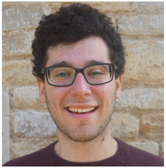
Hierarchical proximal Galerkin: a fast *hp*-FEM solver for variational problems with pointwise inequality constraints

John Papadopoulos[1]

[1] Weierstrass Institute Berlin,

March 13, 2025, WIAS Group 3 Seminar, Berlin



Introduction

Pointwise constraints appear everywhere, e.g. contact mechanics (non-penetration), stress constraints in elasticity, sandpile growth, financial mathematics, pattern formation, engineering design, biological models...

Obstacle problem

Given a forcing term $f \in L^2(\Omega)$ and an obstacle $\varphi \in H^1(\Omega)$, the obstacle problem seeks $u : \Omega \to \mathbb{R}$ minimizing the Dirichlet energy

$$\min_{u \in H_0^1(\Omega)} \int_\Omega \frac{1}{2}|\nabla u|^2 - fu \, \mathrm{d}x \quad \text{subject to } u(x) \leq \varphi(x) \text{ for almost every } x \in \Omega.$$

- primal-dual active set, multigrid, finite-dimensional constrained optimizers (often mesh dependent, confined to low-order[1]).
- penalty methods (infeasible solutions, suboptimal for high-order, ill-conditioning).

[1] With notable exceptions in Kirby & Shapero (2024) and Banz & Schröder (2015).

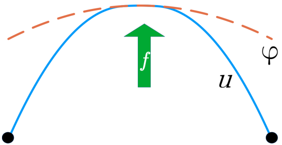
Introduction

Pointwise constraints appear everywhere, e.g. contact mechanics (non-penetration), stress constraints in elasticity, sandpile growth, financial mathematics, pattern formation, engineering design, biological models...

Obstacle problem

Given a forcing term $f \in L^2(\Omega)$ and an obstacle $\varphi \in H^1(\Omega)$, the obstacle problem seeks $u : \Omega \to \mathbb{R}$ minimizing the Dirichlet energy

$$\min_{u \in H_0^1(\Omega)} \int_\Omega \frac{1}{2}|\nabla u|^2 - fu \,\mathrm{d}x \text{ subject to } u(x) \leq \varphi(x) \text{ for almost every } x \in \Omega.$$



- primal-dual active set, multigrid, finite-dimensional constrained optimizers (often mesh dependent, confined to low-order[1]).
- penalty methods (infeasible solutions, suboptimal for high-order, ill-conditioning).

[1] With notable exceptions in Kirby & Shapero (2024) and Banz & Schröder (2015).

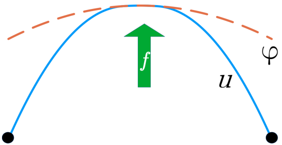
Introduction

Pointwise constraints appear everywhere, e.g. contact mechanics (non-penetration), stress constraints in elasticity, sandpile growth, financial mathematics, pattern formation, engineering design, biological models...

Obstacle problem

Given a forcing term $f \in L^2(\Omega)$ and an obstacle $\varphi \in H^1(\Omega)$, the obstacle problem seeks $u : \Omega \to \mathbb{R}$ minimizing the Dirichlet energy

$$\min_{u \in H_0^1(\Omega)} \int_\Omega \frac{1}{2}|\nabla u|^2 - fu \,\mathrm{d}x \text{ subject to } u(x) \leq \varphi(x) \text{ for almost every } x \in \Omega.$$

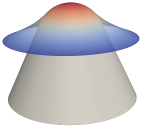


- primal-dual active set, multigrid, finite-dimensional constrained optimizers (often mesh dependent, confined to low-order[1]).
- penalty methods (infeasible solutions, suboptimal for high-order, ill-conditioning).

[1] With notable exceptions in Kirby & Shapero (2024) and Banz & Schröder (2015).

Latent variable proximal point (LVPP)

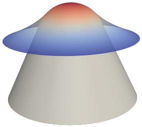
LVPP is a new and powerful framework for solving variational problems with pointwise constraints (`https://arxiv.org/abs/2503.05672`).



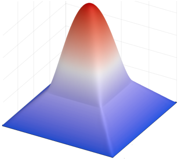
Obstacle, $u \leq \varphi$.

Latent variable proximal point (LVPP)

LVPP is a new and powerful framework for solving variational problems with pointwise constraints (`https://arxiv.org/abs/2503.05672`).



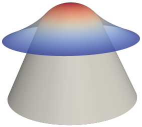
Obstacle, $u \leq \varphi$.



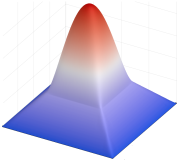
Gradient-type, $|\nabla u| \leq \varphi$.

Latent variable proximal point (LVPP)

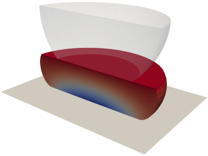
LVPP is a new and powerful framework for solving variational problems with pointwise constraints (`https://arxiv.org/abs/2503.05672`).



Obstacle, $u \leq \varphi$.



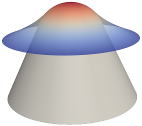
Gradient-type, $|\nabla u| \leq \varphi$.



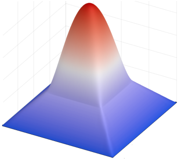
Signorini, $(u \cdot n)|_\Gamma \geq 0$.

Latent variable proximal point (LVPP)

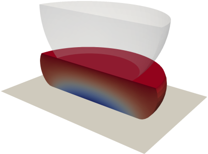
LVPP is a new and powerful framework for solving variational problems with pointwise constraints (`https://arxiv.org/abs/2503.05672`).



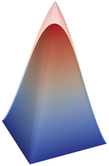
Obstacle, $u \leq \varphi$.



Gradient-type, $|\nabla u| \leq \varphi$.



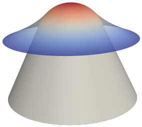
Signorini, $(u \cdot n)|_\Gamma \geq 0$.



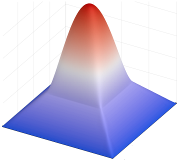
Thermoforming, $u \leq \varphi(u)$.

Latent variable proximal point (LVPP)

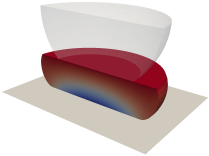
LVPP is a new and powerful framework for solving variational problems with pointwise constraints (https://arxiv.org/abs/2503.05672).



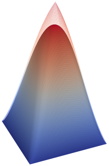
Obstacle, $u \leq \varphi$.



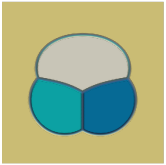
Gradient-type, $|\nabla u| \leq \varphi$.



Signorini, $(u \cdot n)|_\Gamma \geq 0$.



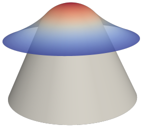
Thermoforming, $u \leq \varphi(u)$.



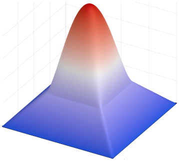
Cahn–Hilliard,
$u_i \geq 0$, $\sum_i u_i = 1$.

Latent variable proximal point (LVPP)

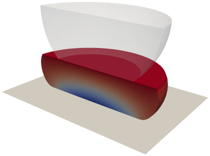
LVPP is a new and powerful framework for solving variational problems with pointwise constraints (`https://arxiv.org/abs/2503.05672`).



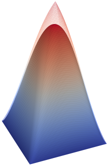
Obstacle, $u \leq \varphi$.



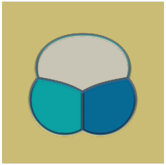
Gradient-type, $|\nabla u| \leq \varphi$.



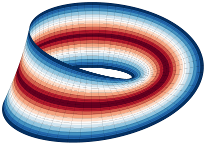
Signorini, $(u \cdot n)|_\Gamma \geq 0$.



Thermoforming, $u \leq \varphi(u)$.



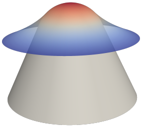
Cahn–Hilliard,
$u_i \geq 0$, $\sum_i u_i = 1$.



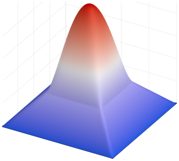
Eikonal, $|\nabla u| = 1$.

Latent variable proximal point (LVPP)

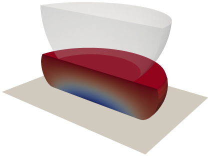
LVPP is a new and powerful framework for solving variational problems with pointwise constraints (`https://arxiv.org/abs/2503.05672`).



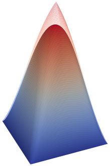
Obstacle, $u \leq \varphi$.



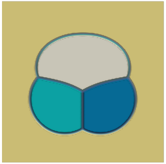
Gradient-type, $|\nabla u| \leq \varphi$.



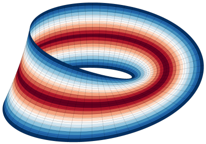
Signorini, $(u \cdot n)|_\Gamma \geq 0$.



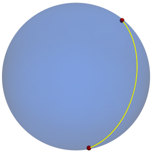
Thermoforming, $u \leq \varphi(u)$.



Cahn–Hilliard, $u_i \geq 0$, $\sum_i u_i = 1$.



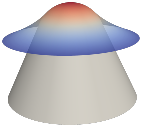
Eikonal, $|\nabla u| = 1$.



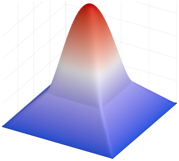
Harmonic map, $|u| = 1$.

Latent variable proximal point (LVPP)

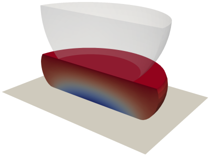
LVPP is a new and powerful framework for solving variational problems with pointwise constraints (`https://arxiv.org/abs/2503.05672`).



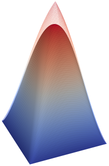
Obstacle, $u \leq \varphi$.



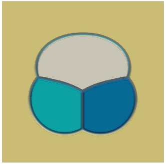
Gradient-type, $|\nabla u| \leq \varphi$.



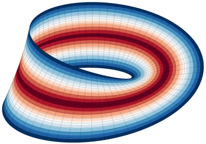
Signorini, $(u \cdot n)|_\Gamma \geq 0$.



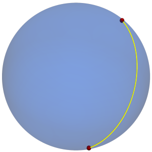
Thermoforming, $u \leq \varphi(u)$.



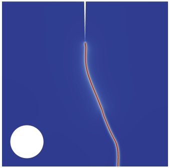
Cahn–Hilliard, $u_i \geq 0$, $\sum_i u_i = 1$.



Eikonal, $|\nabla u| = 1$.



Harmonic map, $|u| = 1$.



Fracture.

Contact problems

Contact problems

Problems of interest

Consider the constrained optimization problem:

$$\min_{u \in U} J(u) \ \text{ subject to } \ Bu(x) \in C(x) \text{ for a.e. } x \in \Omega.$$

Examples

- (Obstacle problem.) Find $u : \Omega \to \mathbb{R}$

$$\min_{u \in H_0^1(\Omega)} \int_\Omega \frac{1}{2}|\nabla u|^2 - fu \,\mathrm{d}x \ \text{ subject to } u(x) \leq \varphi(x).$$

- (Elastic-plastic torsion.) Find $u : \Omega \to \mathbb{R}$,

$$\min_{u \in H_0^1(\Omega)} \int_\Omega \frac{1}{2}|\nabla u|^2 - fu \,\mathrm{d}x \ \text{ subject to } |\nabla u|(x) \leq \varphi(x).$$

Problems of interest

Consider the constrained optimization problem:

$$\min_{u \in U} J(u) \quad \text{subject to} \quad Bu(x) \in C(x) \text{ for a.e. } x \in \Omega.$$

Examples

- (Obstacle problem.) Find $u : \Omega \to \mathbb{R}$

$$\min_{u \in H_0^1(\Omega)} \int_\Omega \frac{1}{2}|\nabla u|^2 - fu \, \mathrm{d}x \quad \text{subject to } u(x) \leq \varphi(x).$$

- (Elastic-plastic torsion.) Find $u : \Omega \to \mathbb{R}$,

$$\min_{u \in H_0^1(\Omega)} \int_\Omega \frac{1}{2}|\nabla u|^2 - fu \, \mathrm{d}x \quad \text{subject to } |\nabla u|(x) \leq \varphi(x).$$

Problems of interest

Consider the constrained optimization problem:

$$\min_{u \in U} J(u) \;\text{ subject to }\; Bu(x) \in C(x) \text{ for a.e. } x \in \Omega.$$

Examples

- (Obstacle problem.) Find $u : \Omega \to \mathbb{R}$

$$\min_{u \in H_0^1(\Omega)} \int_\Omega \frac{1}{2}|\nabla u|^2 - fu \,\mathrm{d}x \;\text{ subject to } u(x) \leq \varphi(x).$$

- (Elastic-plastic torsion.) Find $u : \Omega \to \mathbb{R}$,

$$\min_{u \in H_0^1(\Omega)} \int_\Omega \frac{1}{2}|\nabla u|^2 - fu \,\mathrm{d}x \;\text{ subject to } |\nabla u|(x) \leq \varphi(x).$$

The LVPP algorithm

Obstacle problem

Given $\psi^{k-1} \in L^\infty(\Omega)$, for $k = 1, 2, \dots,$ we seek (u^k, ψ^k) satisfying

$$-\alpha_k \Delta u^k + \psi^k = \alpha_k f + \psi^{k-1},$$
$$u^k + \mathrm{e}^{-\psi^k} = \varphi.$$

Theorem (B. Keith, T. Surowiec, FoCM, 2024)

Suppose that Ω is an open, bounded and Lipschitz domain and $\varphi \in \{\phi \in H^1(\Omega) \cap C(\bar{\Omega}) : \Delta\phi \in L^\infty(\Omega)\}$, then

$$\|u^* - u^k\|_{H^1(\Omega)} \lesssim \left(\sum_{j=1}^{k} \alpha_j\right)^{-1/2}.$$

Note that $u^k \to u^*$ in $H^1(\Omega)$ even if $\alpha_k = 1$ for all $k \in \mathbb{N}$.

The LVPP algorithm

Obstacle problem

Given $\psi^{k-1} \in L^\infty(\Omega)$, for $k = 1, 2, \ldots,$ we seek (u^k, ψ^k) satisfying

$$-\alpha_k \Delta u^k + \psi^k = \alpha_k f + \psi^{k-1},$$
$$u^k + \mathrm{e}^{-\psi^k} = \varphi.$$

Theorem (B. Keith, T. Surowiec, FoCM, 2024)

Suppose that Ω is an open, bounded and Lipschitz domain and $\varphi \in \{\phi \in H^1(\Omega) \cap C(\bar{\Omega}) : \Delta\phi \in L^\infty(\Omega)\}$, then

$$\|u^* - u^k\|_{H^1(\Omega)} \lesssim \left(\sum_{j=1}^{k} \alpha_j\right)^{-1/2}.$$

Note that $u^k \to u^*$ in $H^1(\Omega)$ even if $\alpha_k = 1$ for all $k \in \mathbb{N}$.

The LVPP algorithm

Elastic-plastic torsion problem

Given ψ^{k-1}, for $k = 1, 2, \dots,$ we seek (u^k, ψ^k) satisfying

$$-\alpha_k \Delta u^k - \operatorname{div} \psi^k = \alpha_k f - \operatorname{div} \psi^{k-1},$$

$$u^k - \frac{\varphi \psi^k}{\sqrt{1 + |\psi^k|^2}} = 0.$$

$$\min_{u \in U} J(u) \quad \text{subject to} \quad Bu(x) \in C(x) \text{ for a.e. } x \in \Omega.$$

General LVPP subproblems

Given ψ^{k-1}, for $k = 1, 2, \dots,$ we seek (u^k, ψ^k) satisfying

$$\alpha_k J'(u_k) + B^* \psi^k = B^* \psi^{k-1} \text{ in } U^*$$

$$Bu^k - G(\psi^k) = 0 \text{ a.e.},$$

where $\sum_{j=1}^{\infty} \alpha_j \to \infty$ and G is a pointwise operator chosen such that $G^{-1}(Bu)(x) \to \infty$ as $Bu(x) \to \partial C(x)$.

E.g. $G(\psi) = \varphi - \mathrm{e}^{-\psi} \implies G^{-1}(\mathrm{id}\, u)(x) = -\log(\varphi(x) - u(x)) \to \infty$ as $u(x) \to \varphi(x)$.

The LVPP algorithm

Elastic-plastic torsion problem

Given ψ^{k-1}, for $k = 1, 2, \ldots$, we seek (u^k, ψ^k) satisfying

$$-\alpha_k \Delta u^k - \operatorname{div} \psi^k = \alpha_k f - \operatorname{div} \psi^{k-1},$$

$$u^k - \frac{\varphi \psi^k}{\sqrt{1 + |\psi^k|^2}} = 0.$$

$$\min_{u \in U} J(u) \text{ subject to } Bu(x) \in C(x) \text{ for a.e. } x \in \Omega.$$

General LVPP subproblems

Given ψ^{k-1}, for $k = 1, 2, \ldots$, we seek (u^k, ψ^k) satisfying

$$\alpha_k J'(u_k) + B^* \psi^k = B^* \psi^{k-1} \text{ in } U^*$$

$$Bu^k - G(\psi^k) = 0 \text{ a.e.},$$

where $\sum_{j=1}^{\infty} \alpha_j \to \infty$ and G is a pointwise operator chosen such that $G^{-1}(Bu)(x) \to \infty$ as $Bu(x) \to \partial C(x)$.

E.g. $G(\psi) = \varphi - \mathrm{e}^{-\psi} \implies G^{-1}(\mathrm{id}\, u)(x) = -\log(\varphi(x) - u(x)) \to \infty$ as $u(x) \to \varphi(x)$.

The LVPP algorithm

Elastic-plastic torsion problem

Given ψ^{k-1}, for $k = 1, 2, \ldots$, we seek (u^k, ψ^k) satisfying

$$-\alpha_k \Delta u^k - \operatorname{div} \psi^k = \alpha_k f - \operatorname{div} \psi^{k-1},$$

$$u^k - \frac{\varphi \psi^k}{\sqrt{1 + |\psi^k|^2}} = 0.$$

$$\min_{u \in U} J(u) \;\text{ subject to }\; Bu(x) \in C(x) \text{ for a.e. } x \in \Omega.$$

General LVPP subproblems

Given ψ^{k-1}, for $k = 1, 2, \ldots$, we seek (u^k, ψ^k) satisfying

$$\alpha_k J'(u_k) + B^* \psi^k = B^* \psi^{k-1} \text{ in } U^*$$

$$Bu^k - G(\psi^k) = 0 \text{ a.e.},$$

where $\sum_{j=1}^{\infty} \alpha_j \to \infty$ and G is a pointwise operator chosen such that $G^{-1}(Bu)(x) \to \infty$ as $Bu(x) \to \partial C(x)$.

E.g. $G(\psi) = \varphi - \mathrm{e}^{-\psi} \implies G^{-1}(\mathrm{id}\, u)(x) = -\log(\varphi(x) - u(x)) \to \infty$ as $u(x) \to \varphi(x)$.

The LVPP algorithm

Elastic-plastic torsion problem

Given ψ^{k-1}, for $k = 1, 2, \dots,$ we seek (u^k, ψ^k) satisfying

$$-\alpha_k \Delta u^k - \operatorname{div} \psi^k = \alpha_k f - \operatorname{div} \psi^{k-1},$$
$$u^k - \frac{\varphi \psi^k}{\sqrt{1 + |\psi^k|^2}} = 0.$$

$$\min_{u \in U} J(u) \;\text{ subject to }\; Bu(x) \in C(x) \text{ for a.e. } x \in \Omega.$$

General LVPP subproblems

Given ψ^{k-1}, for $k = 1, 2, \dots,$ we seek (u^k, ψ^k) satisfying

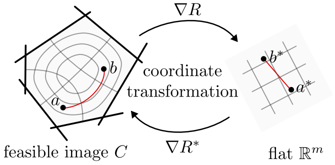
$$\alpha_k J'(u_k) + B^* \psi^k = B^* \psi^{k-1} \text{ in } U^*$$
$$Bu^k - G(\psi^k) = 0 \text{ a.e.,}$$

where $\sum_{j=1}^{\infty} \alpha_j \to \infty$ and G is a pointwise operator chosen such that $G^{-1}(Bu)(x) \to \infty$ as $Bu(x) \to \partial C(x)$.

E.g. $G(\psi) = \varphi - \mathrm{e}^{-\psi} \implies G^{-1}(\operatorname{id} u)(x) = -\log(\varphi(x) - u(x)) \to \infty$ as $u(x) \to \varphi(x)$.

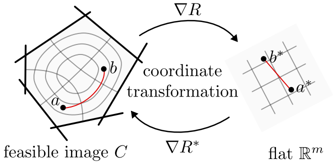
Advantages of the LVPP algorithm

1. It has an infinite-dimensional formulation.
2. Observed discretization-independent number of linear system solves.
3. A simple mechanism for enforcing pointwise constraints on the discrete level (without the need for a projection).
4. Ease of implementation — the algorithm reduces to the repeated solve of a smooth nonlinear system of PDEs *without requiring specialized discretizations*.
5. Robust numerical performance since convergence occurs as α_k can be kept small.



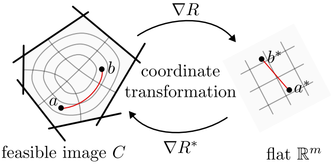
Advantages of the LVPP algorithm

1. It has an infinite-dimensional formulation.
2. Observed discretization-independent number of linear system solves.
3. A simple mechanism for enforcing pointwise constraints on the discrete level (without the need for a projection).
4. Ease of implementation — the algorithm reduces to the repeated solve of a smooth nonlinear system of PDEs *without requiring specialized discretizations*.
5. Robust numerical performance since convergence occurs as α_k can be kept small.



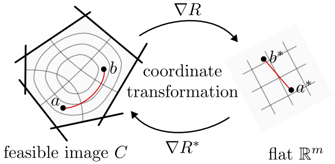
Advantages of the LVPP algorithm

1. It has an infinite-dimensional formulation.
2. Observed discretization-independent number of linear system solves.
3. A simple mechanism for enforcing pointwise constraints on the discrete level (without the need for a projection).
4. Ease of implementation — the algorithm reduces to the repeated solve of a smooth nonlinear system of PDEs *without requiring specialized discretizations*.
5. Robust numerical performance since convergence occurs as α_k can be kept small.



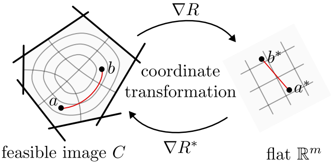
Advantages of the LVPP algorithm

1. It has an infinite-dimensional formulation.
2. Observed discretization-independent number of linear system solves.
3. A simple mechanism for enforcing pointwise constraints on the discrete level (without the need for a projection).
4. Ease of implementation — the algorithm reduces to the repeated solve of a smooth nonlinear system of PDEs *without requiring specialized discretizations*.
5. Robust numerical performance since convergence occurs as α_k can be kept small.



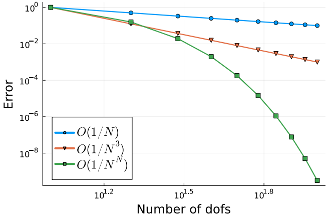
Advantages of the LVPP algorithm

1. It has an infinite-dimensional formulation.
2. Observed discretization-independent number of linear system solves.
3. A simple mechanism for enforcing pointwise constraints on the discrete level (without the need for a projection).
4. Ease of implementation — the algorithm reduces to the repeated solve of a smooth nonlinear system of PDEs *without requiring specialized discretizations*.
5. Robust numerical performance since convergence occurs as α_k can be kept small.



High-order finite element methods

A “high-order” discretization is one where we are approximating the solution with piecewise polynomials of high degree, e.g. $p \geq 4$.



Challenges

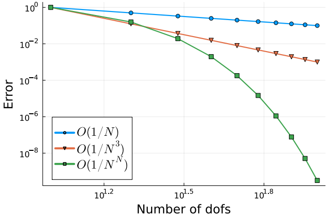
Naive implementations lead to slow quadrature, dense linear systems, capped convergence and, therefore, slow solve times.

Our proposal

Utilize a sparsity-promoting high-order basis that admits fast quadrature via the FFT.

High-order finite element methods

A “high-order” discretization is one where we are approximating the solution with piecewise polynomials of high degree, e.g. $p \geq 4$.



Challenges

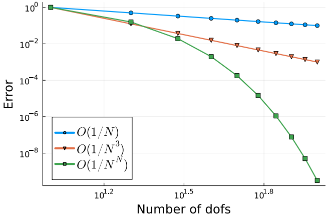
Naive implementations lead to slow quadrature, dense linear systems, capped convergence and, therefore, slow solve times.

Our proposal

Utilize a sparsity-promoting high-order basis that admits fast quadrature via the FFT.

High-order finite element methods

A “high-order” discretization is one where we are approximating the solution with piecewise polynomials of high degree, e.g. $p \geq 4$.



Challenges

Naive implementations lead to slow quadrature, dense linear systems, capped convergence and, therefore, slow solve times.

Our proposal

Utilize a sparsity-promoting high-order basis that admits fast quadrature via the FFT.

Weak form and a finite element discretization

Weak form of LVPP for the obstacle problem

The k^{th} LVPP subproblem seeks $(u^k, \psi^k) \in H_0^1(\Omega) \times L^\infty(\Omega)$ satisfying for all $(v, q) \in H_0^1(\Omega) \times L^\infty(\Omega)$:

$$\alpha_k(\nabla u^k, \nabla v) + (\psi^k, v) = \alpha_k(f, v) + (\psi^{k-1}, v)$$
$$(u^k, q) + (\mathrm{e}^{-\psi^k}, q) = (\varphi, q).$$

FEM discretization

Pick finite-dimensional spaces $V_{hp} \subset H_0^1(\Omega)$, $Q_{hp} \subset L^\infty(\Omega)$ and seek $(u_{hp}^k, \psi_{hp}^k) \in V_{hp} \times Q_{hp}$ satisfying for all $(v_{hp}, q_{hp}) \in V_{hp} \times Q_{hp}$:

$$\alpha_k(\nabla u_{hp}^k, \nabla v_{hp}) + (\psi_{hp}^k, v_{hp}) = \alpha_k(f, v_{hp}) + (\psi_{hp}^{k-1}, v_{hp})$$
$$(u_{hp}^k, q_{hp}) + (\mathrm{e}^{-\psi_{hp}^k}, q_{hp}) = (\varphi, q_{hp}).$$

Nonlinear system of equations... use Newton!

Weak form and a finite element discretization

Weak form of LVPP for the obstacle problem

The k^{th} LVPP subproblem seeks $(u^k, \psi^k) \in H_0^1(\Omega) \times L^\infty(\Omega)$ satisfying for all $(v, q) \in H_0^1(\Omega) \times L^\infty(\Omega)$:

$$\alpha_k(\nabla u^k, \nabla v) + (\psi^k, v) = \alpha_k(f, v) + (\psi^{k-1}, v)$$
$$(u^k, q) + (\mathrm{e}^{-\psi^k}, q) = (\varphi, q).$$

FEM discretization

Pick finite-dimensional spaces $V_{hp} \subset H_0^1(\Omega)$, $Q_{hp} \subset L^\infty(\Omega)$ and seek $(u_{hp}^k, \psi_{hp}^k) \in V_{hp} \times Q_{hp}$ satisfying for all $(v_{hp}, q_{hp}) \in V_{hp} \times Q_{hp}$:

$$\alpha_k(\nabla u_{hp}^k, \nabla v_{hp}) + (\psi_{hp}^k, v_{hp}) = \alpha_k(f, v_{hp}) + (\psi_{hp}^{k-1}, v_{hp})$$
$$(u_{hp}^k, q_{hp}) + (\mathrm{e}^{-\psi_{hp}^k}, q_{hp}) = (\varphi, q_{hp}).$$

Nonlinear system of equations... use Newton!

Weak form and a finite element discretization

Weak form of LVPP for the obstacle problem

The k^{th} LVPP subproblem seeks $(u^k, \psi^k) \in H_0^1(\Omega) \times L^\infty(\Omega)$ satisfying for all $(v, q) \in H_0^1(\Omega) \times L^\infty(\Omega)$:

$$\alpha_k(\nabla u^k, \nabla v) + (\psi^k, v) = \alpha_k(f, v) + (\psi^{k-1}, v)$$
$$(u^k, q) + (\mathrm{e}^{-\psi^k}, q) = (\varphi, q).$$

FEM discretization

Pick finite-dimensional spaces $V_{hp} \subset H_0^1(\Omega)$, $Q_{hp} \subset L^\infty(\Omega)$ and seek $(u_{hp}^k, \psi_{hp}^k) \in V_{hp} \times Q_{hp}$ satisfying for all $(v_{hp}, q_{hp}) \in V_{hp} \times Q_{hp}$:

$$\alpha_k(\nabla u_{hp}^k, \nabla v_{hp}) + (\psi_{hp}^k, v_{hp}) = \alpha_k(f, v_{hp}) + (\psi_{hp}^{k-1}, v_{hp})$$
$$(u_{hp}^k, q_{hp}) + (\mathrm{e}^{-\psi_{hp}^k}, q_{hp}) = (\varphi, q_{hp}).$$

Nonlinear system of equations... use Newton!

Newton linear systems

In matrix-vector form we are solving

$$\begin{pmatrix} \alpha_k A & B \\ B^\top & -D_{\psi^k} \end{pmatrix} \begin{pmatrix} \boldsymbol{\delta}_u \\ \boldsymbol{\delta}_\psi \end{pmatrix} = \begin{pmatrix} \boldsymbol{b}_u \\ \boldsymbol{b}_\psi \end{pmatrix},$$

where for basis function $\phi_i \in V_{hp}$ and $\zeta_i \in Q_{hp}$,

$$A_{ij} = (\nabla \phi_i, \nabla \phi_j), \quad B_{ij} = (\phi_i, \zeta_j), \quad \text{and} \quad [D_\psi]_{ij} = (\zeta_i, \mathrm{e}^{-\psi_{hp}} \zeta_j).$$

Goal

Pick FEM bases $\{\phi_i\} \subset V_{hp}$ and $\{\zeta_j\} \subset Q_{hp}$ that contain high-degree polynomials but also

- Keep A, B and D_ψ sparse.
- Allow for fast assembly or action of D_ψ.

💡 use a discontinuous piecewise Legendre polynomial basis for ψ_{hp} and the (Babuška–Szabó) hierarchical continuous p-FEM basis for u_{hp}.

Newton linear systems

In matrix-vector form we are solving

$$\begin{pmatrix} \alpha_k A & B \\ B^\top & -D_{\psi^k} \end{pmatrix} \begin{pmatrix} \boldsymbol{\delta}_u \\ \boldsymbol{\delta}_\psi \end{pmatrix} = \begin{pmatrix} \boldsymbol{b}_u \\ \boldsymbol{b}_\psi \end{pmatrix},$$

where for basis function $\phi_i \in V_{hp}$ and $\zeta_i \in Q_{hp}$,

$$A_{ij} = (\nabla \phi_i, \nabla \phi_j), \quad B_{ij} = (\phi_i, \zeta_j), \quad \text{and} \quad [D_\psi]_{ij} = (\zeta_i, \mathrm{e}^{-\psi_{hp}} \zeta_j).$$

Goal

Pick FEM bases $\{\phi_i\} \subset V_{hp}$ and $\{\zeta_j\} \subset Q_{hp}$ that contain high-degree polynomials but also

- Keep A, B and D_ψ sparse.
- Allow for fast assembly or action of D_ψ.

💡 use a discontinuous piecewise Legendre polynomial basis for ψ_{hp} and the (Babuška–Szabó) hierarchical continuous p-FEM basis for u_{hp}.

Newton linear systems

In matrix-vector form we are solving

$$\begin{pmatrix} \alpha_k A & B \\ B^\top & -D_{\psi^k} \end{pmatrix} \begin{pmatrix} \boldsymbol{\delta}_u \\ \boldsymbol{\delta}_\psi \end{pmatrix} = \begin{pmatrix} \boldsymbol{b}_u \\ \boldsymbol{b}_\psi \end{pmatrix},$$

where for basis function $\phi_i \in V_{hp}$ and $\zeta_i \in Q_{hp}$,

$$A_{ij} = (\nabla \phi_i, \nabla \phi_j), \quad B_{ij} = (\phi_i, \zeta_j), \quad \text{and} \quad [D_\psi]_{ij} = (\zeta_i, \mathrm{e}^{-\psi_{hp}} \zeta_j).$$

Goal

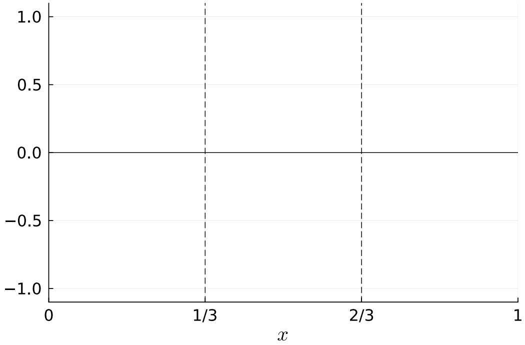
Pick FEM bases $\{\phi_i\} \subset V_{hp}$ and $\{\zeta_j\} \subset Q_{hp}$ that contain high-degree polynomials but also

- Keep A, B and D_ψ sparse.
- Allow for fast assembly or action of D_ψ.

💡 use a discontinuous piecewise Legendre polynomial basis for ψ_{hp} and the (Babuška–Szabó) hierarchical continuous p-FEM basis for u_{hp}.

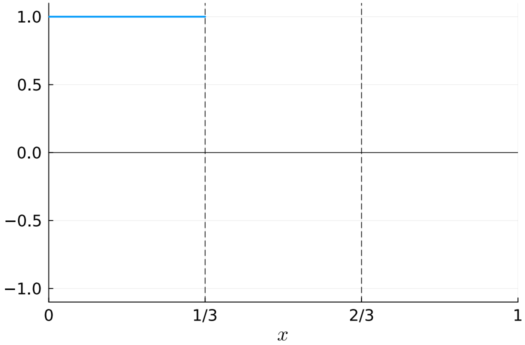
The Legendre polynomials

The Legendre polynomials $P_n(x)$, $n \in \mathbb{N}_0$ satisfy $\int_{-1}^{1} P_n P_m \, \mathrm{d}x \simeq \delta_{nm}$.
We can shift-and-scale the polynomials to construct a 1D basis such that $(\zeta_i, \zeta_j) \simeq \delta_{nm}$ for all basis functions $\zeta_i \in Q_{hp}$. This basis has *fast* transforms.



The Legendre polynomials

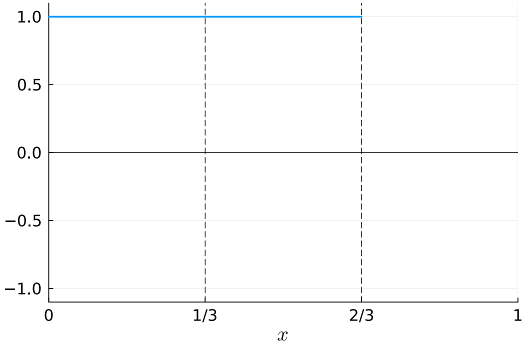
The Legendre polynomials $P_n(x)$, $n \in \mathbb{N}_0$ satisfy $\int_{-1}^{1} P_n P_m \,\mathrm{d}x \simeq \delta_{nm}$.
We can shift-and-scale the polynomials to construct a 1D basis such that $(\zeta_i, \zeta_j) \simeq \delta_{nm}$ for all basis functions $\zeta_i \in Q_{hp}$. This basis has *fast* transforms.



Shift-and-scale constant $P_0(x)$ on each cell.

The Legendre polynomials

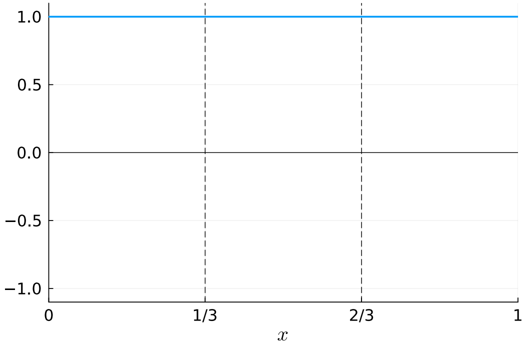
The Legendre polynomials $P_n(x)$, $n \in \mathbb{N}_0$ satisfy $\int_{-1}^{1} P_n P_m \, \mathrm{d}x \simeq \delta_{nm}$.
We can shift-and-scale the polynomials to construct a 1D basis such that $(\zeta_i, \zeta_j) \simeq \delta_{nm}$ for all basis functions $\zeta_i \in Q_{hp}$. This basis has *fast* transforms.



Shift-and-scale constant $P_0(x)$ on each cell.

The Legendre polynomials

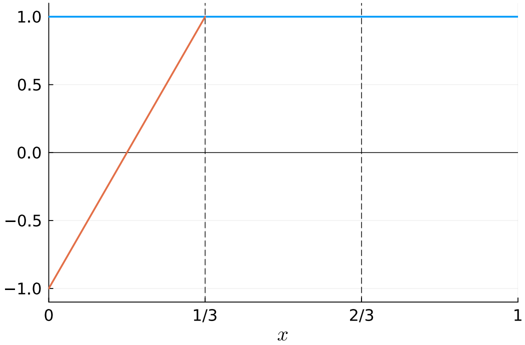
The Legendre polynomials $P_n(x)$, $n \in \mathbb{N}_0$ satisfy $\int_{-1}^{1} P_n P_m \, \mathrm{d}x \simeq \delta_{nm}$.
We can shift-and-scale the polynomials to construct a 1D basis such that $(\zeta_i, \zeta_j) \simeq \delta_{nm}$ for all basis functions $\zeta_i \in Q_{hp}$. This basis has *fast* transforms.



Shift-and-scale constant $P_0(x)$ on each cell.

The Legendre polynomials

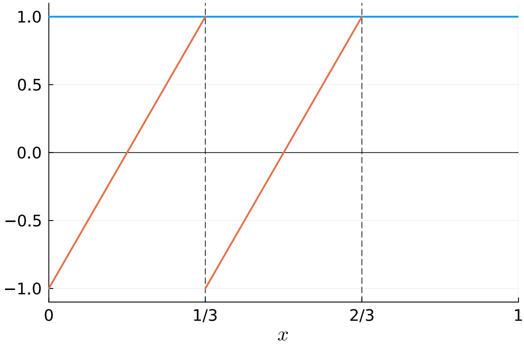
The Legendre polynomials $P_n(x)$, $n \in \mathbb{N}_0$ satisfy $\int_{-1}^{1} P_n P_m \,\mathrm{d}x \simeq \delta_{nm}$.
We can shift-and-scale the polynomials to construct a 1D basis such that $(\zeta_i, \zeta_j) \simeq \delta_{nm}$ for all basis functions $\zeta_i \in Q_{hp}$. This basis has *fast* transforms.



Shift-and-scale linear polynomial $P_1(x)$.

The Legendre polynomials

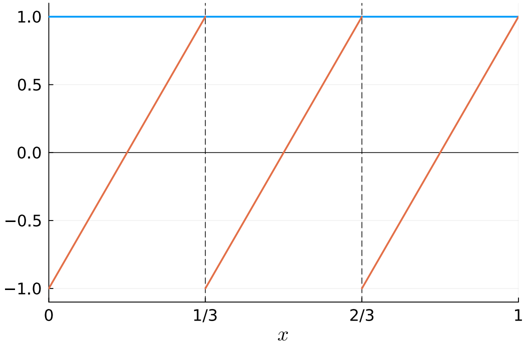
The Legendre polynomials $P_n(x)$, $n \in \mathbb{N}_0$ satisfy $\int_{-1}^{1} P_n P_m \,\mathrm{d}x \simeq \delta_{nm}$.
We can shift-and-scale the polynomials to construct a 1D basis such that $(\zeta_i, \zeta_j) \simeq \delta_{nm}$ for all basis functions $\zeta_i \in Q_{hp}$. This basis has *fast* transforms.



Shift-and-scale linear polynomial $P_1(x)$.

The Legendre polynomials

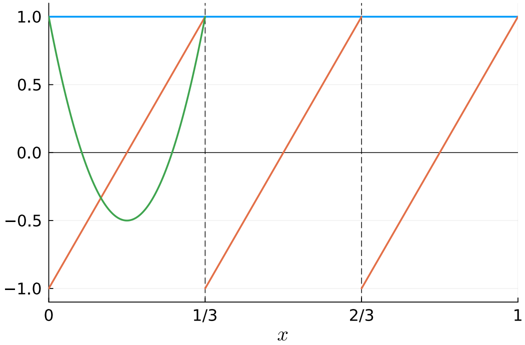
The Legendre polynomials $P_n(x)$, $n \in \mathbb{N}_0$ satisfy $\int_{-1}^{1} P_n P_m \, \mathrm{d}x \simeq \delta_{nm}$.
We can shift-and-scale the polynomials to construct a 1D basis such that $(\zeta_i, \zeta_j) \simeq \delta_{nm}$ for all basis functions $\zeta_i \in Q_{hp}$. This basis has *fast* transforms.



Shift-and-scale linear polynomial $P_1(x)$.

The Legendre polynomials

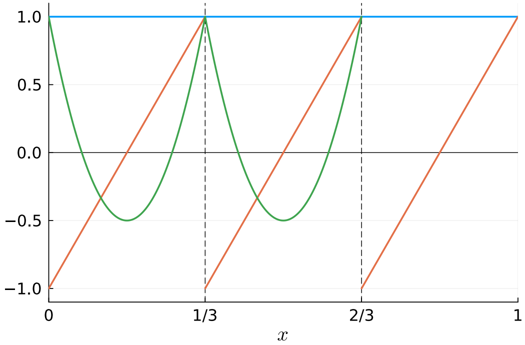
The Legendre polynomials $P_n(x)$, $n \in \mathbb{N}_0$ satisfy $\int_{-1}^{1} P_n P_m \,\mathrm{d}x \simeq \delta_{nm}$.
We can shift-and-scale the polynomials to construct a 1D basis such that $(\zeta_i, \zeta_j) \simeq \delta_{nm}$ for all basis functions $\zeta_i \in Q_{hp}$. This basis has *fast* transforms.



Shift-and-scale quadratic polynomial $P_2(x)$.

The Legendre polynomials

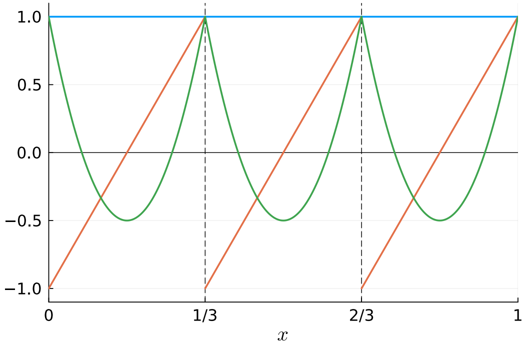
The Legendre polynomials $P_n(x)$, $n \in \mathbb{N}_0$ satisfy $\int_{-1}^{1} P_n P_m \, \mathrm{d}x \simeq \delta_{nm}$.
We can shift-and-scale the polynomials to construct a 1D basis such that $(\zeta_i, \zeta_j) \simeq \delta_{nm}$ for all basis functions $\zeta_i \in Q_{hp}$. This basis has *fast* transforms.



Shift-and-scale quadratic polynomial $P_2(x)$.

The Legendre polynomials

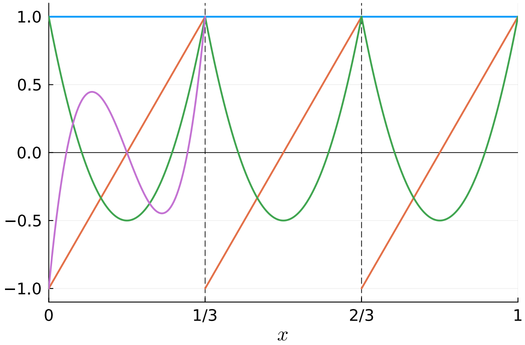
The Legendre polynomials $P_n(x)$, $n \in \mathbb{N}_0$ satisfy $\int_{-1}^{1} P_n P_m \,\mathrm{d}x \simeq \delta_{nm}$.
We can shift-and-scale the polynomials to construct a 1D basis such that $(\zeta_i, \zeta_j) \simeq \delta_{nm}$ for all basis functions $\zeta_i \in Q_{hp}$. This basis has *fast* transforms.



Shift-and-scale quadratic polynomial $P_2(x)$.

The Legendre polynomials

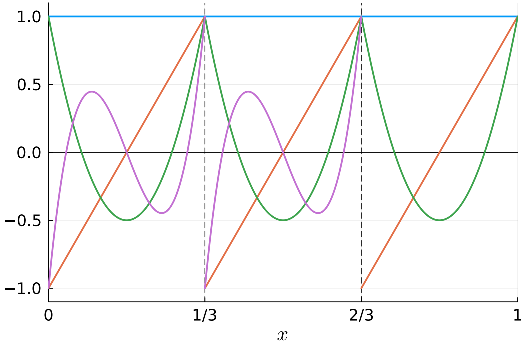
The Legendre polynomials $P_n(x)$, $n \in \mathbb{N}_0$ satisfy $\int_{-1}^{1} P_n P_m \,\mathrm{d}x \simeq \delta_{nm}$.
We can shift-and-scale the polynomials to construct a 1D basis such that $(\zeta_i, \zeta_j) \simeq \delta_{nm}$ for all basis functions $\zeta_i \in Q_{hp}$. This basis has *fast* transforms.



Shift-and-scale cubic polynomial $P_3(x)$ and so on...

The Legendre polynomials

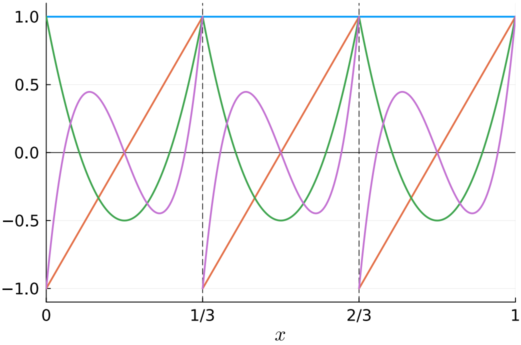
The Legendre polynomials $P_n(x)$, $n \in \mathbb{N}_0$ satisfy $\int_{-1}^{1} P_n P_m \,\mathrm{d}x \simeq \delta_{nm}$.
We can shift-and-scale the polynomials to construct a 1D basis such that $(\zeta_i, \zeta_j) \simeq \delta_{nm}$ for all basis functions $\zeta_i \in Q_{hp}$. This basis has *fast* transforms.



Shift-and-scale cubic polynomial $P_3(x)$ and so on...

The Legendre polynomials

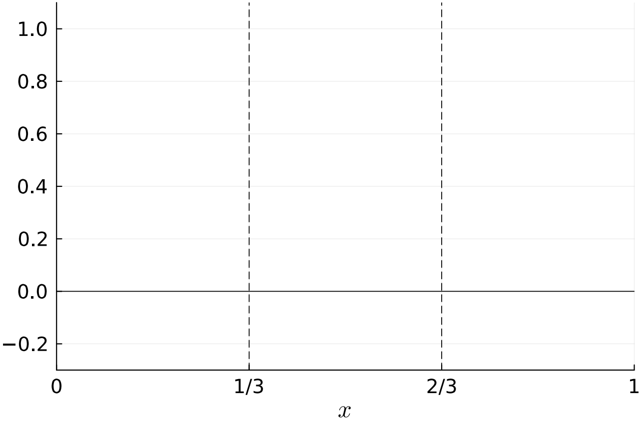
The Legendre polynomials $P_n(x)$, $n \in \mathbb{N}_0$ satisfy $\int_{-1}^{1} P_n P_m \,\mathrm{d}x \simeq \delta_{nm}$.
We can shift-and-scale the polynomials to construct a 1D basis such that $(\zeta_i, \zeta_j) \simeq \delta_{nm}$ for all basis functions $\zeta_i \in Q_{hp}$. This basis has *fast* transforms.



Shift-and-scale cubic polynomial $P_3(x)$ and so on...

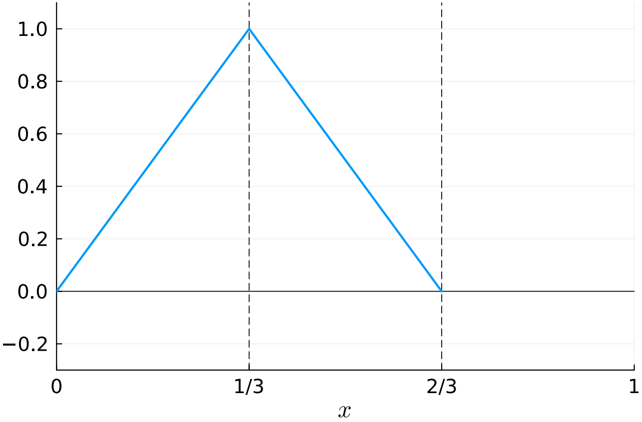
The continuous hierarchical *p*-FEM basis in 1D

We need a continuous FEM basis for u:



The continuous hierarchical p-FEM basis in 1D

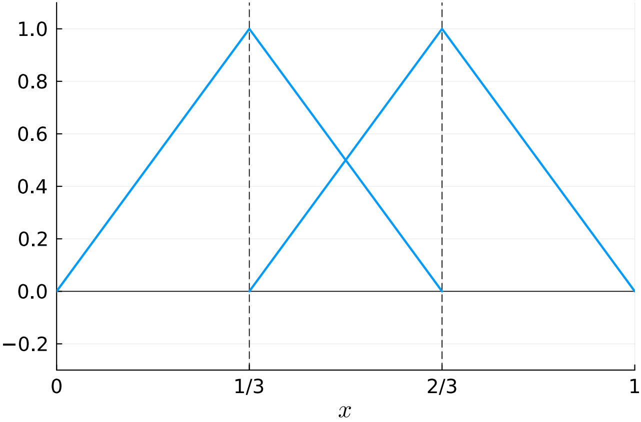
We need a continuous FEM basis for u:



Begin with adding classical FEM piecewise linear “hat” polynomials.

The continuous hierarchical p-FEM basis in 1D

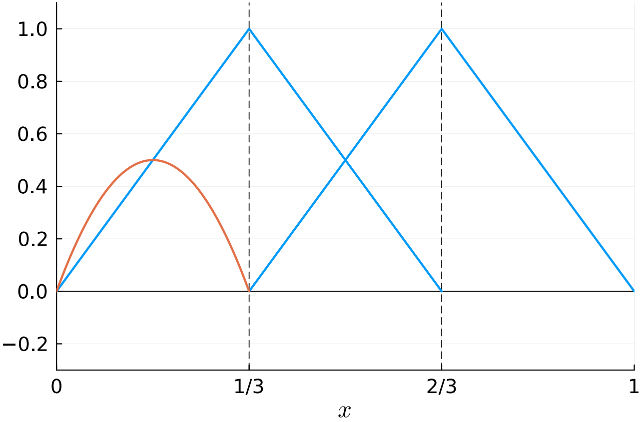
We need a continuous FEM basis for u:



Begin with adding classical FEM piecewise linear “hat” polynomials.

The continuous hierarchical p-FEM basis in 1D

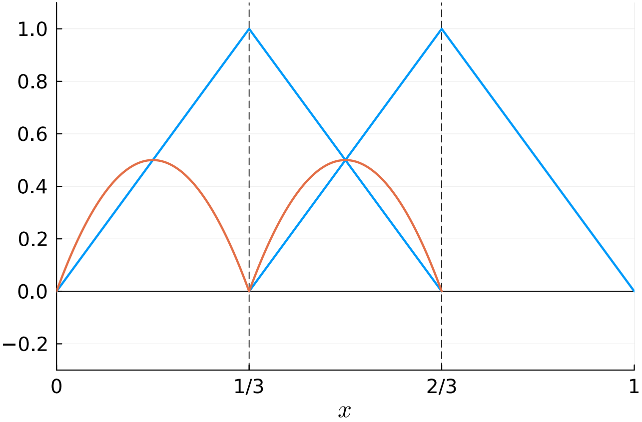
We need a continuous FEM basis for u:



Next add quadratic “bubble” polynomials only supported on one element each.

The continuous hierarchical p-FEM basis in 1D

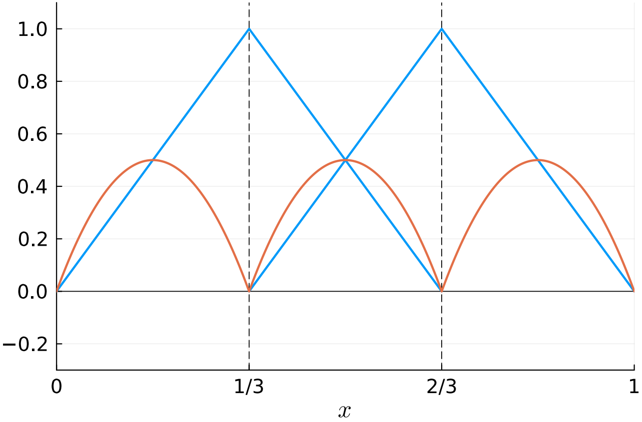
We need a continuous FEM basis for u:



Next add quadratic “bubble” polynomials only supported on one element each.

The continuous hierarchical *p*-FEM basis in 1D

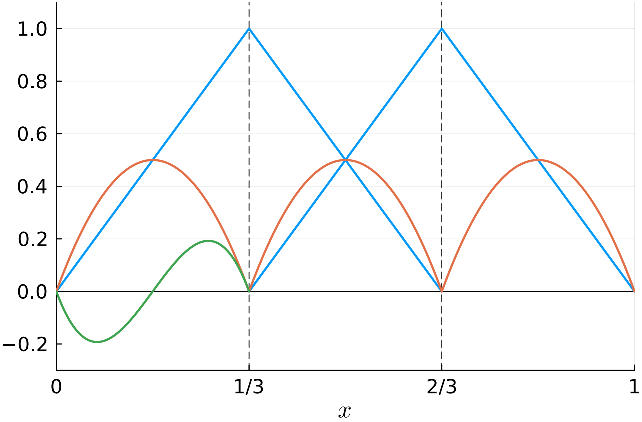
We need a continuous FEM basis for u:



Next add quadratic “bubble” polynomials only supported on one element each.

The continuous hierarchical p-FEM basis in 1D

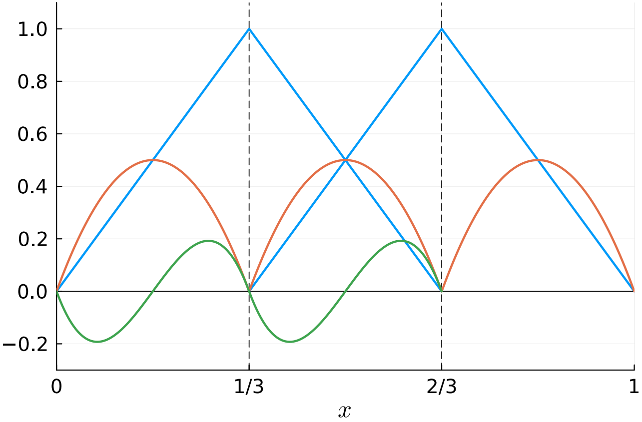
We need a continuous FEM basis for u:



Then add cubic “bubble” polynomials.

The continuous hierarchical p-FEM basis in 1D

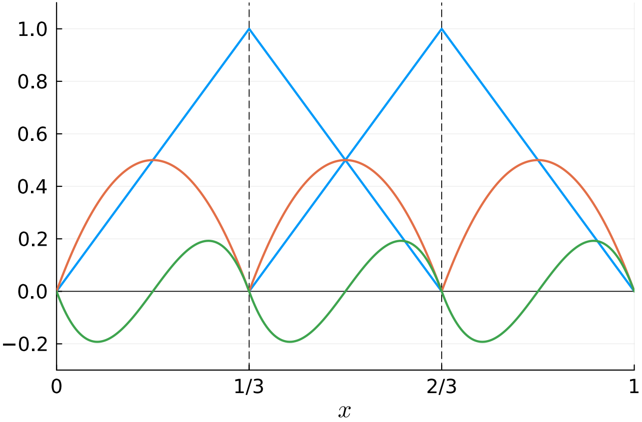
We need a continuous FEM basis for u:



Then add cubic “bubble” polynomials.

The continuous hierarchical p-FEM basis in 1D

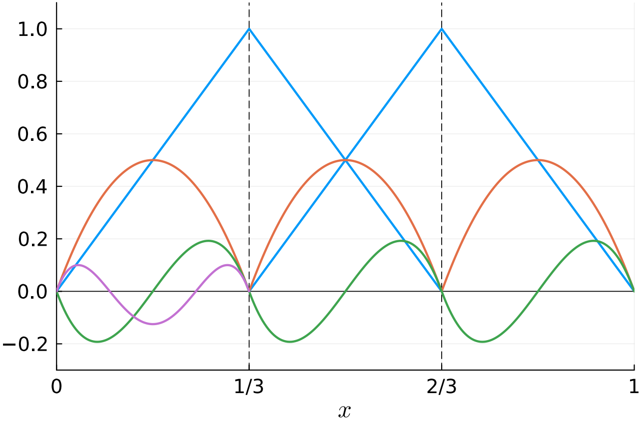
We need a continuous FEM basis for u:



Then add cubic “bubble” polynomials.

The continuous hierarchical *p*-FEM basis in 1D

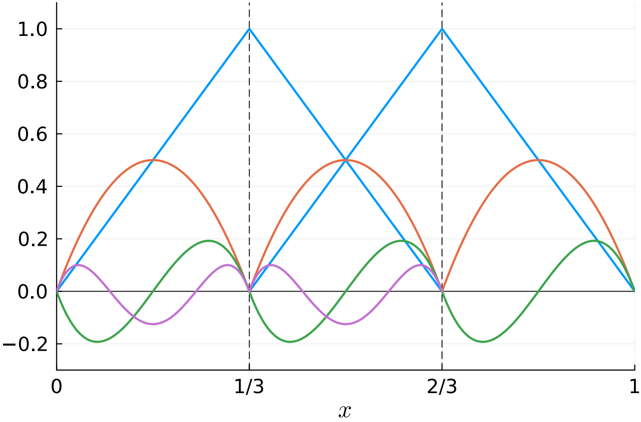
We need a continuous FEM basis for u:



Then add quartic “bubble” polynomials and so on...

The continuous hierarchical *p*-FEM basis in 1D

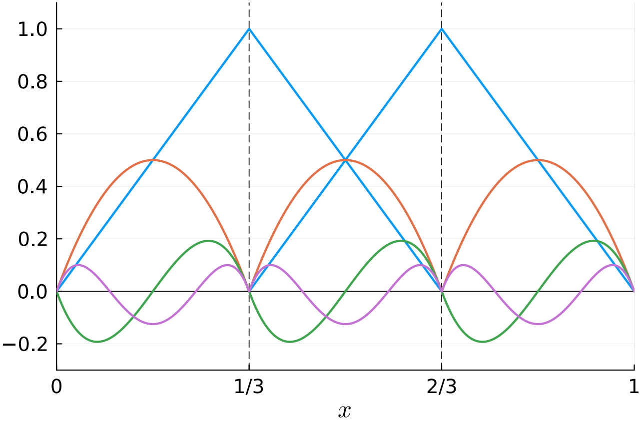
We need a continuous FEM basis for u:



Then add quartic "bubble" polynomials and so on...

The continuous hierarchical p-FEM basis in 1D

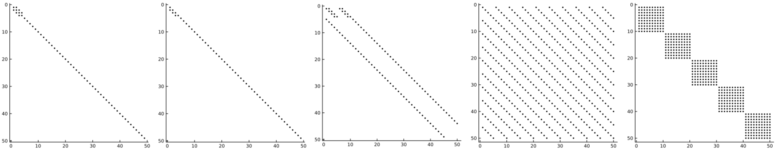
We need a continuous FEM basis for u:



Then add quartic “bubble” polynomials and so on...

Sparsity of A, B and D_ψ

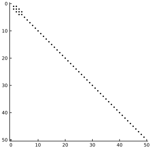
1D, 5 cells, $p = 10$.



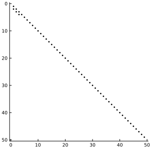
(a) A (b) L^{rchol} (c) B (d) D_ψ (e) Permuted D_ψ

Sparsity of A, B and D_ψ

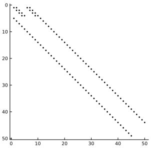
1D, 5 cells, $p = 10$.



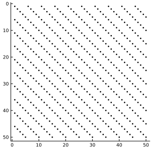
(a) A



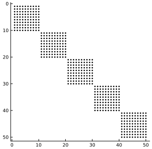
(b) L^{rchol}



(c) B

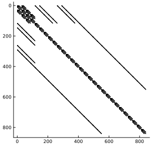


(d) D_ψ

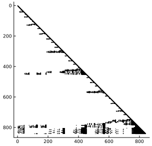


(e) Permuted D_ψ

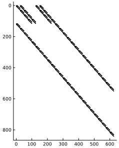
2D, 25 cells, $p = 5$.



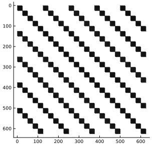
(f) A



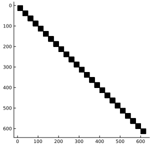
(g) L^{chol}



(h) B



(i) D_ψ



(j) Permuted D_ψ

Example: oscillatory data in 1D

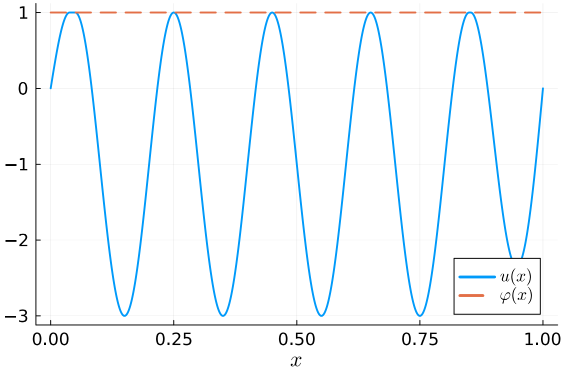
$$\Omega = (0,1), \quad f(x) = 200\pi^2 \sin(10\pi x), \quad \varphi \equiv 1.$$

$$\min_{u \in H_0^1(\Omega)} \int_\Omega \frac{1}{2}|\nabla u|^2 - fu \,\mathrm{d}x \text{ subject to } u(x) \leq \varphi(x).$$

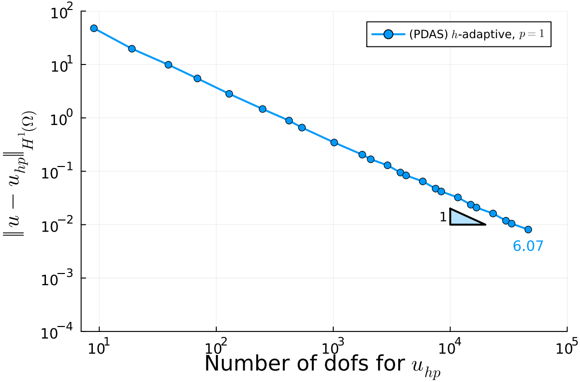
Example: oscillatory data in 1D

$$\Omega = (0,1), \quad f(x) = 200\pi^2 \sin(10\pi x), \quad \varphi \equiv 1.$$

$$\min_{u \in H_0^1(\Omega)} \int_\Omega \frac{1}{2}|\nabla u|^2 - fu \, \mathrm{d}x \ \text{ subject to } u(x) \le \varphi(x).$$

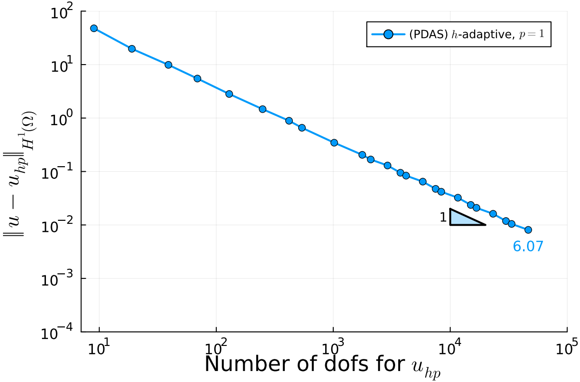


Example: oscillatory data in 1D



Cholesky factorization for the reduced PDAS stiffness matrix.
LU factorization for LVPP Newton systems with $\alpha_1 = 2^{-7}$, $\alpha_{k+1} = \min(\sqrt{2}\alpha_k, 2^{-3})$ and terminate once $\alpha_k = \alpha_{k-1} = 2^{-3}$. LVPP solver exhibits *hp*-independence (20-30 Newton linear system solves).

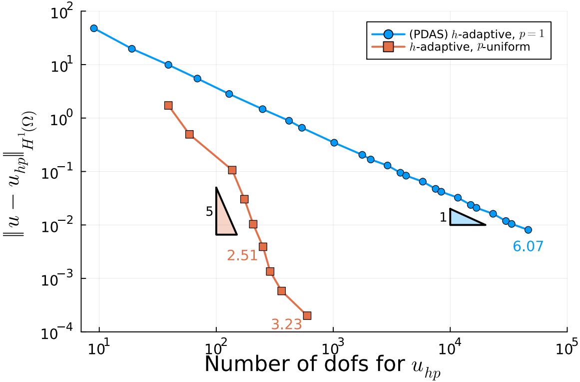
Example: oscillatory data in 1D



Cholesky factorization for the reduced PDAS stiffness matrix.

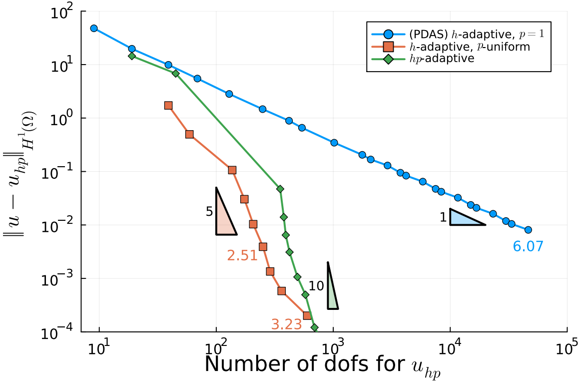
LU factorization for LVPP Newton systems with $\alpha_1 = 2^{-7}$, $\alpha_{k+1} = \min(\sqrt{2}\alpha_k, 2^{-3})$ and terminate once $\alpha_k = \alpha_{k-1} = 2^{-3}$. LVPP solver exhibits *hp*-independence (20-30 Newton linear system solves).

Example: oscillatory data in 1D



Cholesky factorization for the reduced PDAS stiffness matrix.
LU factorization for LVPP Newton systems with $\alpha_1 = 2^{-7}$, $\alpha_{k+1} = \min(\sqrt{2}\alpha_k, 2^{-3})$ and terminate once $\alpha_k = \alpha_{k-1} = 2^{-3}$. LVPP solver exhibits *hp*-independence (20-30 Newton linear system solves).

Example: oscillatory data in 1D

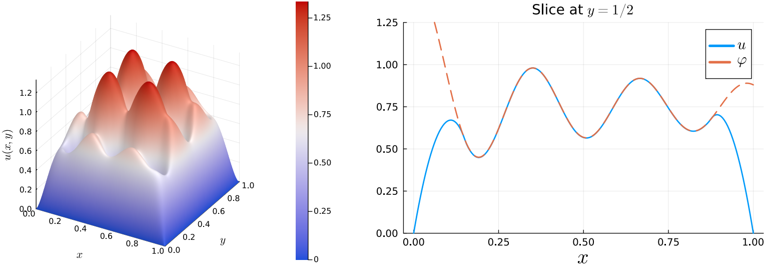


Cholesky factorization for the reduced PDAS stiffness matrix.
LU factorization for LVPP Newton systems with $\alpha_1 = 2^{-7}$, $\alpha_{k+1} = \min(\sqrt{2}\alpha_k, 2^{-3})$ and terminate once $\alpha_k = \alpha_{k-1} = 2^{-3}$. LVPP solver exhibits *hp*-independence (20-30 Newton linear system solves).

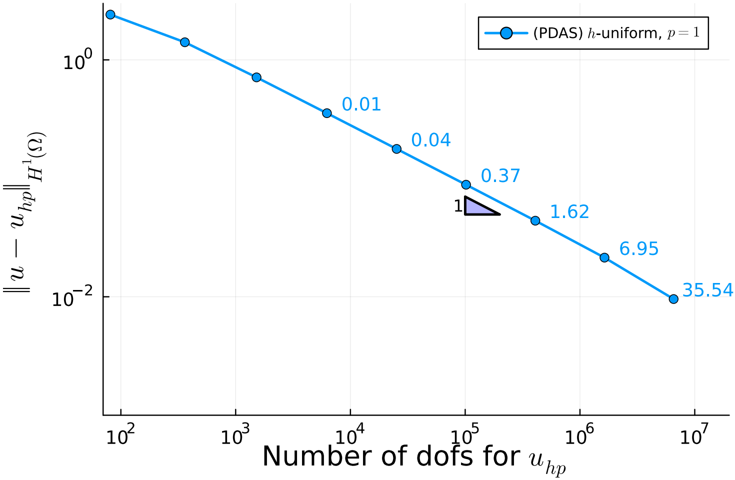
Example: oscillatory obstacle

$$\Omega = (0,1)^2, \quad f(x,y) = 100, \quad \text{and} \quad \varphi(x,y) = (1 + J_0(20x))(1 + J_0(20y)),$$

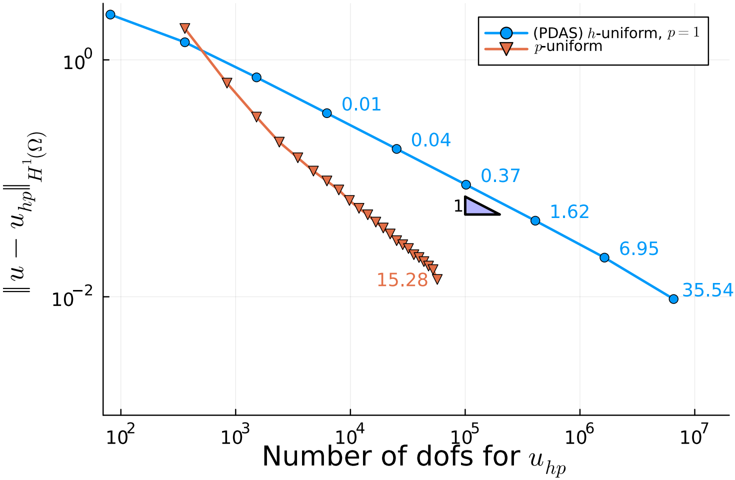
where J_0 denotes the zeroth order Bessel function of the first kind.



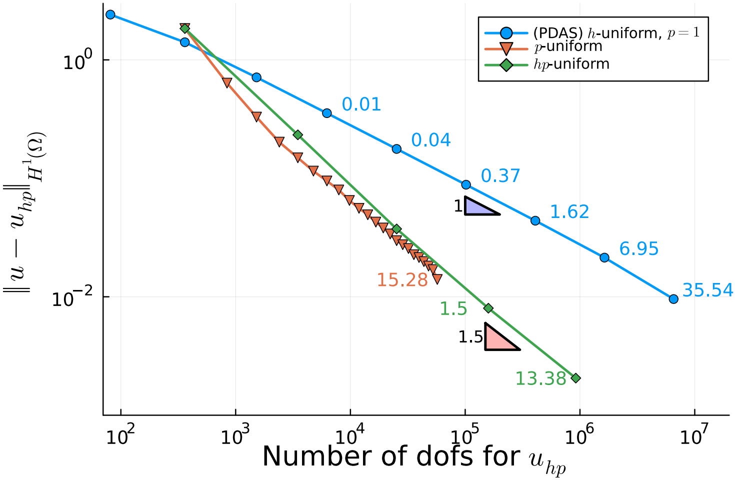
Example: oscillatory obstacle



Example: oscillatory obstacle



Example: oscillatory obstacle



Block preconditioning

Recall we are repeatedly solving (where $A_\alpha := \alpha A$)

$$\begin{pmatrix} A_\alpha & B \\ B^\top & -D_\psi \end{pmatrix} \begin{pmatrix} \boldsymbol{\delta}_u \\ \boldsymbol{\delta}_\psi \end{pmatrix} = \begin{pmatrix} \boldsymbol{b}_u \\ \boldsymbol{b}_\psi \end{pmatrix}.$$

Schur complement factorization

A Schur complement factorization reveals that

$$\boldsymbol{\delta}_u = A_\alpha^{-1}(\boldsymbol{b}_u - B\boldsymbol{\delta}_\psi) \text{ and } \boldsymbol{\delta}_\psi = S^{-1}(\boldsymbol{b}_\psi - B^\top A_\alpha^{-1}\boldsymbol{b}_u),$$

where $S := -(D_\psi + B^\top A_\alpha^{-1} B)$.

Advantages

A_α and B are sparse and A_α admits a cheap Cholesky factorization that we only compute once.

Block preconditioning

Recall we are repeatedly solving (where $A_\alpha := \alpha A$)

$$\begin{pmatrix} A_\alpha & B \\ B^\top & -D_\psi \end{pmatrix} \begin{pmatrix} \boldsymbol{\delta}_u \\ \boldsymbol{\delta}_\psi \end{pmatrix} = \begin{pmatrix} \boldsymbol{b}_u \\ \boldsymbol{b}_\psi \end{pmatrix}.$$

Schur complement factorization

A Schur complement factorization reveals that

$$\boldsymbol{\delta}_u = A_\alpha^{-1}(\boldsymbol{b}_u - B\boldsymbol{\delta}_\psi) \text{ and } \boldsymbol{\delta}_\psi = S^{-1}(\boldsymbol{b}_\psi - B^\top A_\alpha^{-1}\boldsymbol{b}_u),$$

where $S := -(D_\psi + B^\top A_\alpha^{-1} B)$.

Advantages

A_α and B are sparse and A_α admits a cheap Cholesky factorization that we only compute once.

Block preconditioning

Recall we are repeatedly solving (where $A_\alpha := \alpha A$)

$$\begin{pmatrix} A_\alpha & B \\ B^\top & -D_\psi \end{pmatrix} \begin{pmatrix} \boldsymbol{\delta}_u \\ \boldsymbol{\delta}_\psi \end{pmatrix} = \begin{pmatrix} \boldsymbol{b}_u \\ \boldsymbol{b}_\psi \end{pmatrix}.$$

Schur complement factorization

A Schur complement factorization reveals that

$$\boldsymbol{\delta}_u = A_\alpha^{-1}(\boldsymbol{b}_u - B\boldsymbol{\delta}_\psi) \text{ and } \boldsymbol{\delta}_\psi = S^{-1}(\boldsymbol{b}_\psi - B^\top A_\alpha^{-1}\boldsymbol{b}_u),$$

where $S := -(D_\psi + B^\top A_\alpha^{-1} B)$.

Advantages

A_α and B are sparse and A_α admits a cheap Cholesky factorization that we only compute once.

Iterative solver & Schur complement approximation

Complication

$S = -(D_\psi + B^\top A_\alpha^{-1} B)$ is dense — it cannot be assembled and factorized quickly.

However, given a vector $\boldsymbol{y}$ we may compute $S\boldsymbol{y}$ efficiently.

Iterative solver

Solve $S\delta_\psi = (\boldsymbol{b}_\psi - B^\top A_\alpha^{-1} \boldsymbol{b}_u)$ with GMRES preconditioned with a block-diagonal Schur complement approximation $\hat{S}$.

We choose

$$\hat{S} := -(D_\psi + \hat{B}^\top \hat{A}_\alpha^{-1} \hat{B})$$

where $\hat{A}_\alpha$ and $\hat{B}$ are the block-diagonal matrices associated with the basis functions $\{\phi_i\} \subset V_{hp}$ where the hat functions have been dropped.

Iterative solver & Schur complement approximation

Complication

$S = -(D_\psi + B^\top A_\alpha^{-1} B)$ is dense — it cannot be assembled and factorized quickly.

However, given a vector $\boldsymbol{y}$ we may compute $S\boldsymbol{y}$ efficiently.

Iterative solver

Solve $S\delta_\psi = (\boldsymbol{b}_\psi - B^\top A_\alpha^{-1} \boldsymbol{b}_u)$ with GMRES preconditioned with a block-diagonal Schur complement approximation $\hat{S}$.

We choose

$$\hat{S} := -(D_\psi + \hat{B}^\top \hat{A}_\alpha^{-1} \hat{B})$$

where $\hat{A}_\alpha$ and $\hat{B}$ are the block-diagonal matrices associated with the basis functions $\{\phi_i\} \subset V_{hp}$ where the hat functions have been dropped.

Iterative solver & Schur complement approximation

Complication

$S = -(D_\psi + B^\top A_\alpha^{-1} B)$ is dense — it cannot be assembled and factorized quickly.

However, given a vector $\boldsymbol{y}$ we may compute $S\boldsymbol{y}$ efficiently.

Iterative solver

Solve $S\delta_\psi = (\boldsymbol{b}_\psi - B^\top A_\alpha^{-1} \boldsymbol{b}_u)$ with GMRES preconditioned with a block-diagonal Schur complement approximation $\hat{S}$.

We choose

$$\hat{S} := -(D_\psi + \hat{B}^\top \hat{A}_\alpha^{-1} \hat{B})$$

where $\hat{A}_\alpha$ and $\hat{B}$ are the block-diagonal matrices associated with the basis functions $\{\phi_i\} \subset V_{hp}$ where the hat functions have been dropped.

Iterative solver & Schur complement approximation

Complication

$S = -(D_\psi + B^\top A_\alpha^{-1} B)$ is dense — it cannot be assembled and factorized quickly.

However, given a vector $\boldsymbol{y}$ we may compute $S\boldsymbol{y}$ efficiently.

Iterative solver

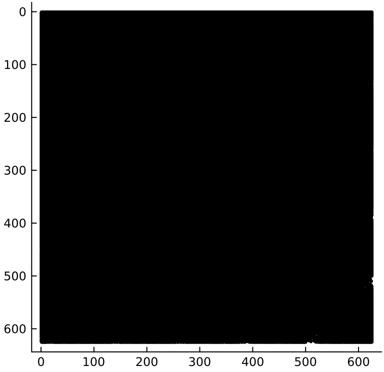
Solve $S\delta_\psi = (\boldsymbol{b}_\psi - B^\top A_\alpha^{-1} \boldsymbol{b}_u)$ with GMRES preconditioned with a block-diagonal Schur complement approximation $\hat{S}$.

We choose

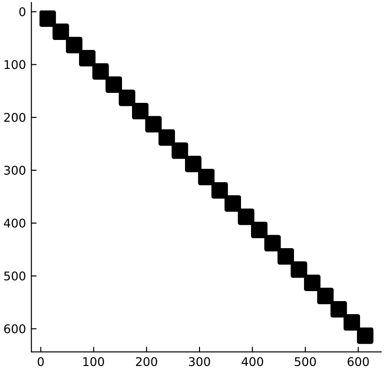
$$\hat{S} := -(D_\psi + \hat{B}^\top \hat{A}_\alpha^{-1} \hat{B})$$

where $\hat{A}_\alpha$ and $\hat{B}$ are the block-diagonal matrices associated with the basis functions $\{\phi_i\} \subset V_{hp}$ where the hat functions have been dropped.

Schur complement approximation



(a) Schur complement S



(b) Block-diagonal approximation $\hat{S}$

Example: thermoforming

The thermoforming quasi-variational inequality seeks u minimizing

$$\min_{u\in H_0^1(\Omega)} \int_\Omega \frac{1}{2}|\nabla u|^2 - fu\,\mathrm{d}x \text{ subject to } u \leq \varphi(T) := \Phi_0 + \xi T, \tag{1}$$

where Φ_0 and ξ are given and T satisfies

$$-\Delta T + \gamma T = g(\Phi_0 + \xi T - u), \;\; \partial_\nu T = 0 \text{ on } \partial\Omega. \tag{2}$$

Solver strategy

We will solve the thermoforming problem via a fixed point approach, i.e. repeatedly solve

1. Freeze T and solve the obstacle subproblem (1) for u,
2. Freeze u and solve the nonlinear PDE (2) for T.

Example: thermoforming

The thermoforming quasi-variational inequality seeks u minimizing

$$\min_{u \in H_0^1(\Omega)} \int_\Omega \frac{1}{2}|\nabla u|^2 - fu \, \mathrm{d}x \text{ subject to } u \leq \varphi(T) := \Phi_0 + \xi T, \qquad (1)$$

where Φ_0 and ξ are given and T satisfies

$$-\Delta T + \gamma T = g(\Phi_0 + \xi T - u), \quad \partial_\nu T = 0 \text{ on } \partial\Omega. \qquad (2)$$

Solver strategy

We will solve the thermoforming problem via a fixed point approach, i.e. repeatedly solve

1. Freeze T and solve the obstacle subproblem (1) for u,
2. Freeze u and solve the nonlinear PDE (2) for T.

Example: thermoforming

The thermoforming quasi-variational inequality seeks u minimizing

$$\min_{u \in H_0^1(\Omega)} \int_\Omega \frac{1}{2}|\nabla u|^2 - fu \,\mathrm{d}x \text{ subject to } u \leq \varphi(T) := \Phi_0 + \xi T, \tag{1}$$

where Φ_0 and ξ are given and T satisfies

$$-\Delta T + \gamma T = g(\Phi_0 + \xi T - u), \quad \partial_\nu T = 0 \text{ on } \partial\Omega. \tag{2}$$

Solver strategy

We will solve the thermoforming problem via a fixed point approach, i.e. repeatedly solve

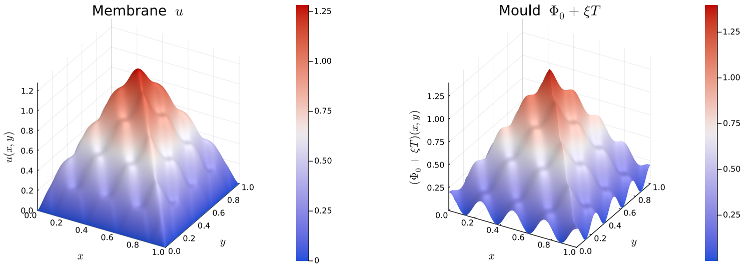
1. Freeze T and solve the obstacle subproblem (1) for u,
2. Freeze u and solve the nonlinear PDE (2) for T.

Example: thermoforming

$$\Omega = (0,1)^2, \qquad f(x,y) = 100, \qquad \xi(x,y) = \sin(\pi x)\sin(\pi y), \qquad \gamma = 1,$$

$$\Phi_0(x,y) = 11/10 - 2\max(|x-1/2|, |y-1/2|) + \cos(8\pi x)\cos(8\pi y)/10,$$

$$g(s) = \begin{cases} 1/5 & \text{if } s \le 0, \\ (1-s)/5 & \text{if } 0 < s < 1, \\ 0 & \text{otherwise.} \end{cases}$$

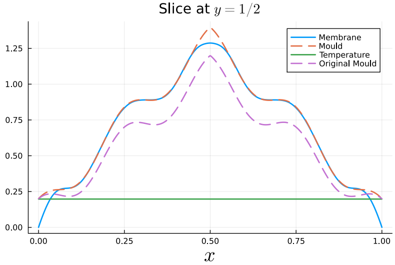


Example: thermoforming

$$\Omega = (0,1)^2, \qquad f(x,y) = 100, \qquad \xi(x,y) = \sin(\pi x)\sin(\pi y), \qquad \gamma = 1,$$

$$\Phi_0(x,y) = 11/10 - 2\max(|x - 1/2|, |y - 1/2|) + \cos(8\pi x)\cos(8\pi y)/10,$$

$$g(s) = \begin{cases} 1/5 & \text{if } s \le 0, \\ (1-s)/5 & \text{if } 0 < s < 1, \\ 0 & \text{otherwise.} \end{cases}$$



Example: thermoforming

		Obstacle subsolve for u		Nonlinear subsolve for T	
p	Fixed point	Avg. Newton	Avg. GMRES	Avg. Newton	Avg. GMRES
6	4	15.00	11.00	1.50	2.83
12	4	15.25	15.85	2.00	3.13
22	4	16.00	19.36	2.00	3.00
32	4	16.00	21.09	2.00	3.00
42	4	15.75	21.75	2.25	3.11
52	4	15.00	22.40	2.00	3.00
62	4	15.00	21.90	2.00	3.00
72	4	15.00	21.90	2.00	3.00
82	4	15.25	21.61	2.00	3.00

p-independent Newton and preconditioned GMRES iteration counts to solve the thermoforming problem. Unbelievable!

Example: thermoforming

		Obstacle subsolve for u		Nonlinear subsolve for T	
p	Fixed point	Avg. Newton	Avg. GMRES	Avg. Newton	Avg. GMRES
6	4	15.00	11.00	1.50	2.83
12	4	15.25	15.85	2.00	3.13
22	4	16.00	19.36	2.00	3.00
32	4	16.00	21.09	2.00	3.00
42	4	15.75	21.75	2.25	3.11
52	4	15.00	22.40	2.00	3.00
62	4	15.00	21.90	2.00	3.00
72	4	15.00	21.90	2.00	3.00
82	4	15.25	21.61	2.00	3.00

Partial degree

p-independent Newton and preconditioned GMRES iteration counts to solve the thermoforming problem. Unbelievable!

Example: thermoforming

		Obstacle subsolve for u		Nonlinear subsolve for T	
p	Fixed point	Avg. Newton	Avg. GMRES	Avg. Newton	Avg. GMRES
6	4	15.00	11.00	1.50	2.83
12	4	15.25	15.85	2.00	3.13
22	4	16.00	19.36	2.00	3.00
32	4	16.00	21.09	2.00	3.00
42	4	15.75	21.75	2.25	3.11
52	4	15.00	22.40	2.00	3.00
62	4	15.00	21.90	2.00	3.00
72	4	15.00	21.90	2.00	3.00
82	4	15.25	21.61	2.00	3.00

Partial degree

Outer loop

p-independent Newton and preconditioned GMRES iteration counts to solve the thermoforming problem. Unbelievable!

Example: thermoforming

		Obstacle subsolve for u		Nonlinear subsolve for T	
p	Fixed point	Avg. Newton	Avg. GMRES	Avg. Newton	Avg. GMRES
6	4	15.00	11.00	1.50	2.83
12	4	15.25	15.85	2.00	3.13
22	4	16.00	19.36	2.00	3.00
32	4	16.00	21.09	2.00	3.00
42	4	15.75	21.75	2.25	3.11
52	4	15.00	22.40	2.00	3.00
62	4	15.00	21.90	2.00	3.00
72	4	15.00	21.90	2.00	3.00
82	4	15.25	21.61	2.00	3.00

Partial degree

Outer loop

Average Newton steps to solve an obstacle subproblem

Average preconditioned GMRES iterations per Newton step

p-independent Newton and preconditioned GMRES iteration counts to solve the thermoforming problem. Unbelievable!

Example: thermoforming

		Obstacle subsolve for u		Nonlinear subsolve for T	
p	Fixed point	Avg. Newton	Avg. GMRES	Avg. Newton	Avg. GMRES
6	4	15.00	11.00	1.50	2.83
12	4	15.25	15.85	2.00	3.13
22	4	16.00	19.36	2.00	3.00
32	4	16.00	21.09	2.00	3.00
42	4	15.75	21.75	2.25	3.11
52	4	15.00	22.40	2.00	3.00
62	4	15.00	21.90	2.00	3.00
72	4	15.00	21.90	2.00	3.00
82	4	15.25	21.61	2.00	3.00

Partial degree

Outer loop

Average Newton steps to solve an obstacle subproblem

Average preconditioned GMRES iterations per Newton step

Average Newton steps to solve a temperature PDE subproblem

Average preconditioned GMRES iterations per Newton step

p-independent Newton and preconditioned GMRES iteration counts to solve the thermoforming problem. Unbelievable!

Conclusions

- Pointwise constraints can be effectively handled by the latent variable proximal point algorithm resulting in a nonlinear system of smooth PDEs.
- The PDE system is linearized with Newton.
- For the obstacle problem, the nonlinearity is confined to the latent variable ψ which can be discretized with a high-order Legendre polynomial DG FEM space that admits fast quadrature via the FFT.
- We discretize the membrane u with the hierarchical continuous p-FEM basis.
- This leads to sparse linear systems which admit simple preconditioners.
- **This leads to fast convergence with competitive wall clock solve times.**

Latent variable proximal point

Jørgen S. Dokken, Patrick E. Farrell, Brendan Keith, I. P., Thomas M. Surowiec, *The latent variable proximal point algorithm for variational problems with inequality constraints* (2025), `https://arxiv.org/abs/2503.05672`.

hp-FEM for obstacle and elastic-plastic torsion problems

I. P, *Hierarchical proximal Galerkin: a fast hp-FEM solver for variational problems with pointwise inequality constraints* (2024), `https://arxiv.org/abs/2412.13733`.

Conclusions

- Pointwise constraints can be effectively handled by the latent variable proximal point algorithm resulting in a nonlinear system of smooth PDEs.
- The PDE system is linearized with Newton.
- For the obstacle problem, the nonlinearity is confined to the latent variable ψ which can be discretized with a high-order Legendre polynomial DG FEM space that admits fast quadrature via the FFT.
- We discretize the membrane u with the hierarchical continuous p-FEM basis.
- This leads to sparse linear systems which admit simple preconditioners.
- **This leads to fast convergence with competitive wall clock solve times.**

Latent variable proximal point

Jørgen S. Dokken, Patrick E. Farrell, Brendan Keith, I. P., Thomas M. Surowiec, *The latent variable proximal point algorithm for variational problems with inequality constraints* (2025), `https://arxiv.org/abs/2503.05672`.

hp-FEM for obstacle and elastic-plastic torsion problems

I. P, *Hierarchical proximal Galerkin: a fast hp-FEM solver for variational problems with pointwise inequality constraints* (2024), `https://arxiv.org/abs/2412.13733`.

Conclusions

- Pointwise constraints can be effectively handled by the latent variable proximal point algorithm resulting in a nonlinear system of smooth PDEs.
- The PDE system is linearized with Newton.
- For the obstacle problem, the nonlinearity is confined to the latent variable ψ which can be discretized with a high-order Legendre polynomial DG FEM space that admits fast quadrature via the FFT.
- We discretize the membrane u with the hierarchical continuous p-FEM basis.
- This leads to sparse linear systems which admit simple preconditioners.
- **This leads to fast convergence with competitive wall clock solve times.**

Latent variable proximal point

Jørgen S. Dokken, Patrick E. Farrell, Brendan Keith, I. P., Thomas M. Surowiec, *The latent variable proximal point algorithm for variational problems with inequality constraints* (2025), https://arxiv.org/abs/2503.05672.

hp-FEM for obstacle and elastic-plastic torsion problems

I. P, *Hierarchical proximal Galerkin: a fast hp-FEM solver for variational problems with pointwise inequality constraints* (2024), https://arxiv.org/abs/2412.13733.

Conclusions

- Pointwise constraints can be effectively handled by the latent variable proximal point algorithm resulting in a nonlinear system of smooth PDEs.
- The PDE system is linearized with Newton.
- For the obstacle problem, the nonlinearity is confined to the latent variable ψ which can be discretized with a high-order Legendre polynomial DG FEM space that admits fast quadrature via the FFT.
- We discretize the membrane u with the hierarchical continuous p-FEM basis.
- This leads to sparse linear systems which admit simple preconditioners.
- **This leads to fast convergence with competitive wall clock solve times.**

Latent variable proximal point

Jørgen S. Dokken, Patrick E. Farrell, Brendan Keith, I. P., Thomas M. Surowiec, *The latent variable proximal point algorithm for variational problems with inequality constraints* (2025), https://arxiv.org/abs/2503.05672.

hp-FEM for obstacle and elastic-plastic torsion problems

I. P, *Hierarchical proximal Galerkin: a fast hp-FEM solver for variational problems with pointwise inequality constraints* (2024), https://arxiv.org/abs/2412.13733.

Conclusions

- Pointwise constraints can be effectively handled by the latent variable proximal point algorithm resulting in a nonlinear system of smooth PDEs.
- The PDE system is linearized with Newton.
- For the obstacle problem, the nonlinearity is confined to the latent variable ψ which can be discretized with a high-order Legendre polynomial DG FEM space that admits fast quadrature via the FFT.
- We discretize the membrane u with the hierarchical continuous p-FEM basis.
- This leads to sparse linear systems which admit simple preconditioners.
- **This leads to fast convergence with competitive wall clock solve times.**

Latent variable proximal point

Jørgen S. Dokken, Patrick E. Farrell, Brendan Keith, I. P., Thomas M. Surowiec, *The latent variable proximal point algorithm for variational problems with inequality constraints* (2025), `https://arxiv.org/abs/2503.05672`.

hp-FEM for obstacle and elastic-plastic torsion problems

I. P, *Hierarchical proximal Galerkin: a fast hp-FEM solver for variational problems with pointwise inequality constraints* (2024), `https://arxiv.org/abs/2412.13733`.

Conclusions

Code availability

`https://github.com/ioannisPApapadopoulos/HierarchicalProximalGalerkin.jl`.

Do you...

- have a problem with pointwise constraints and are looking for a robust solver?
- have ideas for high-order FEM on more general domains?
- have an interest in the infinite-dimensional or numerical analysis of LVPP?

Then please email me at ✉ papadopoulos@wias-berlin.de.

Conclusions

Code availability

`https://github.com/ioannisPApapadopoulos/HierarchicalProximalGalerkin.jl`.

Do you...

- have a problem with pointwise constraints and are looking for a robust solver?
- have ideas for high-order FEM on more general domains?
- have an interest in the infinite-dimensional or numerical analysis of LVPP?

Then please email me at papadopoulos@wias-berlin.de.

Conclusions

Code availability

`https://github.com/ioannisPApapadopoulos/HierarchicalProximalGalerkin.jl`.

Do you...

- have a problem with pointwise constraints and are looking for a robust solver?
- have ideas for high-order FEM on more general domains?
- have an interest in the infinite-dimensional or numerical analysis of LVPP?

Then please email me at papadopoulos@wias-berlin.de.

Conclusions

Code availability

`https://github.com/ioannisPApapadopoulos/HierarchicalProximalGalerkin.jl`.

Do you...

- have a problem with pointwise constraints and are looking for a robust solver?
- have ideas for high-order FEM on more general domains?
- have an interest in the infinite-dimensional or numerical analysis of LVPP?

Then please email me at ✉ papadopoulos@wias-berlin.de.

Conclusions

Code availability

`https://github.com/ioannisPApapadopoulos/HierarchicalProximalGalerkin.jl`.

Do you...

- have a problem with pointwise constraints and are looking for a robust solver?
- have ideas for high-order FEM on more general domains?
- have an interest in the infinite-dimensional or numerical analysis of LVPP?

Then please email me at ✉ papadopoulos@wias-berlin.de.

Thank you for listening!

✉ papadopoulos@wias-berlin.de