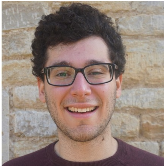
A semismooth Newton method for obstacle-type quasivariational inequalities

John Papadopoulos[1], Amal Alphonse[1], Constantin Christof[2],
Michael Hintermüller[1]

[1]Weierstrass Institute Berlin, [2]TU Munich→Universität Duisburg-Essen

October 22, 2024, WIAS RG8 Seminar, Berlin



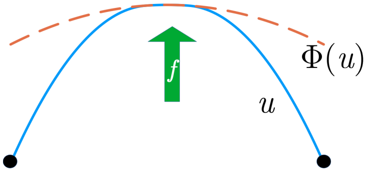

What are obstacle-type QVIs?

Consider the constraint set: $K(u) := \{v \in H_0^1(\Omega) : v \leq \Phi(u) \text{ a.e.}\}$

$$\min_{u \in K(u)} \int_\Omega \frac{1}{2}|\nabla u|^2 - f \cdot u \,\mathrm{d}x.$$

First-order optimality condition is an obstacle-type QVI

Find $u \in H_0^1(\Omega) : (\nabla u, \nabla(v-u))_{L^2(\Omega)} \geq (f, v-u)_{L^2(\Omega)}$ for all $v \in K(u)$.



Applications include contact mechanics, thermoforming, elastic bilayers, image processing, option pricing, fluid flow with pressure constraints... Why doesn't everyone model with QVIs? ...because they are very hard to solve!

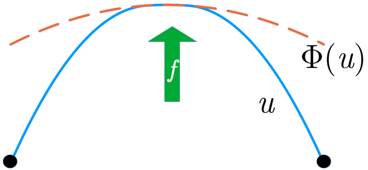
What are obstacle-type QVIs?

Consider the constraint set: $K(u) := \{v \in H_0^1(\Omega) : v \leq \Phi(u) \text{ a.e.}\}$

$$\min_{u \in K(u)} \int_\Omega \frac{1}{2}|\nabla u|^2 - f \cdot u \,\mathrm{d}x.$$

First-order optimality condition is an obstacle-type QVI

Find $u \in H_0^1(\Omega) : (\nabla u, \nabla(v-u))_{L^2(\Omega)} \geq (f, v-u)_{L^2(\Omega)}$ for all $v \in K(u)$.



Applications include contact mechanics, thermoforming, elastic bilayers, image processing, option pricing, fluid flow with pressure constraints... Why doesn't everyone model with QVIs? ...because they are very hard to solve!

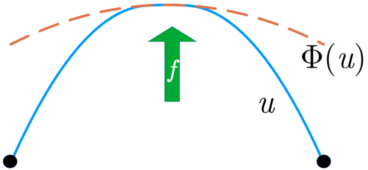
What are obstacle-type QVIs?

Consider the constraint set: $K(u) := \{v \in H^1_0(\Omega) : v \leq \Phi(u) \text{ a.e.}\}$

$$\min_{u \in K(u)} \int_\Omega \frac{1}{2}|\nabla u|^2 - f \cdot u \,\mathrm{d}x.$$

First-order optimality condition is an obstacle-type QVI

Find $u \in H^1_0(\Omega) : (\nabla u, \nabla(v-u))_{L^2(\Omega)} \geq (f, v-u)_{L^2(\Omega)}$ for all $v \in K(u)$.



Applications include contact mechanics, thermoforming, elastic bilayers, image processing, option pricing, fluid flow with pressure constraints... Why doesn't everyone model with QVIs? ...because they are very hard to solve!

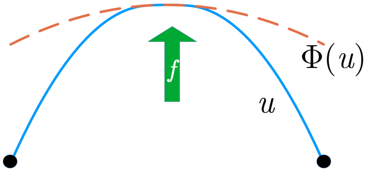
What are obstacle-type QVIs?

Consider the constraint set: $K(u) := \{v \in H_0^1(\Omega) : v \leq \Phi(u) \text{ a.e.}\}$

$$\min_{u \in K(u)} \int_\Omega \frac{1}{2}|\nabla u|^2 - f \cdot u \, \mathrm{d}x.$$

First-order optimality condition is an obstacle-type QVI

Find $u \in H_0^1(\Omega) : (\nabla u, \nabla(v-u))_{L^2(\Omega)} \geq (f, v-u)_{L^2(\Omega)}$ for all $v \in K(u)$.



Applications include contact mechanics, thermoforming, elastic bilayers, image processing, option pricing, fluid flow with pressure constraints... Why doesn't everyone model with QVIs? ...because they are very hard to solve!

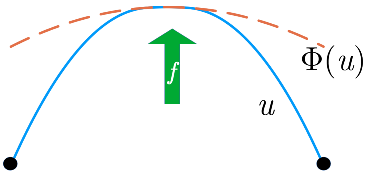
What are obstacle-type QVIs?

Consider the constraint set: $K(u) := \{v \in H_0^1(\Omega) : v \leq \Phi(u) \text{ a.e.}\}$

$$\min_{u \in K(u)} \int_\Omega \frac{1}{2}|\nabla u|^2 - f \cdot u \,\mathrm{d}x.$$

First-order optimality condition is an obstacle-type QVI

Find $u \in H_0^1(\Omega) : (\nabla u, \nabla(v-u))_{L^2(\Omega)} \geq (f, v-u)_{L^2(\Omega)}$ for all $v \in K(u)$.



Applications include contact mechanics, thermoforming, elastic bilayers, image processing, option pricing, fluid flow with pressure constraints... Why doesn't everyone model with QVIs? ...because they are very hard to solve!

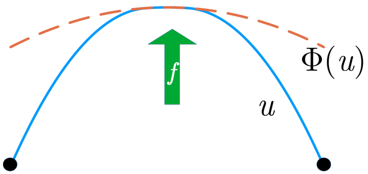
What are obstacle-type QVIs?

Consider the constraint set: $K(u) := \{v \in H_0^1(\Omega) : v \leq \Phi(u) \text{ a.e.}\}$

$$\min_{u \in K(u)} \int_\Omega \frac{1}{2}|\nabla u|^2 - f \cdot u \,\mathrm{d}x.$$

First-order optimality condition is an obstacle-type QVI

Find $u \in H_0^1(\Omega) : (\nabla u, \nabla(v-u))_{L^2(\Omega)} \geq (f, v-u)_{L^2(\Omega)}$ for all $v \in K(u)$.



Applications include contact mechanics, thermoforming, elastic bilayers, image processing, option pricing, fluid flow with pressure constraints... Why doesn't everyone model with QVIs? ...because they are very hard to solve!

Semismooth Newton method

Rewrite the QVI as the fixed point problem

$$u = S(\Phi(u)).$$

Obstacle map $S : H_0^1(\Omega) \to H_0^1(\Omega)$, $S : \phi \mapsto u_\phi$

$u_\phi = S(\phi)$ maps from the obstacle $\to$ solution of the obstacle VI, i.e.

$$(\nabla u_\phi, \nabla(v - u_\phi))_{L^2(\Omega)} \geq (f, v - u_\phi)_{L^2(\Omega)} \text{ for all } v \in H_0^1(\Omega),\ v \leq \phi.$$

⚠Evaluating $\Phi(u)$ might require a nonlinear PDE solve.

⚠Evaluating $S(\phi)$ requires a VI solve.

SSN step for the QVI

Let $R(u) = u - S(\Phi(u))$. The SSN iteration is $u_{i+1} = u_i + \delta$ where

$$G_R(u_i)\delta = -R(u_i).$$

Semismooth Newton method

Rewrite the QVI as the fixed point problem

$$u = S(\Phi(u)).$$

Obstacle map $S : H_0^1(\Omega) \to H_0^1(\Omega)$, $S : \phi \mapsto u_\phi$

$u_\phi = S(\phi)$ maps from the obstacle $\to$ solution of the obstacle VI, i.e.

$$(\nabla u_\phi, \nabla(v - u_\phi))_{L^2(\Omega)} \geq (f, v - u_\phi)_{L^2(\Omega)} \text{ for all } v \in H_0^1(\Omega),\ v \leq \phi.$$

⚠Evaluating $\Phi(u)$ might require a nonlinear PDE solve.

⚠Evaluating $S(\phi)$ requires a VI solve.

SSN step for the QVI

Let $R(u) = u - S(\Phi(u))$. The SSN iteration is $u_{i+1} = u_i + \delta$ where

$$G_R(u_i)\delta = -R(u_i).$$

Semismooth Newton method

Rewrite the QVI as the fixed point problem

$$u = S(\Phi(u)).$$

Obstacle map $S : H^1_0(\Omega) \to H^1_0(\Omega)$, $S : \phi \mapsto u_\phi$

$u_\phi = S(\phi)$ maps from the obstacle $\to$ solution of the obstacle VI, i.e.

$$(\nabla u_\phi, \nabla(v - u_\phi))_{L^2(\Omega)} \geq (f, v - u_\phi)_{L^2(\Omega)} \text{ for all } v \in H^1_0(\Omega),\ v \leq \phi.$$

⚠Evaluating $\Phi(u)$ might require a nonlinear PDE solve.

⚠Evaluating $S(\phi)$ requires a VI solve.

SSN step for the QVI

Let $R(u) = u - S(\Phi(u))$. The SSN iteration is $u_{i+1} = u_i + \delta$ where

$$G_R(u_i)\delta = -R(u_i).$$

Semismooth Newton method

Rewrite the QVI as the fixed point problem

$$u = S(\Phi(u)).$$

Obstacle map $S : H^1_0(\Omega) \to H^1_0(\Omega)$, $S : \phi \mapsto u_\phi$

$u_\phi = S(\phi)$ maps from the obstacle $\to$ solution of the obstacle VI, i.e.

$$(\nabla u_\phi, \nabla(v - u_\phi))_{L^2(\Omega)} \geq (f, v - u_\phi)_{L^2(\Omega)} \text{ for all } v \in H^1_0(\Omega),\ v \leq \phi.$$

⚠Evaluating $\Phi(u)$ might require a nonlinear PDE solve.

⚠Evaluating $S(\phi)$ requires a VI solve.

SSN step for the QVI

Let $R(u) = u - S(\Phi(u))$. The SSN iteration is $u_{i+1} = u_i + \delta$ where

$$G_R(u_i)\delta = -R(u_i).$$

Semismooth Newton method

Rewrite the QVI as the fixed point problem

$$u = S(\Phi(u)).$$

Obstacle map $S : H_0^1(\Omega) \to H_0^1(\Omega)$, $S : \phi \mapsto u_\phi$

$u_\phi = S(\phi)$ maps from the obstacle $\to$ solution of the obstacle VI, i.e.

$$(\nabla u_\phi, \nabla(v - u_\phi))_{L^2(\Omega)} \geq (f, v - u_\phi)_{L^2(\Omega)} \text{ for all } v \in H_0^1(\Omega),\ v \leq \phi.$$

⚠Evaluating $\Phi(u)$ might require a nonlinear PDE solve.

⚠Evaluating $S(\phi)$ requires a VI solve.

SSN step for the QVI

Let $R(u) = u - S(\Phi(u))$. The SSN iteration is $u_{i+1} = u_i + \delta$ where

$$G_R(u_i)\delta = -R(u_i).$$

Evaluating the right-hand side $R(u_i)$

Recall $R(u) = u - S(\Phi(u))$.

Obstacle problem

The difficulty lies in evaluating $S(\phi)$ i.e. find $u_\phi \in H_0^1(\Omega)$ that satisfies:

$$(\nabla u_\phi, \nabla(v - u_\phi))_{L^2(\Omega)} \geq (f, v - u_\phi)_{L^2(\Omega)} \text{ for all } v \in H_0^1(\Omega),\ v \leq \phi.$$

Solver

- Delivers a strictly feasible solution $\tilde{u}_\phi \leq \phi$ a.e.;
- Enjoys mesh independence (the number of iterations does not grow in an unbounded manner as the mesh is refined).

Use a path-following smoothed Moreau–Yosida regularization [mesh independence] followed by a few iterations of the primal-dual active set strategy [strict feasibility].

Evaluating the right-hand side $R(u_i)$

Recall $R(u) = u - S(\Phi(u))$.

Obstacle problem

The difficulty lies in evaluating $S(\phi)$ i.e. find $u_\phi \in H_0^1(\Omega)$ that satisfies:

$$(\nabla u_\phi, \nabla(v - u_\phi))_{L^2(\Omega)} \geq (f, v - u_\phi)_{L^2(\Omega)} \text{ for all } v \in H_0^1(\Omega),\ v \leq \phi.$$

Solver

- Delivers a strictly feasible solution $\tilde{u}_\phi \leq \phi$ a.e.;
- Enjoys mesh independence (the number of iterations does not grow in an unbounded manner as the mesh is refined).

Use a path-following smoothed Moreau–Yosida regularization [mesh independence] followed by a few iterations of the primal-dual active set strategy [strict feasibility].

Evaluating the right-hand side $R(u_i)$

Recall $R(u) = u - S(\Phi(u))$.

Obstacle problem

The difficulty lies in evaluating $S(\phi)$ i.e. find $u_\phi \in H_0^1(\Omega)$ that satisfies:

$$(\nabla u_\phi, \nabla(v - u_\phi))_{L^2(\Omega)} \geq (f, v - u_\phi)_{L^2(\Omega)} \text{ for all } v \in H_0^1(\Omega),\ v \leq \phi.$$

Solver

- Delivers a strictly feasible solution $\tilde{u}_\phi \leq \phi$ a.e.;
- Enjoys mesh independence (the number of iterations does not grow in an unbounded manner as the mesh is refined).

Use a path-following smoothed Moreau–Yosida regularization [mesh independence] followed by a few iterations of the primal-dual active set strategy [strict feasibility].

Evaluating the right-hand side $R(u_i)$

Recall $R(u) = u - S(\Phi(u))$.

Obstacle problem

The difficulty lies in evaluating $S(\phi)$ i.e. find $u_\phi \in H_0^1(\Omega)$ that satisfies:

$$(\nabla u_\phi, \nabla(v - u_\phi))_{L^2(\Omega)} \geq (f, v - u_\phi)_{L^2(\Omega)} \text{ for all } v \in H_0^1(\Omega),\ v \leq \phi.$$

Solver

- Delivers a strictly feasible solution $\tilde{u}_\phi \leq \phi$ a.e.;
- Enjoys mesh independence (the number of iterations does not grow in an unbounded manner as the mesh is refined).

💡 Use a path-following smoothed Moreau–Yosida regularization [mesh independence] followed by a few iterations of the primal-dual active set strategy [strict feasibility].

Semismooth Newton method

Chain rule

$$G_R(u_i) = \mathrm{Id} - G_S(\Phi(u_i))G_\Phi(u_i).$$

Theorem

Let $Y_p := \{\psi \in H_0^1(\Omega) : \Delta\psi \in L^p(\Omega)\}$, $\max(1, 2d/(d+2)) < p \leq \infty$. Then the obstacle map $S : \phi \mapsto u_\phi$, $S : Y_p \to H_0^1(\Omega)$ is *semismooth* with a Newton derivative G_S where $\|G_S\| \leq 1$. Moreover,

$$G_S(\phi)\zeta = \zeta + z_\zeta$$

where $z_\zeta \in H_0^1(\mathcal{I}(\phi))$ satisfies

$$(\nabla z_\zeta - \nabla\zeta, \nabla v) = 0 \ \text{ for all } v \in H_0^1(\mathcal{I}(\phi))$$

and $\mathcal{I}(\phi) = \{x \in \Omega : S(\phi)(x) < \phi(x) \text{ a.e.}\} \subseteq \Omega$.

Semismooth Newton method

Chain rule

$$G_R(u_i) = \mathrm{Id} - G_S(\Phi(u_i))G_\Phi(u_i).$$

Theorem

Let $Y_p := \{\psi \in H^1_0(\Omega) : \Delta\psi \in L^p(\Omega)\}$, $\max(1, 2d/(d+2)) < p \leq \infty$. Then the obstacle map $S : \phi \mapsto u_\phi$, $S : Y_p \to H^1_0(\Omega)$ is *semismooth* with a Newton derivative G_S where $\|G_S\| \leq 1$. Moreover,

$$G_S(\phi)\zeta = \zeta + z_\zeta$$

where $z_\zeta \in H^1_0(\mathcal{I}(\phi))$ satisfies

$$(\nabla z_\zeta - \nabla\zeta, \nabla v) = 0 \ \text{ for all } v \in H^1_0(\mathcal{I}(\phi))$$

and $\mathcal{I}(\phi) = \{x \in \Omega : S(\phi)(x) < \phi(x) \text{ a.e.}\} \subseteq \Omega$.

Semismooth Newton method

Chain rule

$$G_R(u_i) = \mathrm{Id} - G_S(\Phi(u_i))G_\Phi(u_i).$$

Theorem

Let $Y_p := \{\psi \in H_0^1(\Omega) : \Delta\psi \in L^p(\Omega)\}$, $\max(1, 2d/(d+2)) < p \leq \infty$.
Then the obstacle map $S : \phi \mapsto u_\phi$, $S : Y_p \to H_0^1(\Omega)$ is *semismooth* with a Newton derivative G_S where $\|G_S\| \leq 1$. Moreover,

$$G_S(\phi)\zeta = \zeta + z_\zeta$$

where $z_\zeta \in H_0^1(\mathcal{I}(\phi))$ satisfies

$$(\nabla z_\zeta - \nabla\zeta, \nabla v) = 0 \quad \text{for all } v \in H_0^1(\mathcal{I}(\phi))$$

and $\mathcal{I}(\phi) = \{x \in \Omega : S(\phi)(x) < \phi(x) \text{ a.e.}\} \subseteq \Omega$.

Semismooth Newton method

Chain rule

$$G_R(u_i) = \mathrm{Id} - G_S(\Phi(u_i))G_\Phi(u_i).$$

Theorem

Let $Y_p := \{\psi \in H^1_0(\Omega) : \Delta\psi \in L^p(\Omega)\}$, $\max(1, 2d/(d+2)) < p \le \infty$. Then the obstacle map $S : \phi \mapsto u_\phi$, $S : Y_p \to H^1_0(\Omega)$ is *semismooth* with a Newton derivative G_S where $\|G_S\| \le 1$. Moreover,

$$G_S(\phi)\zeta = \zeta + z_\zeta$$

where $z_\zeta \in H^1_0(\mathcal{I}(\phi))$ satisfies

$$(\nabla z_\zeta - \nabla\zeta, \nabla v) = 0 \quad \text{for all } v \in H^1_0(\mathcal{I}(\phi))$$

and $\mathcal{I}(\phi) = \{x \in \Omega : S(\phi)(x) < \phi(x) \text{ a.e.}\} \subseteq \Omega$.

Semismooth Newton method

Chain rule

$$G_R(u_i) = \mathrm{Id} - G_S(\Phi(u_i))G_\Phi(u_i).$$

Theorem

Let $Y_p := \{\psi \in H_0^1(\Omega) : \Delta\psi \in L^p(\Omega)\}$, $\max(1, 2d/(d+2)) < p \leq \infty$. Then the obstacle map $S : \phi \mapsto u_\phi$, $S : Y_p \to H_0^1(\Omega)$ is *semismooth* with a Newton derivative G_S where $\|G_S\| \leq 1$. Moreover,

$$G_S(\phi)\zeta = \zeta + z_\zeta$$

where $z_\zeta \in H_0^1(\mathcal{I}(\phi))$ satisfies

$$(\nabla z_\zeta - \nabla\zeta, \nabla v) = 0 \quad \text{for all } v \in H_0^1(\mathcal{I}(\phi))$$

and $\mathcal{I}(\phi) = \{x \in \Omega : S(\phi)(x) < \phi(x) \text{ a.e.}\} \subseteq \Omega$.

Semismooth Newton method

Chain rule

$$G_R(u_i) = \mathrm{Id} - G_S(\Phi(u_i))G_\Phi(u_i).$$

Theorem

Let $Y_p := \{\psi \in H^1_0(\Omega) : \Delta\psi \in L^p(\Omega)\}$, $\max(1, 2d/(d+2)) < p \leq \infty$. Then the obstacle map $S : \phi \mapsto u_\phi$, $S : Y_p \to H^1_0(\Omega)$ is *semismooth* with a Newton derivative G_S where $\|G_S\| \leq 1$. Moreover,

$$G_S(\phi)\zeta = \zeta + z_\zeta$$

where $z_\zeta \in H^1_0(\mathcal{I}(\phi))$ satisfies

$$(\nabla z_\zeta - \nabla\zeta, \nabla v) = 0 \ \text{ for all } v \in H^1_0(\mathcal{I}(\phi))$$

and $\mathcal{I}(\phi) = \{x \in \Omega : S(\phi)(x) < \phi(x) \text{ a.e.}\} \subseteq \Omega$.

Deriving the SSN System

SSN System

SSN update δ satisfies

$$[\mathrm{Id} - G_S(\Phi(u_i))G_\Phi(u_i)]\delta = -R(u_i).$$

Introduce auxiliary variables $\eta = G_\Phi(u_i)\delta$ and $\mu = G_S(\Phi(u_i))\eta - \eta$. Reformulate SSN system as

$$\begin{pmatrix} \mathrm{Id} & -\mathrm{Id} & -\mathrm{Id}|_{\mathcal{I}(u_i)} \\ G_\Phi(u_i) & -\mathrm{Id} & 0 \\ 0 & (G_S(\Phi(u_i)) - \mathrm{Id})|_{\mathcal{I}(u_i)} & -\mathrm{Id}|_{\mathcal{I}(u_i)} \end{pmatrix} \begin{pmatrix} \delta \\ \eta \\ \mu \end{pmatrix} = -\begin{pmatrix} R(u_i) \\ 0 \\ 0 \end{pmatrix}.$$

where $\mathcal{I}(u_i) := \{x \in \Omega : S(\Phi(u_i))(x) < \Phi(u_i)(x) \text{ a.e.}\}$.

Update: $u_{i+1} = u_i + \delta$.

Deriving the SSN System

SSN System

SSN update δ satisfies

$$[\mathrm{Id} - G_S(\Phi(u_i))G_\Phi(u_i)]\delta = -R(u_i).$$

Introduce auxiliary variables $\eta = G_\Phi(u_i)\delta$ and $\mu = G_S(\Phi(u_i))\eta - \eta$.
Reformulate SSN system as

$$\begin{pmatrix} \mathrm{Id} & -\mathrm{Id} & -\mathrm{Id}|_{\mathcal{I}(u_i)} \\ G_\Phi(u_i) & -\mathrm{Id} & 0 \\ 0 & (G_S(\Phi(u_i)) - \mathrm{Id})|_{\mathcal{I}(u_i)} & -\mathrm{Id}|_{\mathcal{I}(u_i)} \end{pmatrix} \begin{pmatrix} \delta \\ \eta \\ \mu \end{pmatrix} = - \begin{pmatrix} R(u_i) \\ 0 \\ 0 \end{pmatrix}.$$

where $\mathcal{I}(u_i) := \{x \in \Omega : S(\Phi(u_i))(x) < \Phi(u_i)(x) \text{ a.e.}\}$.

Update: $u_{i+1} = u_i + \delta$.

Deriving the SSN System

SSN System

SSN update δ satisfies

$$[\mathrm{Id} - G_S(\Phi(u_i))G_\Phi(u_i)]\delta = -R(u_i).$$

Introduce auxiliary variables $\eta = G_\Phi(u_i)\delta$ and $\mu = G_S(\Phi(u_i))\eta - \eta$. Reformulate SSN system as

$$\begin{pmatrix} \mathrm{Id} & -\mathrm{Id} & -\mathrm{Id}|_{\mathcal{I}(u_i)} \\ G_\Phi(u_i) & -\mathrm{Id} & 0 \\ 0 & (G_S(\Phi(u_i)) - \mathrm{Id})|_{\mathcal{I}(u_i)} & -\mathrm{Id}|_{\mathcal{I}(u_i)} \end{pmatrix} \begin{pmatrix} \delta \\ \eta \\ \mu \end{pmatrix} = - \begin{pmatrix} R(u_i) \\ 0 \\ 0 \end{pmatrix}.$$

where $\mathcal{I}(u_i) := \{x \in \Omega : S(\Phi(u_i))(x) < \Phi(u_i)(x) \text{ a.e.}\}$.

Update: $u_{i+1} = u_i + \delta$.

Deriving the SSN System

SSN System

SSN update δ satisfies

$$[\mathrm{Id} - G_S(\Phi(u_i))G_\Phi(u_i)]\delta = -R(u_i).$$

Introduce auxiliary variables $\eta = G_\Phi(u_i)\delta$ and $\mu = G_S(\Phi(u_i))\eta - \eta$. Reformulate SSN system as

$$\begin{pmatrix} \mathrm{Id} & -\mathrm{Id} & -\mathrm{Id}|_{\mathcal{I}(u_i)} \\ G_\Phi(u_i) & -\mathrm{Id} & 0 \\ 0 & (G_S(\Phi(u_i)) - \mathrm{Id})|_{\mathcal{I}(u_i)} & -\mathrm{Id}|_{\mathcal{I}(u_i)} \end{pmatrix} \begin{pmatrix} \delta \\ \eta \\ \mu \end{pmatrix} = - \begin{pmatrix} R(u_i) \\ 0 \\ 0 \end{pmatrix}.$$

where $\mathcal{I}(u_i) := \{x \in \Omega : S(\Phi(u_i))(x) < \Phi(u_i)(x) \text{ a.e.}\}$.

Update: $u_{i+1} = u_i + \delta$.

Implementing the active-set

After a piecewise (bi)linear FEM discretization, the active set can be directly implemented by deleting the corresponding discrete active set rows and columns.

Active-set SSN system

$$\begin{pmatrix} M & -M & -M_{:,\mathfrak{I}} \\ A & -M & 0 \\ 0 & B_{\mathfrak{I},:} & -M_{\mathfrak{I},\mathfrak{I}} \end{pmatrix} \begin{pmatrix} \boldsymbol{\delta} \\ \boldsymbol{\eta} \\ \boldsymbol{\mu}_{\mathfrak{I}} \end{pmatrix} = \begin{pmatrix} -\boldsymbol{r} \\ 0 \\ 0 \end{pmatrix}, \quad \boldsymbol{\mu}_{\mathrm{dofs}\setminus\mathfrak{I}} = 0,$$

$\mathfrak{I} = \{j \in \mathrm{dofs} : [S(\Phi(u_i))]_j < [\Phi(u_i)]_j\}$, $A \approx G_\Phi(u_i)$ and $B \approx G_S(\Phi(u_i)) - \mathrm{Id}$.

Globalization & inexactness

Globalization

For $u, v \in X$ and a $\gamma \in [0, 1)$ suppose that

$$\|S(\Phi(u)) - S(\Phi(v))\|_X \leq \gamma \|u - v\|_X,$$
$$\sup_{u \in X} \|G_S(\Phi(u)) G_\Phi(u)\| \leq \gamma.$$

A simple safeguarding technique that ensures globalization is take the next iterate as

$$u_{i+1} = \begin{cases} u_i + \delta & \text{if } \|R(u_i + \delta)\|_X \leq \|R(S(\Phi(u_i)))\|_X, \\ S(\Phi(u_i)) & \text{if } \|R(S(\Phi(u_i)))\|_X < \|R(u_i + \delta)\|_X. \end{cases}$$

⚠Globalization techniques require efficient evaluations of $R(u)$. Here each evaluation requires an obstacle problem solve.

Globalization & inexactness

Globalization

For $u, v \in X$ and a $\gamma \in [0, 1)$ suppose that

$$\|S(\Phi(u)) - S(\Phi(v))\|_X \leq \gamma \|u - v\|_X,$$
$$\sup_{u \in X} \|G_S(\Phi(u)) G_\Phi(u)\| \leq \gamma.$$

A simple safeguarding technique that ensures globalization is take the next iterate as

$$u_{i+1} = \begin{cases} u_i + \delta & \text{if } \|R(u_i + \delta)\|_X \leq \|R(S(\Phi(u_i)))\|_X, \\ S(\Phi(u_i)) & \text{if } \|R(S(\Phi(u_i)))\|_X < \|R(u_i + \delta)\|_X. \end{cases}$$

⚠Globalization techniques require efficient evaluations of $R(u)$. Here each evaluation requires an obstacle problem solve.

Globalization & inexactness

Globalization

For $u, v \in X$ and a $\gamma \in [0, 1)$ suppose that

$$\|S(\Phi(u)) - S(\Phi(v))\|_X \leq \gamma \|u - v\|_X,$$
$$\sup_{u \in X} \|G_S(\Phi(u)) G_\Phi(u)\| \leq \gamma.$$

A simple safeguarding technique that ensures globalization is take the next iterate as

$$u_{i+1} = \begin{cases} u_i + \delta & \text{if } \|R(u_i + \delta)\|_X \leq \|R(S(\Phi(u_i)))\|_X, \\ S(\Phi(u_i)) & \text{if } \|R(S(\Phi(u_i)))\|_X < \|R(u_i + \delta)\|_X. \end{cases}$$

⚠Globalization techniques require efficient evaluations of $R(u)$. Here each evaluation requires an obstacle problem solve.

Globalization & inexactness

Inexactness

The SSN update δ does not need to be computed exactly. An inexact strategy considers the updates

$$\|R(u_i) + G_R(u_i)\delta\|_X \leq \rho_i \|R(u_i)\|_X.$$

If $\rho_i \to 0$ as $i \to \infty$ then the SSN strategy converges with a local superlinear rate to the solution.

This means that we do not need to compute the RHS $R(u_i)$ exactly at each SSN step.

Globalization & inexactness

Inexactness

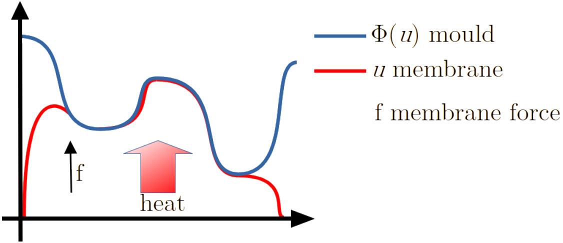
The SSN update δ does not need to be computed exactly. An inexact strategy considers the updates

$$\|R(u_i) + G_R(u_i)\delta\|_X \leq \rho_i \|R(u_i)\|_X.$$

If $\rho_i \to 0$ as $i \to \infty$ then the SSN strategy converges with a local superlinear rate to the solution.

This means that we do not need to compute the RHS $R(u_i)$ exactly at each SSN step.

Thermoforming: an obstacle-type QVI



Model

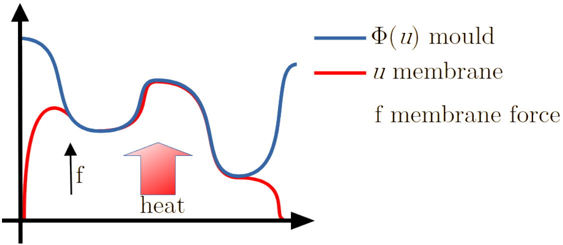
Find $u \in H_0^1(\Omega)$ satisfying $u \leq \Phi(u) := \Phi_0 + \psi T$ and

$$(\nabla u, \nabla(v-u))_{L^2(\Omega)} - (f, v-u)_{L^2(\Omega)} \geq 0 \quad \text{for all } v \in H_0^1(\Omega), v \leq \Phi(u),$$

with T as the solution of

$$kT - \Delta T = g(\Phi_0 + \psi T - u) \text{ in } \Omega, \qquad \partial_\nu T = 0 \text{ on } \partial\Omega,$$

Thermoforming: an obstacle-type QVI



Model

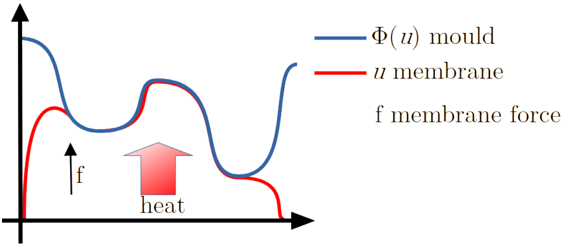
Find $u \in H_0^1(\Omega)$ satisfying $u \leq \Phi(u) := \Phi_0 + \psi T$ and

$$(\nabla u, \nabla(v-u))_{L^2(\Omega)} - (f, v-u)_{L^2(\Omega)} \geq 0 \quad \text{for all } v \in H_0^1(\Omega), v \leq \Phi(u),$$

with T as the solution of

$$kT - \Delta T = g(\Phi_0 + \psi T - u) \text{ in } \Omega, \qquad \partial_\nu T = 0 \text{ on } \partial\Omega,$$

Thermoforming: an obstacle-type QVI



Model

Find $u \in H_0^1(\Omega)$ satisfying $u \leq \Phi(u) := \Phi_0 + \psi T$ and

$$(\nabla u, \nabla(v-u))_{L^2(\Omega)} - (f, v-u)_{L^2(\Omega)} \geq 0 \quad \text{for all } v \in H_0^1(\Omega), v \leq \Phi(u),$$

with T as the solution of

$$kT - \Delta T = g(\Phi_0 + \psi T - u) \text{ in } \Omega, \qquad \partial_\nu T = 0 \text{ on } \partial\Omega,$$

Some properties of the thermoforming problem

Assumptions

- $\Omega \subset \mathbb{R}^d$, $d \in \mathbb{N}$, is a Lipschitz domain;
- $\Phi_0 \in L^{2+\epsilon}(\Omega)$ for some $\epsilon > 0$ and $\psi \in C^2(\bar{\Omega})$ and $\psi = 0$ on $\partial\Omega$;
- $g : \mathbb{R} \to \mathbb{R}$ is globally Lipschitz continuous, nonincreasing, and Newton differentiable.

Results

- There exists a solution (u, T) to the thermoforming problem;
- Φ is Newton differentiable from $H_0^1(\Omega)$ to Y_2;
- Φ is locally Lipschitz from $H_0^1(\Omega)$ to Y_2,

and the "contraction" coefficient is given by

$$\gamma = C_P(\Omega)\mathrm{Lip}(g)\left(\|\psi\|_{L^\infty(\Omega)}k^{-1/2} + \|\nabla\psi\|_{L^\infty(\Omega)}k^{-1}\right).$$

Some properties of the thermoforming problem

Assumptions

- $\Omega \subset \mathbb{R}^d$, $d \in \mathbb{N}$, is a Lipschitz domain;
- $\Phi_0 \in L^{2+\epsilon}(\Omega)$ for some $\epsilon > 0$ and $\psi \in C^2(\bar{\Omega})$ and $\psi = 0$ on $\partial\Omega$;
- $g : \mathbb{R} \to \mathbb{R}$ is globally Lipschitz continuous, nonincreasing, and Newton differentiable.

Results

- There exists a solution (u, T) to the thermoforming problem;
- Φ is Newton differentiable from $H_0^1(\Omega)$ to Y_2;
- Φ is locally Lipschitz from $H_0^1(\Omega)$ to Y_2,

and the "contraction" coefficient is given by

$$\gamma = C_P(\Omega)\mathrm{Lip}(g)\left(\|\psi\|_{L^\infty(\Omega)}k^{-1/2} + \|\nabla\psi\|_{L^\infty(\Omega)}k^{-1}\right).$$

FEM discretization & algorithm

Piecewise (bi)linear FEM discretization for u and T.

Algorithm for SSN step

`Step 1.` Compute $R(u_i) = u_i - S(\Phi(u_i))$.

`Step 1.1.` $\Phi(u_i)$ requires a nonlinear solve [Newton].

`Step 1.2.` $S(\Phi(u_i))$ is a VI solve [smoothed Moreau–Yosida path-following + feasibility restoration with the primal-dual active set strategy].

`Step 2.` Compute SSN update coefficient vector δ by solving:

$$\begin{pmatrix} \mathrm{Id} & -\mathrm{Id} & -\mathrm{Id}|_I \\ g' & -\Delta + k + g' \cdot \varphi & 0 \\ 0 & (\varphi\nabla + \nabla(\varphi)\mathrm{Id})|_I & (-\Delta)|_I \end{pmatrix} \begin{pmatrix} \delta_h \\ \xi_h \\ \mu_h \end{pmatrix} \approx \begin{pmatrix} M & -M & -M_{:,\mathfrak{I}} \\ D & B & 0 \\ 0 & C_{\mathfrak{I},:} & A_{\mathfrak{I},\mathfrak{I}} \end{pmatrix} \begin{pmatrix} \boldsymbol{\delta} \\ \boldsymbol{\xi} \\ \boldsymbol{\mu} \end{pmatrix} = \begin{pmatrix} -\boldsymbol{r} \\ 0 \\ 0 \end{pmatrix}$$

$\mathfrak{I} = \{j \in \mathrm{dofs} : [S(\Phi(u_i))]_j < [\Phi_0 + \Phi(u_0)]_j\}$.

`Step 3.` $u_{i+1} = u_i + \delta$.

FEM discretization & algorithm

Piecewise (bi)linear FEM discretization for u and T.

Algorithm for SSN step

`Step 1.` Compute $R(u_i) = u_i - S(\Phi(u_i))$.

`Step 1.1.` $\Phi(u_i)$ requires a nonlinear solve [Newton].

`Step 1.2.` $S(\Phi(u_i))$ is a VI solve [smoothed Moreau–Yosida path-following + feasibility restoration with the primal-dual active set strategy].

`Step 2.` Compute SSN update coefficient vector δ by solving:

$$\begin{pmatrix} \mathrm{Id} & -\mathrm{Id} & -\mathrm{Id}|_I \\ g' & -\Delta + k + g' \cdot \varphi & 0 \\ 0 & (\varphi\nabla + \nabla(\varphi)\mathrm{Id})|_I & (-\Delta)|_I \end{pmatrix} \begin{pmatrix} \delta_h \\ \xi_h \\ \mu_h \end{pmatrix} \approx \begin{pmatrix} M & -M & -M_{:,\mathfrak{I}} \\ D & B & 0 \\ 0 & C_{\mathfrak{I},:} & A_{\mathfrak{I},\mathfrak{I}} \end{pmatrix} \begin{pmatrix} \delta \\ \xi \\ \mu \end{pmatrix} = \begin{pmatrix} -r \\ 0 \\ 0 \end{pmatrix}$$

$\mathfrak{I} = \{j \in \mathrm{dofs} : [S(\Phi(u_i))]_j < [\Phi_0 + \Phi(u_0)]_j\}$.

`Step 3.` $u_{i+1} = u_i + \delta$.

FEM discretization & algorithm

Piecewise (bi)linear FEM discretization for u and T.

Algorithm for SSN step

`Step 1.` Compute $R(u_i) = u_i - S(\Phi(u_i))$.

`Step 1.1.` $\Phi(u_i)$ requires a nonlinear solve [Newton].

`Step 1.2.` $S(\Phi(u_i))$ is a VI solve [smoothed Moreau–Yosida path-following + feasibility restoration with the primal-dual active set strategy].

`Step 2.` Compute SSN update coefficient vector δ by solving:

$$\begin{pmatrix} \mathrm{Id} & -\mathrm{Id} & -\mathrm{Id}|_I \\ g' & -\Delta + k + g' \cdot \varphi & 0 \\ 0 & (\varphi\nabla + \nabla(\varphi)\mathrm{Id})|_I & (-\Delta)|_I \end{pmatrix} \begin{pmatrix} \delta_h \\ \xi_h \\ \mu_h \end{pmatrix} \approx \begin{pmatrix} M & -M & -M_{:,\mathfrak{I}} \\ D & B & 0 \\ 0 & C_{\mathfrak{I},:} & A_{\mathfrak{I},\mathfrak{I}} \end{pmatrix} \begin{pmatrix} \delta \\ \xi \\ \mu \end{pmatrix} = \begin{pmatrix} -r \\ 0 \\ 0 \end{pmatrix}$$

$\mathfrak{I} = \{j \in \mathrm{dofs} : [S(\Phi(u_i))]_j < [\Phi_0 + \Phi(u_0)]_j\}$.

`Step 3.` $u_{i+1} = u_i + \delta$.

FEM discretization & algorithm

Piecewise (bi)linear FEM discretization for u and T.

Algorithm for SSN step

`Step 1.` Compute $R(u_i) = u_i - S(\Phi(u_i))$.

`Step 1.1.` $\Phi(u_i)$ requires a nonlinear solve [Newton].

`Step 1.2.` $S(\Phi(u_i))$ is a VI solve [smoothed Moreau–Yosida path-following + feasibility restoration with the primal-dual active set strategy].

`Step 2.` Compute SSN update coefficient vector δ by solving:

$$\begin{pmatrix} \mathrm{Id} & -\mathrm{Id} & -\mathrm{Id}|_I \\ g' & -\Delta + k + g' \cdot \varphi & 0 \\ 0 & (\varphi\nabla + \nabla(\varphi)\mathrm{Id})|_I & (-\Delta)|_I \end{pmatrix} \begin{pmatrix} \delta_h \\ \xi_h \\ \mu_h \end{pmatrix} \approx \begin{pmatrix} M & -M & -M_{:,\mathfrak{I}} \\ D & B & 0 \\ 0 & C_{\mathfrak{I},:} & A_{\mathfrak{I},\mathfrak{I}} \end{pmatrix} \begin{pmatrix} \delta \\ \xi \\ \mu \end{pmatrix} = \begin{pmatrix} -r \\ 0 \\ 0 \end{pmatrix}$$

$\mathfrak{I} = \{j \in \mathrm{dofs} : [S(\Phi(u_i))]_j < [\Phi_0 + \Phi(u_0)]_j\}$.

`Step 3.` $u_{i+1} = u_i + \delta$.

FEM discretization & algorithm

Piecewise (bi)linear FEM discretization for u and T.

Algorithm for SSN step

`Step 1.` Compute $R(u_i) = u_i - S(\Phi(u_i))$.
`Step 1.1.` $\Phi(u_i)$ requires a nonlinear solve [Newton].
`Step 1.2.` $S(\Phi(u_i))$ is a VI solve [smoothed Moreau–Yosida path-following + feasibility restoration with the primal-dual active set strategy].

`Step 2.` Compute SSN update coefficient vector δ by solving:

$$\begin{pmatrix} \mathrm{Id} & -\mathrm{Id} & -\mathrm{Id}|_I \\ g' & -\Delta + k + g' \cdot \varphi & 0 \\ 0 & (\varphi\nabla + \nabla(\varphi)\mathrm{Id})|_I & (-\Delta)|_I \end{pmatrix} \begin{pmatrix} \delta_h \\ \xi_h \\ \mu_h \end{pmatrix} \approx \begin{pmatrix} M & -M & -M_{:,\mathfrak{I}} \\ D & B & 0 \\ 0 & C_{\mathfrak{I},:} & A_{\mathfrak{I},\mathfrak{I}} \end{pmatrix} \begin{pmatrix} \boldsymbol{\delta} \\ \boldsymbol{\xi} \\ \boldsymbol{\mu} \end{pmatrix} = \begin{pmatrix} -\boldsymbol{r} \\ 0 \\ 0 \end{pmatrix}$$

$\mathfrak{I} = \{j \in \mathrm{dofs} : [S(\Phi(u_i))]_j < [\Phi_0 + \Phi(u_0)]_j\}$.

`Step 3.` $u_{i+1} = u_i + \delta$.

FEM discretization & algorithm

Piecewise (bi)linear FEM discretization for u and T.

Algorithm for SSN step

`Step 1.` Compute $R(u_i) = u_i - S(\Phi(u_i))$.
`Step 1.1.` $\Phi(u_i)$ requires a nonlinear solve [Newton].
`Step 1.2.` $S(\Phi(u_i))$ is a VI solve [smoothed Moreau–Yosida path-following + feasibility restoration with the primal-dual active set strategy].

`Step 2.` Compute SSN update coefficient vector δ by solving:

$$\begin{pmatrix} \mathrm{Id} & -\mathrm{Id} & -\mathrm{Id}|_I \\ g' & -\Delta + k + g' \cdot \varphi & 0 \\ 0 & (\varphi\nabla + \nabla(\varphi)\mathrm{Id})|_I & (-\Delta)|_I \end{pmatrix} \begin{pmatrix} \delta_h \\ \xi_h \\ \mu_h \end{pmatrix} \approx \begin{pmatrix} M & -M & -M_{:,\mathfrak{I}} \\ D & B & 0 \\ 0 & C_{\mathfrak{I},:} & A_{\mathfrak{I},\mathfrak{I}} \end{pmatrix} \begin{pmatrix} \boldsymbol{\delta} \\ \boldsymbol{\xi} \\ \boldsymbol{\mu} \end{pmatrix} = \begin{pmatrix} -\boldsymbol{r} \\ 0 \\ 0 \end{pmatrix}$$

$\mathfrak{I} = \{j \in \mathrm{dofs} : [S(\Phi(u_i))]_j < [\Phi_0 + \Phi(u_0)]_j\}$.

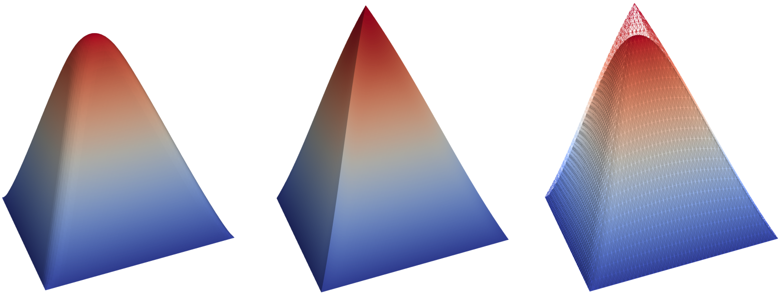
`Step 3.` $u_{i+1} = u_i + \delta$.

Example 1: setup

$$\Omega = (0,1)^2, \qquad \Phi_0(x_1,x_2) = 1 - 2\max(|x_1 - 0.5|, |x_2 - 0.5|),$$

$$f(x_1,x_2) = 25, \qquad \psi(x_1,x_2) = \sin(\pi x_1)\sin(\pi x_2), \qquad k = 1,$$

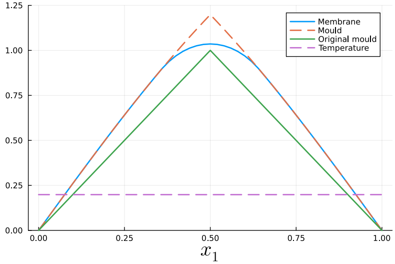
$$g(s) = \begin{cases} 1/5 & \text{if } s \le 0, \\ (1-s)/5 & \text{if } 0 < s < 1, \\ 0 & \text{otherwise.} \end{cases}$$



(a) Membrane u (b) Mould $\Phi_0 + \psi T$ (c) Membrane & Mould

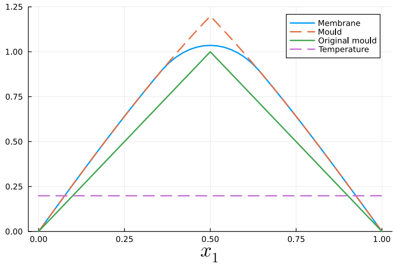
Example 1: convergence

- MY-Newton: Regularize the QVI with a smoothed Moreau–Yosida penalty in the obstacle problem [Solution is not strictly feasible].
- Fixed point method: $u_{i+1} = S(\Phi(u_i))$ [Converges linearly].



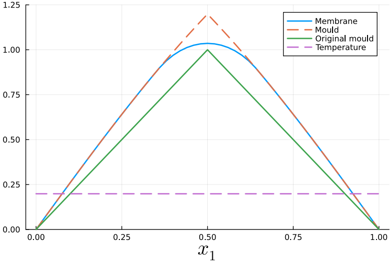
Example 1: convergence

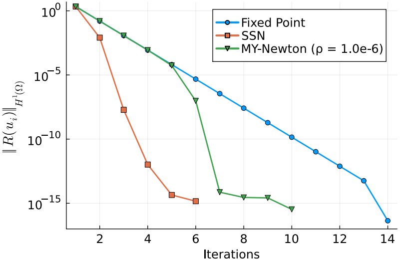
- MY-Newton: Regularize the QVI with a smoothed Moreau–Yosida penalty in the obstacle problem [Solution is not strictly feasible].
- Fixed point method: $u_{i+1} = S(\Phi(u_i))$ [Converges linearly].



Example 1: convergence

- MY-Newton: Regularize the QVI with a smoothed Moreau–Yosida penalty in the obstacle problem [Solution is not strictly feasible].
- Fixed point method: $u_{i+1} = S(\Phi(u_i))$ [Converges linearly].





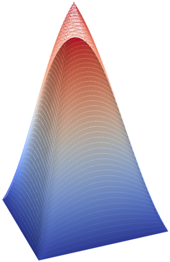
Example 1: mesh independence

	Outer loop	Evaluate Φ	Evaluate S	
h	SSN	Newton	PFMY	$+$PDAS
0.04	4	9	159	10
0.02	4	9	185	17
0.01	3	8	150	11
0.00667	3	8	158	11
0.005	3	8	158	17
0.004	4	8	199	21
0.00333	4	7	184	21

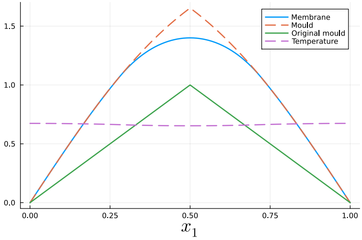
Table: Mesh independence of the SSN.

Example 2: setup

Only change: $$g(s) = \begin{cases} 1 & \text{if } s \leq 0, \\ (1 - 100s) & \text{if } 0 < s < 1/100, \\ 0 & \text{otherwise.} \end{cases}$$

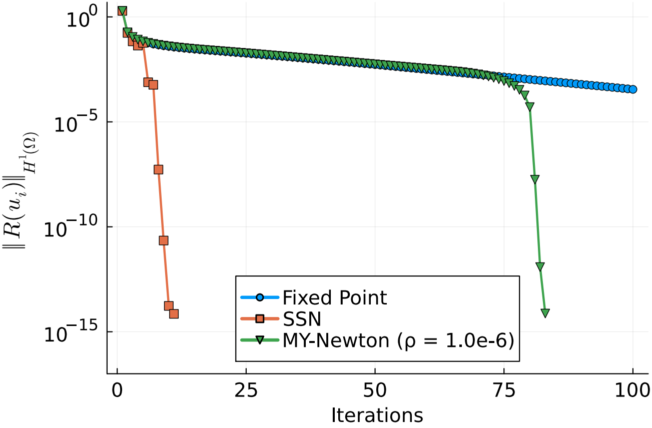


(a) Membrane & Mould



(b) Slice at $x_2 = 1/2$

Example 2: convergence



Conclusions

- A semismooth Newton method for solving obstacle-type QVIs;
- An active-set strategy implemented in Gridap & Firedrake;
- Theory relies on recent semismooth results for the obstacle map S.

A globalized inexact semismooth Newton method for nonsmooth fixed point equations involving variational inequalities

A. Alphonse, C. Christof, M. Hintermüller, I. P. A. Papadopoulos, 2024, https://arxiv.org/abs/2409.19637.

Software packages

https://github.com/ioannisPApapadopoulos/semismoothQVIs

https://github.com/ioannisPApapadopoulos/SemismoothQVIs.jl

Conclusions

- A semismooth Newton method for solving obstacle-type QVIs;
- An active-set strategy implemented in Gridap & Firedrake;
- Theory relies on recent semismooth results for the obstacle map S.

A globalized inexact semismooth Newton method for nonsmooth fixed point equations involving variational inequalities

A. Alphonse, C. Christof, M. Hintermüller, I. P. A. Papadopoulos, 2024, `https://arxiv.org/abs/2409.19637`.

Software packages

`https://github.com/ioannisPApapadopoulos/semismoothQVIs`

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

Conclusions

- A semismooth Newton method for solving obstacle-type QVIs;
- An active-set strategy implemented in Gridap & Firedrake;
- Theory relies on recent semismooth results for the obstacle map S.

A globalized inexact semismooth Newton method for nonsmooth fixed point equations involving variational inequalities

A. Alphonse, C. Christof, M. Hintermüller, I. P. A. Papadopoulos, 2024, `https://arxiv.org/abs/2409.19637`.

Software packages

`https://github.com/ioannisPApapadopoulos/semismoothQVIs`

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

Thank you for listening!

✉ papadopoulos@wias-berlin.de