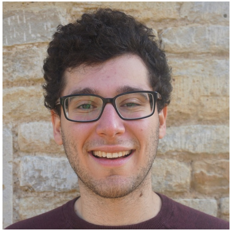
The latent variable proximal point algorithm for variational problems with inequality constraints

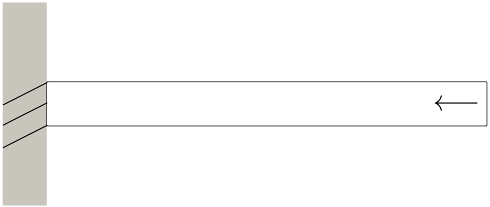
Patrick E. Farrell[1,2] Brendan Keith[3] Thomas Surowiec[4,3] Jørgen S. Dokken[4]
Ioannis P. A. Papadopoulos[5]

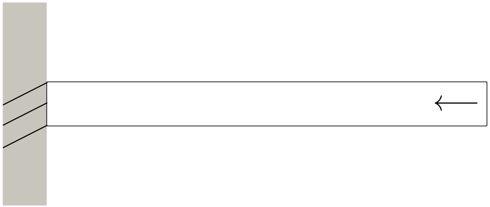
[1]University of Oxford [2]Charles University [3]Brown University [4]Simula Research Laboratory [5]WIAS

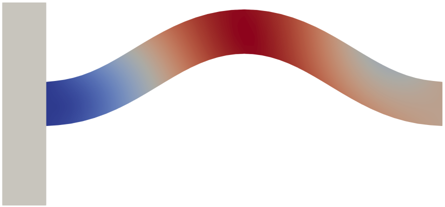
11 September 2025

Problems with inequality constraints are formulated as *variational inequalities* (VIs).

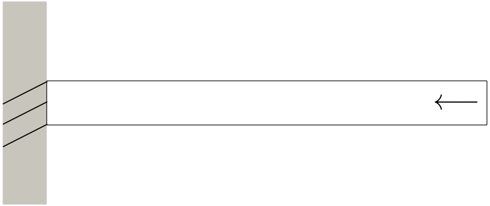


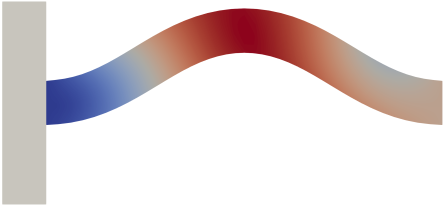
Problems with inequality constraints are formulated as *variational inequalities* (VIs).





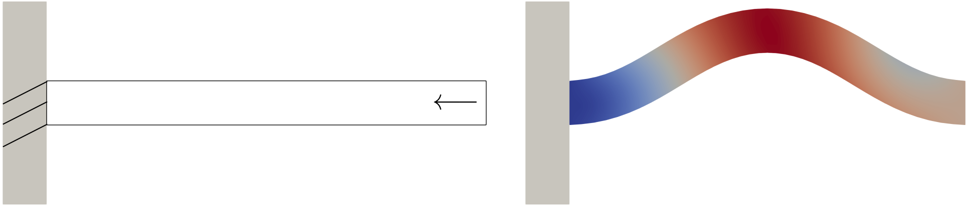
Problems with inequality constraints are formulated as *variational inequalities* (VIs).



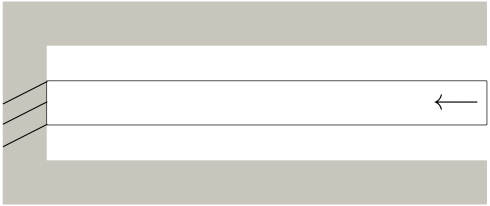


PDE

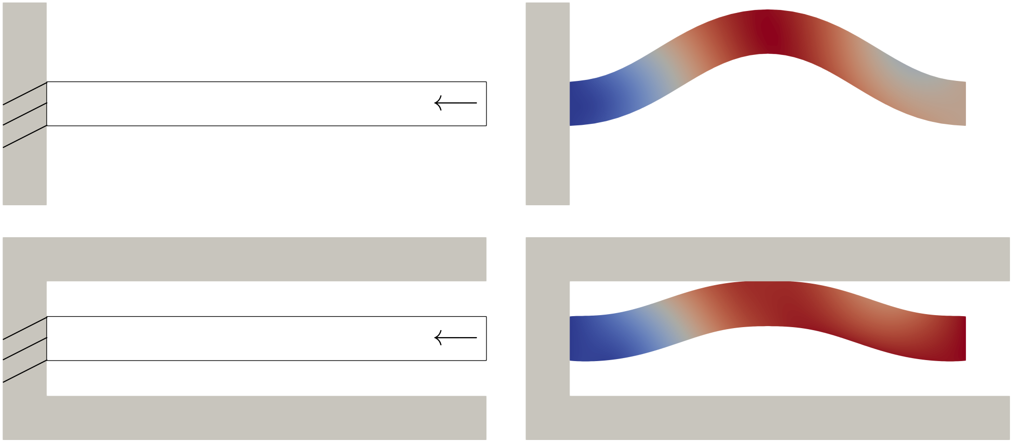
Problems with inequality constraints are formulated as *variational inequalities* (VIs).



PDE

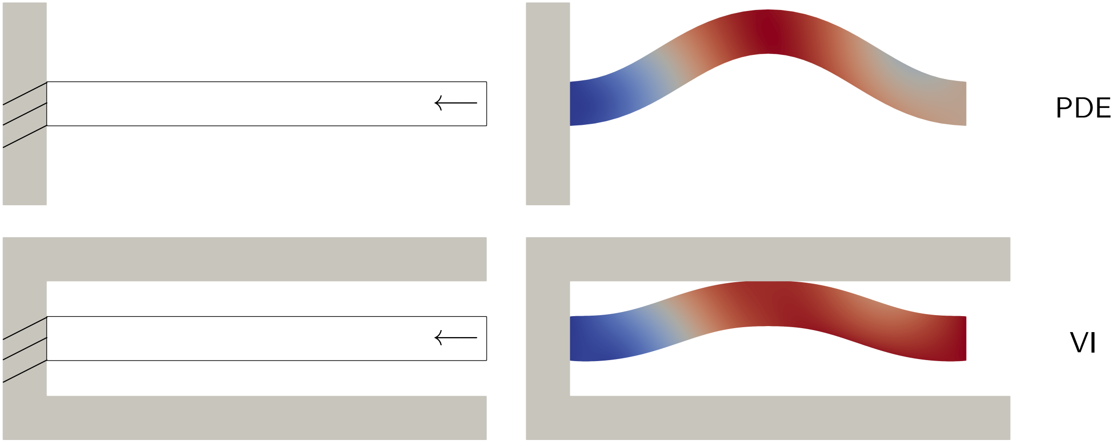


Problems with inequality constraints are formulated as *variational inequalities* (VIs).



PDE

Problems with inequality constraints are formulated as *variational inequalities* (VIs).



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

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 substantial advantages over existing methods.

Let's see an example of a VI. 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

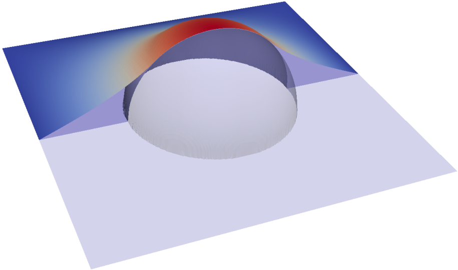
$$K = \left\{v \in H_0^1(\Omega) \mid v \geq \phi \text{ a.e. in } \Omega\right\} \subsetneq V \coloneqq H_0^1(\Omega).$$

Let's see an example of a VI. 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

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



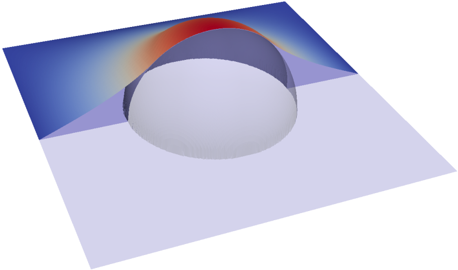
$f = 0, \phi = \frown$

Let's see an example of a VI. The obstacle problem is to minimise the energy

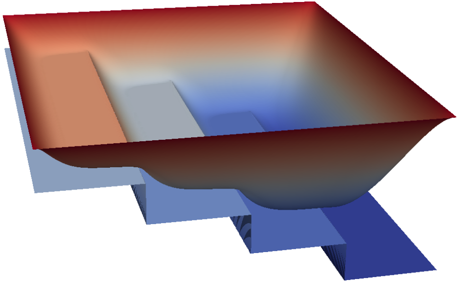
$$J(u) = \frac{1}{2}\int_\Omega \nabla u \cdot \nabla u \,\mathrm{d}x - \int_\Omega f u \,\mathrm{d}x$$

over the constrained set

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



$f = 0, \phi =$



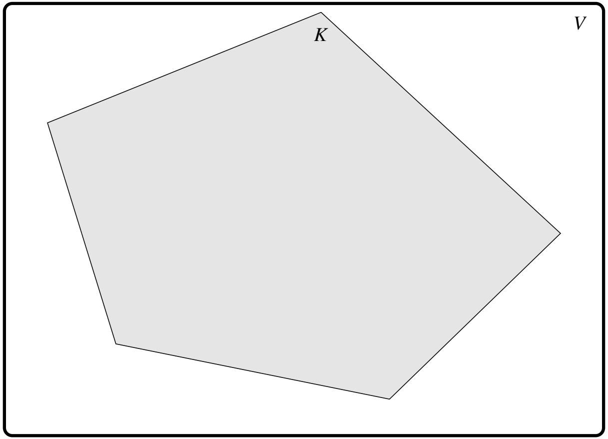
$f = -10, \phi =$

The optimality condition for this problem is the variational inequality

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

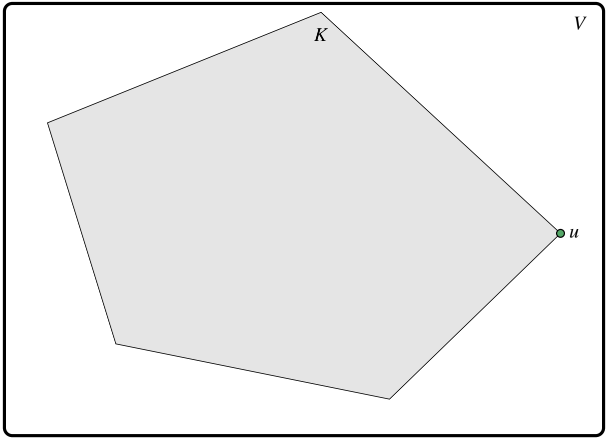
The optimality condition for this problem is the variational inequality

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



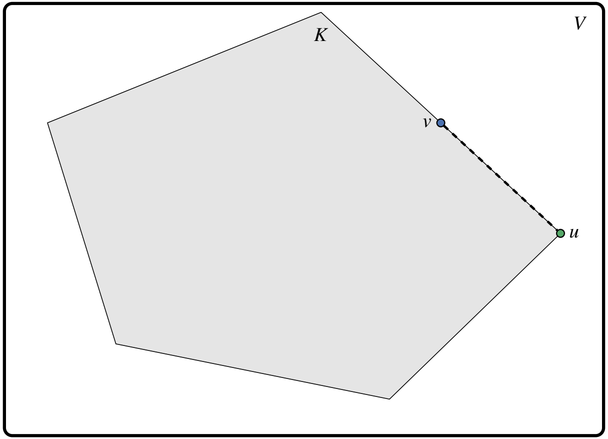
The optimality condition for this problem is the variational inequality

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



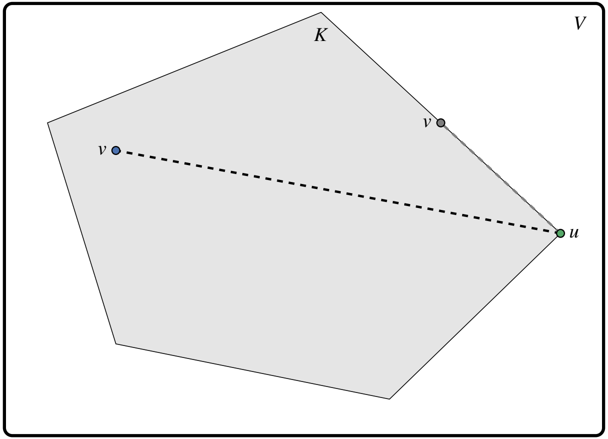
The optimality condition for this problem is the variational inequality

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



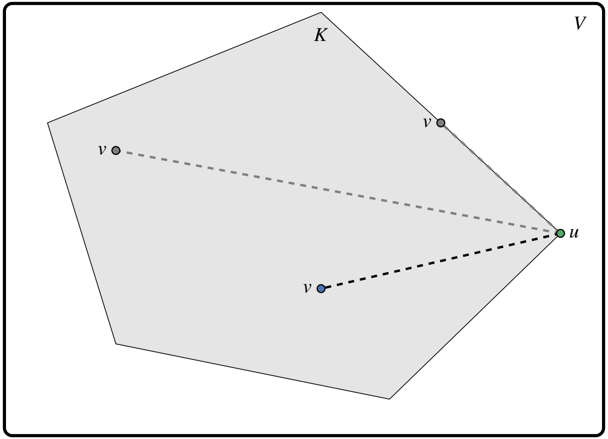
The optimality condition for this problem is the variational inequality

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



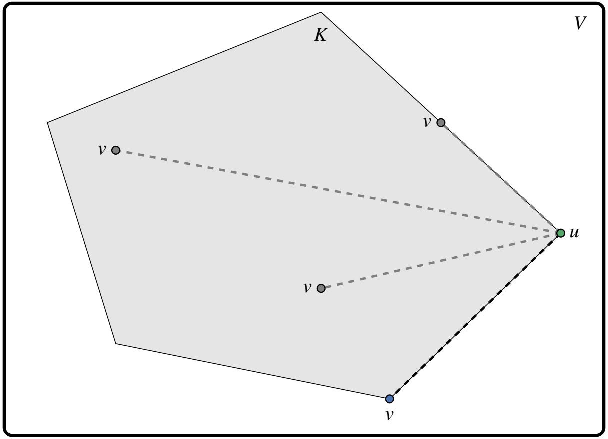
The optimality condition for this problem is the variational inequality

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



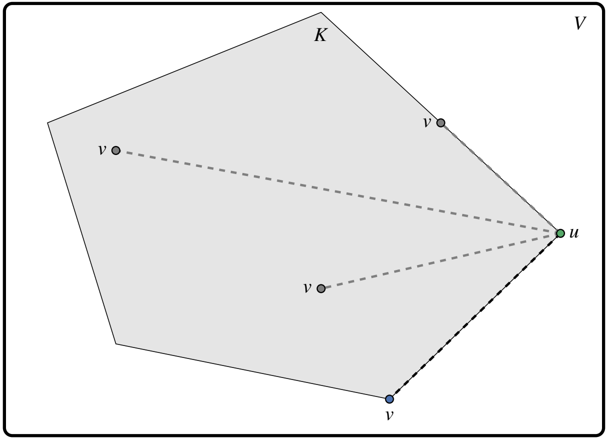
The optimality condition for this problem is the variational inequality

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



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* $v - u$.

Section 2

Latent variable proximal point

Latent variable proximal point (LVPP) for solving VIs

is a mixed reformulation of the *Bregman proximal point* algorithm.

Latent variable proximal point (LVPP) for solving VIs

is a mixed reformulation of the *Bregman proximal point* algorithm.

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

Latent variable proximal point (LVPP) for solving VIs

is a mixed reformulation of the *Bregman proximal point* algorithm.

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 mixed reformulation of the *Bregman proximal point* algorithm.

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

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

The mixed reformulation is equivalent to a primal one, but much better for discretisation.

Latent variable proximal point (LVPP) for solving VIs

is a mixed reformulation of the *Bregman proximal point* algorithm.

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

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

The mixed reformulation is equivalent to a primal one, but much better for discretisation.

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 mixed reformulation of the *Bregman proximal point* algorithm.

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

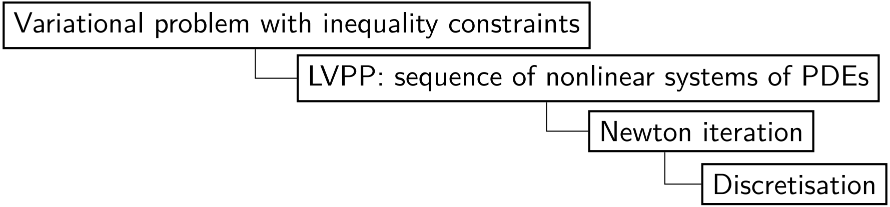
The Legendre function encodes the geometry of the constraints at each point.

The mixed reformulation is equivalent to a primal one, but much better for discretisation.

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.



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

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

The hyperelastic beam problem has feasible set

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

The obstacle problem has feasible set

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

The hyperelastic beam problem has feasible set

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

A problem with gradient constraints might have feasible set

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

The obstacle problem has feasible set

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

The hyperelastic beam problem has feasible set

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

A problem with gradient constraints might have feasible set

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

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

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

The hyperelastic beam problem has feasible set

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

A problem with gradient constraints might have feasible set

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

Our general feasible set

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

Here $C(x) \subset \mathbb{R}^m, \operatorname{int} C(x) \neq \emptyset, C(x)$ convex is the *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.

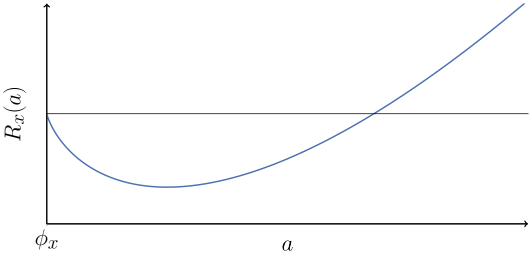
Legendre functions are a class of proper convex functions. They *encode the geometry of the feasible image* $C_x = C(x)$.

$R_x : C_x \to \mathbb{R} \cup \{\infty\}$ 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:

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



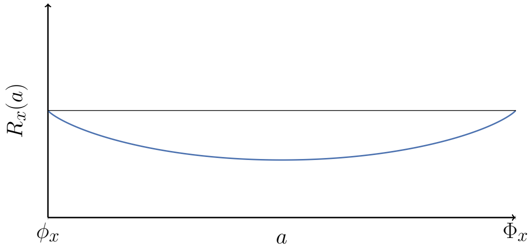
Legendre functions are a class of proper convex functions. They *encode the geometry of the feasible image* $C_x = C(x)$.

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

Example

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

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



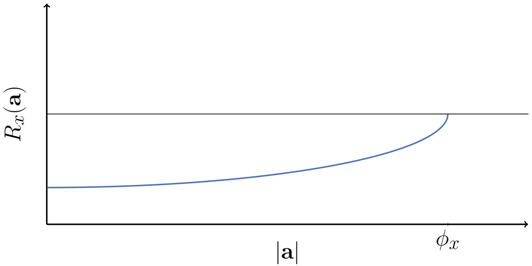
Legendre functions are a class of proper convex functions. They *encode the geometry of the feasible image* $C_x = C(x)$.

$R_x : C_x \to \mathbb{R} \cup \{\infty\}$ 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 useful?

Why is this useful?

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

Why is this useful?

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

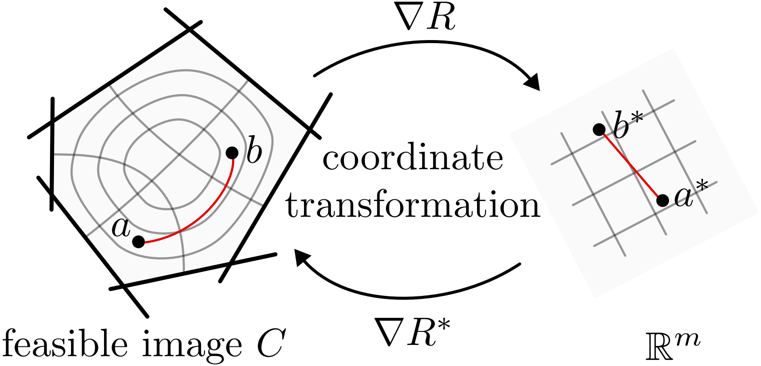
Theorem (Rockafellar (1967))

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

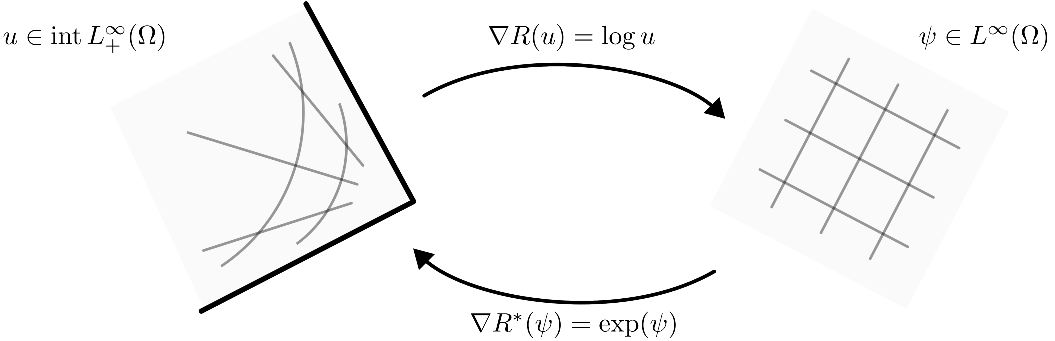
$$\nabla R \colon \operatorname{int}(\operatorname{dom} R) \to \operatorname{int}(\operatorname{dom} R^*)$$

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

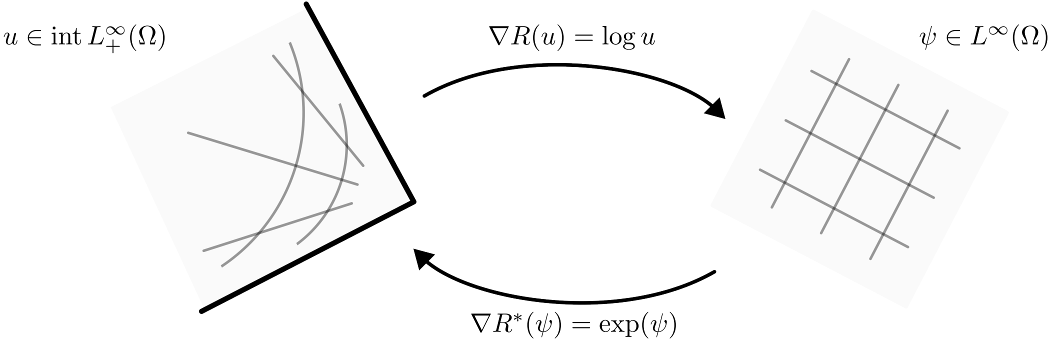
R. Tyrell Rockafellar


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

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



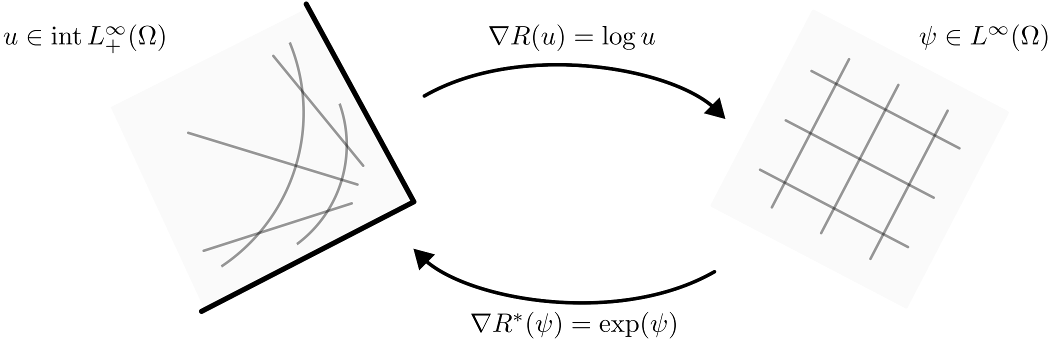
Applying this idea at every point, for the obstacle problem with $\phi = 0$, we have



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



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

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

we iterate

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

Bernard Martinet

Osman Güler

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

To solve

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

we iterate

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

Bernard Martinet

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

Osman Güler

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

Bernard Martinet

To solve

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

we iterate

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

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

Amazingly, for convex J, this converges in J 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*.

Bad news

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

The regularisation

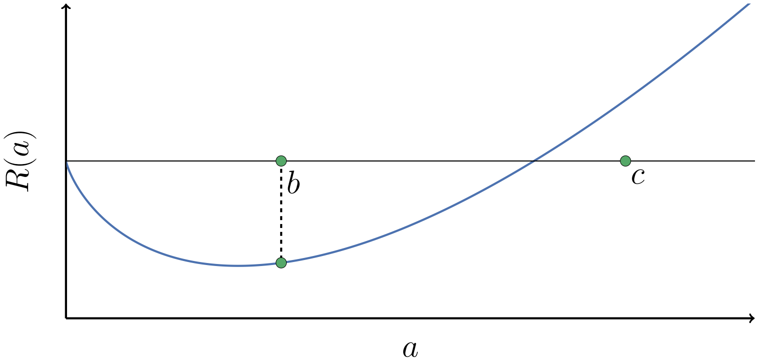
$$\frac{1}{\alpha^k}\|v - u_{k-1}\|_V^2$$

does *not respect the geometry of the constraints*.

Good news

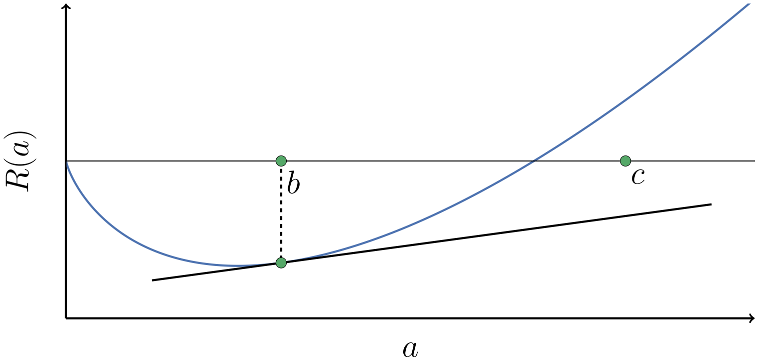
The Legendre function gives a notion of distance where the subproblems *do* simplify.

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



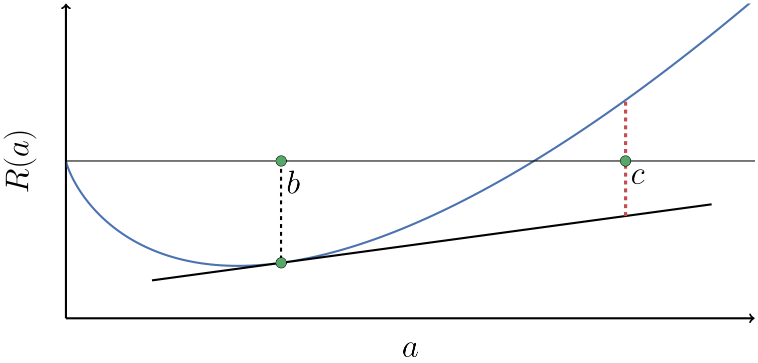
Start with the Legendre function R.

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



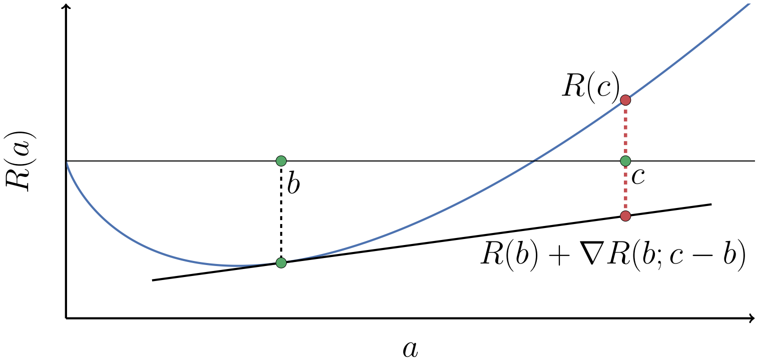
Build the tangent at b.

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



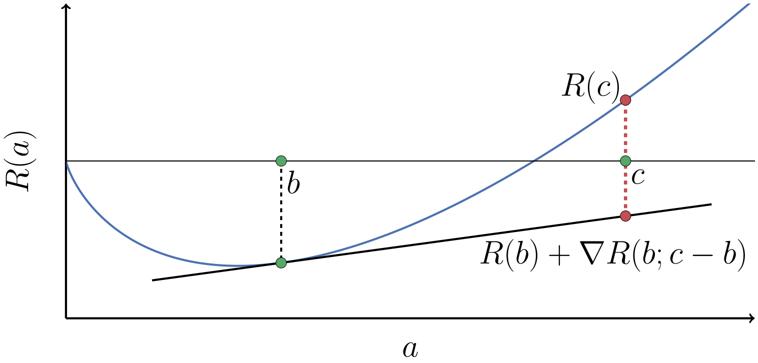
Measure distance between tangent and R at c.

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

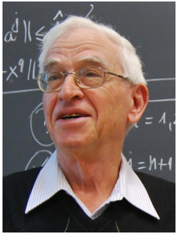


$$D_R(c, b) = R(c) - R(b) - \nabla R(b; c - b).$$

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

To solve

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



Yair Censor

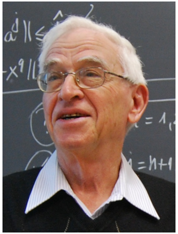
Stavros Zenios

To solve

$$u \in \underset{v \in K}{\text{argmin}}\; J(v)$$

with Bregman proximal point, we iterate

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



Yair Censor

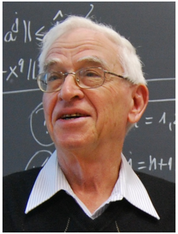
Stavros Zenios

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*:

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

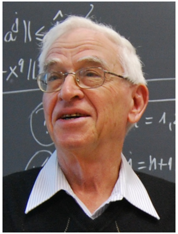
Stavros Zenios

To solve

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

with Bregman proximal point, we iterate

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



Yair Censor

Good news

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

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

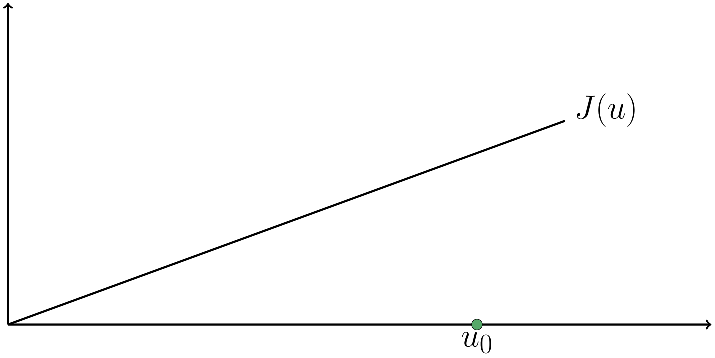
This breaks down the VI into a sequence of nonlinear PDEs!

Stavros Zenios

Consider the toy problem

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

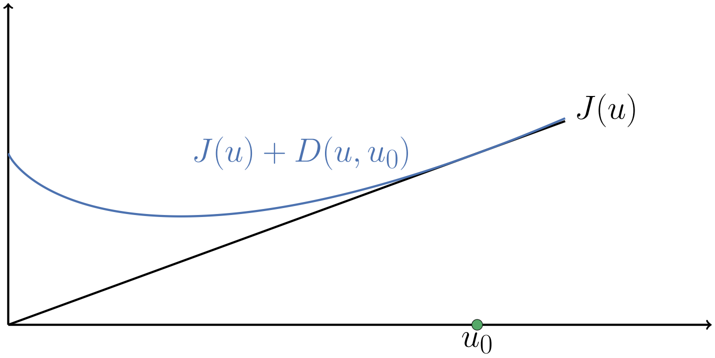
with solution $u = 0$.



Consider the toy problem

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

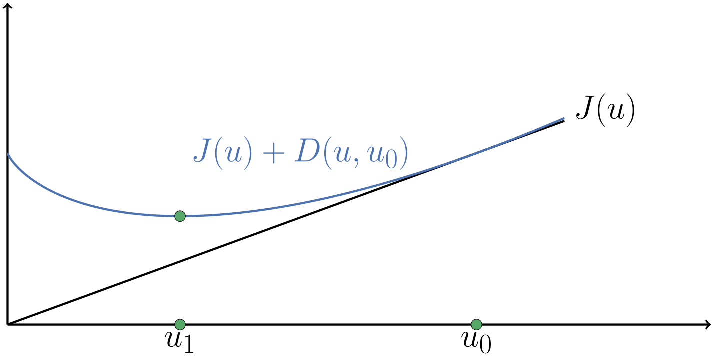
with solution $u = 0$.



Consider the toy problem

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

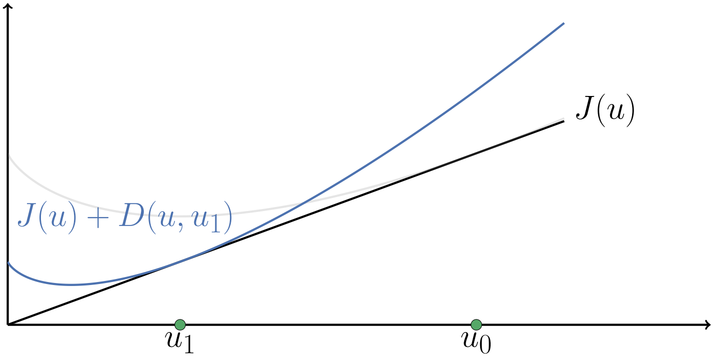
with solution $u = 0$.



Consider the toy problem

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

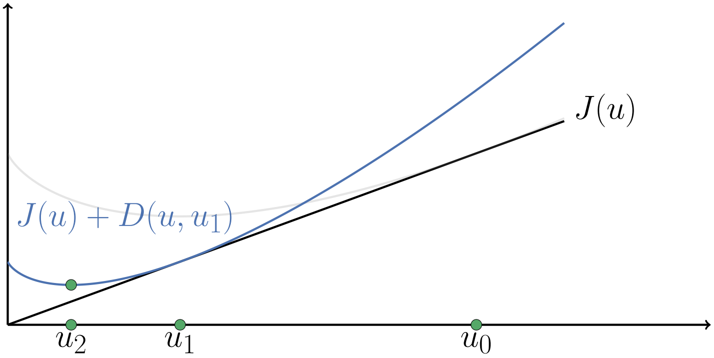
with solution $u = 0$.



Consider the toy problem

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

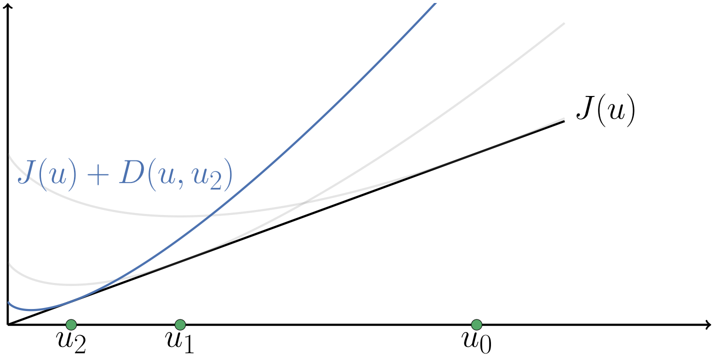
with solution $u = 0$.



Consider the toy problem

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

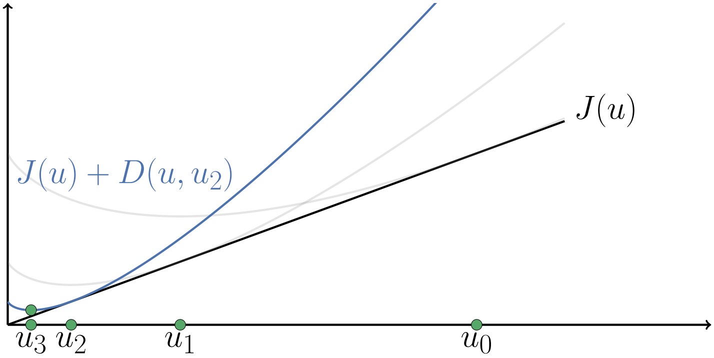
with solution $u = 0$.



Consider the toy problem

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

with solution $u = 0$.



Subsection 3

Mixed formulation

Good news

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

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

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 $u \in K$. This is really hard.

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 $u \in K$. This is really hard.

How do I do this at high order? How do we add on updates and retain feasibility?

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 $u \in K$. This is really hard.

How do I do this at high order? How do we add on updates and retain feasibility?

Idea

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

Latent variable proximal point

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

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

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

The mixed formulation only requires discretising $u \in V$, not $u \in K$!

Latent variable proximal point

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

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

Important observation

The mixed formulation only requires discretising $u \in V$, not $u \in K$!

Two approximations

When discretised, this gives *two* approximations of the observable Bu:

$$Bu_h \neq \nabla R^*(\psi_h).$$

In particular, if $B = I$, $\nabla R^*(\psi_h)$ is strictly feasible (while u_h probably is not).

Section 3

Bound constraints

We consider again the obstacle problem:

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

for feasible set

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

We consider again the obstacle problem:

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

for feasible set

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

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

We can discretise the LVPP iterations with a Galerkin scheme:

$$\text{LVPP} + \text{Galerkin discretisation} = \text{proximal Galerkin.}$$

We can discretise the LVPP iterations with a Galerkin scheme:

$$\text{LVPP} + \text{Galerkin discretisation} = \text{proximal Galerkin.}$$

As it is a mixed problem, we need to satisfy inf-sup conditions when discretising.

We can discretise the LVPP iterations with a Galerkin scheme:

$$\text{LVPP} + \text{Galerkin discretisation} = \text{proximal Galerkin}.$$

As it is a mixed problem, we need to satisfy inf-sup conditions when discretising.

Equal order CG_p-CG_p elements work, but require quasi-uniformity.

We can discretise the LVPP iterations with a Galerkin scheme:

$$\text{LVPP} + \text{Galerkin discretisation} = \text{proximal Galerkin}.$$

As it is a mixed problem, we need to satisfy inf-sup conditions when discretising.

Equal order CG_p-CG_p elements work, but require quasi-uniformity.

$(\mathrm{CG}_p \oplus \mathrm{B}_{p+2})$-$\mathrm{DG}_{p-1}$ works without quasi-uniformity, but $\nabla R^*(\psi_h)$ is not conforming.

We can discretise the LVPP iterations with a Galerkin scheme:

$$\text{LVPP} + \text{Galerkin discretisation} = \text{proximal Galerkin}.$$

As it is a mixed problem, we need to satisfy inf-sup conditions when discretising.

Equal order CG_p-CG_p elements work, but require quasi-uniformity.

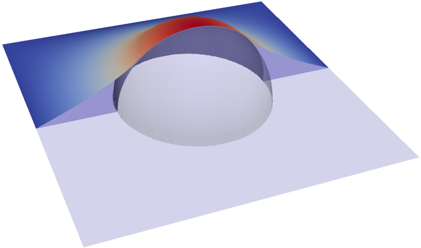
$(\mathrm{CG}_p \oplus \mathrm{B}_{p+2})$-$\mathrm{DG}_{p-1}$ works without quasi-uniformity, but $\nabla R^*(\psi_h)$ is not conforming.

Open question

What is the 'best' inf-sup stable finite element pair?

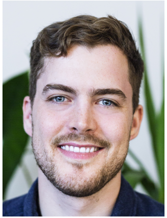
	Degree $p=1$			Degree $p=2$		
Method	h	$h/2$	$h/4$	h	$h/2$	$h/4$
Proximal Galerkin	15	13	12	15	16	12
Active Set (PETSc RSLS)	11	16	25	Not bound preserving		
Trust-Region (GALAHAD)	6	12	19			
Interior Point (IPOPT)	9	9	8			
IPOPT without Hessian	90	260	500			

Total number of linear system solves for popular solvers using various mesh sizes h.

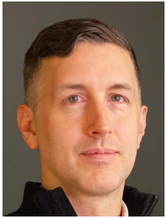


For the proof that the Bregman proximal point iterations for the obstacle problem are PDEs, not VIs, see

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.



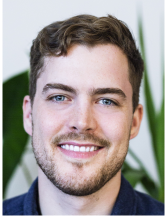
Brendan Keith



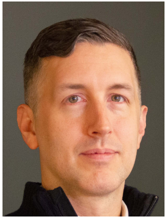
Thomas Surowiec

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

You can easily apply other discretisations, too.

Mesh size h	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}
Finite Difference	10	15	13	15	16	16

Degree p	8	16	24	32	40	48
Spectral Method	16	17	16	16	16	15

Total number of linear system solves for other discretisations.

Subsection 2

Comparison

There are lots of 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	✓*	✓	✓	✓

Active set/NCP function + semismooth Newton

Often mesh-dependent if applied directly because Lagrange multipliers are not smooth enough.

There are lots of 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	✓*	✓	✓	✓

Penalty/augmented Lagrangian methods

Always approximate from outside the feasible set; penalty $\to \infty$ even with AL in ∞-dim.

There are lots of 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	✓*	✓	✓	✓

Monotone multigrid

Inherently discrete; requires hierarchy; specialised components required for each problem.

There are lots of 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	✓*	✓	✓	✓

Interior point

Often involves mixed subproblems; line search very finicky as $h \to 0$, barrier $\to 0$.

There are lots of 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

Applies to very wide class of problems; requires mixed subproblems; $B \neq I$ not feasible.

Subsection 3

A quasi-variational inequality

Quasi-variational inequalities (QVIs) are even harder.

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

Quasi-variational inequalities (QVIs) are even harder.

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

QVI: $$u \in K(u) \subsetneq V : \; F(u; v - u) \geq 0 \qquad \forall\, v \in K(u)$$

Quasi-variational inequalities (QVIs) are even harder.

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

QVI: $$u \in K(u) \subsetneq V : \; F(u; v-u) \geq 0 \qquad \forall\, v \in K(u)$$

QVIs model contact. Your seat deforms under your weight, but also deforms you.

Quasi-variational inequalities (QVIs) are even harder.

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

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

QVIs model contact. Your seat deforms under your weight, but also deforms you.

Good news

LVPP extends *very cleanly* to solving (some) QVIs.

Quasi-variational inequalities (QVIs) are even harder.

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

QVI: $$u \in K(u) \subsetneq V : \; F(u; v - u) \geq 0 \qquad \forall\, v \in K(u)$$

QVIs model contact. Your seat deforms under your weight, but also deforms you.

Good news

LVPP extends *very cleanly* to solving (some) QVIs.

The obstacle explicitly appears in the LVPP equations, so it can depend on other variables.

In a thermoforming QVI, a heated membrane is pushed upwards into a mold with force f.

The mold deforms with temperature:

$$K(T) = \left\{ v \in H_0^1(\Omega) \mid v \leq \Phi \coloneqq \Phi_0 + \xi T \right\}.$$

In a thermoforming QVI, a heated membrane is pushed upwards into a mold with force f.

The mold deforms with temperature:

$$K(T) = \left\{ v \in H_0^1(\Omega) \mid v \leq \Phi \coloneqq \Phi_0 + \xi T \right\}.$$

More heat transfer occurs between membrane and mold where they are in contact (given by g).

In a thermoforming QVI, a heated membrane is pushed upwards into a mold with force f.

The mold deforms with temperature:

$$K(T) = \left\{ v \in H_0^1(\Omega) \mid v \leq \Phi := \Phi_0 + \xi T \right\}.$$

More heat transfer occurs between membrane and mold where they are in contact (given by g).

The thermoforming QVI is to find $(u, T) \in K(T) \times H^1(\Omega)$ s. t.

$$\begin{aligned} (\nabla T, \nabla q) + \beta(T, q) &= (g(\Phi_0 + \xi T - u), q), \\ (\nabla u, \nabla(v - u)) &\geq (f, v - u), \end{aligned}$$

for all $(v, q) \in K(T) \times H^1(\Omega)$.

We again use the Shannon entropy (with signs switched).

The LVPP iterations become: find $(u^k, \psi^k, T^k) \in H_0^1(\Omega) \times L^\infty(\Omega) \times H^1(\Omega)$ s. t.

$$
\begin{aligned}
(\nabla T^k, \nabla q) + \beta(T^k, q) &= (g(\exp(-\psi^k)), q), \\
\alpha_k(\nabla u^k, \nabla v) + (\psi^k, v) &= \alpha_k(f, v) + (\psi^{k-1}, v), \\
(u^k, w) + (\exp(-\psi^k), w) &= (\Phi_0 + \xi T^k, w).
\end{aligned}
$$

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

We again use the Shannon entropy (with signs switched).

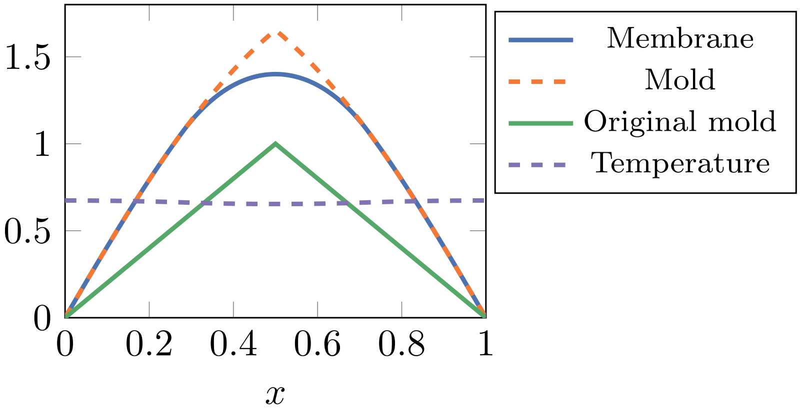
The LVPP iterations become: find $(u^k, \psi^k, T^k) \in H_0^1(\Omega) \times L^\infty(\Omega) \times H^1(\Omega)$ s. t.

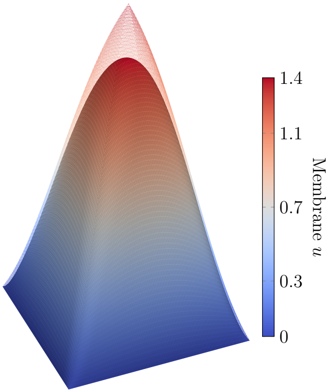
$$
\begin{aligned}
(\nabla T^k, \nabla q) + \beta(T^k, q) &= (g(\exp(-\psi^k)), q), \\
\alpha_k(\nabla u^k, \nabla v) + (\psi^k, v) &= \alpha_k(f, v) + (\psi^{k-1}, v), \\
(u^k, w) + (\exp(-\psi^k), w) &= (\Phi_0 + \xi T^k, w).
\end{aligned}
$$

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

Good news

Mesh-independent convergence and small iteration counts ($\sim$ 20 linear solves total).


1.5
1
0.5
0
0
0.2
0.4
0.6
0.8
1
x
Membrane
Mold
Original mold
Temperature


1.4
1.1
0.7
0.3
0
Membrane u

Section 4

Gradient constraints

We consider again the Dirichlet energy:

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

We consider 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 *gradient* constraints:

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

Here $B = \nabla$ and $C(x)$ is the Euclidean ball of radius $\Phi(x)$.

We consider again the Dirichlet energy:

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

but now impose *gradient* constraints:

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

Here $B = \nabla$ and $C(x)$ is the Euclidean ball of radius $\Phi(x)$.

Numerical methods are scarce.

We use the modified Hellinger entropy $R(\mathbf{a}) = -\sqrt{\Phi^2 - |\mathbf{a}|^2}$.

The LVPP iteration becomes: find $(u^k, \Psi^k) \in H_0^1(\Omega) \times L^\infty(\Omega, \mathbb{R}^n)$ ($n =$ spatial dimension) s. t.

$$\alpha_k(\nabla u^k, \nabla v) + (\Psi^k, \nabla v) = \alpha_k(f, v) + (\Psi^{k-1}, \nabla v),$$

$$(\nabla u^k, W) - \left(\frac{\Phi\,\Psi^k}{\sqrt{1 + |\Psi^k|^2}}, W\right) = 0$$

for all $(v, W) \in H_0^1(\Omega) \times L^\infty(\Omega, \mathbb{R}^n)$.

We use the modified Hellinger entropy $R(\mathbf{a}) = -\sqrt{\Phi^2 - |\mathbf{a}|^2}$.

The LVPP iteration becomes: find $(u^k, \Psi^k) \in H_0^1(\Omega) \times L^\infty(\Omega, \mathbb{R}^n)$ ($n =$ spatial dimension) s. t.

$$\alpha_k(\nabla u^k, \nabla v) + (\Psi^k, \nabla v) = \alpha_k(f, v) + (\Psi^{k-1}, \nabla v),$$

$$(\nabla u^k, W) - \left(\frac{\Phi\,\Psi^k}{\sqrt{1 + |\Psi^k|^2}}, W\right) = 0$$

for all $(v, W) \in H_0^1(\Omega) \times L^\infty(\Omega, \mathbb{R}^n)$.

Good news

Again, we observe (but do not yet prove) robust mesh-independent convergence.

LVPP extends to *intersections of constraints*. Take again the Dirichlet energy

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

LVPP extends to *intersections of constraints*. Take again the Dirichlet energy

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

but now impose *both obstacle and gradient* constraints:

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

LVPP extends to *intersections of constraints*. Take again the Dirichlet energy

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

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.

LVPP extends to *intersections of constraints*. Take again the Dirichlet energy

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

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:

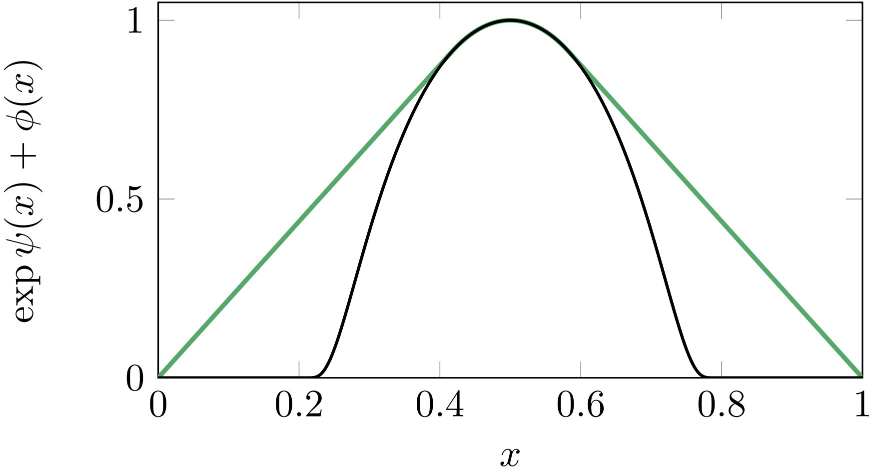
$$\nabla R^*\left((a^*, \mathbf{a}^*)\right) = \begin{pmatrix} \phi + \exp a^* \\ \dfrac{\Phi\, \mathbf{a}^*}{\sqrt{1 + |\mathbf{a}^*|^2}} \end{pmatrix}.$$

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.

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

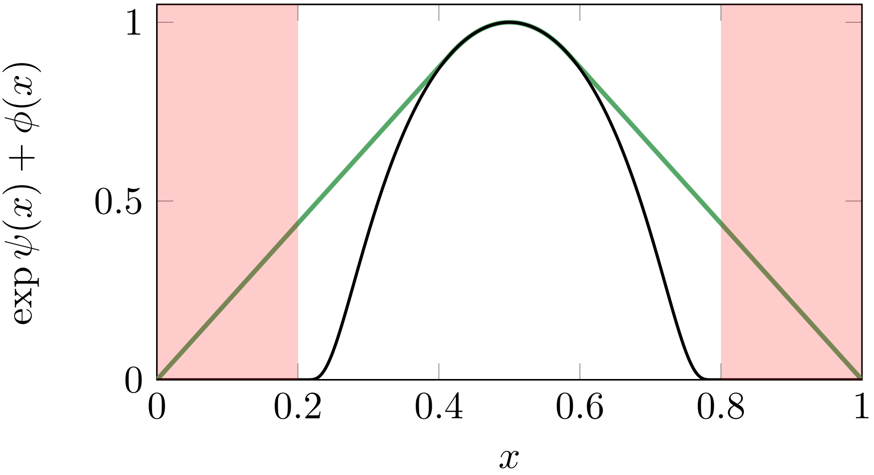
for all $(v, w, W) \in H^1(\Omega) \times L^\infty(\Omega) \times L^\infty(\Omega, \mathbb{R}^n)$.

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



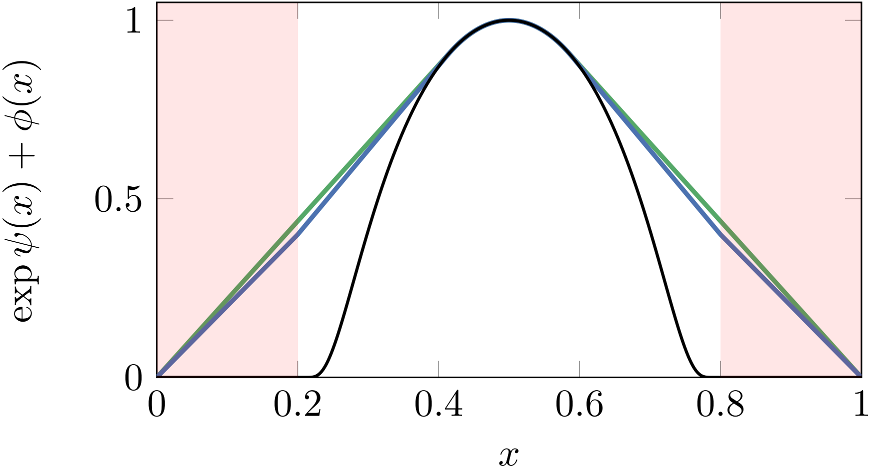
No gradient constraints.

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



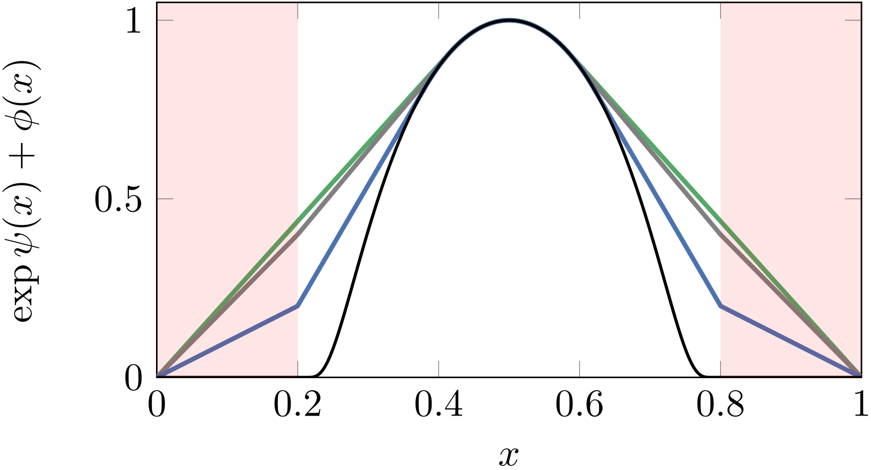
Apply gradient constraints on $[0, 0.2] \cup [0.8, 1]$.

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



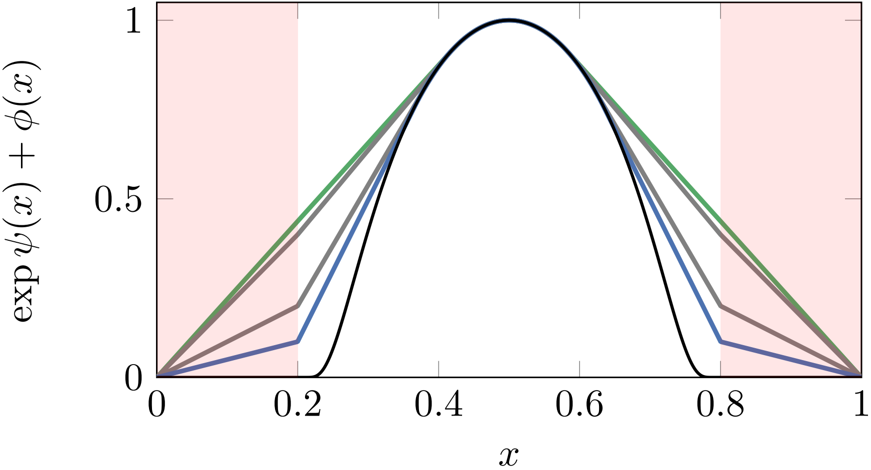
Light gradient constraints.

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



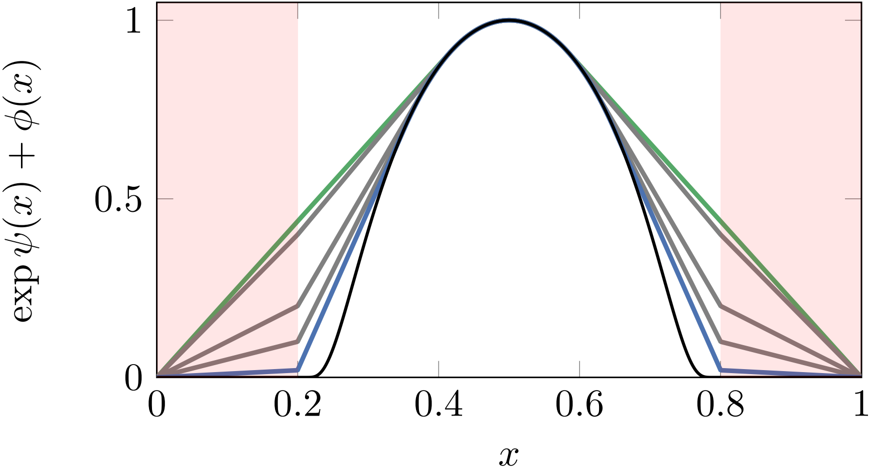
Medium gradient constraints.

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

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

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

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

The LVPP iteration becomes: find $(Q^k, \psi^k) \in H^1_D(\Omega, \mathbb{R}^{2\times 2}_{\mathsf{sym,tr}}) \times L^\infty(\Omega, \mathbb{R}^{2\times 2}_{\mathrm{sym,tr}})$ 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 $(V, w) \in H^1_0(\Omega, \mathbb{R}^{2\times 2}_{\mathsf{sym,tr}}) \times L^\infty(\Omega, \mathbb{R}^{2\times 2}_{\mathrm{sym,tr}})$.

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.

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

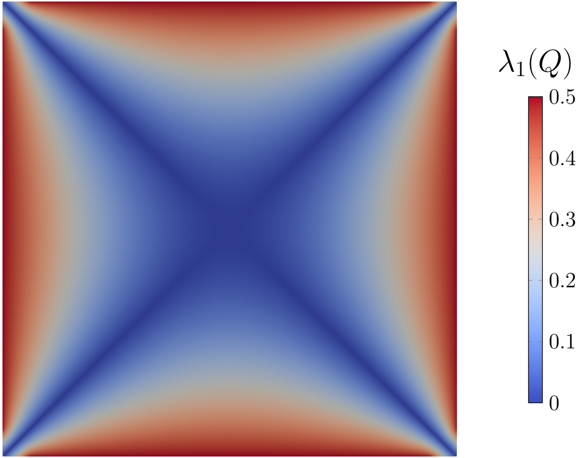
$$(Q, w) - (\frac{1}{2}\operatorname{tanhm}(\psi/2), w) = 0$$

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

Conclusions

Conclusion

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

J. S. Dokken, P. E. Farrell, B. Keith, I. P. A. Papadopoulos, and T. M. Surowiec. "The latent variable proximal point algorithm for variational problems with inequality constraints". In: *Computer Methods in Applied Mechanics and Engineering* 445 (2025), p. 118181. DOI: 10.1016/j.cma.2025.118181.

Conclusion

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

J. S. Dokken, P. E. Farrell, B. Keith, I. P. A. Papadopoulos, and T. M. Surowiec. "The latent variable proximal point algorithm for variational problems with inequality constraints". In: *Computer Methods in Applied Mechanics and Engineering* 445 (2025), p. 118181. DOI: 10.1016/j.cma.2025.118181.

Good news

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

Section 7

Monge–Ampère

For uniformly positive $\rho \in C(\overline{\Omega})$ and $g \in C^3(\overline{\Omega})$, find the unique $u \in K$ such that

$$\det(\nabla^2 u) = \rho \ \text{ in } \Omega\,, \qquad u = g \ \text{ on } \partial\Omega\,.$$

For uniformly positive $\rho \in C(\overline{\Omega})$ and $g \in C^3(\overline{\Omega})$, find the unique $u \in K$ such that

$$\det(\nabla^2 u) = \rho \ \text{ in } \Omega\,, \qquad u = g \ \text{ on } \partial\Omega\,.$$

where

$$K = \{u \in H^2(\Omega) \cap H^1_g(\Omega) \mid \nabla^2 u \succeq 0 \text{ a.e. in } \Omega\}$$

over a smooth, bounded, convex set $\Omega \subset \mathbb{R}^n$.

For uniformly positive $\rho \in C(\overline{\Omega})$ and $g \in C^3(\overline{\Omega})$, find the unique $u \in K$ such that

$$\det(\nabla^2 u) = \rho \ \text{ in } \Omega\,, \qquad u = g \ \text{ on } \partial\Omega\,.$$

where

$$K = \{u \in H^2(\Omega) \cap H^1_g(\Omega) \mid \nabla^2 u \succeq 0 \text{ a.e. in } \Omega\}$$

over a smooth, bounded, convex set $\Omega \subset \mathbb{R}^n$.

This set has Legendre function

$$R(A) = \operatorname{tr}(A \ln A - A)\,, \quad \text{with } \nabla R^*(A^*) = \exp A^*.$$

So, introduce a latent variable

$$\psi = \ln \nabla^2 u \qquad \Longleftrightarrow \qquad \exp \psi = \nabla^2 u.$$

So, introduce a latent variable

$$\psi = \ln \nabla^2 u \quad \Longleftrightarrow \quad \exp \psi = \nabla^2 u.$$

Using the identity $\det(\exp \psi) = \exp(\operatorname{tr} \psi)$, the main PDE becomes an algebraic relation:

$$\operatorname{tr} \psi = \ln \rho.$$

So, introduce a latent variable

$$\psi = \ln \nabla^2 u \quad \iff \quad \exp \psi = \nabla^2 u.$$

Using the identity $\det(\exp \psi) = \exp(\operatorname{tr} \psi)$, the main PDE becomes an algebraic relation:

$$\operatorname{tr} \psi = \ln \rho.$$

This leads to the variational formulation: find $u \in H^2(\Omega) \cap H^1_g(\Omega)$ and $\psi \in L^\infty(\Omega; \mathbb{R}^{n\times n}_{\mathsf{sym}})$ s. t.

$$\begin{aligned} (\nabla^2 u, w) - (\exp \psi, w) &= 0, \\ (\operatorname{tr} \psi, v) &= (\ln \rho, v), \end{aligned}$$

for all $v \in H^2(\Omega) \cap H^1_0(\Omega)$ and $w \in L^\infty(\Omega; \mathbb{R}^{n\times n}_{\mathsf{sym}})$.

Most codes cannot discretise $u \in H^2$, so introduce $T = \nabla u$:

Find $u \in H^1_g(\Omega)$, $T \in H^1(\Omega, \mathbb{R}^n)$, and $\psi \in L^\infty(\Omega; \mathbb{R}^{n\times n}_{\text{sym}})$ s. t.

$$\begin{aligned}(T, S) - (\nabla u, S) &= 0,\\ (\nabla T, w) - (\exp \psi, w) &= 0,\\ (\operatorname{tr} \psi, v) &= (\ln \rho, v),\end{aligned}$$

for all $v \in H^1_0(\Omega)$, $S \in H^1(\Omega, \mathbb{R}^n)$, and $w \in L^\infty(\Omega; \mathbb{R}^{n\times n}_{\text{sym}})$.

Most codes cannot discretise $u \in H^2$, so introduce $T = \nabla u$:

Find $u \in H^1_g(\Omega)$, $T \in H^1(\Omega, \mathbb{R}^n)$, and $\psi \in L^\infty(\Omega; \mathbb{R}^{n\times n}_{\text{sym}})$ s. t.

$$\begin{aligned} (T, S) - (\nabla u, S) &= 0, \\ (\nabla T, w) - (\exp \psi, w) &= 0, \\ (\operatorname{tr} \psi, v) &= (\ln \rho, v), \end{aligned}$$

for all $v \in H^1_0(\Omega)$, $S \in H^1(\Omega, \mathbb{R}^n)$, and $w \in L^\infty(\Omega; \mathbb{R}^{n\times n}_{\text{sym}})$.

Numerical experiments with CG_p-$[\mathrm{CG}_{p+1}]^n$-$[\mathrm{CG}_p]^{n\times n}_{\text{sym}}$ show excellent convergence, for $p \geq 2$.

Good news

Robust Newton convergence, error of $\sim 10^{-13}$ for $p = 14$.

Section 8

Fracture

Variational fracture is an important model with a non-convex energy.

Variational fracture is an important model with a non-convex energy.

We consider the anti-plane shear test and only solve for vertical component of displacement:

$$(u, c) \in \operatorname*{argmin}_{(v,d)\in K} J(v, d) = \frac{G}{2} \int_\Omega (\epsilon + (1-\epsilon)(1-d)^2)|\nabla v|^2 \, \mathrm{d}x + \frac{G_c}{2} \int_\Omega \ell |\nabla d|^2 + \ell^{-1} d^2 \, \mathrm{d}x,$$

for feasible set

$$K = \left\{ (u, c) \in H^1_D(\Omega) \times H^1(\Omega) \mid 0 \le c_{\text{prev}} \le c \le 1 \text{ a.e. in } \Omega \right\}.$$

Variational fracture is an important model with a non-convex energy.

We consider the anti-plane shear test and only solve for vertical component of displacement:

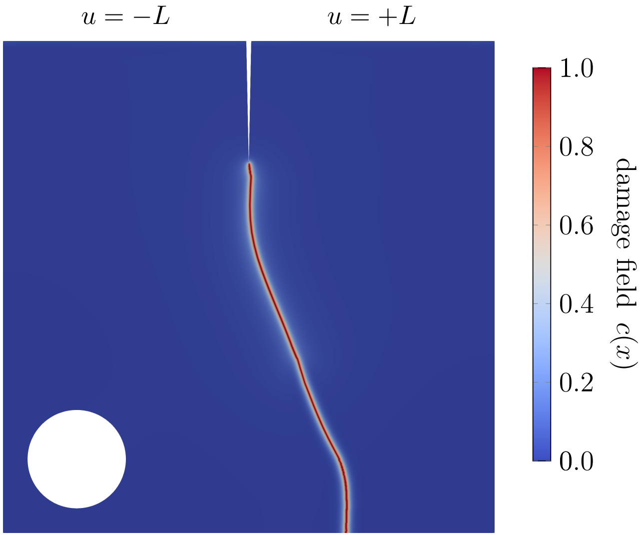
$$(u, c) \in \operatorname*{argmin}_{(v,d)\in K} J(v, d) = \frac{G}{2}\int_\Omega (\epsilon + (1-\epsilon)(1-d)^2)|\nabla v|^2 \,\mathrm{d}x + \frac{G_c}{2}\int_\Omega \ell|\nabla d|^2 + \ell^{-1}d^2 \,\mathrm{d}x,$$

for feasible set

$$K = \left\{(u, c) \in H^1_D(\Omega) \times H^1(\Omega) \mid 0 \le c_{\text{prev}} \le c \le 1 \text{ a.e. in } \Omega\right\}.$$

To put this in our general abstraction, we take

$$B = (0, \text{id}), \quad \Omega_d = \Omega, \quad C(x) = \mathbb{R} \times [c_{\text{prev}}(x), 1].$$



The final damage field for the fracture problem considered.

We introduce a latent variable for the bound constraint on c.

The LVPP system becomes: find $(u^k, c^k, \psi^k) \in H^1_D(\Omega) \times H^1(\Omega) \times L^\infty(\Omega)$ s. t.

$$
\begin{aligned}
\alpha_k G((\epsilon + (1-\epsilon)(1-c^k)^2)\nabla u^k, \nabla v) &= 0\,, \\
-\alpha_k G((1-\epsilon)(1-c^k)|\nabla u^k|^2, d) + \alpha_k G_c(\ell(\nabla c^k, \nabla d) + \ell^{-1}(c^k, d)) + (\psi^k, d) &= (\psi^{k-1}, d)\,, \\
(c^k, w) - \left(\frac{c_{\text{prev}} + \exp(\psi^k)}{\exp(\psi^k) + 1}, w \right) &= 0\,,
\end{aligned}
$$

for all $(v, d, w) \in H^1_0(\Omega) \times H^1(\Omega) \times L^\infty(\Omega)$.

We introduce a latent variable for the bound constraint on c.

The LVPP system becomes: find $(u^k, c^k, \psi^k) \in H^1_D(\Omega) \times H^1(\Omega) \times L^\infty(\Omega)$ s. t.

$$
\begin{aligned}
\alpha_k G((\epsilon + (1-\epsilon)(1-c^k)^2)\nabla u^k, \nabla v) &= 0\,,\\
-\alpha_k G((1-\epsilon)(1-c^k)|\nabla u^k|^2, d) + \alpha_k G_c(\ell(\nabla c^k, \nabla d) + \ell^{-1}(c^k, d)) + (\psi^k, d) &= (\psi^{k-1}, d)\,,\\
(c^k, w) - \left(\frac{c_{\text{prev}} + \exp(\psi^k)}{\exp(\psi^k) + 1}, w\right) &= 0\,,
\end{aligned}
$$

for all $(v, d, w) \in H^1_0(\Omega) \times H^1(\Omega) \times L^\infty(\Omega)$.

CG_1-CG_1-CG_1 discretisation

Each loading step takes an average of 2.85 proximal iterations (but with large variance).

Each proximal iteration takes an average of 5.44 Newton iterations.