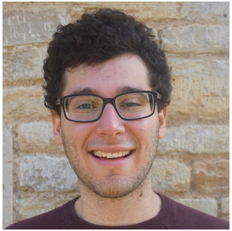
The latent variable proximal point algorithm for variational problems with inequality constraints

Patrick E. Farrell[1,2] Brendan Keith[3] Thomas Surowiec[4,3] Jørgen S. Dokken[4]
Ioannis P. A. Papadopoulos[5]



[1]University of Oxford [2]Charles University [3]Brown University [4]Simula Research Laboratory [5]WIAS

13 April 2026

Optimisation problems with inequality constraints lead to *variational inequalities* (VIs).

Optimisation problems with inequality constraints lead to *variational inequalities* (VIs).

Let's see an example. On a domain $\Omega \subset \mathbb{R}^n$, the obstacle problem is to minimise the energy

$$J(u) = \frac{1}{2}\int_\Omega \nabla u \cdot \nabla u \,\mathrm{d}x - \int_\Omega f u \,\mathrm{d}x$$

over the constrained set

$$K = \left\{ v \in H_0^1(\Omega) \mid v \geq \phi \text{ a.e. in } \Omega \right\}.$$

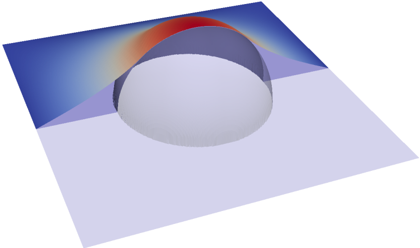
Optimisation problems with inequality constraints lead to *variational inequalities* (VIs).

Let's see an example. On a domain $\Omega \subset \mathbb{R}^n$, the obstacle problem is to minimise the energy

$$J(u) = \frac{1}{2}\int_\Omega \nabla u \cdot \nabla u \; \mathrm{d}x - \int_\Omega f u \; \mathrm{d}x$$

over the constrained set

$$K = \left\{ v \in H^1_0(\Omega) \mid v \geq \phi \text{ a.e. in } \Omega \right\}.$$



$f = 0, \phi = \frown$

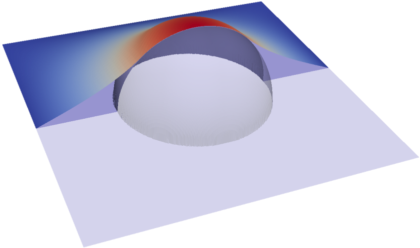
Optimisation problems with inequality constraints lead to *variational inequalities* (VIs).

Let's see an example. On a domain $\Omega \subset \mathbb{R}^n$, the obstacle problem is to minimise the energy

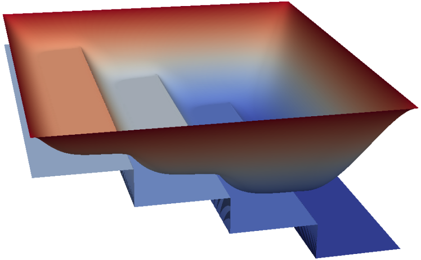
$$J(u) = \frac{1}{2}\int_\Omega \nabla u \cdot \nabla u \, \mathrm{d}x - \int_\Omega f u \, \mathrm{d}x$$

over the constrained set

$$K = \left\{ v \in H_0^1(\Omega) \mid v \geq \phi \text{ a.e. in } \Omega \right\}.$$



$f = 0, \phi = \frown$



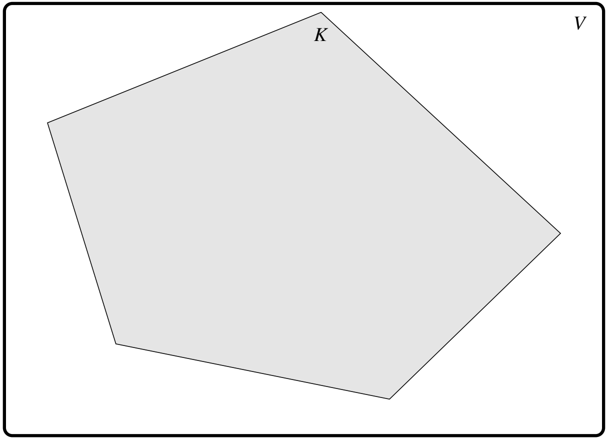
$f = -10, \phi =$ ⌝

The optimality condition for this problem is the variational inequality

$$u \in K : \quad J'(u; v-u) \geq 0 \;\forall\, v \in K.$$

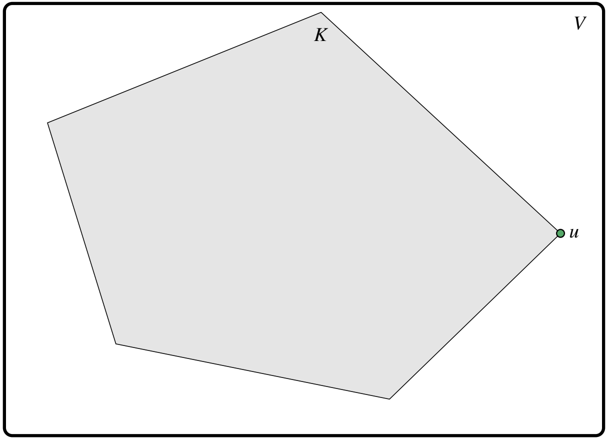
The optimality condition for this problem is the variational inequality

$$u \in K : \quad J'(u; v - u) \geq 0 \; \forall \, v \in K.$$



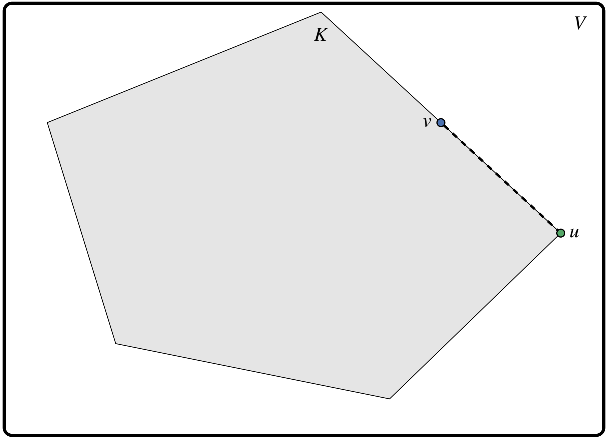
The optimality condition for this problem is the variational inequality

$$u \in K : \quad J'(u; v-u) \geq 0 \; \forall \, v \in K.$$



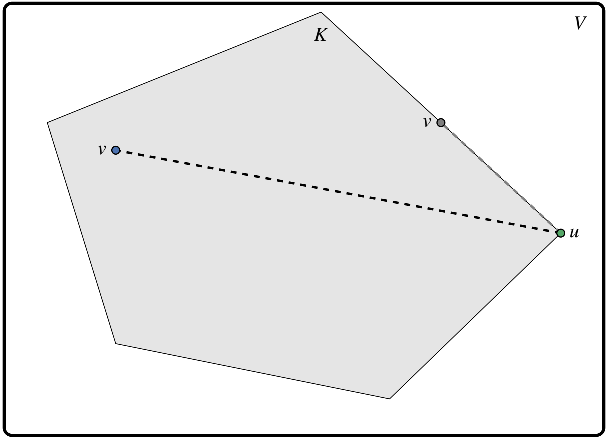
The optimality condition for this problem is the variational inequality

$$u \in K : \quad J'(u; v-u) \geq 0 \; \forall \, v \in K.$$



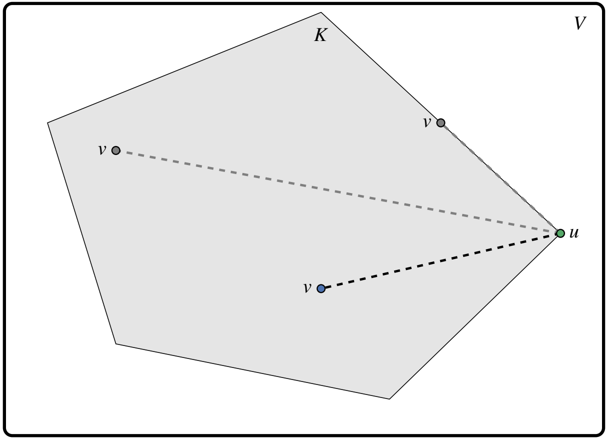
The optimality condition for this problem is the variational inequality

$$u \in K : \quad J'(u; v-u) \geq 0 \;\forall\, v \in K.$$



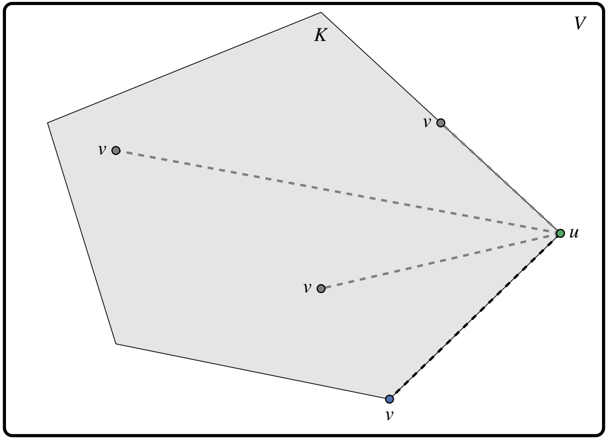
The optimality condition for this problem is the variational inequality

$$u \in K : \quad J'(u; v - u) \geq 0 \; \forall \, v \in K.$$



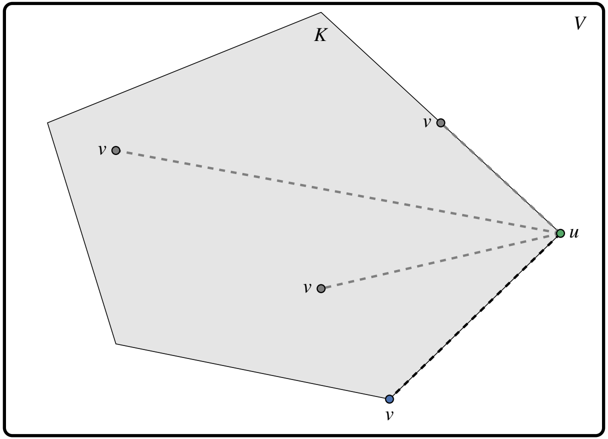
The optimality condition for this problem is the variational inequality

$$u \in K : \quad J'(u; v-u) \geq 0 \; \forall \, v \in K.$$



The optimality condition for this problem is the variational inequality

$$u \in K : \quad J'(u; v - u) \geq 0 \; \forall \, v \in K.$$



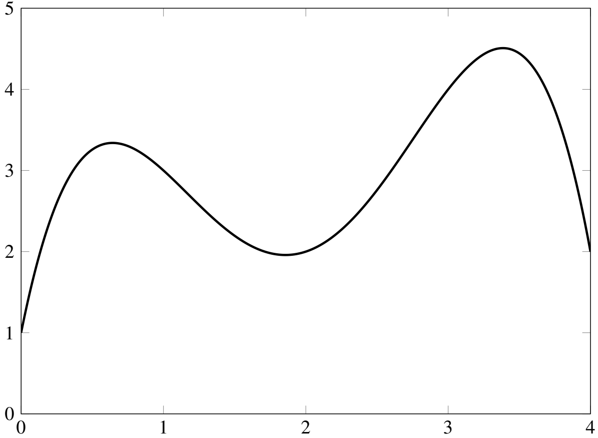
At a solution u, the energy must not decrease *along any feasible direction* $v - u$.

To minimise $f \in C^1(\mathbb{R}, \mathbb{R})$ over a nonempty, closed, convex interval $I \subset \mathbb{R}$, the necessary optimality condition is

$$x \in I : f'(x)(y - x) \geq 0 \, \forall \, y \in I.$$

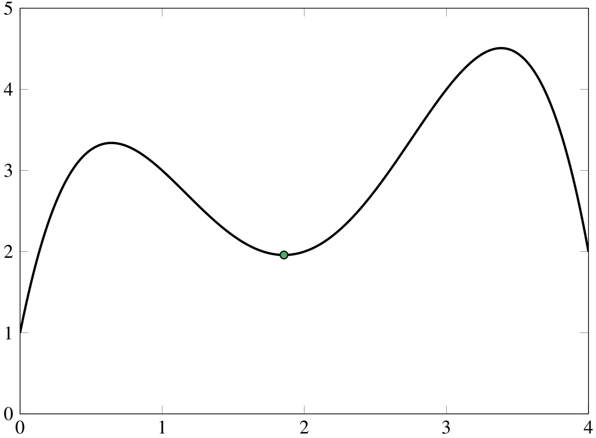
To minimise $f \in C^1(\mathbb{R}, \mathbb{R})$ over a nonempty, closed, convex interval $I \subset \mathbb{R}$, the necessary optimality condition is

$$x \in I : f'(x)(y - x) \geq 0 \,\forall\, y \in I.$$



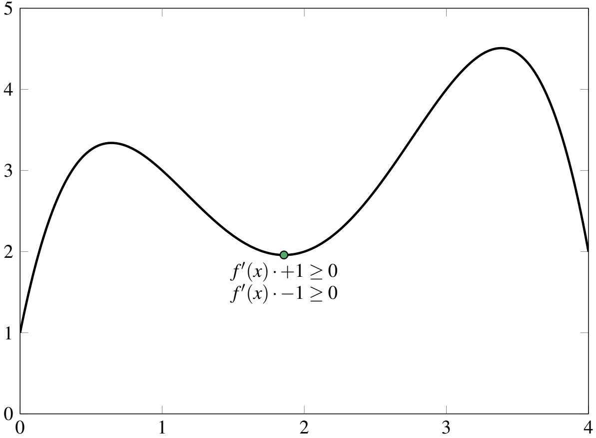
To minimise $f \in C^1(\mathbb{R}, \mathbb{R})$ over a nonempty, closed, convex interval $I \subset \mathbb{R}$, the necessary optimality condition is

$$x \in I : f'(x)(y - x) \geq 0 \, \forall \, y \in I.$$



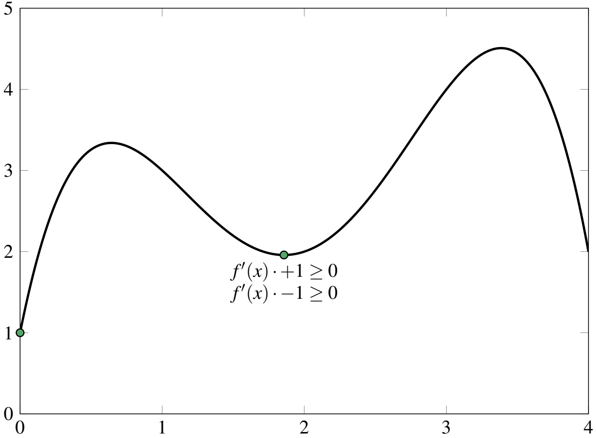
To minimise $f \in C^1(\mathbb{R}, \mathbb{R})$ over a nonempty, closed, convex interval $I \subset \mathbb{R}$, the necessary optimality condition is

$$x \in I : f'(x)(y - x) \geq 0 \,\forall\, y \in I.$$



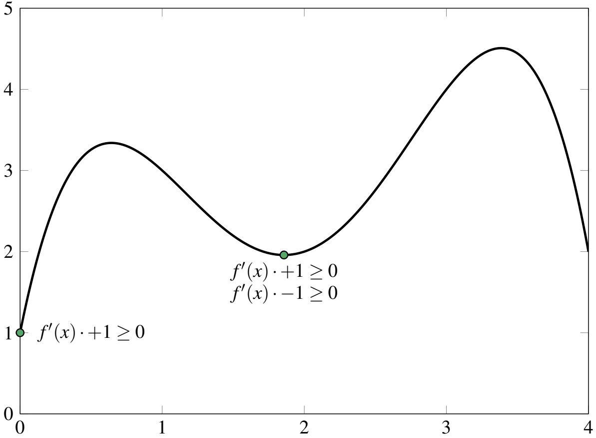
To minimise $f \in C^1(\mathbb{R}, \mathbb{R})$ over a nonempty, closed, convex interval $I \subset \mathbb{R}$, the necessary optimality condition is

$$x \in I : f'(x)(y - x) \geq 0 \,\forall\, y \in I.$$



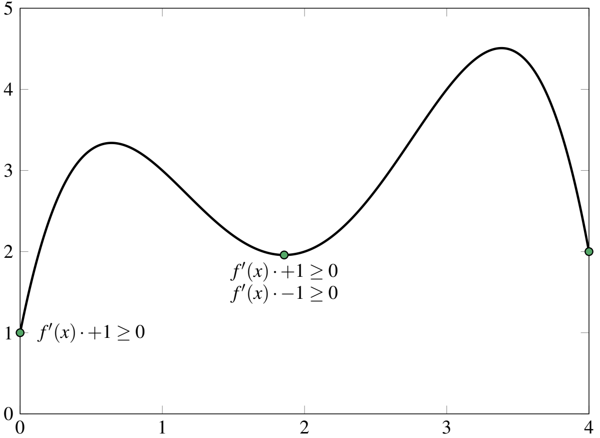
To minimise $f \in C^1(\mathbb{R}, \mathbb{R})$ over a nonempty, closed, convex interval $I \subset \mathbb{R}$, the necessary optimality condition is

$$x \in I : f'(x)(y - x) \geq 0 \, \forall \, y \in I.$$



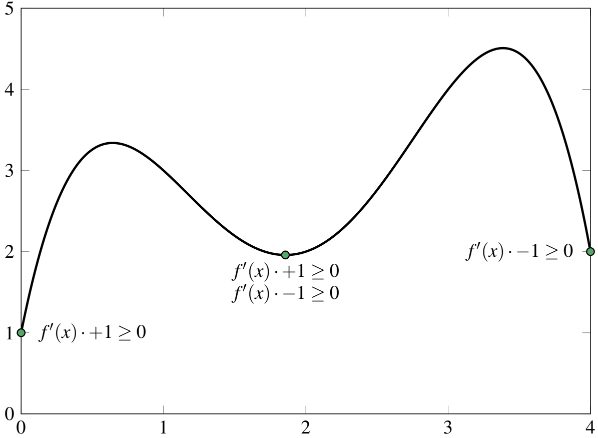
To minimise $f \in C^1(\mathbb{R}, \mathbb{R})$ over a nonempty, closed, convex interval $I \subset \mathbb{R}$, the necessary optimality condition is

$$x \in I : f'(x)(y - x) \geq 0 \,\forall\, y \in I.$$



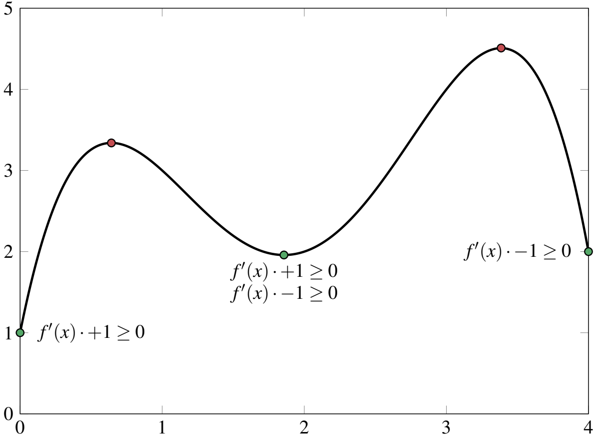
To minimise $f \in C^1(\mathbb{R}, \mathbb{R})$ over a nonempty, closed, convex interval $I \subset \mathbb{R}$, the necessary optimality condition is

$$x \in I : f'(x)(y - x) \geq 0 \, \forall \, y \in I.$$



To minimise $f \in C^1(\mathbb{R}, \mathbb{R})$ over a nonempty, closed, convex interval $I \subset \mathbb{R}$, the necessary optimality condition is

$$x \in I : f'(x)(y - x) \geq 0 \,\forall\, y \in I.$$



Three levels of difficulty:

linear PDE: $u \in V :$ $a(u, v) = L(v)\ \forall\, v \in V$

Three levels of difficulty:

linear PDE: $u \in V: \quad a(u,v) = L(v) \ \forall\, v \in V$

nonlinear PDE: $u \in V: \quad F(u;v) = 0 \quad \forall\, v \in V$

Three levels of difficulty:

linear PDE: $u \in V: \quad a(u,v) = L(v) \; \forall v \in V$

nonlinear PDE: $u \in V: \quad F(u;v) = 0 \quad \forall v \in V$

nonlinear VI: $u \in K \subsetneq V: \quad F(u; v-u) \geq 0 \quad \forall v \in K$

Three levels of difficulty:

linear PDE: $u \in V: \quad a(u,v) = L(v) \ \forall v \in V$

nonlinear PDE: $u \in V: \quad F(u;v) = 0 \quad \forall v \in V$

nonlinear VI: $u \in K \subsetneq V: \ F(u; v-u) \geq 0 \quad \forall v \in K$

This talk

A new framework for solving infinite-dimensional variational inequalities ...

...with major advantages over existing algorithms.

Section 2

Latent variable proximal point

Latent variable proximal point (LVPP) for solving VIs

is a reformulation of the *Bregman proximal point* algorithm to solve for several variables.

Latent variable proximal point (LVPP) for solving VIs

is a reformulation of the *Bregman proximal point* algorithm to solve for several variables.

The *Bregman distance* is induced by a *Legendre function*.

Latent variable proximal point (LVPP) for solving VIs

is a reformulation of the *Bregman proximal point* algorithm to solve for several variables.

The *Bregman distance* is induced by a *Legendre function*.

The Legendre function encodes the geometry of the constraints at each point.

Latent variable proximal point (LVPP) for solving VIs

is a reformulation of the *Bregman proximal point* algorithm to solve for several variables.

The *Bregman distance* is induced by a *Legendre function*.

The Legendre function encodes the geometry of the constraints at each point.

The reformulation is equivalent to the original one, but much better for approximation.

Latent variable proximal point (LVPP) for solving VIs

is a reformulation of the *Bregman proximal point* algorithm to solve for several variables.

The *Bregman distance* is induced by a *Legendre function*.

The Legendre function encodes the geometry of the constraints at each point.

The reformulation is equivalent to the original one, but much better for approximation.

How it works

LVPP breaks down a VI into a sequence of nonlinear PDE solves.

Latent variable proximal point (LVPP) for solving VIs

is a reformulation of the *Bregman proximal point* algorithm to solve for several variables.

The *Bregman distance* is induced by a *Legendre function*.

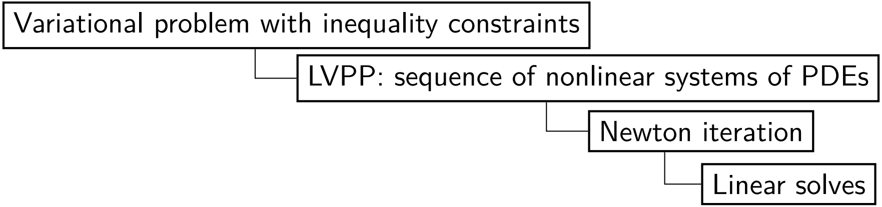
The Legendre function encodes the geometry of the constraints at each point.

The reformulation is equivalent to the original one, but much better for approximation.

How it works

LVPP breaks down a VI into a sequence of nonlinear PDE solves.

We then use Newton's method to break down nonlinear PDE solves into linear PDE solves.



- Variational problem with inequality constraints
  - LVPP: sequence of nonlinear systems of PDEs
    - Newton iteration
      - Linear solves

Schematic solver diagram.

Subsection 1

Legendre functions

The obstacle problem has feasible set

$$K = \left\{ v \in H_0^1(\Omega) \mid v(x) \geq \phi(x) \text{ a.e. in } \Omega \right\}.$$

The obstacle problem has feasible set

$$K = \left\{ v \in H_0^1(\Omega) \mid v(x) \geq \phi(x) \text{ a.e. in } \Omega \right\}.$$

A hyperelastic beam might have feasible set

$$K = \left\{ v \in H^1(\Omega) \otimes \mathbb{R}^2 \mid (\operatorname{tr} v_2)(x) \in [-\phi(x), \phi(x)] \text{ a.e. on } \partial\Omega \right\}.$$

The obstacle problem has feasible set

$$K = \left\{ v \in H_0^1(\Omega) \mid v(x) \geq \phi(x) \text{ a.e. in } \Omega \right\}.$$

A hyperelastic beam might have feasible set

$$K = \left\{ v \in H^1(\Omega) \otimes \mathbb{R}^2 \mid (\operatorname{tr} v_2)(x) \in [-\phi(x), \phi(x)] \text{ a.e. on } \partial\Omega \right\}.$$

A problem with gradient constraints might have feasible set like

$$K = \left\{ v \in H_0^1(\Omega) \mid |\nabla v(x)| \leq \phi(x) \text{ a.e. in } \Omega \right\}.$$

The obstacle problem has feasible set

$$K = \left\{ v \in H_0^1(\Omega) \mid v(x) \geq \phi(x) \text{ a.e. in } \Omega \right\}.$$

A hyperelastic beam might have feasible set

$$K = \left\{ v \in H^1(\Omega) \otimes \mathbb{R}^2 \mid (\operatorname{tr} v_2)(x) \in [-\phi(x), \phi(x)] \text{ a.e. on } \partial\Omega \right\}.$$

A problem with gradient constraints might have feasible set like

$$K = \left\{ v \in H_0^1(\Omega) \mid |\nabla v(x)| \leq \phi(x) \text{ a.e. in } \Omega \right\}.$$

Our general feasible set

$$K = \left\{ v \in V \mid Bv \in C(x) \text{ a.e. in } \Omega_d \subset \overline{\Omega} \right\}.$$

The obstacle problem has feasible set

$$K = \left\{ v \in H_0^1(\Omega) \mid v(x) \geq \phi(x) \text{ a.e. in } \Omega \right\}.$$

A hyperelastic beam might have feasible set

$$K = \left\{ v \in H^1(\Omega) \otimes \mathbb{R}^2 \mid (\operatorname{tr} v_2)(x) \in [-\phi(x), \phi(x)] \text{ a.e. on } \partial\Omega \right\}.$$

A problem with gradient constraints might have feasible set like

$$K = \left\{ v \in H_0^1(\Omega) \mid |\nabla v(x)| \leq \phi(x) \text{ a.e. in } \Omega \right\}.$$

Our general feasible set

$$K = \left\{ v \in V \mid Bv \in C(x) \text{ a.e. in } \Omega_d \subset \overline{\Omega} \right\}.$$

Here $C(x) \subset \mathbb{R}^m$ is the nonempty, closed, convex *feasible image* at x.

Legendre functions are a class of proper convex functions. They *encode the geometry of the feasible image* $C_x = C(x)$.

Legendre functions are a class of proper convex functions. They *encode the geometry of the feasible image* $C_x = C(x)$.

$R_x : C_x \to \mathbb{R} \cup \{\infty\}$ has domain C_x and *singular gradient* ∇R_x *on* ∂C_x.

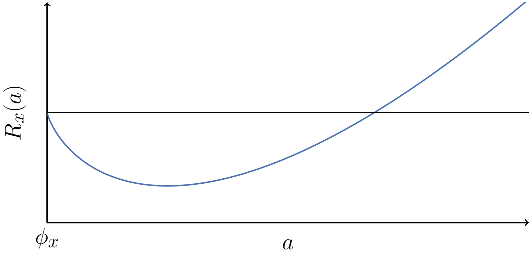
Legendre functions are a class of proper convex functions. They *encode the geometry of the feasible image* $C_x = C(x)$.

$R_x : C_x \to \mathbb{R} \cup \{\infty\}$ has domain C_x and *singular gradient* ∇R_x *on* ∂C_x.

Example

For the obstacle problem, $C_x = [\phi_x, \infty)$, and we choose a modified Shannon entropy:

$$R_x(a) = (a - \phi_x)\log(a - \phi_x) - (a - \phi_x), \quad \nabla R_x(a) = \log(a - \phi_x).$$



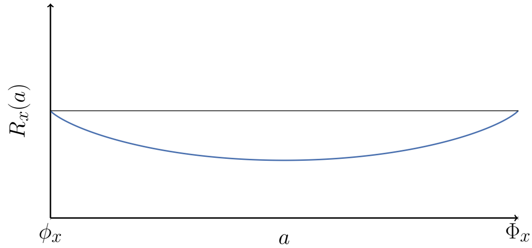
Legendre functions are a class of proper convex functions. They *encode the geometry of the feasible image* $C_x = C(x)$.

$R_x : C_x \to \mathbb{R} \cup \{\infty\}$ has domain C_x and *singular gradient* ∇R_x *on* ∂C_x.

Example

For a *double obstacle* problem, $C_x = [\phi_x, \Phi_x]$, and we choose the Fermi–Dirac entropy:

$$R_x(a) = (a - \phi_x) \log (a - \phi_x) + (\Phi_x - a) \log (\Phi_x - a), \quad \nabla R_x(a) = \log (a - \phi_x) + \log (\Phi_x - a).$$



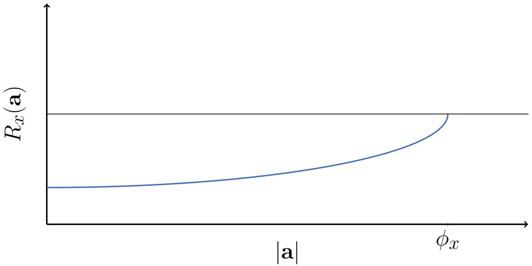
Legendre functions are a class of proper convex functions. They *encode the geometry of the feasible image* $C_x = C(x)$.

$R_x : C_x \to \mathbb{R} \cup \{\infty\}$ has domain C_x and *singular gradient* ∇R_x *on* ∂C_x.

Example

For gradient constraints, $B = \nabla$, $C_x = \mathcal{B}(0, \phi_x)$, and we choose a modified Hellinger entropy:

$$R_x(\mathbf{a}) = -\sqrt{\phi_x^2 - |\mathbf{a}|^2}, \quad \nabla R_x(\mathbf{a}) = \mathbf{a}/\sqrt{\phi_x^2 - |\mathbf{a}|^2}.$$



Why is this description useful?

Why is this description useful?

Because R_x induces an isomorphism between $\operatorname{int} C_x$ and $\mathbb{R}^m$!

Why is this description useful?

Because R_x induces an isomorphism between $\operatorname{int} C_x$ and $\mathbb{R}^m$!

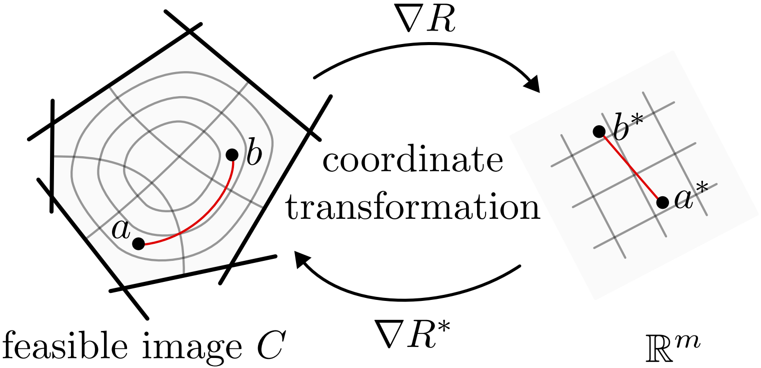
Theorem (Rockafellar (1967))

A proper convex function R is a Legendre function if and only if its convex conjugate R^* is also a Legendre function. Moreover,

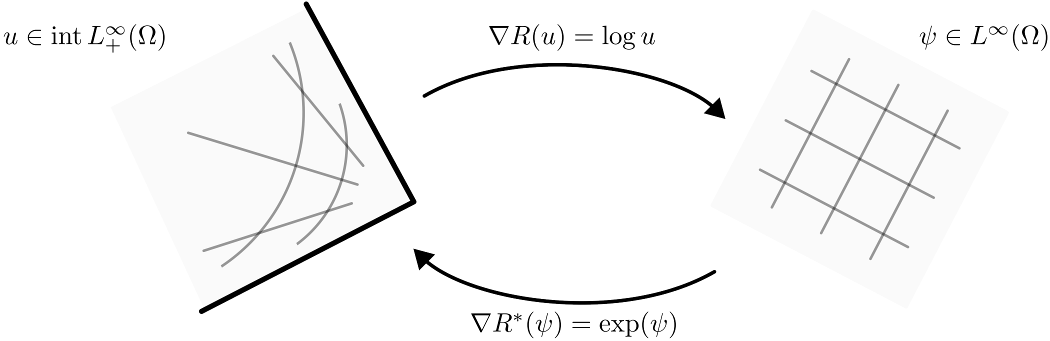
$$\nabla R\colon \operatorname{int}(\operatorname{dom} R) \to \operatorname{int}(\operatorname{dom} R^*)$$

is a topological isomorphism with $(\nabla R)^{-1} = \nabla R^*$.

R. Tyrell Rockafellar


∇R
coordinate transformation
∇R^*
a
b
b^*
a^*
feasible image C
$\mathbb{R}^m$

Applying this idea at every point, for the obstacle problem with $\phi = 0$, we have



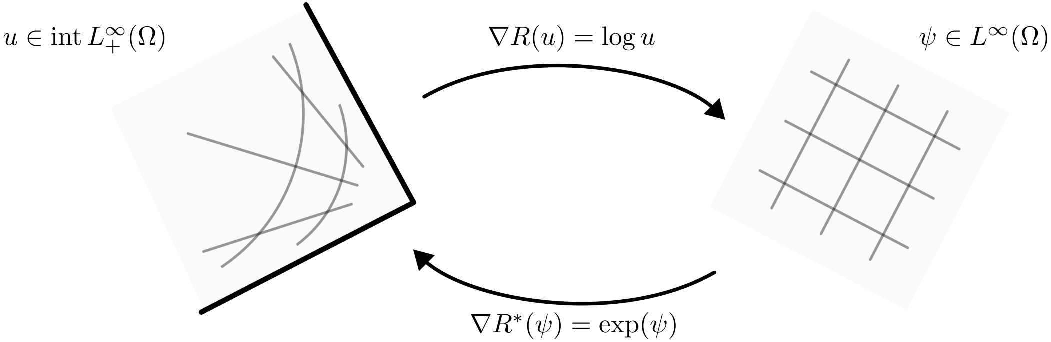
$u \in \operatorname{int} L^\infty_+(\Omega)$

$\nabla R(u) = \log u$

$\psi \in L^\infty(\Omega)$

$\nabla R^*(\psi) = \exp(\psi)$

Applying this idea at every point, for the obstacle problem with $\phi = 0$, we have



$u \in \operatorname{int} L^\infty_+(\Omega)$

$\nabla R(u) = \log u$

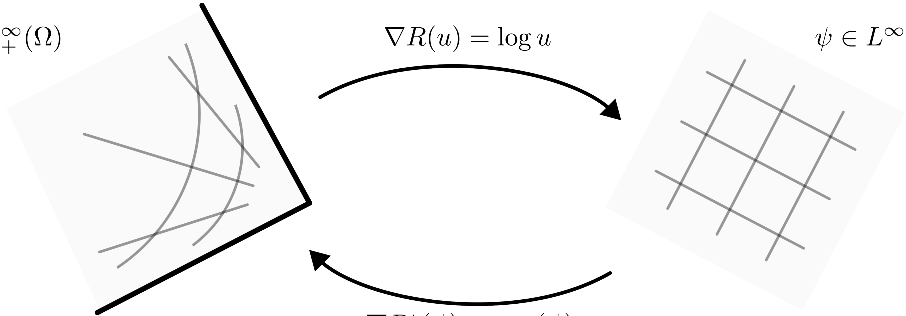
$\psi \in L^\infty(\Omega)$

$\nabla R^*(\psi) = \exp(\psi)$

Good news

We can represent any feasible function with a *latent variable* in a $\mathbb{R}^m$-valued Banach space!

Applying this idea at every point, for the obstacle problem with $\phi = 0$, we have



$u \in \operatorname{int} L^\infty_+(\Omega)$

$\nabla R(u) = \log u$

$\psi \in L^\infty(\Omega)$

$\nabla R^*(\psi) = \exp(\psi)$

Good news

We can represent any feasible function with a *latent variable* in a $\mathbb{R}^m$-valued Banach space!

...or more precisely any *strictly* feasible function.

Subsection 2

Proximal point

Proximal point is a fundamental algorithm in nonsmooth, convex optimisation.

To solve

$$u \in \operatorname*{argmin}_{v \in K} \; J(v)$$

Bernard Martinet

Osman Güler

Proximal point is a fundamental algorithm in nonsmooth, convex optimisation.

To solve

$$u \in \operatorname*{argmin}_{v \in K} J(v)$$

we iterate

$$u^k \in \operatorname*{argmin}_{v \in K} \left\{ J(v) + \frac{1}{\alpha^k} \|v - u_{k-1}\|_V^2 \right\} \quad \text{for} \quad \{\alpha^k\},\ \alpha^k > 0.$$

Bernard Martinet

Osman Güler

Proximal point is a fundamental algorithm in nonsmooth, convex optimisation.

To solve

$$u \in \operatorname*{argmin}_{v \in K} J(v)$$

we iterate

$$u^k \in \operatorname*{argmin}_{v \in K} \left\{ J(v) + \frac{1}{\alpha^k} \|v - u_{k-1}\|_V^2 \right\} \quad \text{for} \quad \{\alpha^k\},\ \alpha^k > 0.$$

The idea is to *adaptively regularise the problem* with information from our current iterate.

Bernard Martinet

Osman Güler

Proximal point is a fundamental algorithm in nonsmooth, convex optimisation.

To solve

$$u \in \operatorname*{argmin}_{v \in K} \; J(v)$$

we iterate

$$u^k \in \operatorname*{argmin}_{v \in K} \left\{ J(v) + \frac{1}{\alpha^k} \|v - u_{k-1}\|_V^2 \right\} \quad \text{for} \quad \{\alpha^k\},\ \alpha^k > 0.$$

Bernard Martinet

The idea is to *adaptively regularise the problem* with information from our current iterate.

Amazingly, for strictly convex J, this converges arbitrarily quickly:

$$J(u^k) - J(u) \leq \frac{\|u^0 - u\|_V^2}{\sum_{i=1}^k \alpha_i}.$$

Osman Güler

Bad news

Applying proximal point to the obstacle problem gives *subproblems that are as hard to solve.*

Bad news

Applying proximal point to the obstacle problem gives *subproblems that are as hard to solve.*

The regularisation

$$\frac{1}{\alpha^k}\|v - u_{k-1}\|_V^2$$

does *not respect the geometry of the constraints*. The subproblems remain VIs.

Bad news

Applying proximal point to the obstacle problem gives *subproblems that are as hard to solve.*

The regularisation

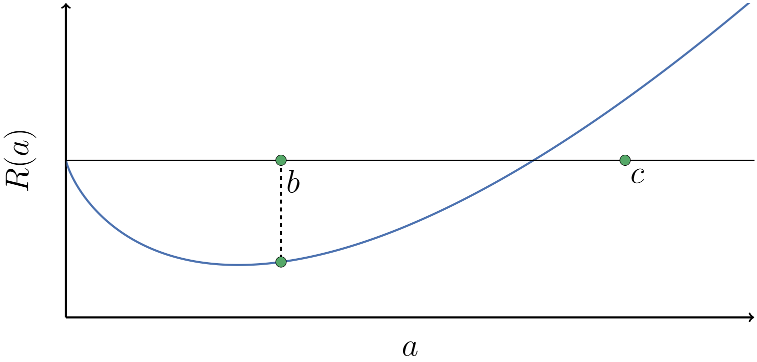
$$\frac{1}{\alpha^k}\|v - u_{k-1}\|_V^2$$

does *not respect the geometry of the constraints*. The subproblems remain VIs.

Good news

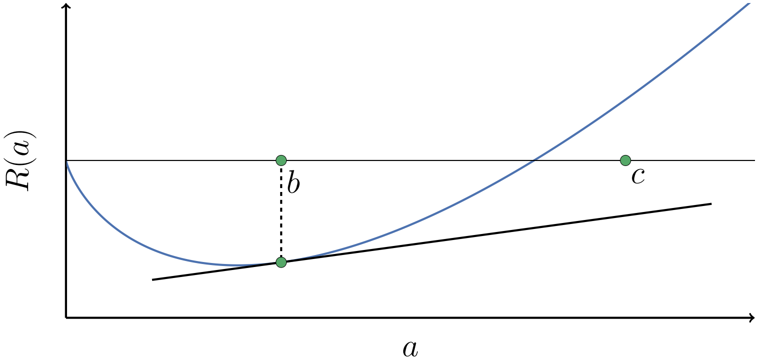
The Legendre function gives a notion of distance where the subproblems *do* simplify.

To define the *Bregman distance* $D_R(c, b)$ between base b and c, proceed as follows.



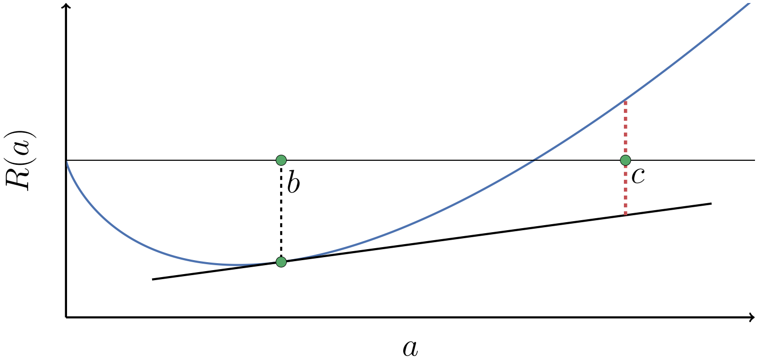
Start with the Legendre function R.

To define the *Bregman distance* $D_R(c, b)$ between base b and c, proceed as follows.



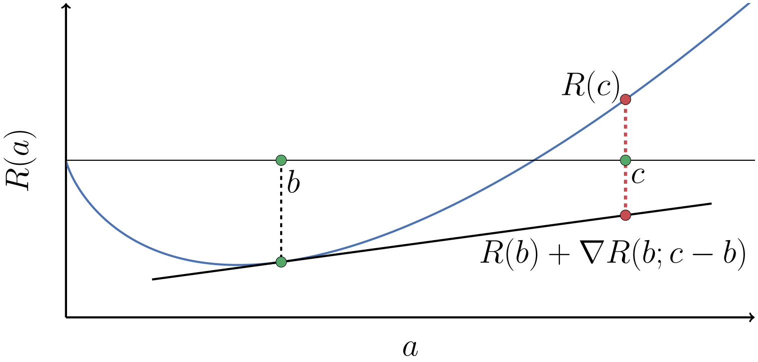
Build the tangent at b.

To define the *Bregman distance* $D_R(c,b)$ between base b and c, proceed as follows.



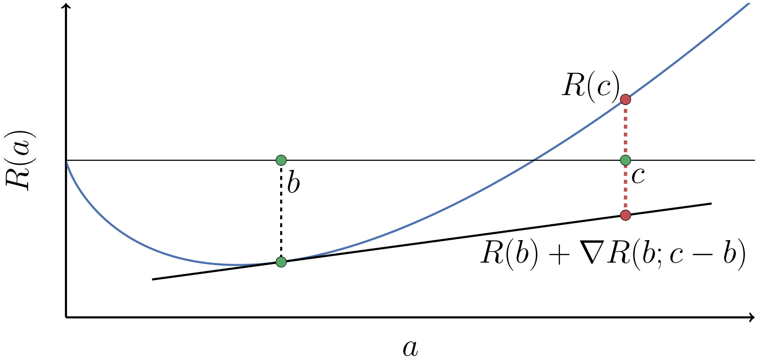
Measure distance between tangent and R at c.

To define the *Bregman distance* $D_R(c, b)$ between base b and c, proceed as follows.



$$D_R(c, b) = R(c) - R(b) - \nabla R(b; c - b).$$

To define the *Bregman distance* $D_R(c, b)$ between base b and c, proceed as follows.



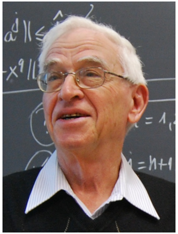
$$D_R(c, b) = R(c) - R(b) - \nabla R(b; c - b).$$

Example

If R is the Shannon entropy, then D_R is the Kullback–Leibler divergence.

To solve

$$u \in \operatorname*{argmin}_{v\in K} \; J(v)$$



Yair Censor

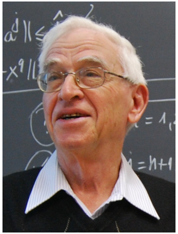
Stavros Zenios

To solve

$$u \in \operatorname*{argmin}_{v \in K} \; J(v)$$

with Bregman proximal point, we iterate

$$u^k \in \operatorname*{argmin}_{v \in K} \left\{ J(v) + \frac{1}{\alpha^k} \int_{\Omega_d} D_R(Bv, Bu^{k-1}) \, \mathrm{d}x \right\} \quad \text{for} \quad \{\alpha^k\}, \; \alpha^k > 0.$$



Yair Censor

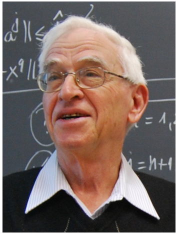
Stavros Zenios

To solve

$$u \in \operatorname*{argmin}_{v \in K} J(v)$$

with Bregman proximal point, we iterate

$$u^k \in \operatorname*{argmin}_{v \in K} \left\{ J(v) + \frac{1}{\alpha^k} \int_{\Omega_d} D_R(Bv, Bu^{k-1}) \, \mathrm{d}x \right\} \quad \text{for} \quad \{\alpha^k\},\ \alpha^k > 0.$$



Yair Censor

Good news

For many problems, this forces u^k to be strictly feasible, and the subproblem optimality condition *becomes a PDE*:

$$u^k \in K : \alpha^k J'(u^k) + B^* \nabla R(Bu^k) - B^* \nabla R(Bu^{k-1}) = 0.$$

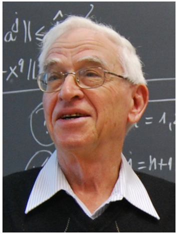
Stavros Zenios

To solve

$$u \in \operatorname*{argmin}_{v \in K} J(v)$$

with Bregman proximal point, we iterate

$$u^k \in \operatorname*{argmin}_{v \in K} \left\{ J(v) + \frac{1}{\alpha^k} \int_{\Omega_d} D_R(Bv, Bu^{k-1}) \, \mathrm{d}x \right\} \quad \text{for} \quad \{\alpha^k\},\ \alpha^k > 0.$$



Yair Censor

Good news

For many problems, this forces u^k to be strictly feasible, and the subproblem optimality condition *becomes a PDE*:

$$u^k \in K : \alpha^k J'(u^k) + B^* \nabla R(Bu^k) - B^* \nabla R(Bu^{k-1}) = 0.$$

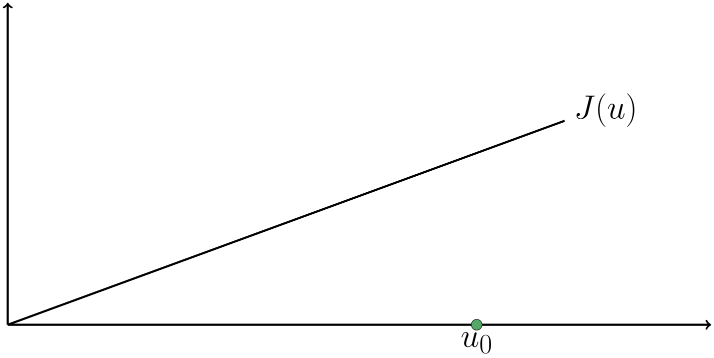
Stavros Zenios

This breaks down the VI into a sequence of nonlinear PDEs!

Consider the toy problem

$$u \in \operatorname*{argmin}_{v \in [0,\infty)} J(v) = v$$

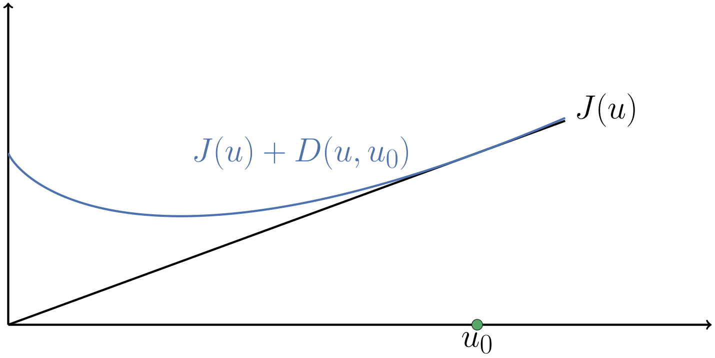
with solution $u = 0$.



Consider the toy problem

$$u \in \operatorname*{argmin}_{v \in [0,\infty)} J(v) = v$$

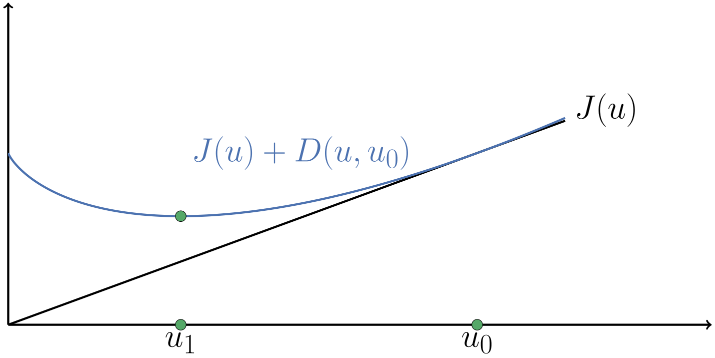
with solution $u = 0$.



Consider the toy problem

$$u \in \operatorname*{argmin}_{v \in [0,\infty)} J(v) = v$$

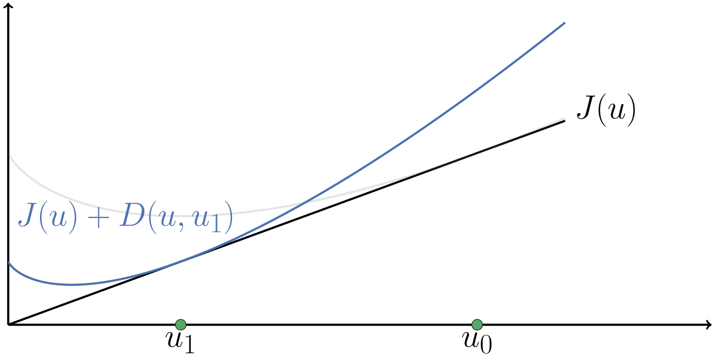
with solution $u = 0$.



Consider the toy problem

$$u \in \operatorname*{argmin}_{v \in [0,\infty)} J(v) = v$$

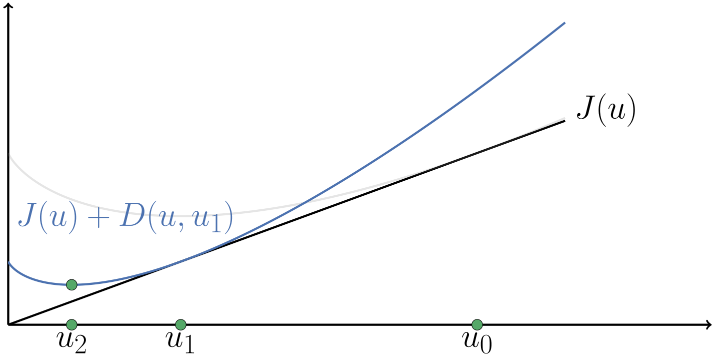
with solution $u = 0$.



Consider the toy problem

$$u \in \operatorname*{argmin}_{v\in[0,\infty)} J(v) = v$$

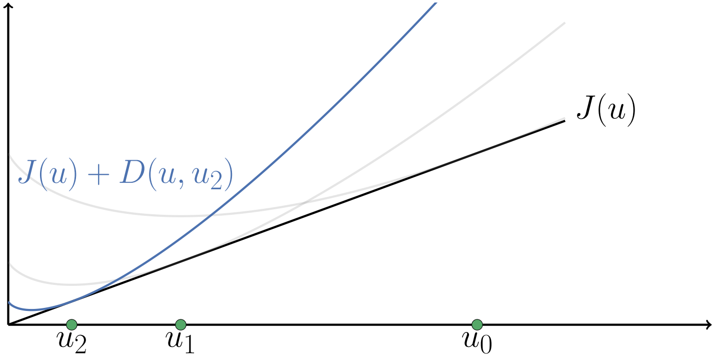
with solution $u = 0$.



Consider the toy problem

$$u \in \underset{v \in [0,\infty)}{\text{argmin}}\ J(v) = v$$

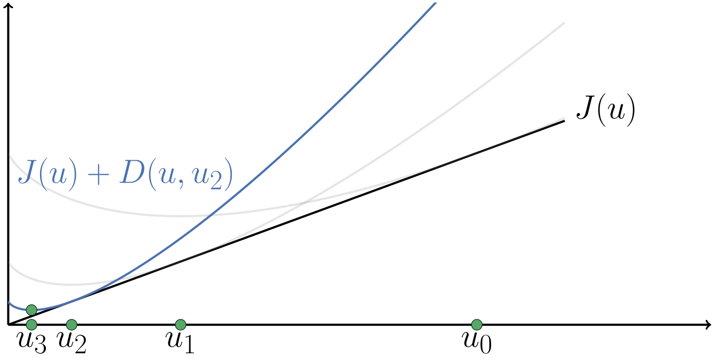
with solution $u = 0$.



Consider the toy problem

$$u \in \operatorname*{argmin}_{v \in [0,\infty)} J(v) = v$$

with solution $u = 0$.

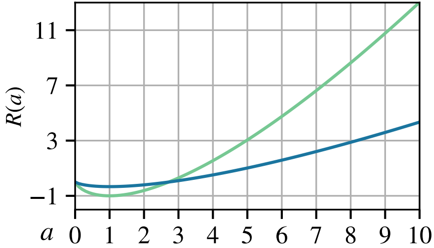


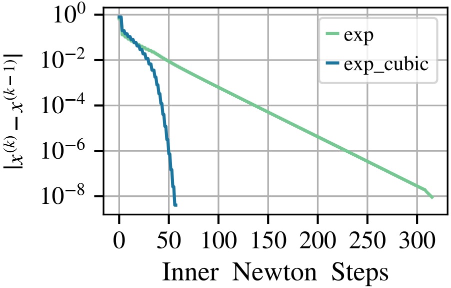
Observation

The choice of Legendre function is important!

Observation

The choice of Legendre function is important!

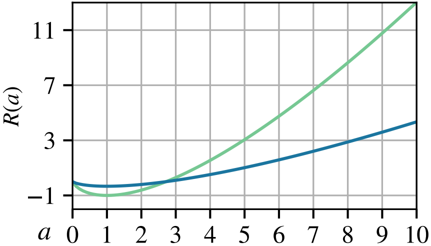


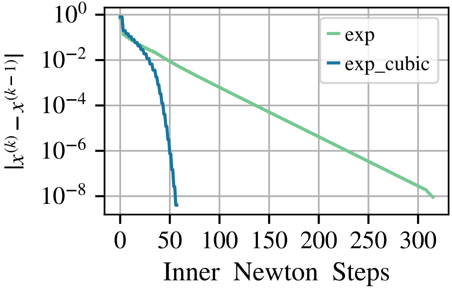


Jongho Park

Observation

The choice of Legendre function is important!





Jongho Park

Analysis

Less steep gradient at the boundary $\partial C(x)$ means faster convergence.

Subsection 3

Mixed formulation

Good news

The iterates are strictly feasible, so the optimality condition becomes a PDE:

$$u^k \in K : \alpha^k J'(u^k) + B^* \nabla R(B u^k) - B^* \nabla R(B u^{k-1}) = 0.$$

Good news

The iterates are strictly feasible, so the optimality condition becomes a PDE:

$$u^k \in K : \alpha^k J'(u^k) + B^* \nabla R(Bu^k) - B^* \nabla R(Bu^{k-1}) = 0.$$

Bad news

We still need to discretise $u \in K$, because R is only defined for feasible Bu.

Good news

The iterates are strictly feasible, so the optimality condition becomes a PDE:

$$u^k \in K : \alpha^k J'(u^k) + B^*\nabla R(Bu^k) - B^*\nabla R(Bu^{k-1}) = 0.$$

Bad news

We still need to discretise $u \in K$, because R is only defined for feasible Bu.

Idea

Introduce a latent variable $\psi \in W$ and express $Bu = \nabla R^*(\psi)$!

Latent variable proximal point

For some $\psi^0 \in W$, find $(u^k, \psi^k) \in V \times W$ s. t.

$$\alpha_k J'(u^k) + B^*\psi^k = B^*\psi^{k-1},$$
$$Bu^k - \nabla R^*(\psi^k) = 0.$$

Latent variable proximal point

For some $\psi^0 \in W$, find $(u^k, \psi^k) \in V \times W$ s. t.

$$\alpha_k J'(u^k) + B^* \psi^k = B^* \psi^{k-1},$$
$$B u^k - \nabla R^*(\psi^k) = 0.$$

Important observation

This new formulation only requires discretising $u \in V$, not $u \in K$!

This is much, much simpler, because V is a Banach space but K is not.

Section 3

Bound constraints

Consider again the obstacle problem:

$$u \in \underset{v \in K}{\operatorname{argmin}}\ J(v) = \frac{1}{2} \int_\Omega \nabla v \cdot \nabla v \ \mathrm{d}x - \int_\Omega f v \ \mathrm{d}x,$$

for feasible set

$$K = \left\{ v \in H_0^1(\Omega) \mid v \geq \phi \text{ a.e. in } \Omega \right\}.$$

Consider again the obstacle problem:

$$u \in \underset{v \in K}{\text{argmin}}\ J(v) = \frac{1}{2}\int_\Omega \nabla v \cdot \nabla v \,\mathrm{d}x - \int_\Omega f v \,\mathrm{d}x,$$

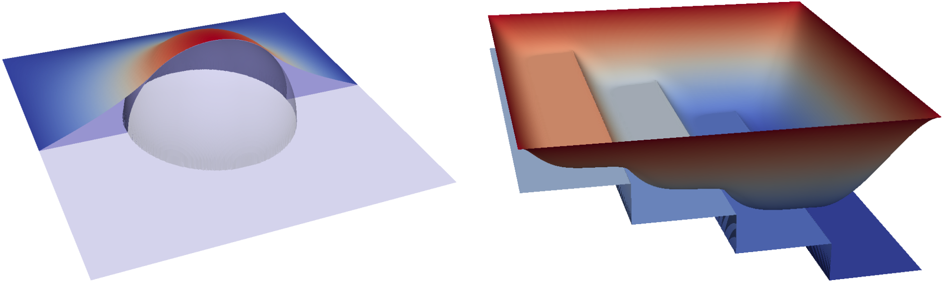
for feasible set

$$K = \left\{ v \in H_0^1(\Omega) \mid v \geq \phi \text{ a.e. in } \Omega \right\}.$$

The LVPP formulation becomes: for $\psi^0 = 0$, find $(u^k, \psi^k) \in H_0^1(\Omega) \times L^\infty(\Omega)$ s. t.

$$\begin{aligned} \alpha_k(\nabla u^k, \nabla v) + (\psi^k, v) &= \alpha_k(f, v) + (\psi^{k-1}, v), \\ (u^k, w) - (\exp(\psi^k) + \phi, w) &= 0, \end{aligned}$$

for all $(v, w) \in H_0^1(\Omega) \times L^\infty(\Omega)$.

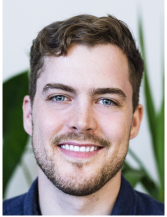


Good news

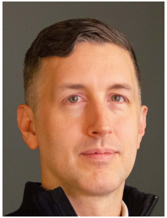
Complete mesh independence *and* strict feasibility for *any* approximation order.

For the proof that the Bregman proximal point iterations for the obstacle problem are PDEs, not VIs, see

B. Keith and T. M. Surowiec. “Proximal Galerkin: a structure-preserving finite element method for pointwise bound constraints”. In: *Foundations of Computational Mathematics* (2024). DOI: 10.1007/s10208-024-09681-8.



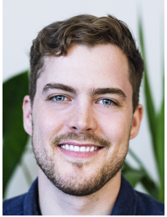
Brendan Keith



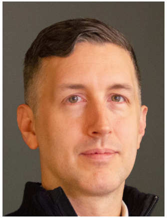
Thomas Surowiec

For the proof that the Bregman proximal point iterations for the obstacle problem are PDEs, not VIs, see

B. Keith and T. M. Surowiec. “Proximal Galerkin: a structure-preserving finite element method for pointwise bound constraints”. In: *Foundations of Computational Mathematics* (2024). DOI: 10.1007/s10208-024-09681-8.



Brendan Keith



Thomas Surowiec

For the proof that the convergence of proximal Galerkin is mesh-independent for the obstacle and Signorini problems, see

B. Keith, R. Masri, and M. Zeinhofer. *A priori error analysis of the proximal Galerkin method.* arXiv:2507.13516. 2025.

Rami Masri

Marius Zeinhofer

There are several algorithms for obstacle-type VIs. How does LVPP compare?

	feasible?	inf-dim?	mesh-indep?	no param to $0/\infty$?
active set/semismooth Newton	✓	✓	✗	✓
penalty/augmented Lagrangian	✗	✓	✓	✗
monotone multigrid	✓	✗	✓	✓
interior point	✓	✓	✓	✗
latent variable proximal point	✓	✓	✓	✓

Latent variable proximal point

Only method that combines all desirable properties!

Section 4

Gradient constraints

Take again the Dirichlet energy

$$u \in \operatorname*{argmin}_{v \in K} J(v) = \frac{1}{2} \int_\Omega \nabla v \cdot \nabla v \, \mathrm{d}x - \int_\Omega f v \, \mathrm{d}x,$$

Take again the Dirichlet energy

$$u \in \operatorname*{argmin}_{v \in K} J(v) = \frac{1}{2} \int_\Omega \nabla v \cdot \nabla v \, \mathrm{d}x - \int_\Omega f v \, \mathrm{d}x,$$

but now impose *both obstacle and gradient* constraints:

$$K = \left\{ v \in H_0^1(\Omega) \mid v \geq \phi \text{ and } |\nabla v| \leq \Phi \text{ a.e. in } \Omega \right\}.$$

Take again the Dirichlet energy

$$u \in \operatorname*{argmin}_{v \in K} J(v) = \frac{1}{2}\int_\Omega \nabla v \cdot \nabla v \,\mathrm{d}x - \int_\Omega f v \,\mathrm{d}x,$$

but now impose *both obstacle and gradient* constraints:

$$K = \left\{ v \in H_0^1(\Omega) \mid v \geq \phi \text{ and } |\nabla v| \leq \Phi \text{ a.e. in } \Omega \right\}.$$

Now $B = (\mathrm{id}, \nabla)^\top$ and $C(x) = [\phi(x), \infty) \times \mathcal{B}(0, \Phi(x))$, $\mathcal{B}$ the Euclidean ball.

Take again the Dirichlet energy

$$u \in \operatorname*{argmin}_{v \in K} J(v) = \frac{1}{2} \int_\Omega \nabla v \cdot \nabla v \, \mathrm{d}x - \int_\Omega f v \, \mathrm{d}x,$$

but now impose *both obstacle and gradient* constraints:

$$K = \left\{ v \in H_0^1(\Omega) \mid v \geq \phi \text{ and } |\nabla v| \leq \Phi \text{ a.e. in } \Omega \right\}.$$

Now $B = (\mathrm{id}, \nabla)^\top$ and $C(x) = [\phi(x), \infty) \times \mathcal{B}(0, \Phi(x))$, $\mathcal{B}$ the Euclidean ball.

Legendre functions for intersection

Legendre functions for intersections of sets are additive:

$$R(a, \mathbf{a}) = (a - \phi) \log (a - \phi) - (a - \phi) - \sqrt{\Phi^2 - |\mathbf{a}|^2}.$$

The induced isomorphism has two components:

$$\nabla R^* \left((a^*, \mathbf{a}^*) \right) = \begin{pmatrix} \phi + \exp a^* \\ \dfrac{\Phi \, \mathbf{a}^*}{\sqrt{1 + |\mathbf{a}^*|^2}} \end{pmatrix} .$$

The induced isomorphism has two components:

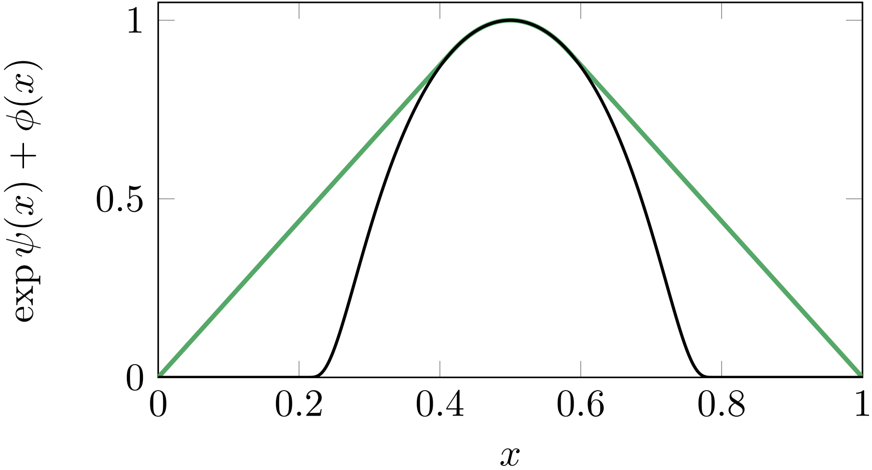
$$\nabla R^*\left((a^*, \mathbf{a}^*)\right) = \begin{pmatrix} \phi + \exp a^* \\ \dfrac{\Phi\,\mathbf{a}^*}{\sqrt{1 + |\mathbf{a}^*|^2}} \end{pmatrix}.$$

The LVPP iteration becomes: find $(u^k, \psi^k, \Psi^k) \in H^1(\Omega) \times L^\infty(\Omega) \times L^\infty(\Omega, \mathbb{R}^n)$ s. t.

$$\begin{aligned} \alpha_k(\nabla u^k, \nabla v) + (\psi^k, v) + (\Psi^k, \nabla v) &= (\psi^{k-1}, v) + (\Psi^{k-1}, \nabla v) \\ (u^k, w) - (\exp\psi^k + \phi, w) &= 0 \\ (\nabla u^k, W) - \left(\frac{\Phi\,\Psi^k}{\sqrt{1 + |\Psi^k|^2}}, W\right) &= 0 \end{aligned}$$

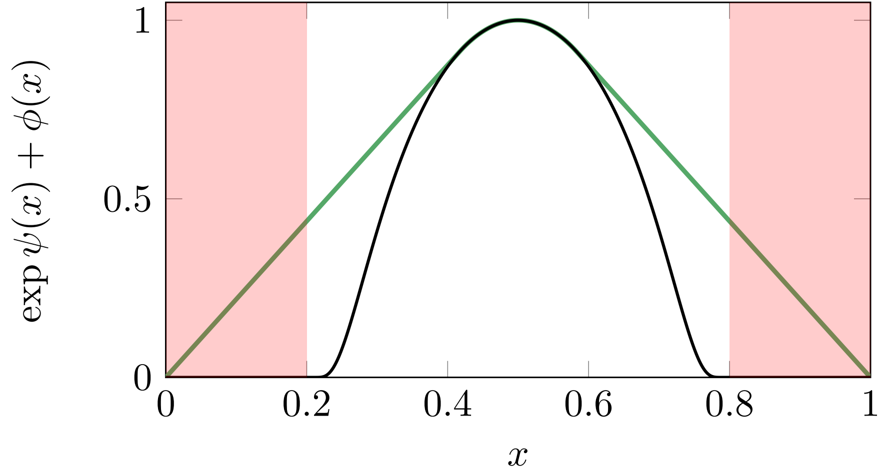
for all $(v, w, W) \in H^1(\Omega) \times L^\infty(\Omega) \times L^\infty(\Omega, \mathbb{R}^n)$.

We set $f = 0$ and vary the gradient constraints.



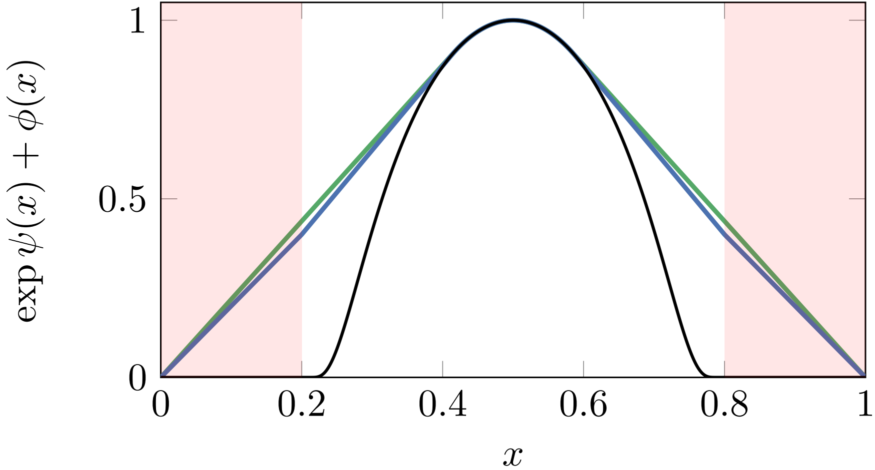
No gradient constraints.

We set $f = 0$ and vary the gradient constraints.



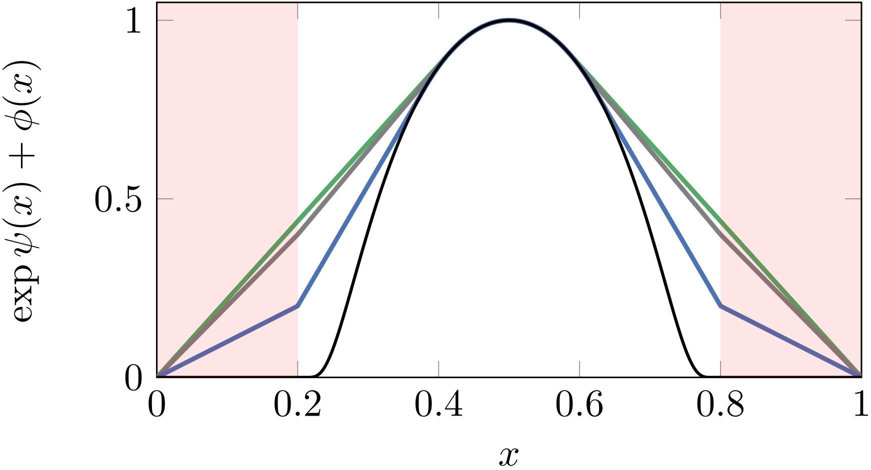
Apply gradient constraints on $[0, 0.2] \cup [0.8, 1]$.

We set $f = 0$ and vary the gradient constraints.



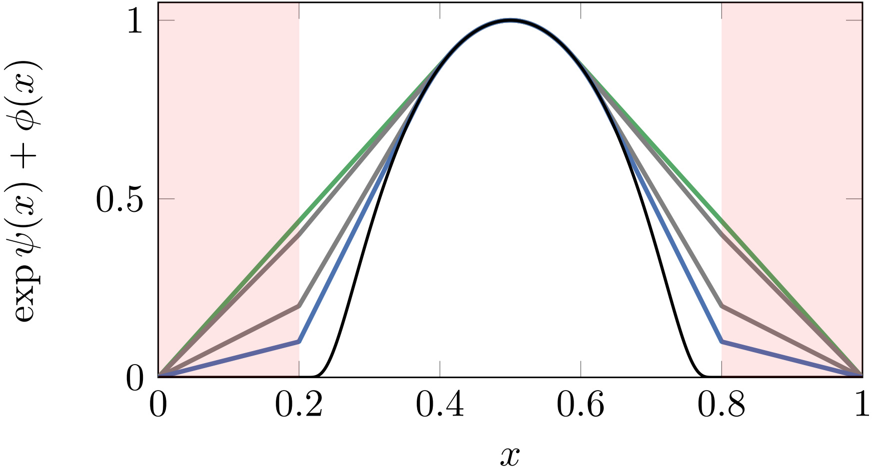
Light gradient constraints.

We set $f = 0$ and vary the gradient constraints.



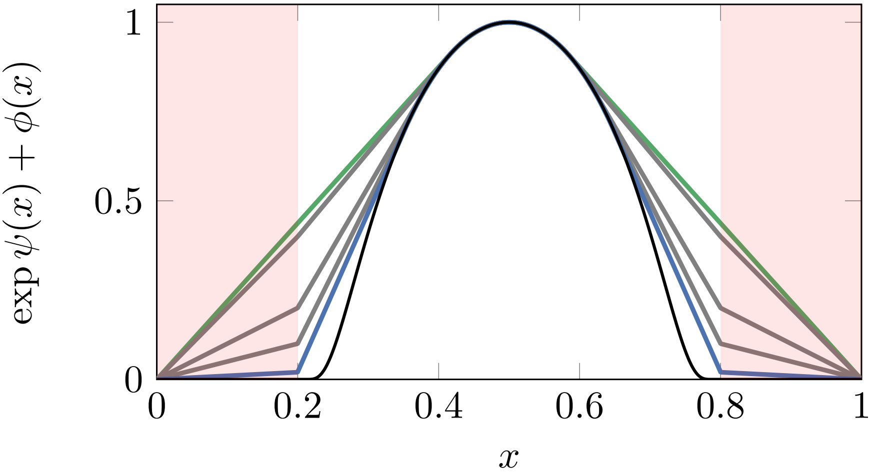
Medium gradient constraints.

We set $f = 0$ and vary the gradient constraints.



Heavy gradient constraints.

We set $f = 0$ and vary the gradient constraints.



Extreme gradient constraints.

Section 5

Eigenvalue constraints

Eigenvalue constraints are very important, but very difficult to enforce numerically.

Eigenvalue constraints are very important, but very difficult to enforce numerically.

The Landau–de Gennes model of nematic liquid crystals minimises

$$J(Q) = \frac{1}{2}\int_\Omega \nabla Q \,\vdots\, \nabla Q \ \mathrm{d}x + \frac{1}{2}\int_\Omega A\,\mathrm{tr}(Q^2) \ \mathrm{d}x + \frac{1}{4}\int_\Omega C(\mathrm{tr}(Q^2)^2) \ \mathrm{d}x$$

for a symmetric traceless matrix field Q.

Eigenvalue constraints are very important, but very difficult to enforce numerically.

The Landau–de Gennes model of nematic liquid crystals minimises

$$J(Q) = \frac{1}{2}\int_\Omega \nabla Q \vdots \nabla Q \ \mathrm{d}x + \frac{1}{2}\int_\Omega A\,\mathrm{tr}(Q^2)\ \mathrm{d}x + \frac{1}{4}\int_\Omega C(\mathrm{tr}(Q^2)^2)\ \mathrm{d}x$$

for a symmetric traceless matrix field Q.

To be physical, Q must satisfy eigenvalue constraints ($n =$ spatial dimension)

$$\lambda_i(Q) \in [-1/n, (n-1)/n], \quad i = 1, \ldots, n,$$

but this is usually ignored as too difficult.

Fix $n = 2$ for simplicity. We employ as Legendre function

$$R(A) = \operatorname{tr}\big((A + I/2)\log(A + I/2) + (I/2 - A)\log(I/2 - A)\big),$$

$$\text{with } \nabla R^*(A^*) = \operatorname{tanhm}(A^*/2)/2,$$

where $\log$ and tanhm are the matrix logarithm and hyperbolic tangent functions.

Fix $n = 2$ for simplicity. We employ as Legendre function

$$R(A) = \operatorname{tr}\left((A + I/2)\log(A + I/2) + (I/2 - A)\log(I/2 - A)\right),$$

with $\nabla R^*(A^*) = \operatorname{tanhm}(A^*/2)/2,$

where $\log$ and tanhm are the matrix logarithm and hyperbolic tangent functions.

The LVPP iteration becomes: find $(Q^k, \psi^k) \in H^1_D(\Omega, \mathbb{R}^{2\times 2}_{\text{sym,tr}}) \times L^\infty(\Omega, \mathbb{R}^{2\times 2}_{\mathrm{sym,tr}})$ s. t.

$$\alpha_k J'(Q; V) + (\psi^k, V) = (\psi^{k-1}, V)$$
$$(Q, w) - (\frac{1}{2}\operatorname{tanhm}(\psi/2), w) = 0$$

for all $(V, w) \in H^1_0(\Omega, \mathbb{R}^{2\times 2}_{\text{sym,tr}}) \times L^\infty(\Omega, \mathbb{R}^{2\times 2}_{\mathrm{sym,tr}})$.

Fix $n = 2$ for simplicity. We employ as Legendre function

$$R(A) = \operatorname{tr}\big((A + I/2)\log(A + I/2) + (I/2 - A)\log(I/2 - A)\big),$$

$$\text{with } \nabla R^*(A^*) = \operatorname{tanhm}(A^*/2)/2,$$

where $\log$ and tanhm are the matrix logarithm and hyperbolic tangent functions.

The LVPP iteration becomes: find $(Q^k, \psi^k) \in H^1_D(\Omega, \mathbb{R}^{2\times 2}_{\text{sym,tr}}) \times L^\infty(\Omega, \mathbb{R}^{2\times 2}_{\text{sym,tr}})$ s. t.

$$\alpha_k J'(Q; V) + (\psi^k, V) = (\psi^{k-1}, V)$$

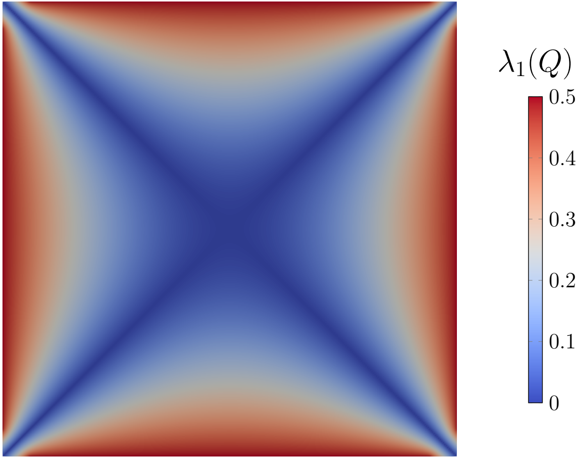
$$(Q, w) - (\frac{1}{2}\operatorname{tanhm}(\psi/2), w) = 0$$

for all $(V, w) \in H^1_0(\Omega, \mathbb{R}^{2\times 2}_{\text{sym,tr}}) \times L^\infty(\Omega, \mathbb{R}^{2\times 2}_{\text{sym,tr}})$.

All of this extends straightforwardly to $n > 2$.

Good news

Mesh-independent convergence, $\sim$ 6 proximal steps, $\sim$ 11 Newton iterations.



The larger eigenvalue $\lambda_1(Q)$. Both eigenvalues satisfy the inequality constraints.

Section 6

Multiple solutions

Many VIs support multiple solutions.

Many VIs support multiple solutions.

Deflation can compute multiple solutions of nonlinear PDEs. Can we use this?

Many VIs support multiple solutions.

Deflation can compute multiple solutions of nonlinear PDEs. Can we use this?

Idea: Voronoi–Bregman regularisation (primal)

Given a *set* S of solutions to the previous iteration, solve

$$u^k \in \operatorname*{argmin}_{v \in K} J(v) + \frac{1}{\alpha^k} \min_{s \in S} \int_{\Omega_d} D_R(Bv, Bs) \, \mathrm{d}x.$$

Chenghao Dong

Many VIs support multiple solutions.

Deflation can compute multiple solutions of nonlinear PDEs. Can we use this?

Idea: Voronoi–Bregman regularisation (primal)

Given a *set* S of solutions to the previous iteration, solve

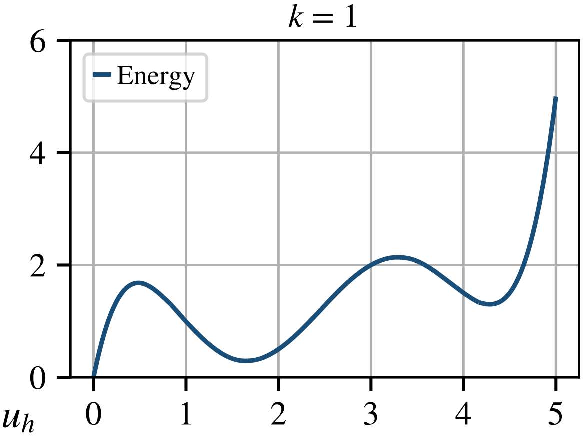
$$u^k \in \operatorname*{argmin}_{v \in K} J(v) + \frac{1}{\alpha^k} \min_{s \in S} \int_{\Omega_d} D_R(Bv, Bs) \, \mathrm{d}x.$$

Chenghao Dong

We can search for multiple solutions of this nonsmooth PDE at every LVPP iteration.

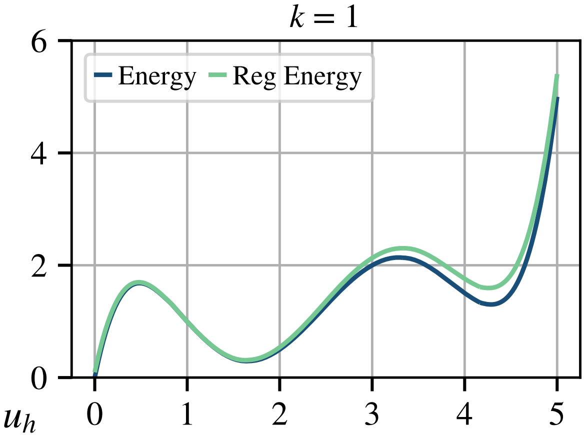
Use $u = 1$ as initial guess for the toy problem

$$u \in \underset{v \in [0,\infty)}{\text{argmin}} \; J(v)$$



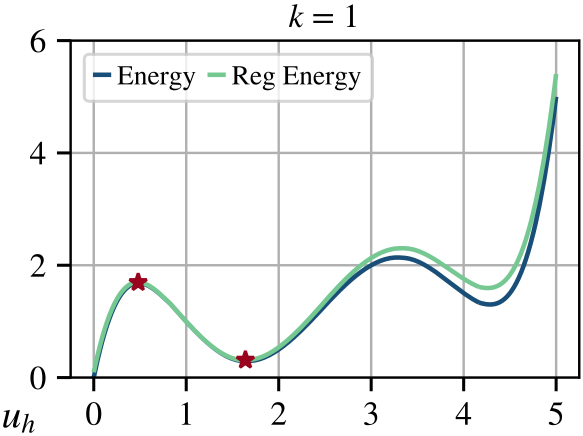
Use $u = 1$ as initial guess for the toy problem

$$u \in \operatorname*{argmin}_{v \in [0,\infty)} J(v)$$



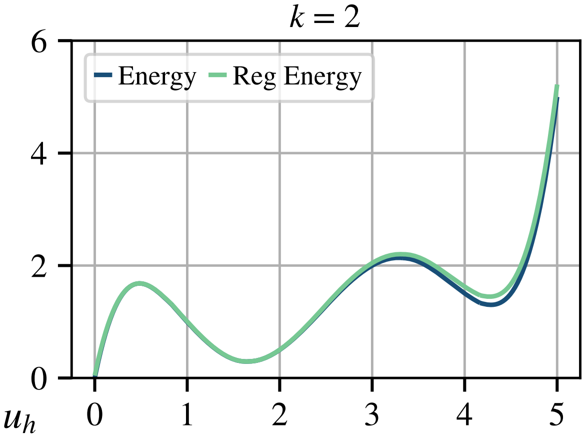
Use $u = 1$ as initial guess for the toy problem

$$u \in \operatorname*{argmin}_{v \in [0,\infty)} J(v)$$



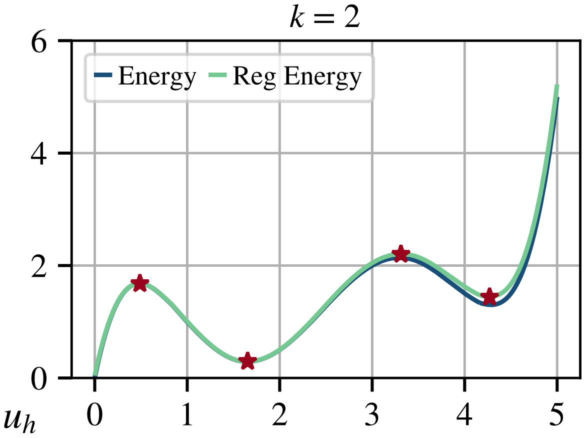
Use $u = 1$ as initial guess for the toy problem

$$u \in \operatorname*{argmin}_{v\in[0,\infty)} J(v)$$



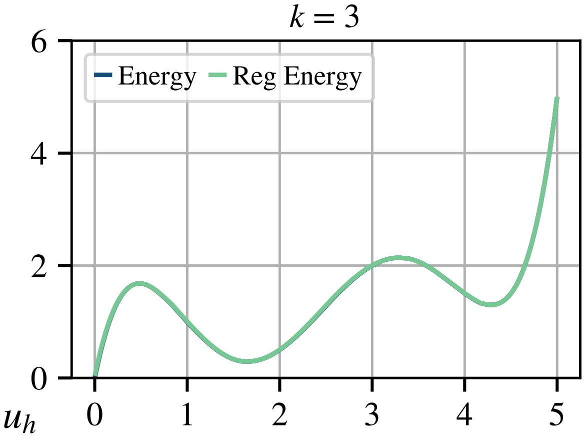
Use $u = 1$ as initial guess for the toy problem

$$u \in \operatorname*{argmin}_{v \in [0,\infty)} J(v)$$



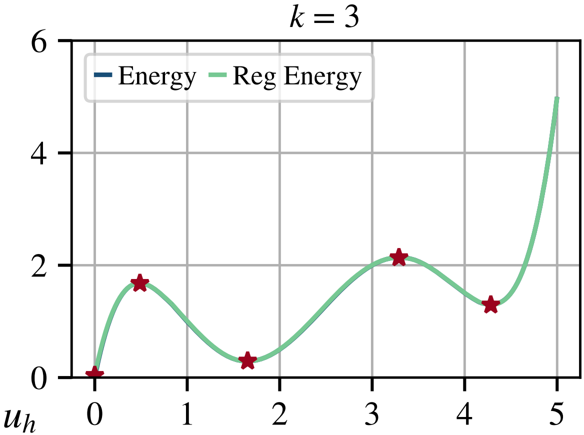
Use $u = 1$ as initial guess for the toy problem

$$u \in \operatorname*{argmin}_{v \in [0,\infty)} J(v)$$



Use $u = 1$ as initial guess for the toy problem

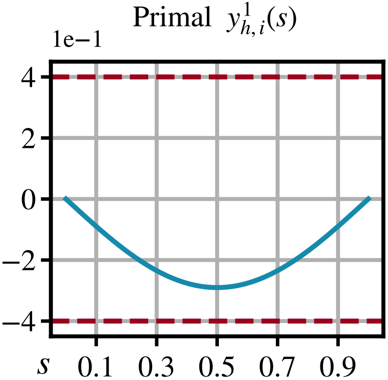
$$u \in \operatorname*{argmin}_{v \in [0,\infty)} J(v)$$



Consider the Zeidler beam problem

$$u \in \underset{v \in H^2(0,L) \cap H^1_0(0,L)}{\text{argmin}} \quad J(v) = \int_0^L B(v'')^2 - P(v')^2 - \rho g v \, \mathrm{d}x$$

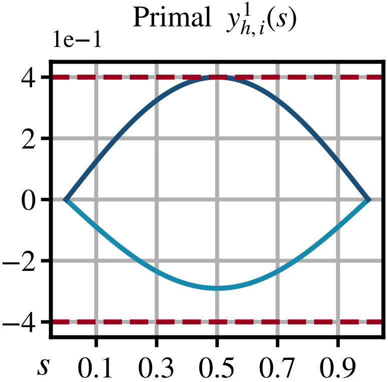
subject to $|y| \leq \alpha = 0.4$.



Consider the Zeidler beam problem

$$u \in \operatorname*{argmin}_{v \in H^2(0,L) \cap H_0^1(0,L)} J(v) = \int_0^L B(v'')^2 - P(v')^2 - \rho g v \,\mathrm{d}x$$

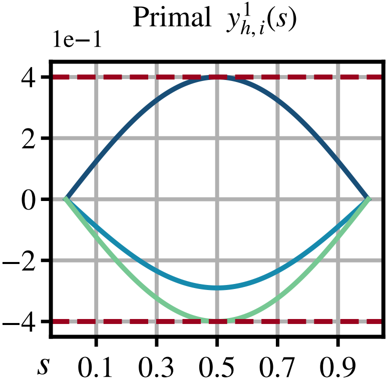
subject to $|y| \leq \alpha = 0.4$.



Consider the Zeidler beam problem

$$u \in \underset{v \in H^2(0,L) \cap H_0^1(0,L)}{\text{argmin}} \quad J(v) = \int_0^L B(v'')^2 - P(v')^2 - \rho g v \,\mathrm{d}x$$

subject to $|y| \leq \alpha = 0.4$.



Section 7

Conclusions

Conclusion

Latent variable proximal point is a powerful framework for problems with inequality constraints.

Conclusion

Latent variable proximal point is a powerful framework for problems with inequality constraints.

Good news

Many open questions remain! Proofs, discretisations, solvers, nonconvex constraints, ….