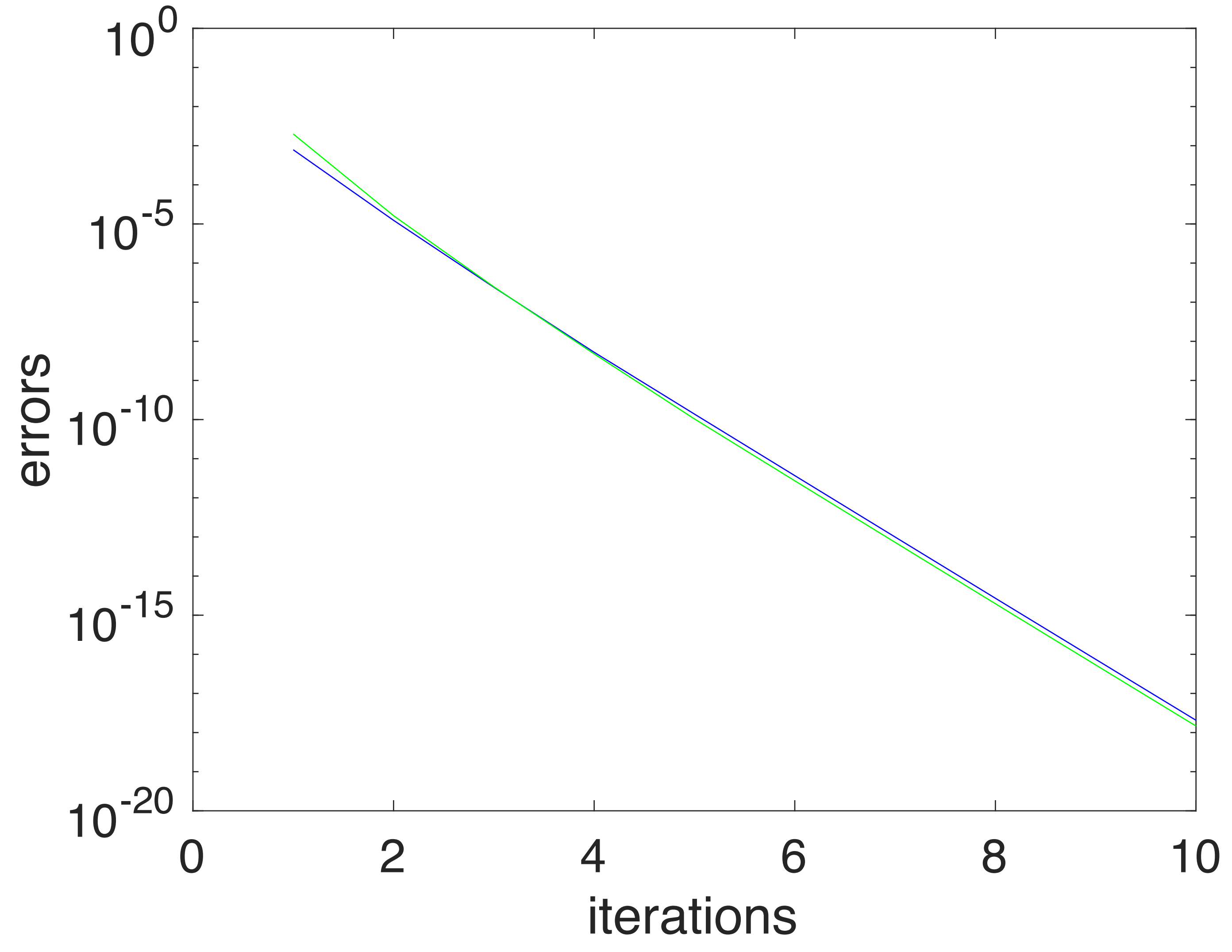
$$\frac{\nu}{2} \int_{T_K}^{T_{K+1}} \int_0^{\ell} u_K^2 dx dt + \frac{1}{2\sigma} \int_0^{\ell} \left((y_K(T_K) - \mu_{K,K+1})^2 + h_{K,K-1}^2 \right) dx$$

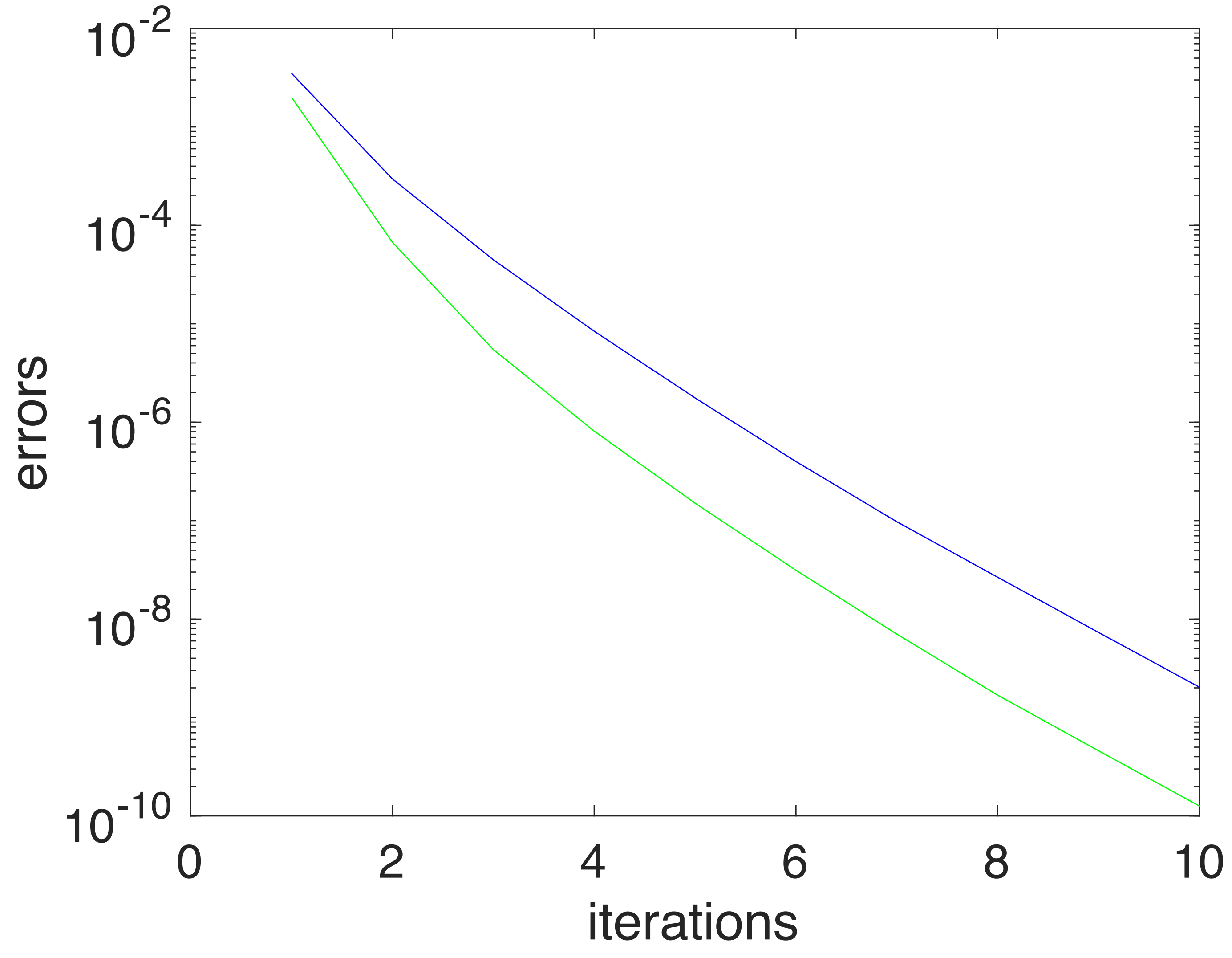
s. t.

$$\partial_t \beta_K(y_K) - \partial_x(\beta_K(\partial_x(y_K))) = u_K, \quad \text{in } (T_K, T_{K+1}) \times (0, \ell)$$

$$\beta_k(y_K)(T_K) = \mu_{K,K-1}^n + h_{K,K-1}, \quad \text{in } (0, \ell),$$

TDD via virtual controls





DEEP-PINN-DDM

Sturm-Liouville problem on an interval: Robin-Robin-DDM

• 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.
• In this talk, we apply PINN to the P.L. Lions Robin-Robin-type approach to a two-link problem.
• 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.*

Implementation (simplistic first approach)

Specification of training parameters

$Ni = 201;$ %Number of grid points for the solution domain$[0, 1]$

$Ne = 2000;$ %#of Epochs (1 Epoch contains Tb training batches)

$Tb = 600;$ %#of training batches (# or corrections during 1 Epoch)

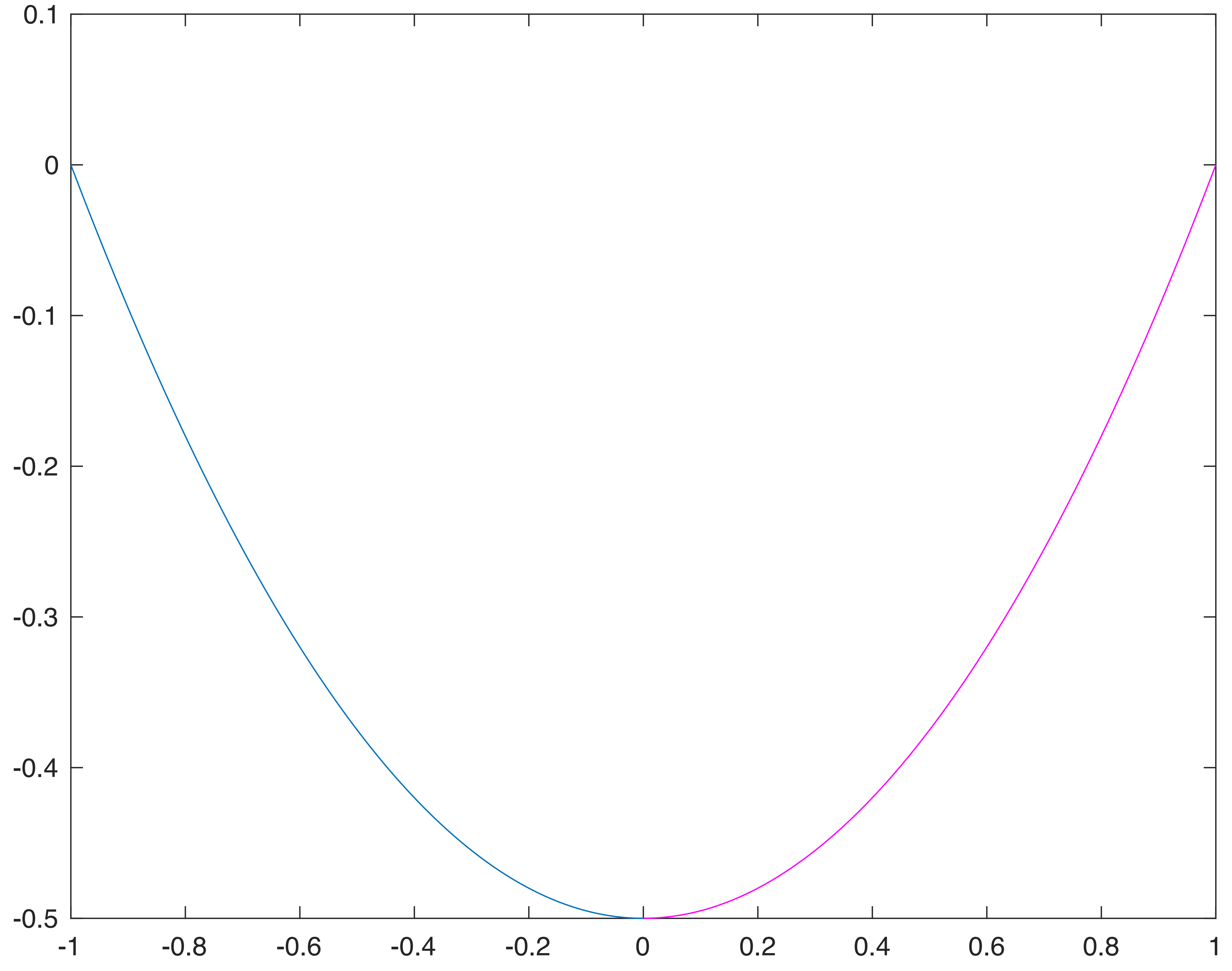
$lr = 0.005;$ %Learning rate coefficient (relaxation for the update)

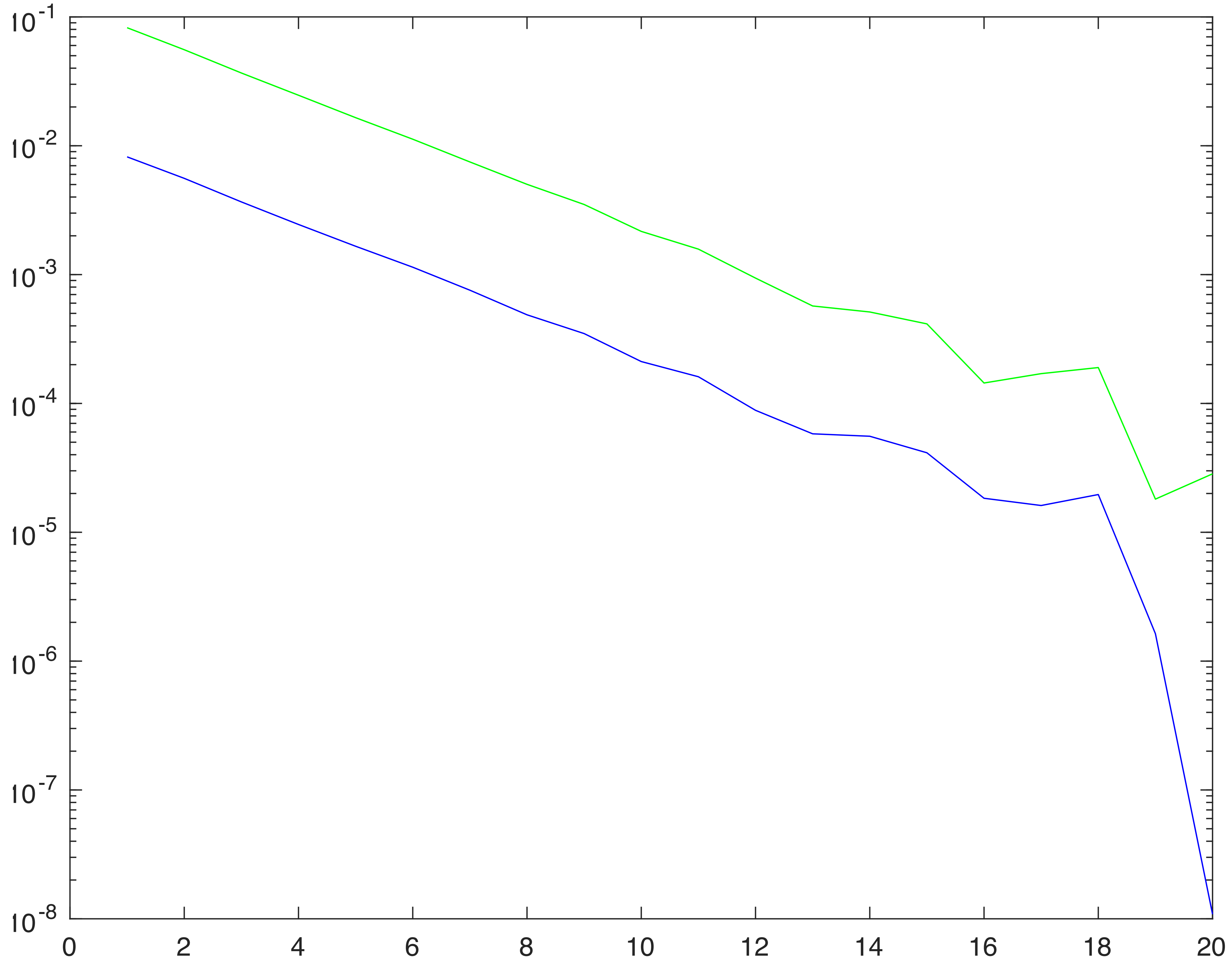
$Nn = 10;$ %Number of nodes in the 1st hidden layer

$Tt = 1e - 30;$ %Training tolerance N.B. redundant in the current version

```
Code for the Sturm–Liouville problem on a single interval developed from a MATLAB
code by
Andreas Almquist
Luleå University of Technology
Departement of Machine Elements
2020-01-01
```

Numerical results for Deep-PINN





Further results and outlook

1. We have a similar result for the time-domain-decomposition problem (again, the proof only for for $\alpha = 2$)
2. The simultaneous space-time-domain decomposition is open (fine for the p-elliptic case)
3. The (β_α, β_p)-problem is open (as far as the proof is concerned)
4. Constrained control can be included, however, this has not yet been proved (just a matter of writing it up)
5. State constraints are completely open.
6. One may use PINN (XPINN) on subnetworks as surrogate models and perform interface learning (in preparation)
7. Final goal: **Ne**twork **T**earing and **I**nterconnection, a formal analogue of FETI.

Thank you for your attention!