

Computing multiple solutions of nonlinear problems with deflation

Patrick E. Farrell

University of Oxford and Charles University

March 23 2026

Multiple solutions

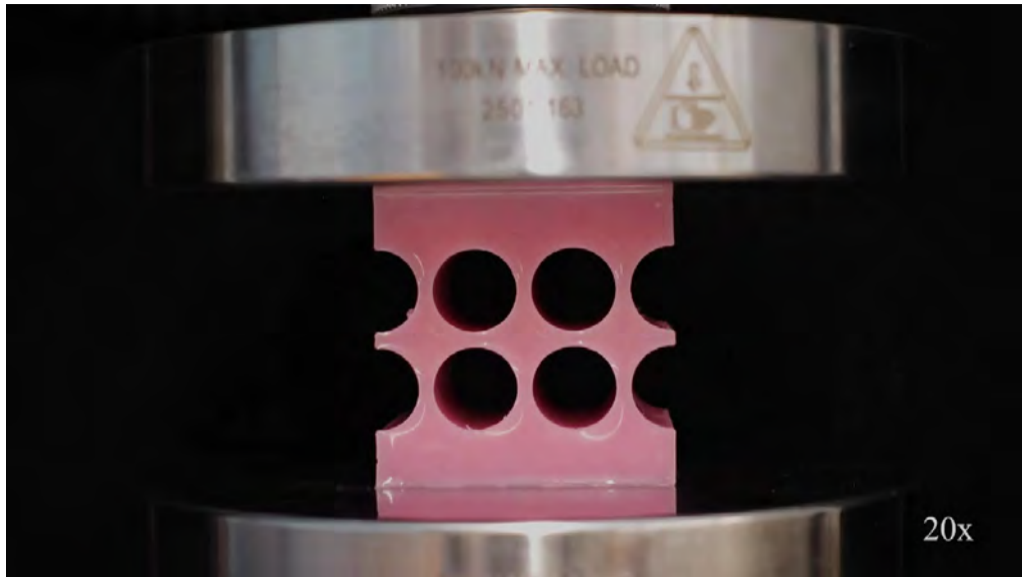
Nonlinear PDEs can support multiple, isolated solutions.

Multiple solutions

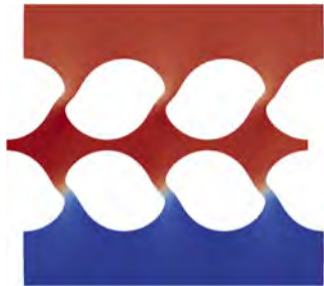
Nonlinear PDEs can support multiple, isolated solutions.

This talk

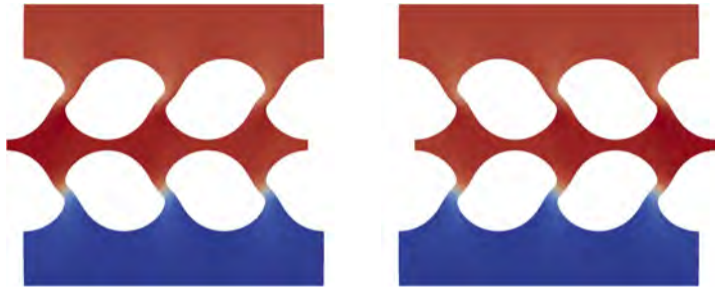
An algorithm for computing multiple solutions with Newton's method.



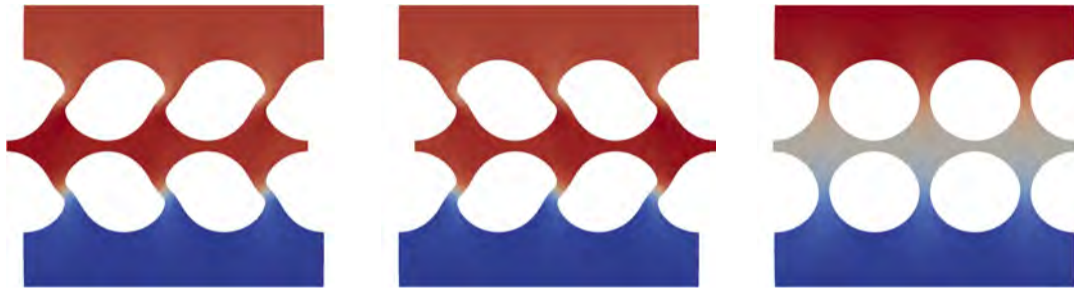
From Medina, Farrell, Bertoldi, Rycroft (2020).



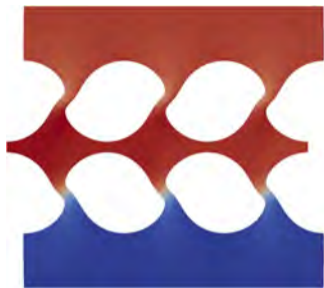
Solutions of a hyperelastic PDE.



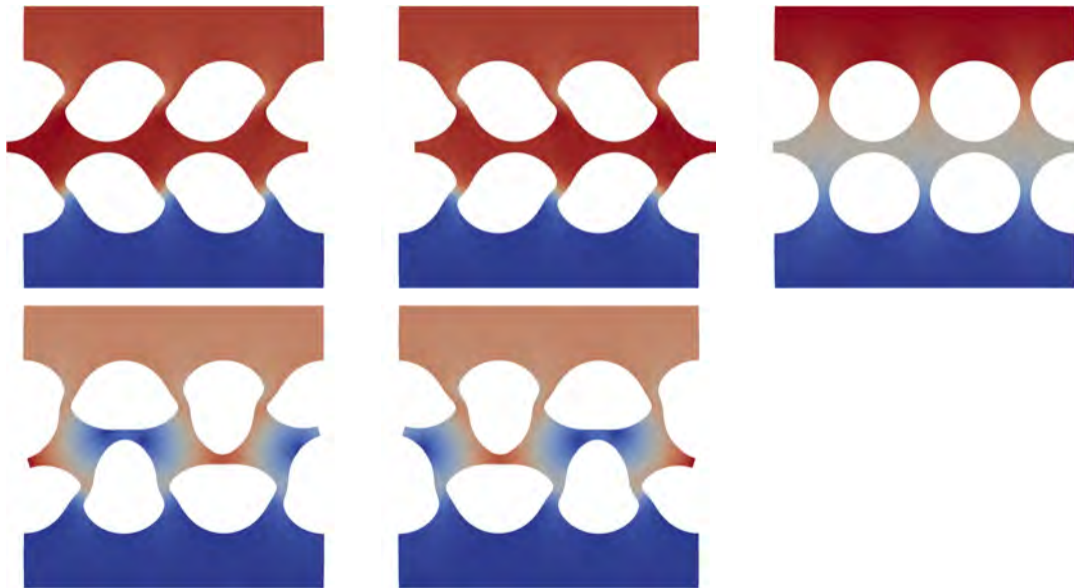
Solutions of a hyperelastic PDE.



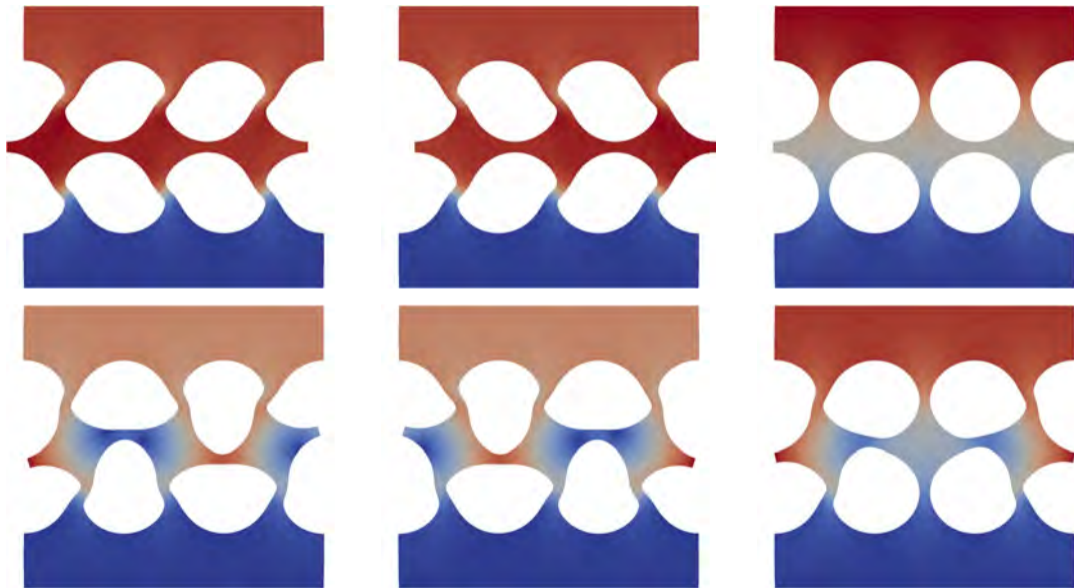
Solutions of a hyperelastic PDE.



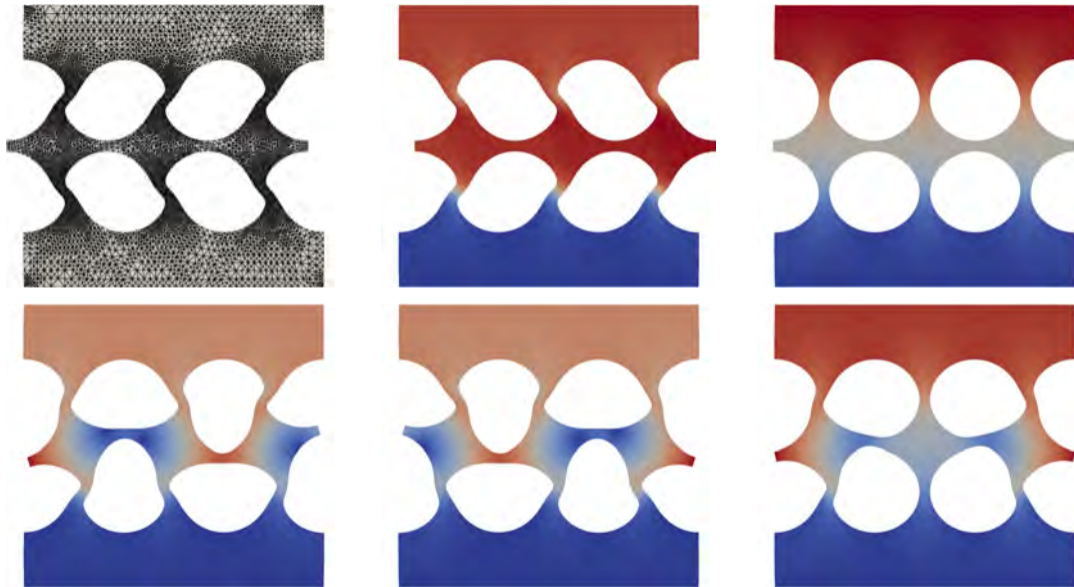
Solutions of a hyperelastic PDE.



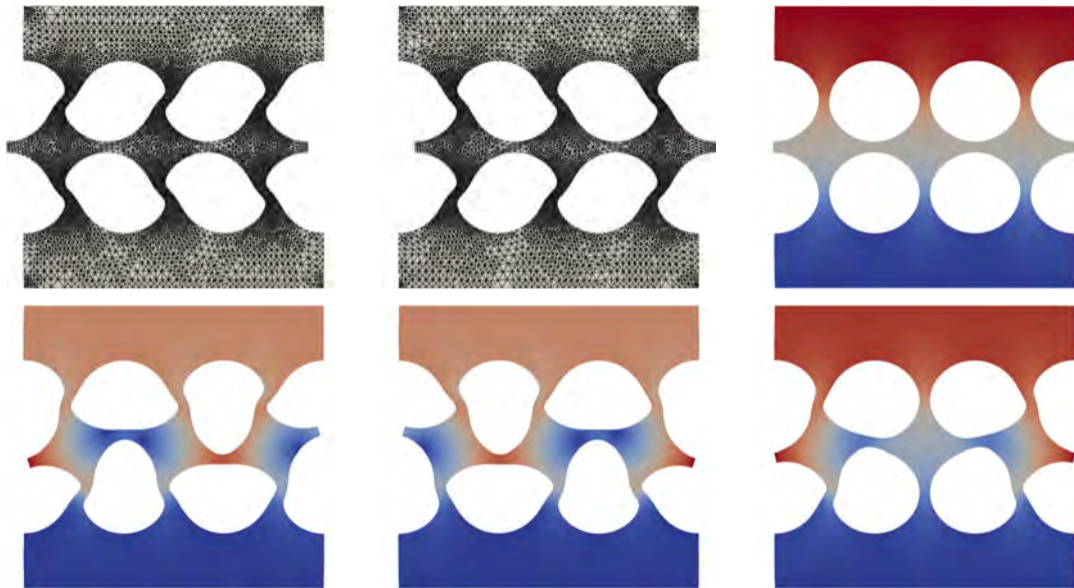
Solutions of a hyperelastic PDE.



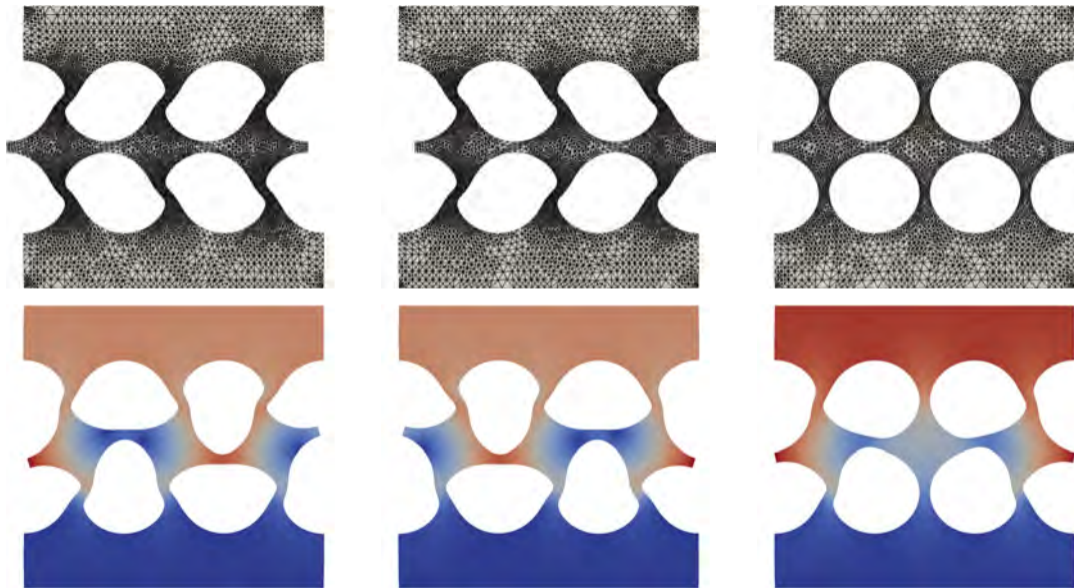
Solutions of a hyperelastic PDE.



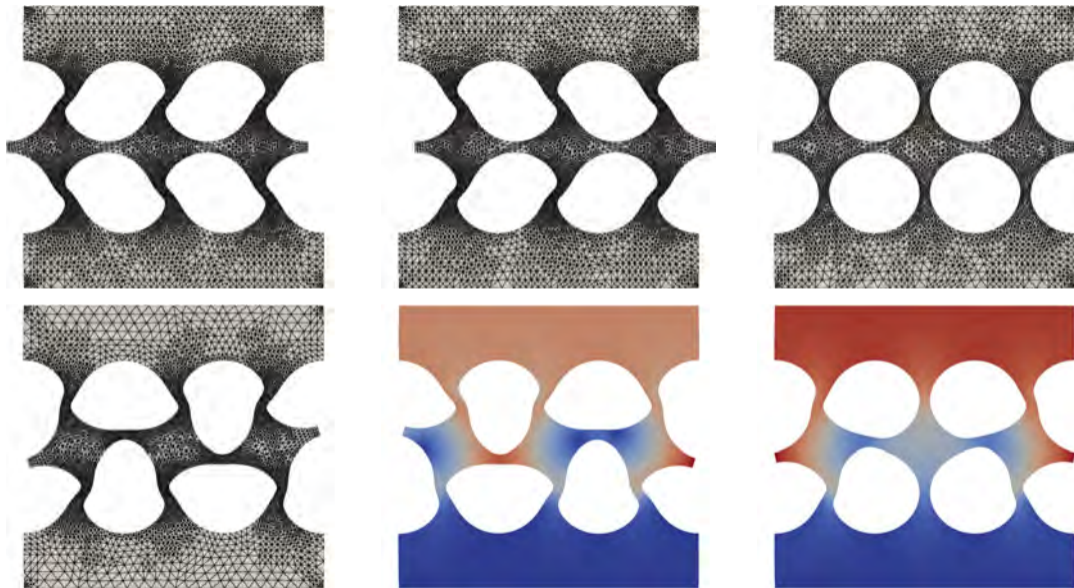
Adapted meshes (automated DWR + GMG, $|E(u) - E(u_h)| \leq 10^{-2}$).



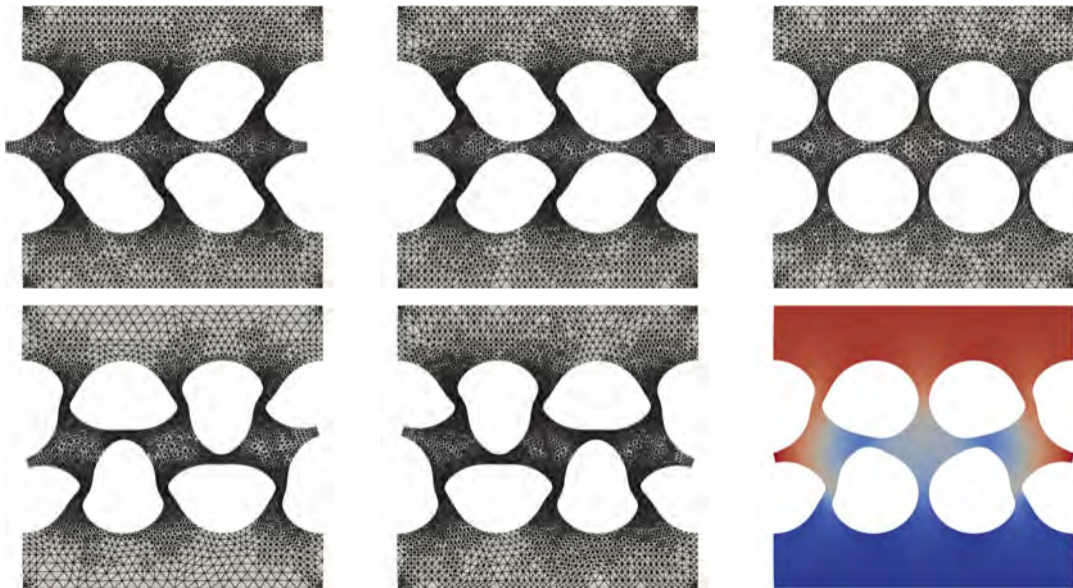
Adapted meshes (automated DWR + GMG, $|E(u) - E(u_h)| \leq 10^{-2}$).



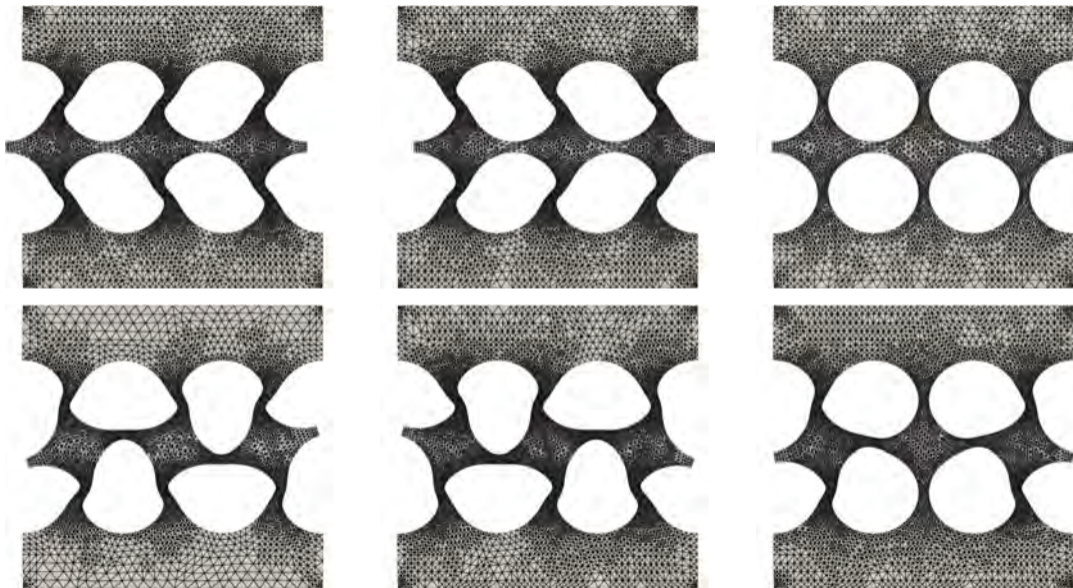
Adapted meshes (automated DWR + GMG, $|E(u) - E(u_h)| \leq 10^{-2}$).



Adapted meshes (automated DWR + GMG, $|E(u) - E(u_h)| \leq 10^{-2}$).

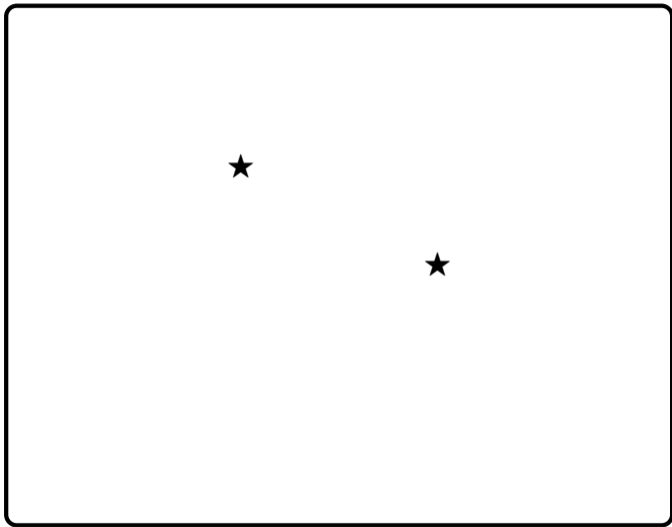


Adapted meshes (automated DWR + GMG, $|E(u) - E(u_h)| \leq 10^{-2}$).



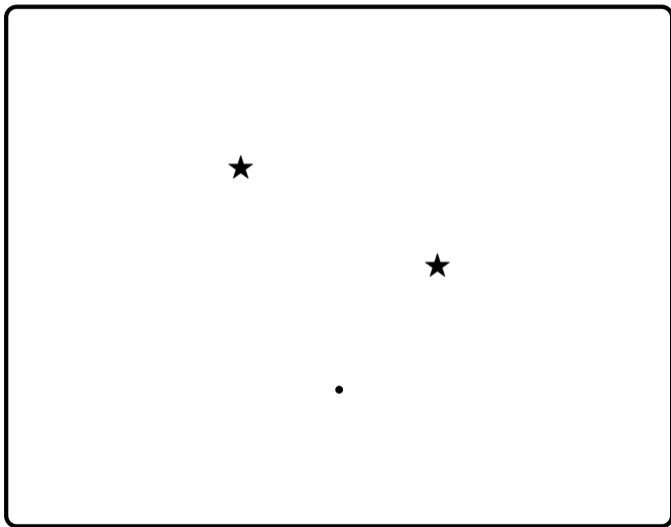
Adapted meshes (automated DWR + GMG, $|E(u) - E(u_h)| \leq 10^{-2}$).

Should we care?



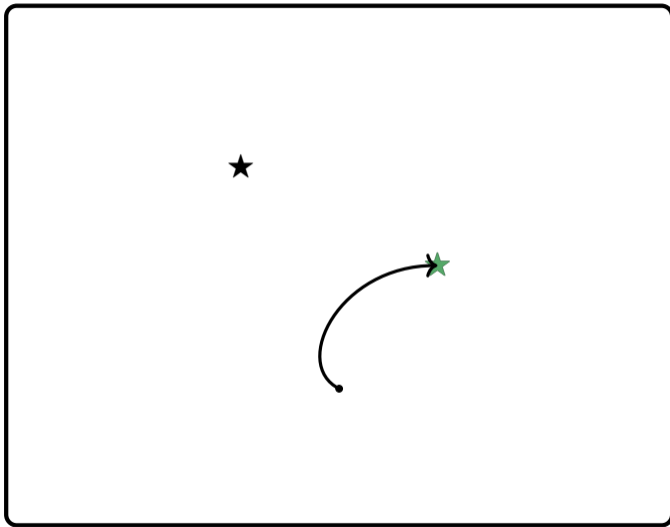
A PDE with two unknown solutions.

Should we care?



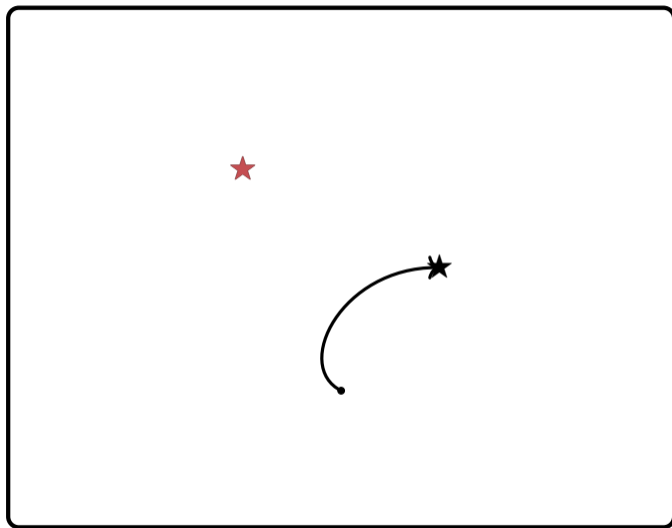
Start Newton from some initial guess.

Should we care?



We converge to one solution, our prediction.

Should we care?



But nature has chosen another (unknown) solution!

Warning

All your calculations can be correct, but you still *get the wrong answer*.

Warning

All your calculations can be correct, but you still *get the wrong answer*.

Mathematical formulation

Compute the multiple *solutions* u^* of a stationary nonlinear equation

$$F(u^*, \lambda) = 0$$

$$F \in C^1(X \times \mathbb{R}, Y)$$

as a function of a parameter $\lambda \in \mathbb{R}$. (X, Y Banach spaces.)

Warning

All your calculations can be correct, but you still *get the wrong answer*.

Mathematical formulation

Compute the multiple *solutions* u^* of a stationary nonlinear equation

$$F(u^*, \lambda) = 0$$
$$F \in C^1(X \times \mathbb{R}, Y)$$

as a function of a parameter $\lambda \in \mathbb{R}$. (X, Y Banach spaces.)

Limitation

We (usually) can't guarantee to find *all* solutions. But finding many is better than finding one.

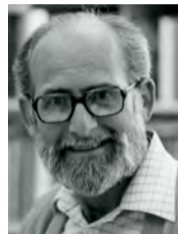
Section 1

Classical techniques

Basic idea of numerical bifurcation analysis:

procedure ANALYSE(u_0, λ_0)

end procedure

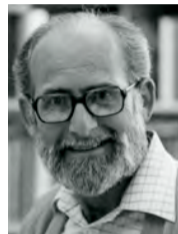


Herbert Keller

Basic idea of numerical bifurcation analysis:

procedure ANALYSE(u_0, λ_0)
continue branch of solutions;

end procedure



Herbert Keller

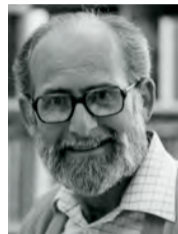
Continuation

Extending our knowledge of the branch to other values of λ .

Basic idea of numerical bifurcation analysis:

procedure ANALYSE(u_0, λ_0)
continue branch of solutions;
detect bifurcations on the branch;

end procedure



Herbert Keller

Bifurcation detection

Discovering when a bifurcation has occurred on the branch.

Basic idea of numerical bifurcation analysis:

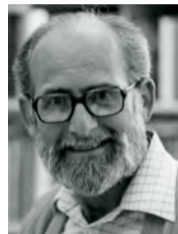
procedure ANALYSE(u_0, λ_0)

continue branch of solutions;

detect bifurcations on the branch;

localise bifurcations;

end procedure



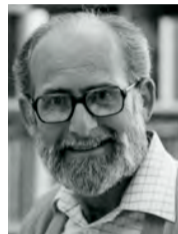
Herbert Keller

Bifurcation localisation

Identifying precisely the bifurcation point.

Basic idea of numerical bifurcation analysis:

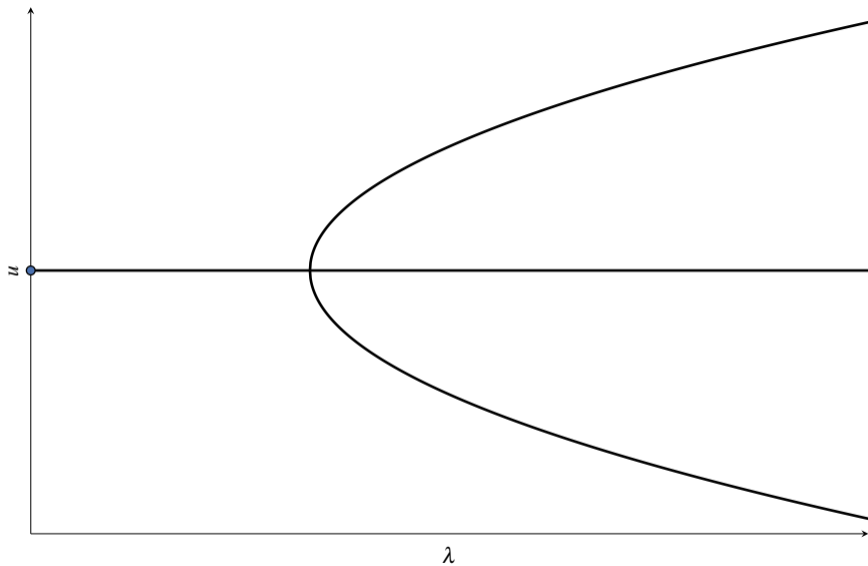
procedure ANALYSE(u_0, λ_0)
continue branch of solutions;
detect bifurcations on the branch;
localise bifurcations;
switch branches at bifurcations, and recurse.
end procedure



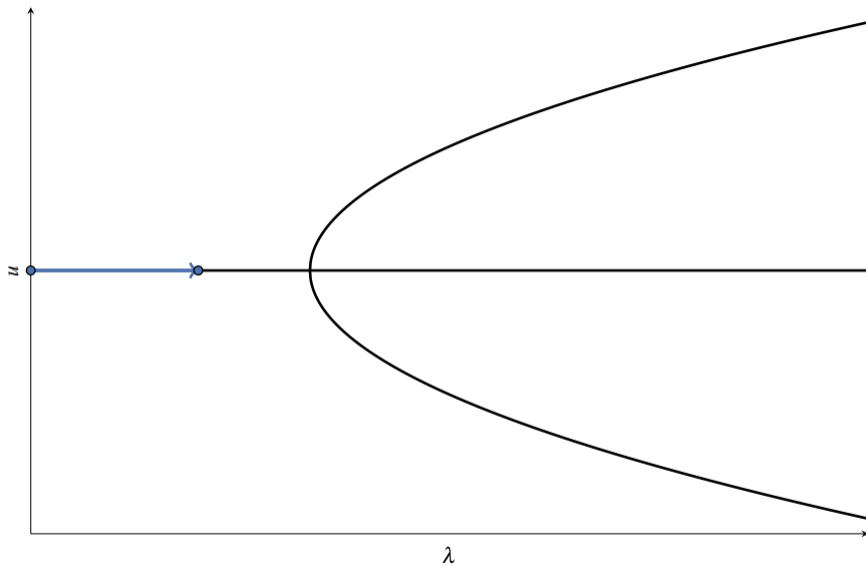
Herbert Keller

Branch switching

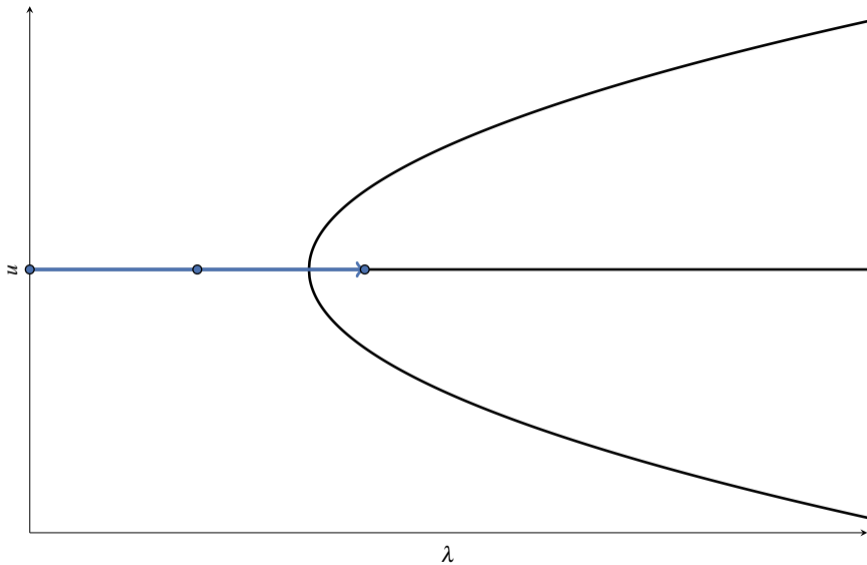
Constructing the emanating branches, and analysing them recursively.



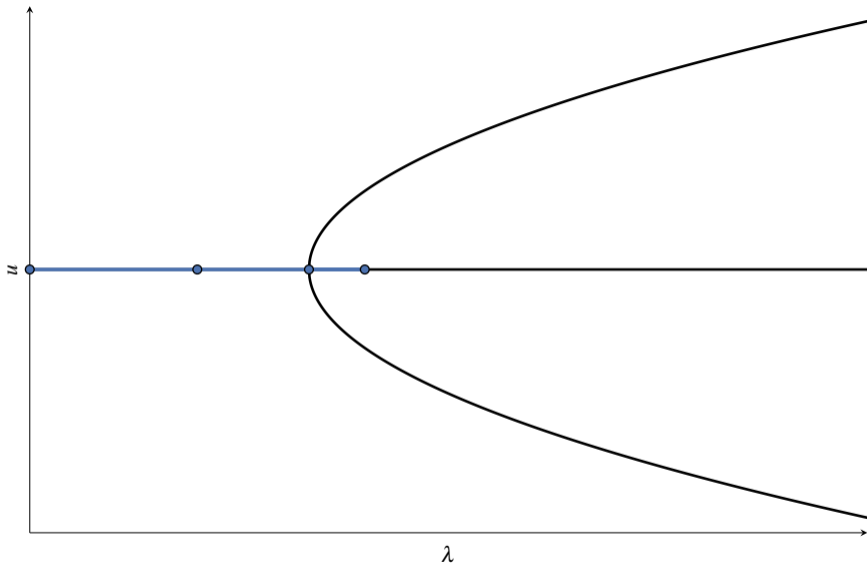
Start with one known solution (u_0, λ_0) .



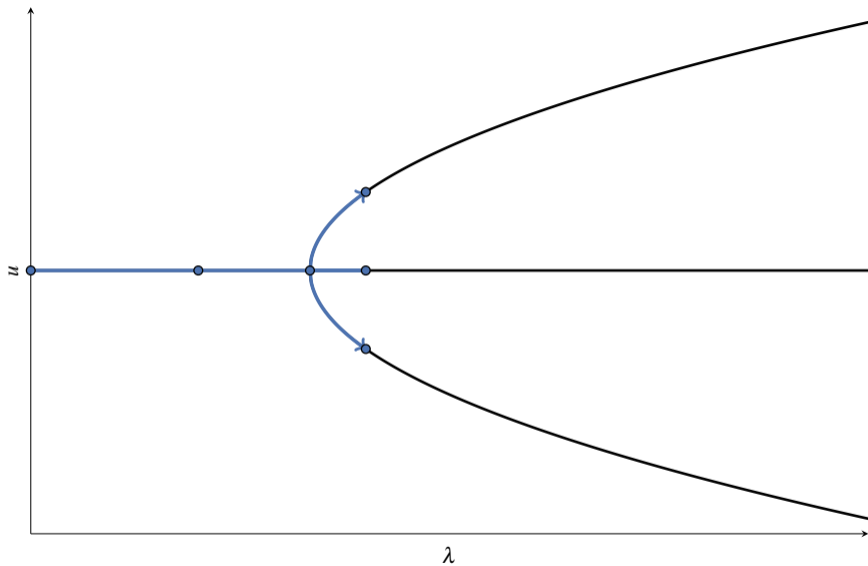
Continue the branch, finding a solution for the next parameter value.



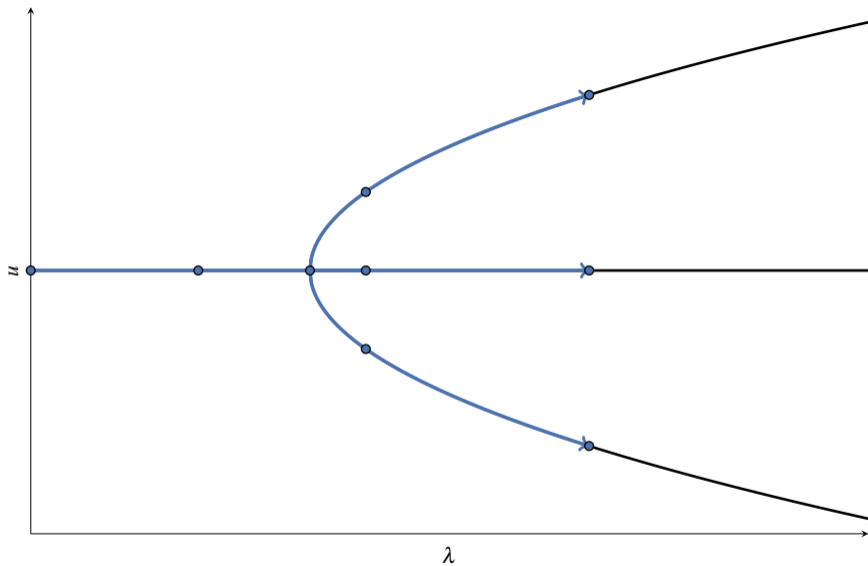
Detect we have passed a bifurcation, where the number of solutions varies.



Localise bifurcation point, by solving a system of equations.



Switch branches.



Apply recursively.

Good news

The combination of continuation and branch switching is very powerful.

Good news

The combination of continuation and branch switching is very powerful.

Bad news

However, it has some disadvantages and weaknesses, too.

Downside A

You have to solve a lot of different problems.

Downside A

You have to solve a lot of different problems.

We work for years to develop a good solver for

$$F(u, \lambda) = 0 \dots$$

Downside A

You have to solve a lot of different problems.

We work for years to develop a good solver for

$$F(u, \lambda) = 0 \dots$$

but now we need to solve related but different problems like

$$\begin{bmatrix} F(u, \lambda) \\ p(u, \lambda, s) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad F_u(u, \lambda)v = \lambda v \quad \begin{bmatrix} F(u, \lambda) \\ F_u(u, \lambda)v \\ \|v\|^2 - 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Downside A

You have to solve a lot of different problems.

We work for years to develop a good solver for

$$F(u, \lambda) = 0 \dots$$

but now we need to solve related but different problems like

$$\begin{bmatrix} F(u, \lambda) \\ p(u, \lambda, s) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad F_u(u, \lambda)v = \lambda v \quad \begin{bmatrix} F(u, \lambda) \\ F_u(u, \lambda)v \\ \|v\|^2 - 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Resolution

This is OK when you can afford direct solvers, but not at scale.

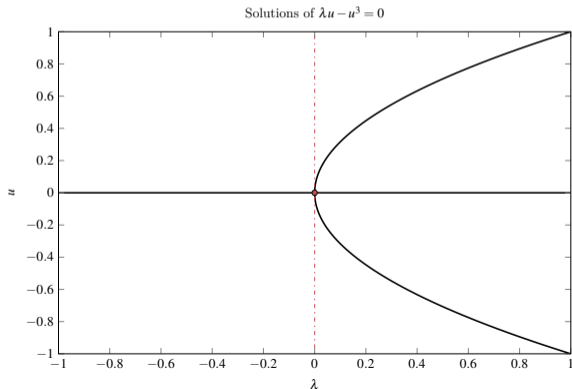
Downside B

We can only find branches *connected* to our initial data.

Downside B

We can only find branches *connected* to our initial data.

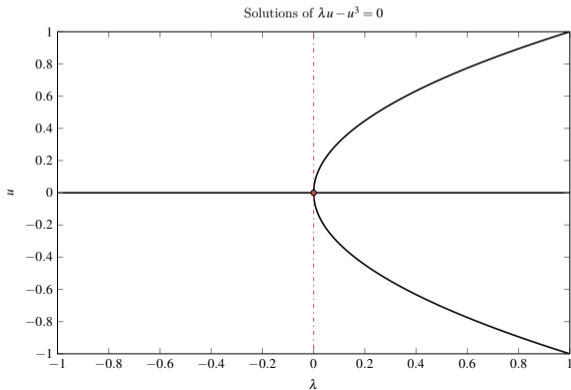
This works fine ...



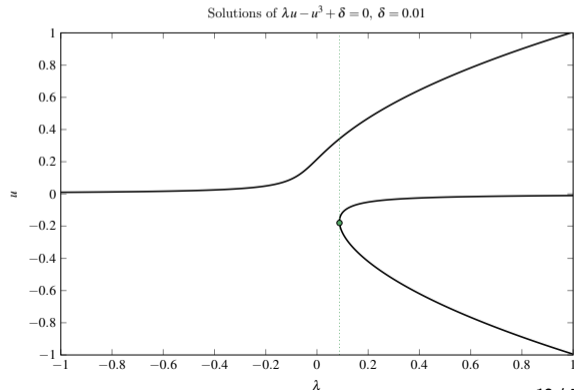
Downside B

We can only find branches *connected* to our initial data.

This works fine ...



...but this does not.



The standard approach to deal with this is to

- (a) modify the problem to restore connectedness;

The standard approach to deal with this is to

- (a) modify the problem to restore connectedness;
- (b) apply continuation + branch switching;

The standard approach to deal with this is to

- (a) modify the problem to restore connectedness;
- (b) apply continuation + branch switching;
- (c) continue the branches you find back to the problem you care about.

The standard approach to deal with this is to

- (a) modify the problem to restore connectedness;
- (b) apply continuation + branch switching;
- (c) continue the branches you find back to the problem you care about.

Problem A

You have to know to look for the missing branches.

The standard approach to deal with this is to

- (a) modify the problem to restore connectedness;
- (b) apply continuation + branch switching;
- (c) continue the branches you find back to the problem you care about.

Problem A

You have to know to look for the missing branches.

Problem B

Executing this is manual and tedious.

The standard approach to deal with this is to

- (a) modify the problem to restore connectedness;
- (b) apply continuation + branch switching;
- (c) continue the branches you find back to the problem you care about.

Problem A

You have to know to look for the missing branches.

Problem B

Executing this is manual and tedious.

Problem C

Restoring connectedness is not always possible!

In this talk I describe a complementary approach.

Disconnected diagrams

An algorithm that can compute **disconnected bifurcation diagrams**.

In this talk I describe a complementary approach.

Disconnected diagrams

An algorithm that can compute **disconnected bifurcation diagrams**.

Simplicity & scaling

It is a straightforward modification of Newton's method.

Section 2

Deflation

Deflation

Fix parameter λ . Given

- ▶ a Fréchet differentiable residual $F : X \rightarrow Y$
- ▶ a solution $u^* \in X$, $F(u^*) = 0$, $F_u(u^*)$ nonsingular

Deflation

Fix parameter λ . Given

- ▶ a Fréchet differentiable residual $F : X \rightarrow Y$
- ▶ a solution $u^* \in X$, $F(u^*) = 0$, $F_u(u^*)$ nonsingular

construct a **new nonlinear problem** $G : X \setminus \{u^*\} \rightarrow Y$ such that:

Deflation

Fix parameter λ . Given

- ▶ a Fréchet differentiable residual $F : X \rightarrow Y$
- ▶ a solution $u^* \in X$, $F(u^*) = 0$, $F_u(u^*)$ nonsingular

construct a **new nonlinear problem** $G : X \setminus \{u^*\} \rightarrow Y$ such that:

- ▶ (Preservation of solutions) $F(\tilde{u}) = 0 \iff G(\tilde{u}) = 0 \forall \tilde{u} \neq u^*$;

Deflation

Fix parameter λ . Given

- ▶ a Fréchet differentiable residual $F : X \rightarrow Y$
- ▶ a solution $u^* \in X$, $F(u^*) = 0$, $F_u(u^*)$ nonsingular

construct a **new nonlinear problem** $G : X \setminus \{u^*\} \rightarrow Y$ such that:

- ▶ (Preservation of solutions) $F(\tilde{u}) = 0 \iff G(\tilde{u}) = 0 \forall \tilde{u} \neq u^*$;
- ▶ (Deflation property) Newton's method applied to G will never converge to u^* again, starting from any initial guess.

Deflation

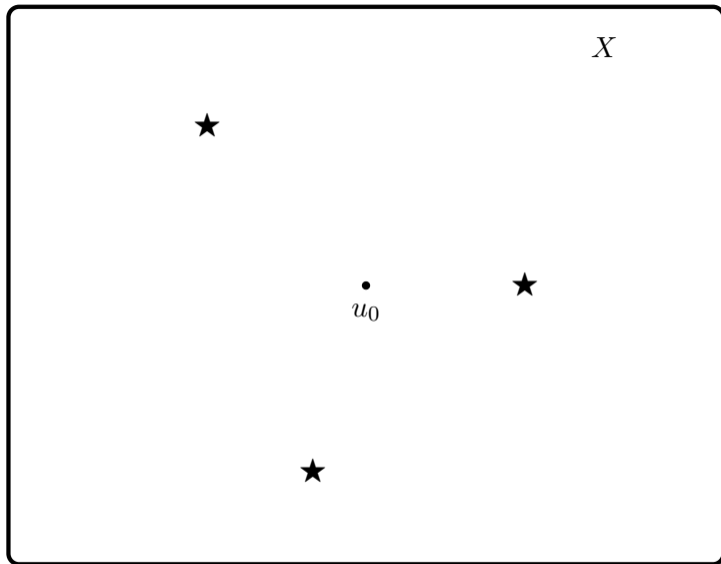
Fix parameter λ . Given

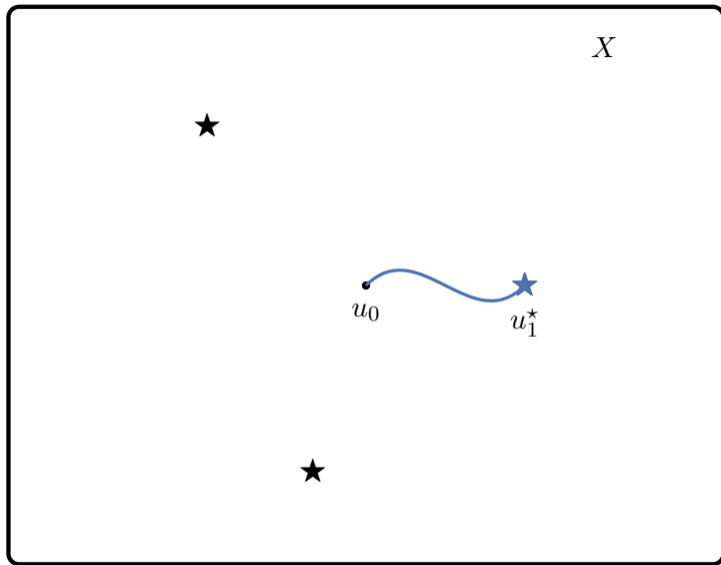
- ▶ a Fréchet differentiable residual $F : X \rightarrow Y$
- ▶ a solution $u^* \in X$, $F(u^*) = 0$, $F_u(u^*)$ nonsingular

construct a **new nonlinear problem** $G : X \setminus \{u^*\} \rightarrow Y$ such that:

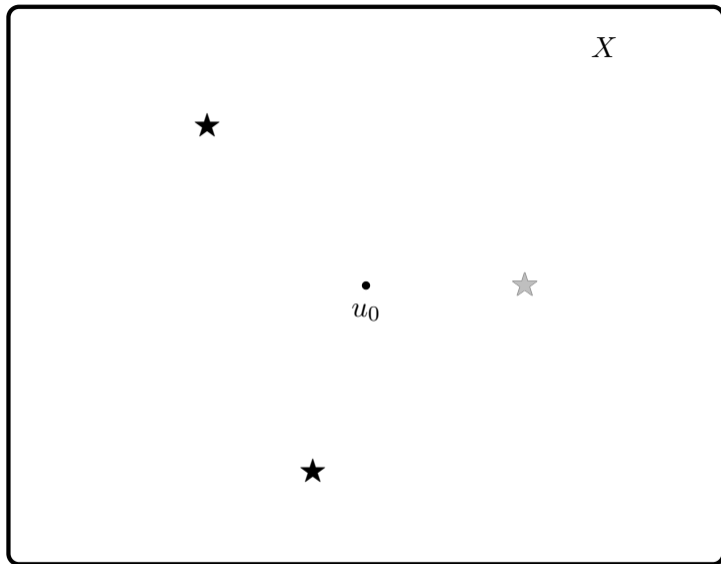
- ▶ (Preservation of solutions) $F(\tilde{u}) = 0 \iff G(\tilde{u}) = 0 \forall \tilde{u} \neq u^*$;
- ▶ (Deflation property) Newton's method applied to G will never converge to u^* again, starting from any initial guess.

Find more solutions, starting from the same initial guess.

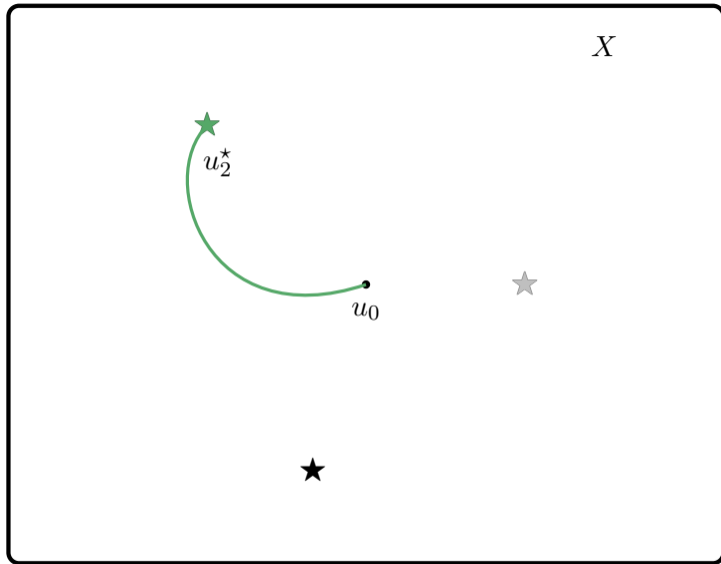




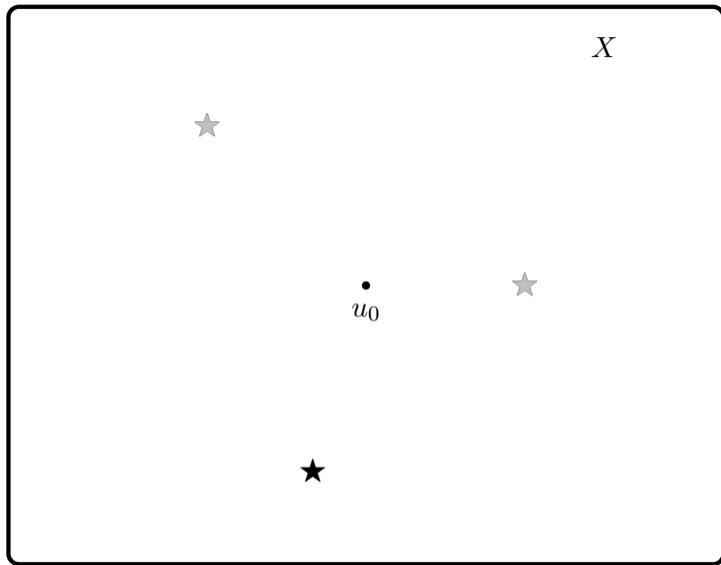
Newton from initial guess.



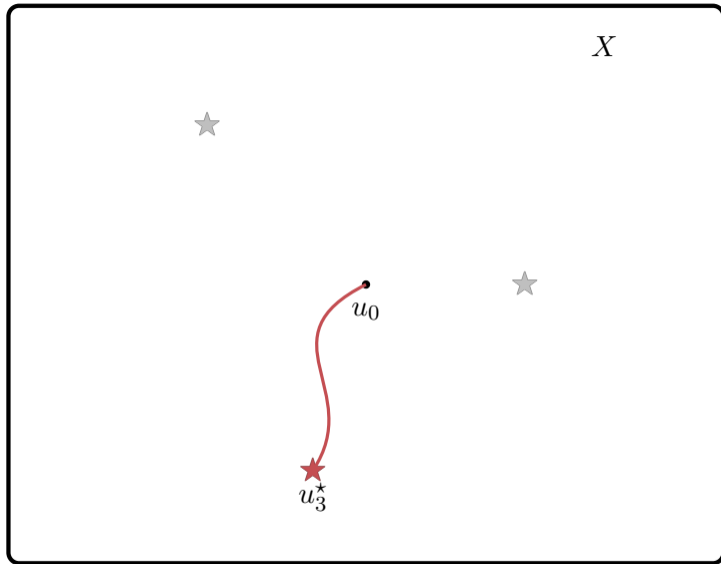
Deflate solution found.



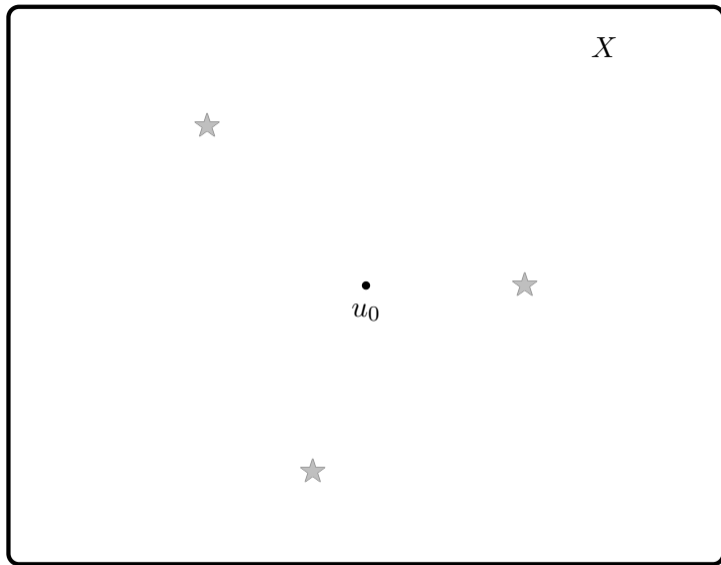
Newton from initial guess.



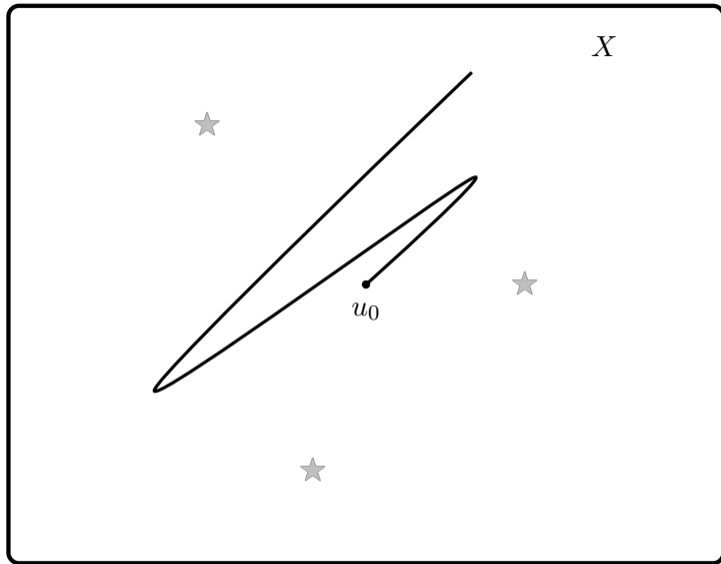
Deflate solution found.



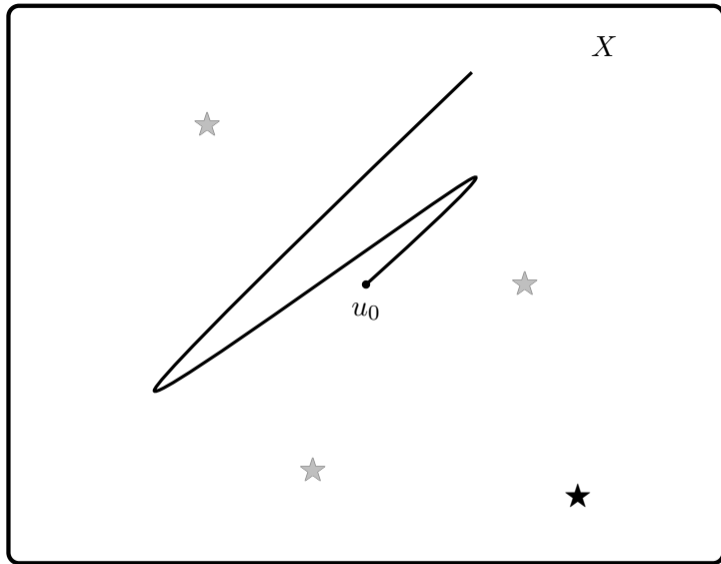
Newton from initial guess.



Deflate solution found.



Terminate on nonconvergence.



Terminate on nonconvergence.

How can we do it?

Definition (Brown & Gearhart, 1971)

We say that $M(u; u^*) \in C^1(X, Y)$ is a *deflation operator* if

$$\liminf_{u \rightarrow u^*} \|G(u)\| := \liminf_{u \rightarrow u^*} \|M(u; u^*)F(u)\| > 0.$$



Kenneth Brown



Bill Gearhart

How can we do it?

Definition (Brown & Gearhart, 1971)

We say that $M(u; u^*) \in C^1(X, Y)$ is a *deflation operator* if

$$\liminf_{u \rightarrow u^*} \|G(u)\| := \liminf_{u \rightarrow u^*} \|M(u; u^*)F(u)\| > 0.$$

Newton's method is looking for sequences (u_k) that send $G(u_k) \rightarrow 0$.
But a deflation operator hides the existence of the root as $u \rightarrow u^*$.



Kenneth Brown



Bill Gearhart

How can we do it?

Definition (Brown & Gearhart, 1971)

We say that $M(u; u^*) \in C^1(X, Y)$ is a *deflation operator* if

$$\liminf_{u \rightarrow u^*} \|G(u)\| := \liminf_{u \rightarrow u^*} \|M(u; u^*)F(u)\| > 0.$$

Newton's method is looking for sequences (u_k) that send $G(u_k) \rightarrow 0$.
But a deflation operator hides the existence of the root as $u \rightarrow u^*$.

However, this has not been entirely successful ...

Numerical experience with deflation has shown it is often a matter of seeming chance whether one obtains an additional solution.

(Allgower & Georg, 1990)



Kenneth Brown



Bill Gearhart

The main deflation operator proposed by Brown & Gearhart was

$$M(u; u^*) = \frac{1}{\|u - u^*\|}.$$

The main deflation operator proposed by Brown & Gearhart was

$$M(u; u^*) = \frac{1}{\|u - u^*\|}.$$

This has the right scaling as

$$\|u - u^*\| \rightarrow 0.$$

The main deflation operator proposed by Brown & Gearhart was

$$M(u; u^*) = \frac{1}{\|u - u^*\|}.$$

This has the right scaling as

$$\|u - u^*\| \rightarrow 0.$$

But it has the wrong scaling as

$$\|u - u^*\| \rightarrow \infty.$$

The main deflation operator proposed by Brown & Gearhart was

$$M(u; u^*) = \frac{1}{\|u - u^*\|}.$$

This has the right scaling as

$$\|u - u^*\| \rightarrow 0.$$

But it has the wrong scaling as

$$\|u - u^*\| \rightarrow \infty.$$

This is why deflation did not work well: if F does not blow up as $\|u - u^*\| \rightarrow \infty$, then Newton discovers it can achieve

$$\|G(u)\|_Y < \text{tol}$$

for any tol, by taking $\|u - u^*\|$ large enough.

The deflation operator we propose

$$M(u; u^*) := \left(\frac{1}{\|u - u^*\|^p} + 1 \right), \quad p \geq 1.$$



Ásgeir Birkisson



Simon Funke

The deflation operator we propose

$$M(u; u^*) := \left(\frac{1}{\|u - u^*\|^p} + 1 \right), \quad p \geq 1.$$

This has the right behaviour *both* as

$$\begin{aligned} \|u - u^*\| &\rightarrow 0, \\ \|u - u^*\| &\rightarrow \infty, \end{aligned}$$

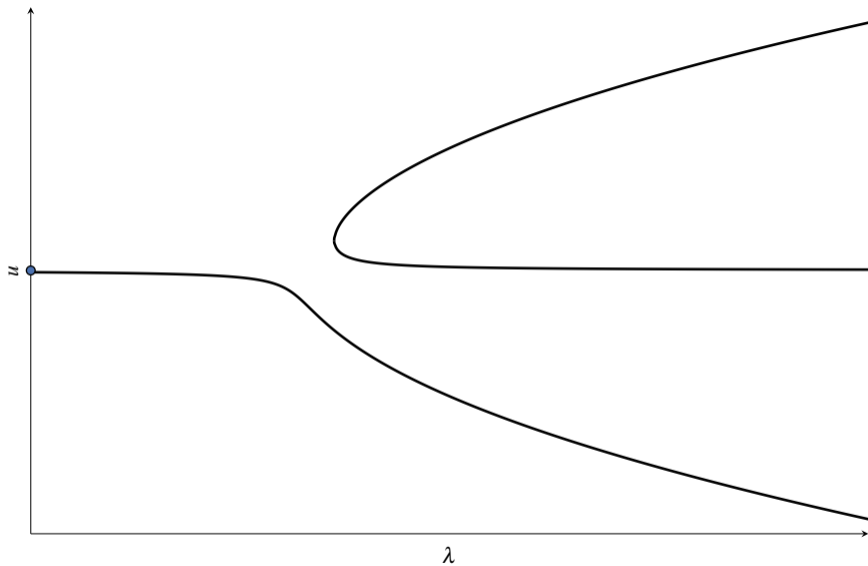
which makes deflation much more reliable.



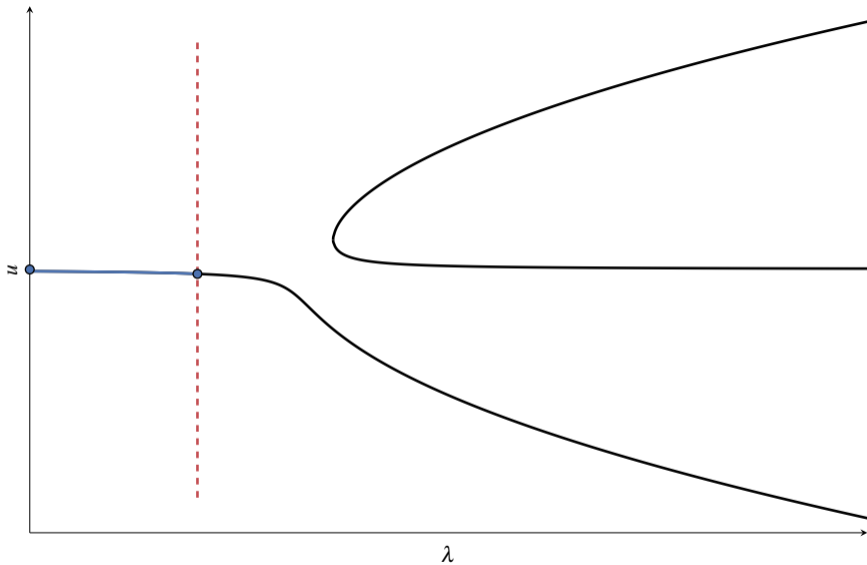
Ásgeir Birgisson



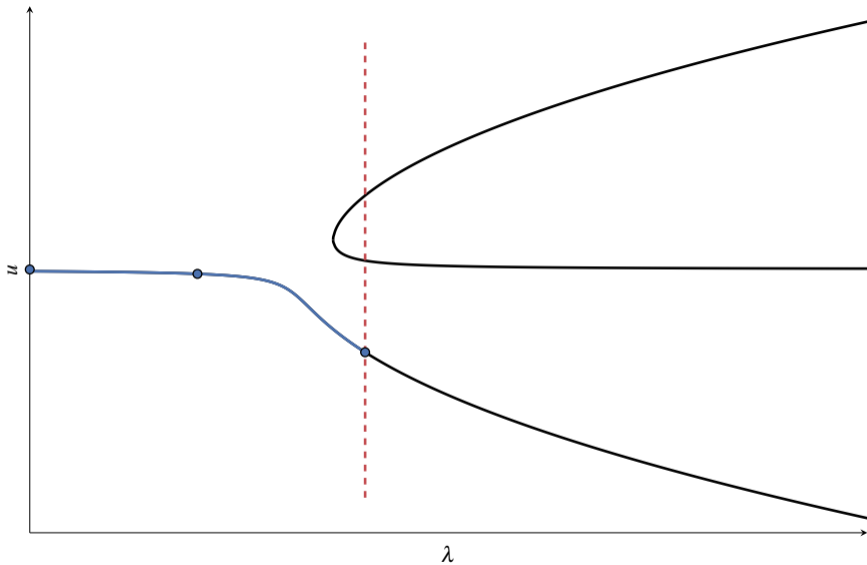
Simon Funke



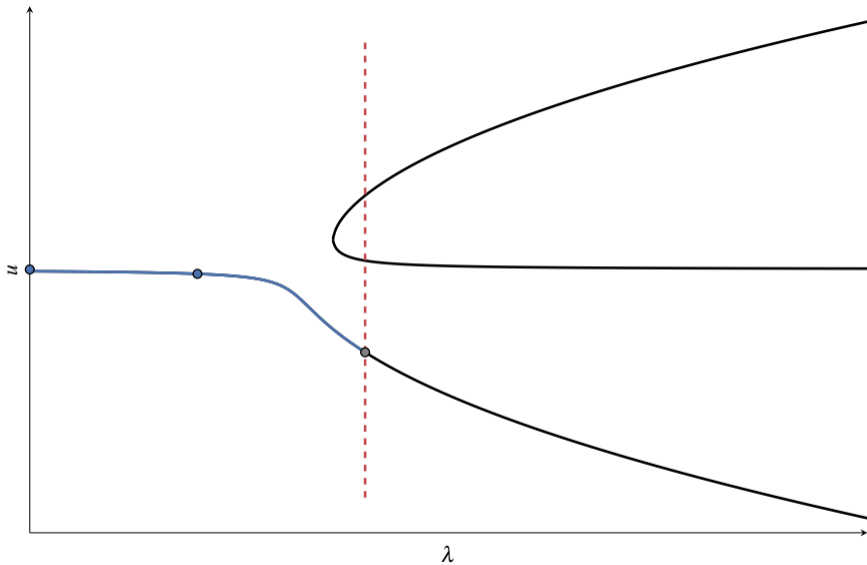
Start with (u_0, λ_0) .



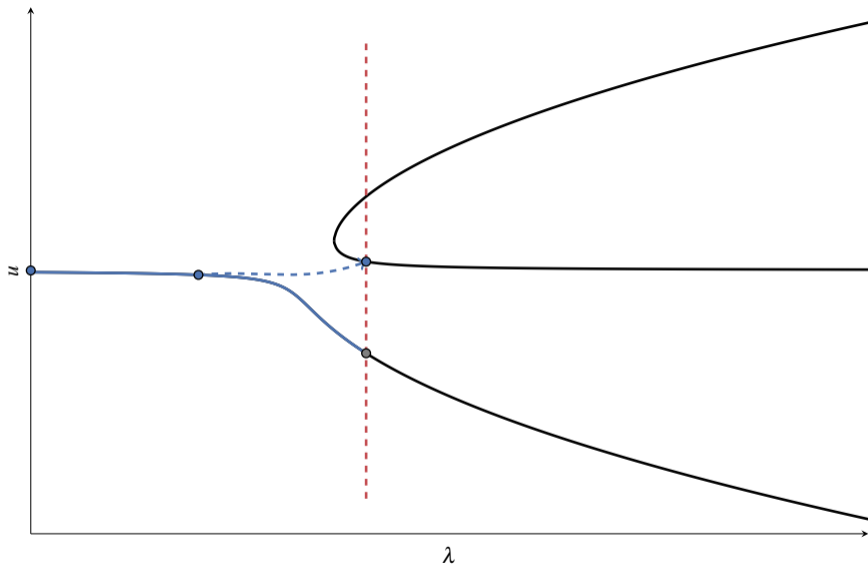
Perform a continuation step.



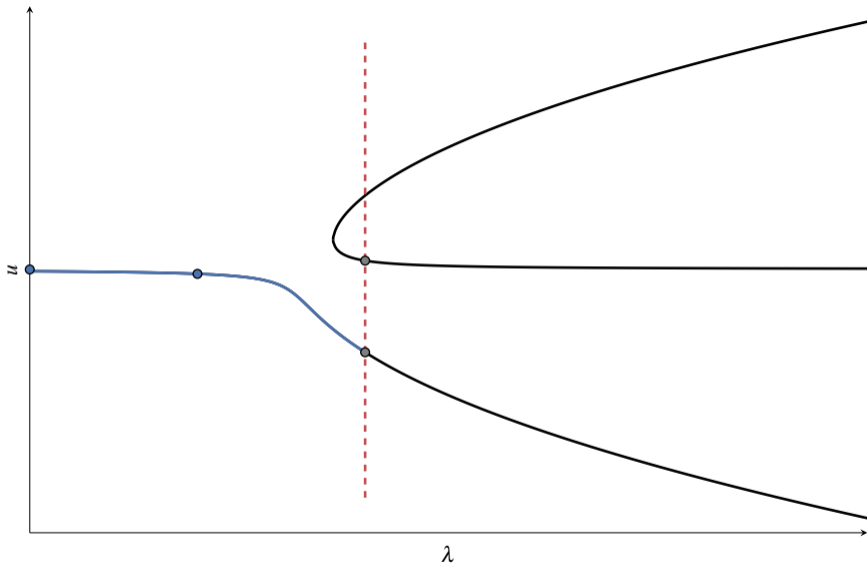
Perform another continuation step.



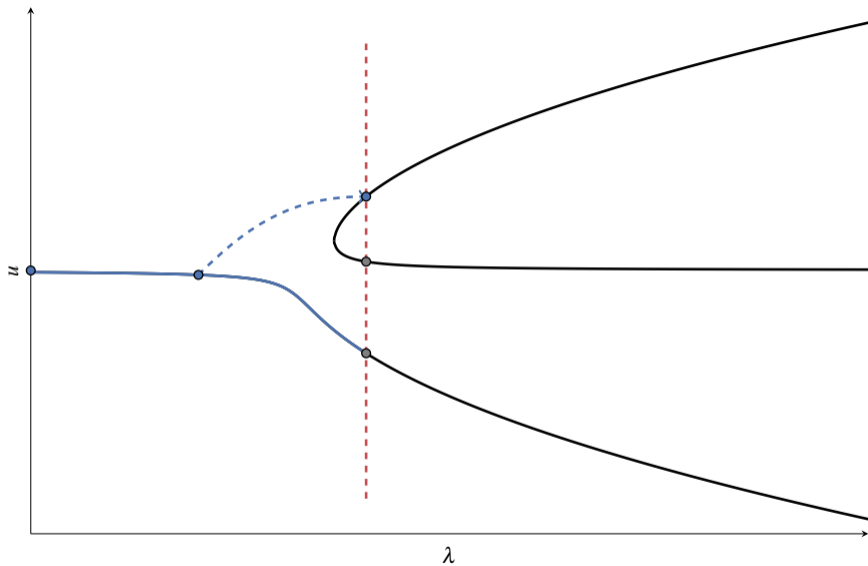
Deflate the solution found.



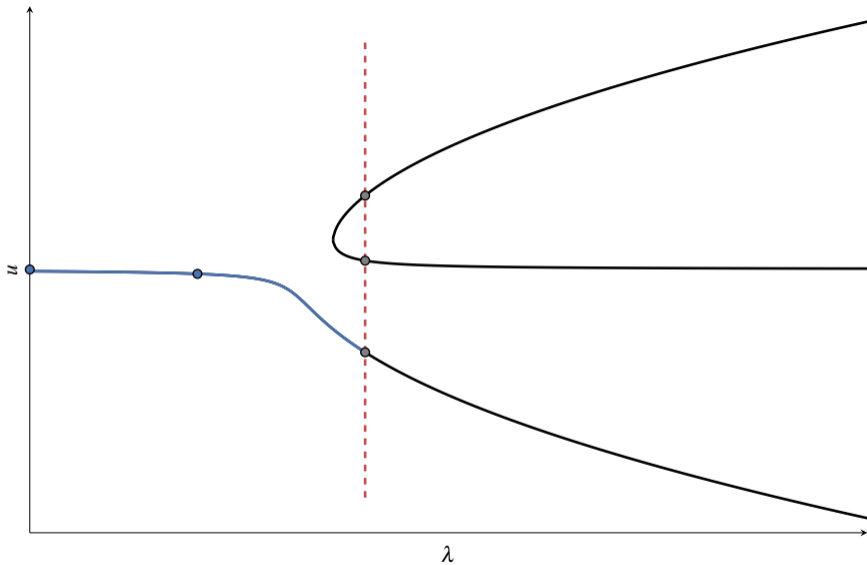
Solve again.



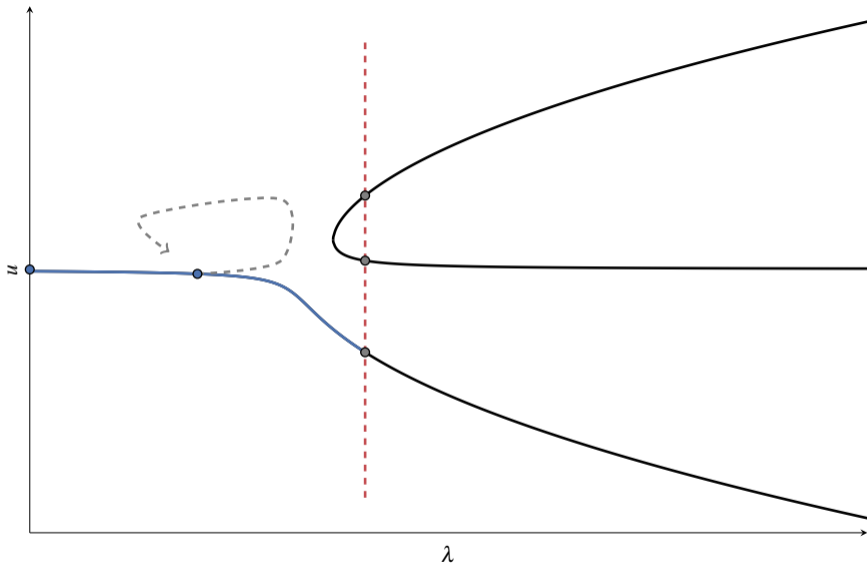
Deflate the solution found.



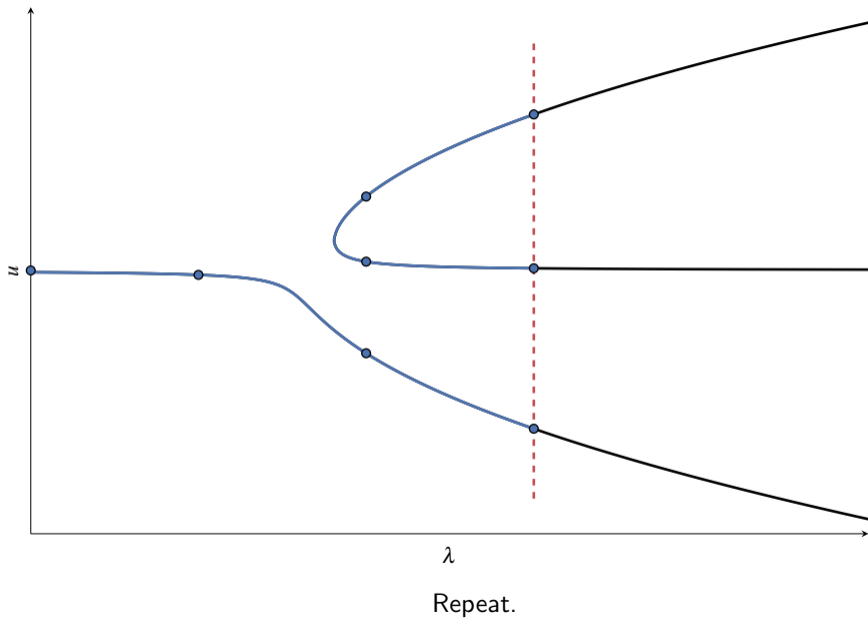
Solve again.



Deflate the solution found.



Search again, unsuccessfully.



Good news

Deflation lets us discover disconnected branches!

Section 3

Solving the deflated problem

In PDEs we aim to have a good linear solver for the discretised Newton step

$$F_u(u, \lambda)\delta u_F = -F(u, \lambda), \quad F \in C^1(\mathbb{R}^N \times \mathbb{R}, \mathbb{R}^N).$$

In PDEs we aim to have a good linear solver for the discretised Newton step

$$F_u(u, \lambda)\delta u_F = -F(u, \lambda), \quad F \in C^1(\mathbb{R}^N \times \mathbb{R}, \mathbb{R}^N).$$

We now want to solve

$$G_u(u, \lambda)\delta u_G = -G(u, \lambda)$$

where

$$G(u, \lambda) = M(u; u_1)M(u; u_2) \cdots M(u; u_n)F(u, \lambda) =: M(u)F(u, \lambda).$$

In PDEs we aim to have a good linear solver for the discretised Newton step

$$F_u(u, \lambda)\delta u_F = -F(u, \lambda), \quad F \in C^1(\mathbb{R}^N \times \mathbb{R}, \mathbb{R}^N).$$

We now want to solve

$$G_u(u, \lambda)\delta u_G = -G(u, \lambda)$$

where

$$G(u, \lambda) = M(u; u_1)M(u; u_2) \cdots M(u; u_n)F(u, \lambda) =: M(u)F(u, \lambda).$$

Good news

You can compute δu_G easily from δu_F with Sherman–Morrison!

Solving the deflated problem

To solve

$$G_u \delta u_G = -G,$$

do the following:

Solving the deflated problem

To solve

$$G_u \delta u_G = -G,$$

do the following:

1. Solve

$$F_u \delta u_F = -F.$$

Solving the deflated problem

To solve

$$G_u \delta u_G = -G,$$

do the following:

1. Solve

$$F_u \delta u_F = -F.$$

2. Evaluate

$$p = M_u^\top \delta u_F.$$

Solving the deflated problem

To solve

$$G_u \delta u_G = -G,$$

do the following:

1. Solve

$$F_u \delta u_F = -F.$$

2. Evaluate

$$p = M_u^\top \delta u_F.$$

3. Evaluate

$$\tau = 1 + \frac{M^{-1}p}{1 - M^{-1}p}.$$

Solving the deflated problem

To solve

$$G_u \delta u_G = -G,$$

do the following:

1. Solve

$$F_u \delta u_F = -F.$$

2. Evaluate

$$p = M_u^\top \delta u_F.$$

3. Evaluate

$$\tau = 1 + \frac{M^{-1}p}{1 - M^{-1}p}.$$

4. Return

$$\delta u_G = \tau \delta u_F.$$

Section 4

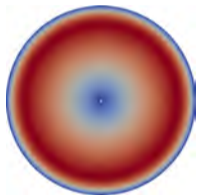
Examples

Yamabe equation

$$F(u, \lambda) = -8\lambda^2 \nabla^2 u + r^{-3} u^5 - u/10 = 0, \quad u = 1 \text{ on } \partial\Omega.$$

Yamabe equation

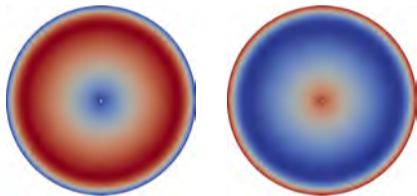
$$F(u, \lambda) = -8\lambda^2 \nabla^2 u + r^{-3} u^5 - u/10 = 0, \quad u = 1 \text{ on } \partial\Omega.$$



Solutions found starting from $u = \pm 1$.

Yamabe equation

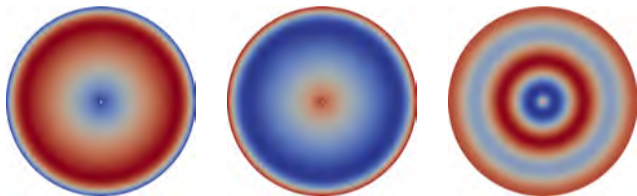
$$F(u, \lambda) = -8\lambda^2 \nabla^2 u + r^{-3} u^5 - u/10 = 0, \quad u = 1 \text{ on } \partial\Omega.$$



Solutions found starting from $u = \pm 1$.

Yamabe equation

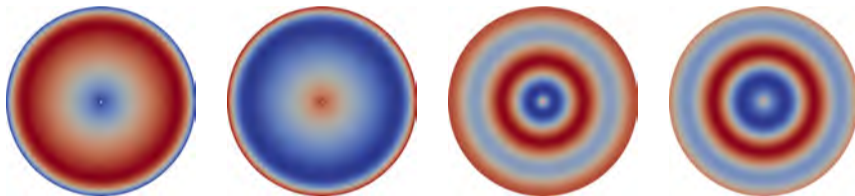
$$F(u, \lambda) = -8\lambda^2 \nabla^2 u + r^{-3} u^5 - u/10 = 0, \quad u = 1 \text{ on } \partial\Omega.$$



Solutions found starting from $u = \pm 1$.

Yamabe equation

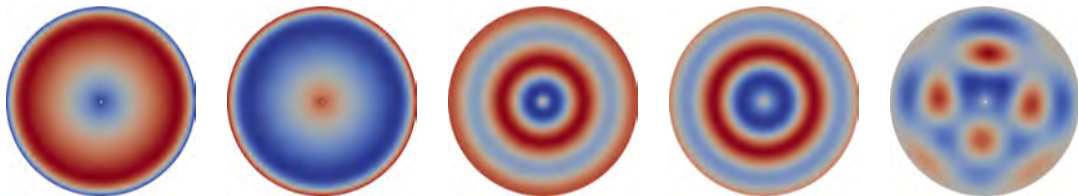
$$F(u, \lambda) = -8\lambda^2 \nabla^2 u + r^{-3} u^5 - u/10 = 0, \quad u = 1 \text{ on } \partial\Omega.$$



Solutions found starting from $u = \pm 1$.

Yamabe equation

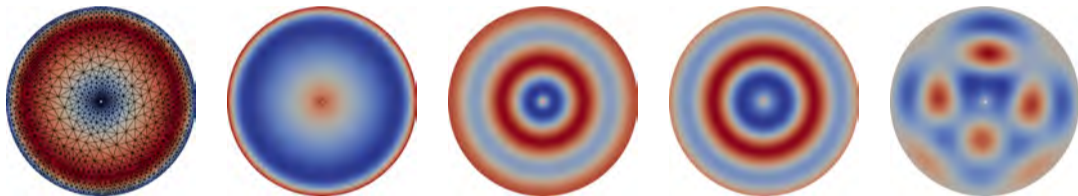
$$F(u, \lambda) = -8\lambda^2 \nabla^2 u + r^{-3} u^5 - u/10 = 0, \quad u = 1 \text{ on } \partial\Omega.$$



Solutions found starting from $u = \pm 1$.

Yamabe equation

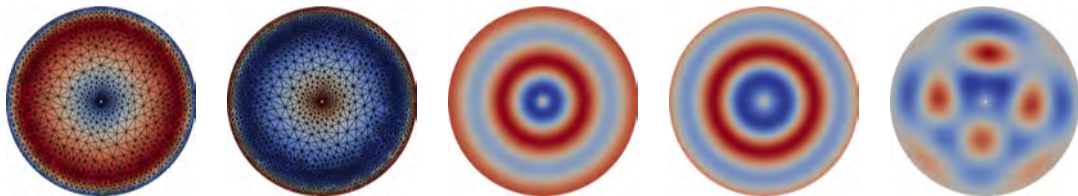
$$F(u, \lambda) = -8\lambda^2 \nabla^2 u + r^{-3} u^5 - u/10 = 0, \quad u = 1 \text{ on } \partial\Omega.$$



Solutions found starting from $u = \pm 1$.

Yamabe equation

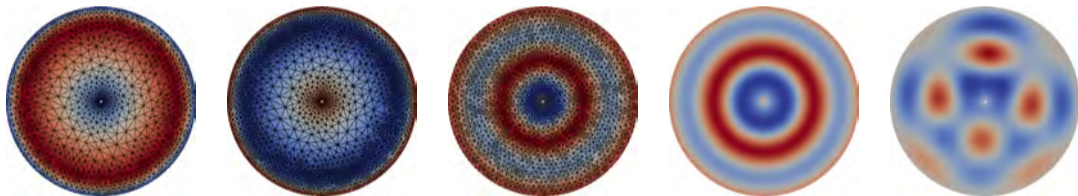
$$F(u, \lambda) = -8\lambda^2 \nabla^2 u + r^{-3} u^5 - u/10 = 0, \quad u = 1 \text{ on } \partial\Omega.$$



Solutions found starting from $u = \pm 1$.

Yamabe equation

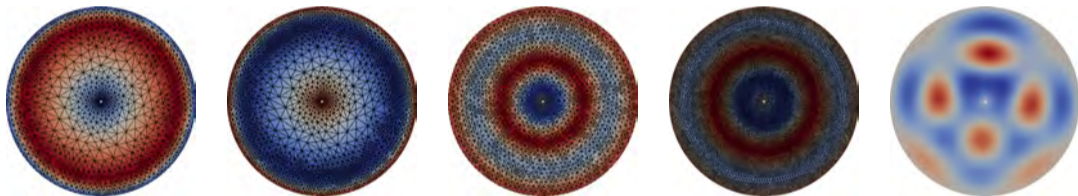
$$F(u, \lambda) = -8\lambda^2 \nabla^2 u + r^{-3} u^5 - u/10 = 0, \quad u = 1 \text{ on } \partial\Omega.$$



Solutions found starting from $u = \pm 1$.

Yamabe equation

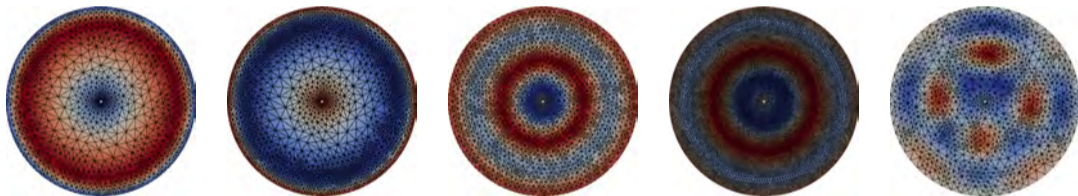
$$F(u, \lambda) = -8\lambda^2 \nabla^2 u + r^{-3} u^5 - u/10 = 0, \quad u = 1 \text{ on } \partial\Omega.$$



Solutions found starting from $u = \pm 1$.

Yamabe equation

$$F(u, \lambda) = -8\lambda^2 \nabla^2 u + r^{-3} u^5 - u/10 = 0, \quad u = 1 \text{ on } \partial\Omega.$$



Solutions found starting from $u = \pm 1$.

Aside: in Firedrake there is now an *automated* implementation of DWR.

A standard Firedrake code might look like

```
from firedrake import *
...
problem = NonlinearVariationalProblem(F, solution, bcs)
solver = NonlinearVariationalSolver(problem, solver_parameters)
```

and we can now we can do

```
from firedrake import *
from netgen.occ import * # use Netgen to build the mesh
...
problem = NonlinearVariationalProblem(F, solution, bcs)
solver = GoalAdaptiveNonlinearVariationalSolver(problem, J, tolerance,
                                                solver_parameters)
```

Firedrake could previously automatically derive the adjoint equation.



Marie Rognes



Anders Logg

Firedrake could previously automatically derive the adjoint equation.

We need to automatically compute a localised residual on cells/facets.



Marie Rognes



Anders Logg

Firedrake could previously automatically derive the adjoint equation.

We need to automatically compute a localised residual on cells/facets.

Set b_K to be a bubble function on every cell. We solve e.g.: find $R_K \in DG_p$ such that

$$(R_K, b_K v) = \rho(u_h; b_K v) \text{ for all } v \in DG_p.$$



Marie Rognes



Anders Logg

Firedrake could previously automatically derive the adjoint equation.

We need to automatically compute a localised residual on cells/facets.

Set b_K to be a bubble function on every cell. We solve e.g.: find $R_K \in DG_p$ such that

$$(R_K, b_K v) = \rho(u_h; b_K v) \text{ for all } v \in DG_p.$$

We solve analogous problems for the primal facet residuals, and for the dual cell and facet residuals.



Marie Rognes



Anders Logg

Firedrake could previously automatically derive the adjoint equation.

We need to automatically compute a localised residual on cells/facets.

Set b_K to be a bubble function on every cell. We solve e.g.: find $R_K \in DG_p$ such that

$$(R_K, b_K v) = \rho(u_h; b_K v) \text{ for all } v \in DG_p.$$

We solve analogous problems for the primal facet residuals, and for the dual cell and facet residuals.

These give polynomial approximations for DWR localisation.



Marie Rognes



Anders Logg

Firedrake could previously automatically derive the adjoint equation.

We need to automatically compute a localised residual on cells/facets.

Set b_K to be a bubble function on every cell. We solve e.g.: find $R_K \in DG_p$ such that

$$(R_K, b_K v) = \rho(u_h; b_K v) \text{ for all } v \in DG_p.$$

We solve analogous problems for the primal facet residuals, and for the dual cell and facet residuals.

These give polynomial approximations for DWR localisation.

M. E. Rognes and A. Logg. “Automated goal-oriented error control I: stationary variational problems”. In: *SIAM Journal on Scientific Computing* 35.3 (2013), pp. C173–C193



Marie Rognes



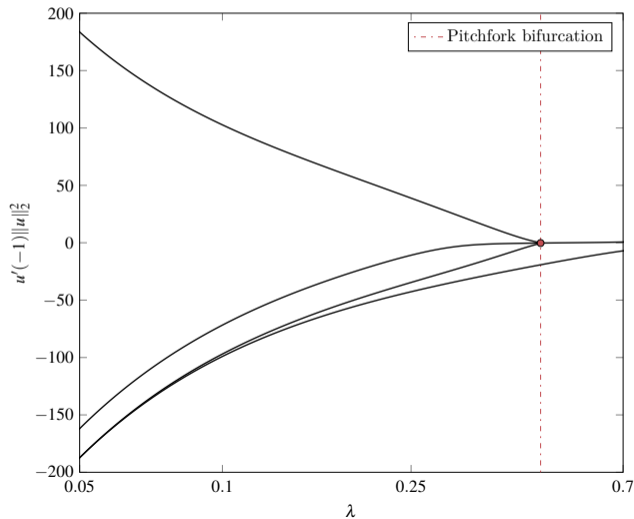
Anders Logg

Carrier's equation

$$F(u, \lambda) = \lambda^2 u'' + 2(1 - x^2)u + u^2 - 1 = 0, \quad u(-1) = 0 = u(1).$$

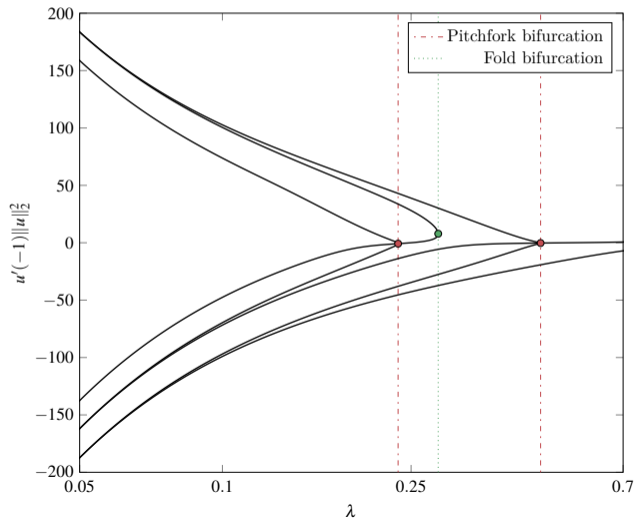
Carrier's equation

$$F(u, \lambda) = \lambda^2 u'' + 2(1 - x^2)u + u^2 - 1 = 0, \quad u(-1) = 0 = u(1).$$



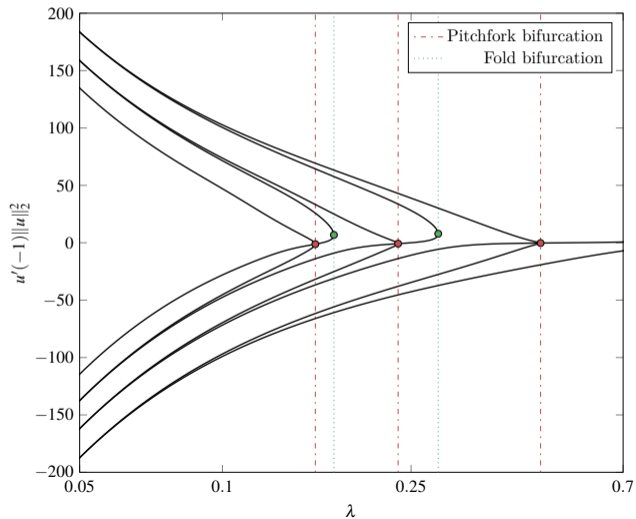
Carrier's equation

$$F(u, \lambda) = \lambda^2 u'' + 2(1 - x^2)u + u^2 - 1 = 0, \quad u(-1) = 0 = u(1).$$



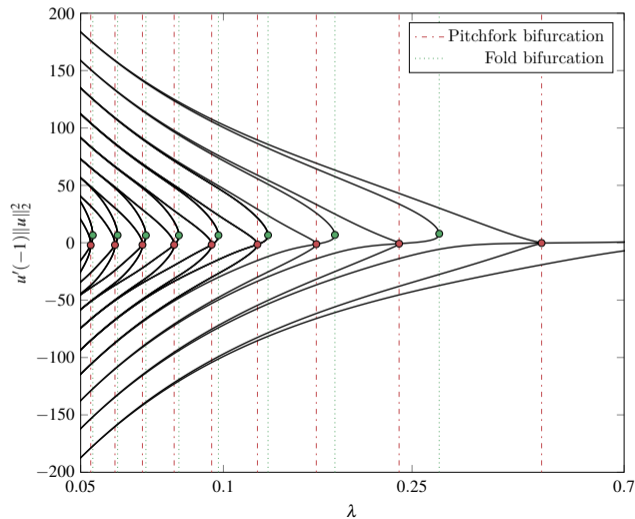
Carrier's equation

$$F(u, \lambda) = \lambda^2 u'' + 2(1 - x^2)u + u^2 - 1 = 0, \quad u(-1) = 0 = u(1).$$



Carrier's equation

$$F(u, \lambda) = \lambda^2 u'' + 2(1 - x^2)u + u^2 - 1 = 0, \quad u(-1) = 0 = u(1).$$

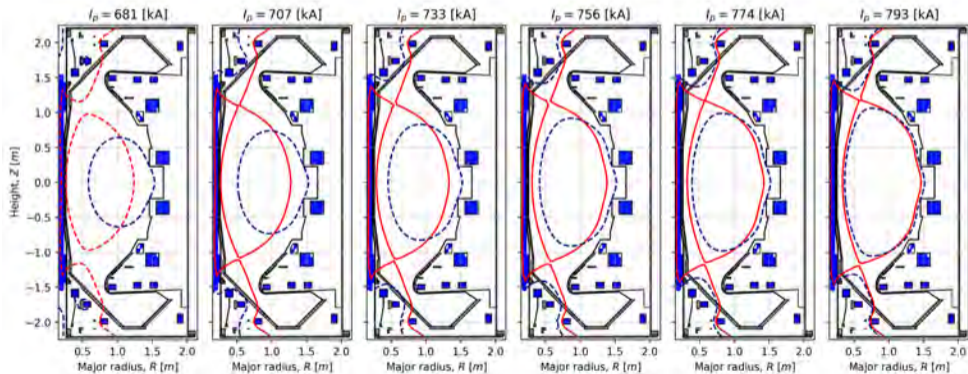


Grad–Shafranov equation for tokamak fusion reactors

$$(R\partial_R R^{-1}\partial_R + \partial_{ZZ})\psi = -\mu_0 R J_\phi(\psi, R, Z).$$

Grad-Shafranov equation for tokamak fusion reactors

$$(R\partial_R R^{-1}\partial_R + \partial_{ZZ})\psi = -\mu_0 R J_\phi(\psi, R, Z).$$



Two different solutions as a function of plasma current for a physical configuration of MAST-U.

Section 5

Symmetries

Symmetries

What if the equation has a continuous symmetry group?

Symmetries

What if the equation has a continuous symmetry group?

Philosophy

The fundamental structures are the distinct **orbits** of solutions.

Symmetries

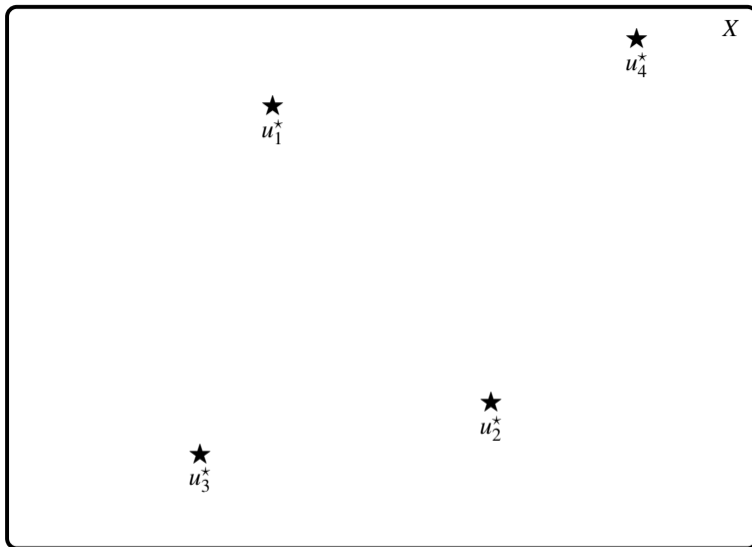
What if the equation has a continuous symmetry group?

Philosophy

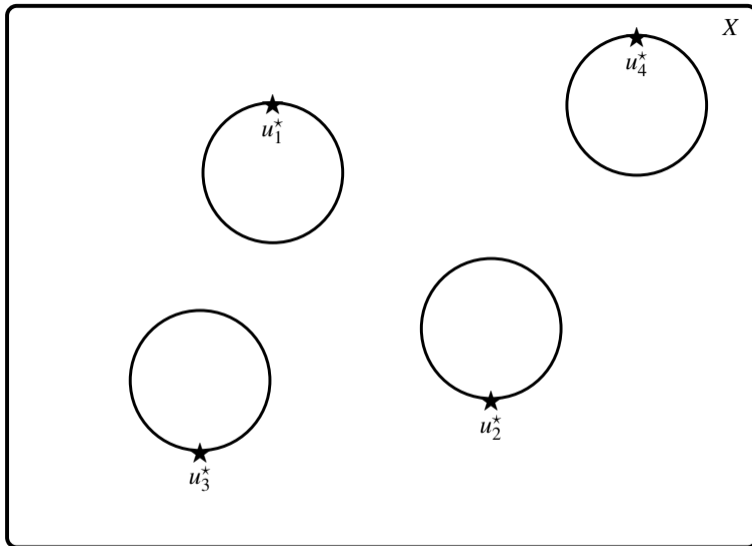
The fundamental structures are the distinct **orbits** of solutions.

Key idea

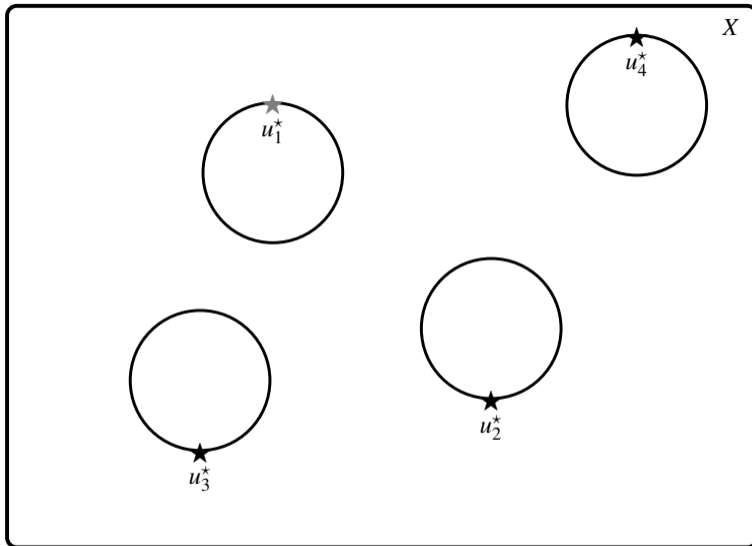
Construct a deflation operator invariant under the action of the Lie group.



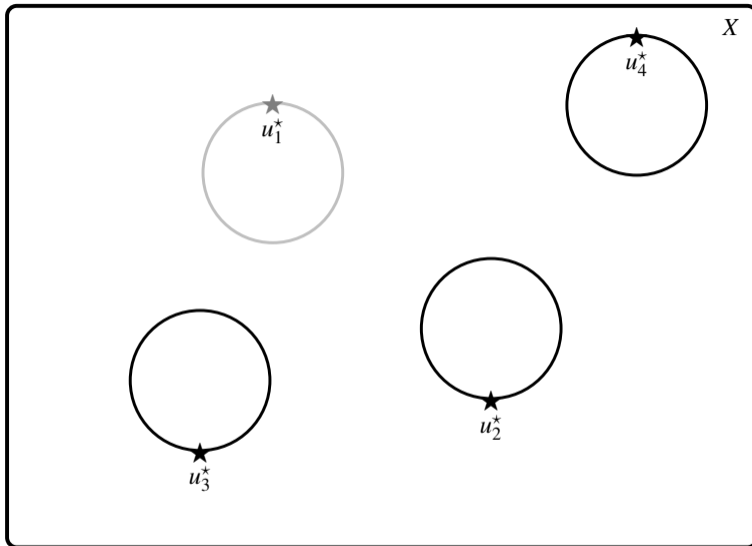
Four solutions, not related by the symmetry group.



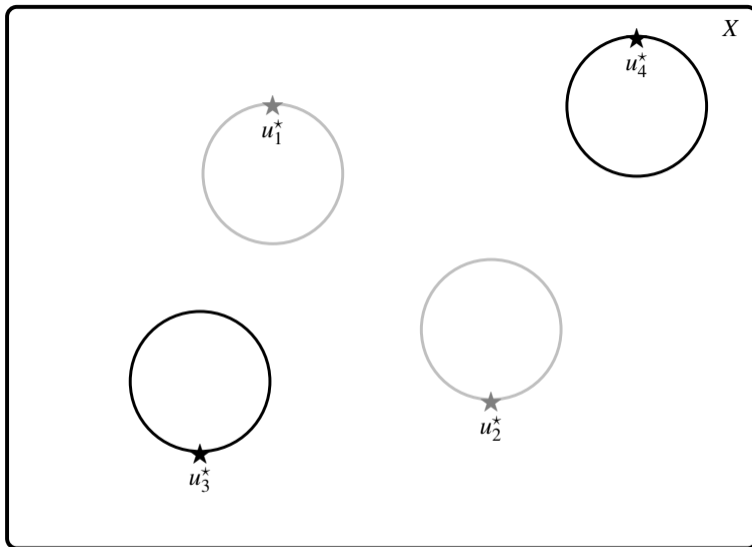
Each solution induces a *group orbit* of solutions, related by symmetry.



Not enough to deflate the solution—must deflate the entire orbit.



Design a deflation operator that deflates the entire orbit.



Design a deflation operator that deflates the entire orbit.

Gross–Pitaevskii equation

$$-\frac{1}{2}\Delta u + \frac{x^2 + y^2 + z^2}{2}u - \mu u + |u|^2u = 0, \quad u|_{\partial\Omega} = 0.$$

Gross–Pitaevskii equation

$$-\frac{1}{2}\Delta u + \frac{x^2 + y^2 + z^2}{2}u - \mu u + |u|^2u = 0, \quad u|_{\partial\Omega} = 0.$$

First symmetry group $\text{SO}(2)$: phase shifts

$$u(\vec{x}) \mapsto e^{i\theta}u(\vec{x}), \quad \theta \in \mathbb{R}.$$

Gross–Pitaevskii equation

$$-\frac{1}{2}\Delta u + \frac{x^2 + y^2 + z^2}{2}u - \mu u + |u|^2u = 0, \quad u|_{\partial\Omega} = 0.$$

First symmetry group $SO(2)$: phase shifts

$$u(\vec{x}) \mapsto e^{i\theta}u(\vec{x}), \quad \theta \in \mathbb{R}.$$

Invariant deflation operator

$$M(u; r) = \left\| |u|^2 - |r|^2 \right\|^{-2} + 1.$$

Gross–Pitaevskii equation

$$-\frac{1}{2}\Delta u + \frac{x^2 + y^2 + z^2}{2}u - \mu u + |u|^2u = 0, \quad u|_{\partial\Omega} = 0.$$

Second symmetry group $\text{SO}(3)$: spatial rotations

$$u(\vec{x}) \mapsto u(R\vec{x}), \quad R^{-1} = R^T, \quad \det(R) = 1.$$

Gross–Pitaevskii equation

$$-\frac{1}{2}\Delta u + \frac{x^2 + y^2 + z^2}{2}u - \mu u + |u|^2u = 0, \quad u|_{\partial\Omega} = 0.$$

Second symmetry group $SO(3)$: spatial rotations

$$u(\vec{x}) \mapsto u(R\vec{x}), \quad R^{-1} = R^T, \quad \det(R) = 1.$$

Invariant deflation operator

$$M(u; r) = \|\bar{u} - \bar{r}\|^{-2} + 1,$$

where

$\bar{u}(r, \theta, \psi)$ averages u over the sphere of radius r .

Gross–Pitaevskii equation

$$-\frac{1}{2}\Delta u + \frac{x^2 + y^2 + z^2}{2}u - \mu u + |u|^2u = 0, \quad u|_{\partial\Omega} = 0.$$

Solutions for $\mu = 6$.

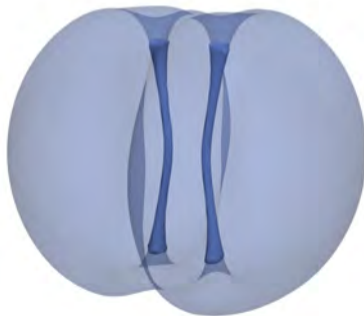


A vortex line and a planar dark soliton.

Gross–Pitaevskii equation

$$-\frac{1}{2}\Delta u + \frac{x^2 + y^2 + z^2}{2}u - \mu u + |u|^2u = 0, \quad u|_{\partial\Omega} = 0.$$

Solutions for $\mu = 6$.

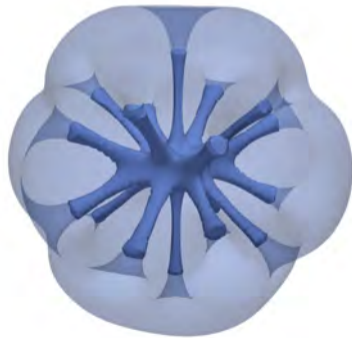


A pair of vortex lines.

Gross–Pitaevskii equation

$$-\frac{1}{2}\Delta u + \frac{x^2 + y^2 + z^2}{2}u - \mu u + |u|^2u = 0, \quad u|_{\partial\Omega} = 0.$$

Solutions for $\mu = 6$.

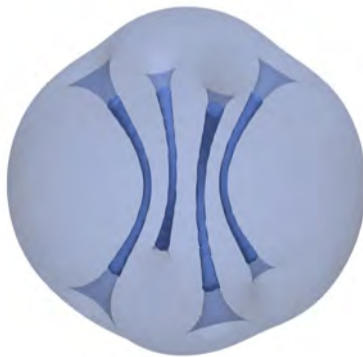


A vortex star.

Gross–Pitaevskii equation

$$-\frac{1}{2}\Delta u + \frac{x^2 + y^2 + z^2}{2}u - \mu u + |u|^2u = 0, \quad u|_{\partial\Omega} = 0.$$

Solutions for $\mu = 6$.

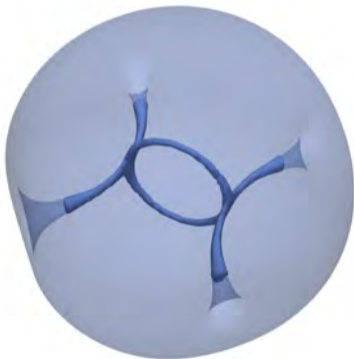


Four vortex lines of alternating charge.

Gross–Pitaevskii equation

$$-\frac{1}{2}\Delta u + \frac{x^2 + y^2 + z^2}{2}u - \mu u + |u|^2u = 0, \quad u|_{\partial\Omega} = 0.$$

Solutions for $\mu = 6$.

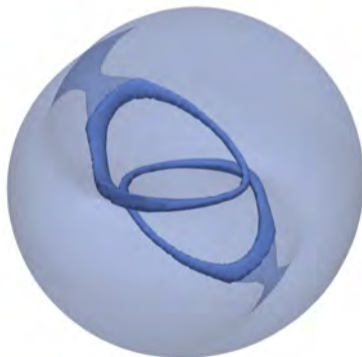


A vortex ring with two “handles”.

Gross–Pitaevskii equation

$$-\frac{1}{2}\Delta u + \frac{x^2 + y^2 + z^2}{2}u - \mu u + |u|^2u = 0, \quad u|_{\partial\Omega} = 0.$$

Solutions for $\mu = 6$.



Two bent vortex rings?

Gross–Pitaevskii equation

$$-\frac{1}{2}\Delta u + \frac{x^2 + y^2 + z^2}{2}u - \mu u + |u|^2u = 0, \quad u|_{\partial\Omega} = 0.$$

Solutions for $\mu = 6$.

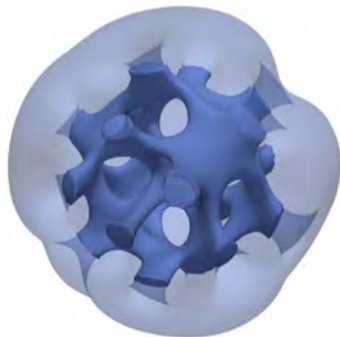


Two vortex rings and five lines?

Gross–Pitaevskii equation

$$-\frac{1}{2}\Delta u + \frac{x^2 + y^2 + z^2}{2}u - \mu u + |u|^2u = 0, \quad u|_{\partial\Omega} = 0.$$

Solutions for $\mu = 6$.



A vortex ring cage?

Section 6

Problems with inequality constraints

Many problems feature *inequality constraints*.



Flowing glaciers cannot penetrate the bedrock.

Many problems feature *inequality constraints*.



Continua in contact cannot penetrate each other.

Many problems feature *inequality constraints*.



Traffic equilibrium: no driver can improve travel time by unilaterally changing route.

Many problems feature *inequality constraints*.



Market equilibrium: no participant can unilaterally improve profit.

Many problems feature *inequality constraints*.



Holder of American options exercise early only if profitable.

Good news

Deflation works for these problems too.

Good news

Deflation works for these problems too.

Theorem (F., Croci, Surowiec, 2020)

Under the same assumptions that are required for superlinear convergence of semismooth Newton, deflation works the same.



Matteo Croci



Gould gives an optimisation problem where interior point methods struggle:

Nonconvex quadratic programming problem

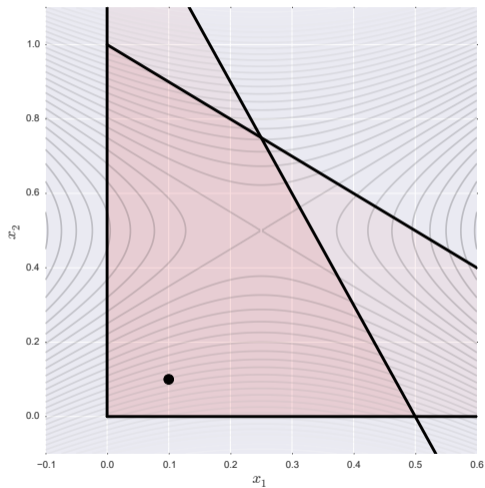
$$\underset{x \in \mathbb{R}^2}{\text{minimise}} \quad -2(x_1 - 0.25)^2 + 2(x_2 - 0.5)^2$$

$$\begin{aligned} \text{subject to} \quad & x_1 + x_2 \leq 1 \\ & 3x_1 + x_2 \leq 1.5 \\ & x_1 \geq 0 \\ & x_2 \geq 0 \end{aligned}$$



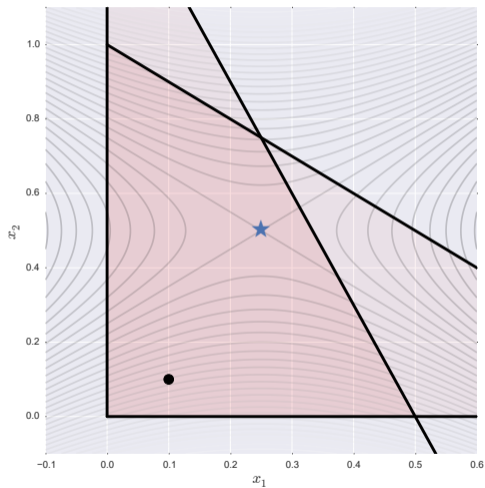
Nick Gould

Gould gives an optimization problem where interior point methods struggle:



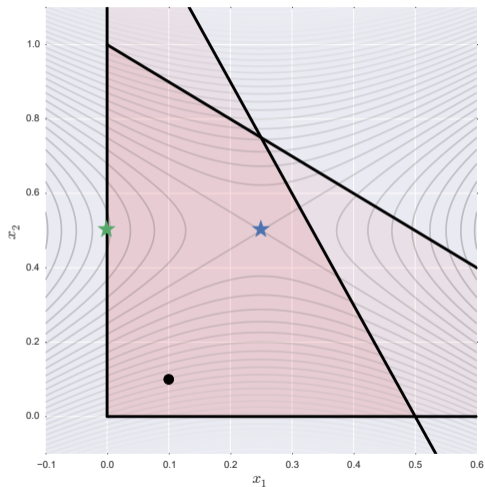
Deflation finds both minima and the saddle point.

Gould gives an optimization problem where interior point methods struggle:



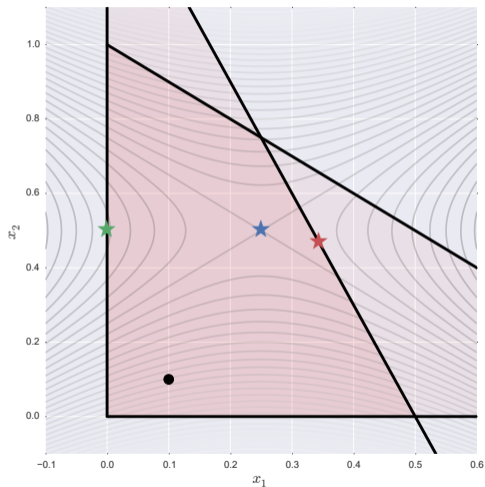
Deflation finds both minima and the saddle point.

Gould gives an optimization problem where interior point methods struggle:



Deflation finds both minima and the saddle point.

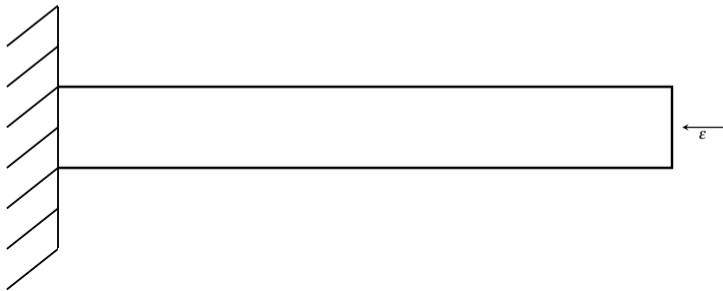
Gould gives an optimization problem where interior point methods struggle:



Deflation finds both minima and the saddle point.

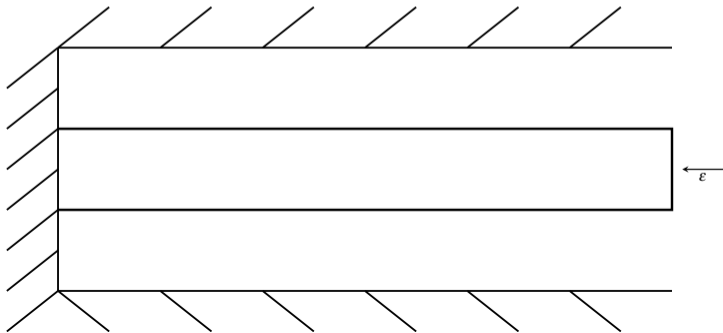
Buckling of a hyperelastic beam with contact constraints

$$\begin{aligned} & \underset{u \in H^1(\Omega; \mathbb{R}^2)}{\text{minimise}} && \Pi(u) = \int_{\Omega} \psi(u) \, dx - \int_{\Omega} B \cdot u \, dx \\ & \text{subject to} && u|_{\text{left}} = (0, 0), \quad u|_{\text{right}} = (-\varepsilon, 0), \\ & && \text{tr}(u_y) \in [a, b] \text{ a.e. in } \Gamma_{\text{top}}, \Gamma_{\text{bottom}}. \end{aligned}$$



Buckling of a hyperelastic beam with contact constraints

$$\begin{aligned} & \underset{u \in H^1(\Omega; \mathbb{R}^2)}{\text{minimise}} && \Pi(u) = \int_{\Omega} \psi(u) \, dx - \int_{\Omega} B \cdot u \, dx \\ & \text{subject to} && u|_{\text{left}} = (0, 0), \quad u|_{\text{right}} = (-\varepsilon, 0), \\ & && \text{tr}(u_y) \in [a, b] \text{ a.e. in } \Gamma_{\text{top}}, \Gamma_{\text{bottom}}. \end{aligned}$$



Buckling of a hyperelastic beam with contact constraints

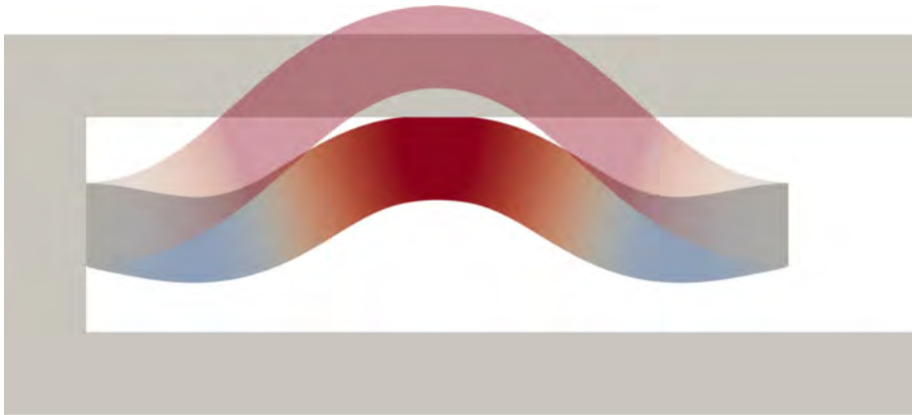
$$\begin{aligned} & \underset{u \in H^1(\Omega; \mathbb{R}^2)}{\text{minimise}} & \Pi(u) &= \int_{\Omega} \psi(u) \, dx - \int_{\Omega} B \cdot u \, dx \\ & \text{subject to} & u|_{\text{left}} &= (0, 0), \quad u|_{\text{right}} = (-\varepsilon, 0), \\ & & \text{tr}(u_y) &\in [a, b] \text{ a.e. in } \Gamma_{\text{top}}, \Gamma_{\text{bottom}}. \end{aligned}$$

Neo-Hookean strain energy density

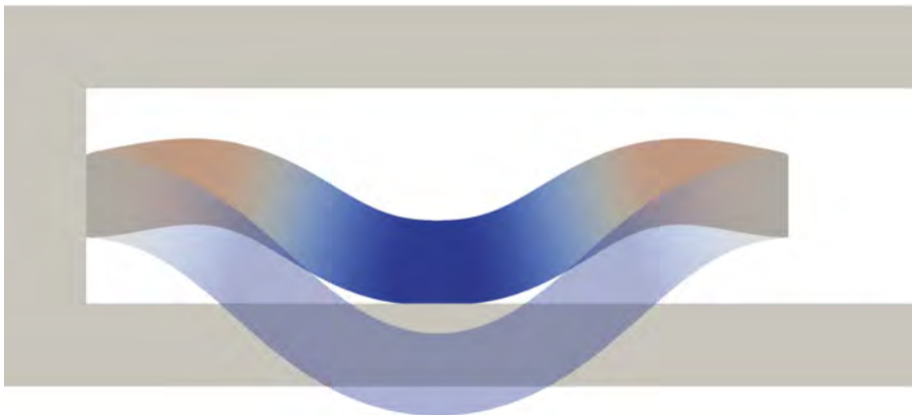
$$\psi(u) = \frac{\mu}{2}(\text{tr}(C) - 2) - \mu \log(\det(C)) + \frac{\lambda}{2} \log(\det(C))^2,$$

where

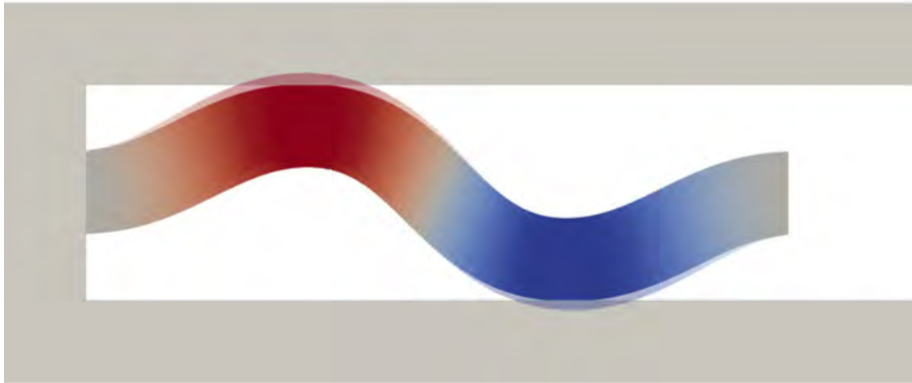
$$C = (I + \nabla u)^{\top} (I + \nabla u).$$



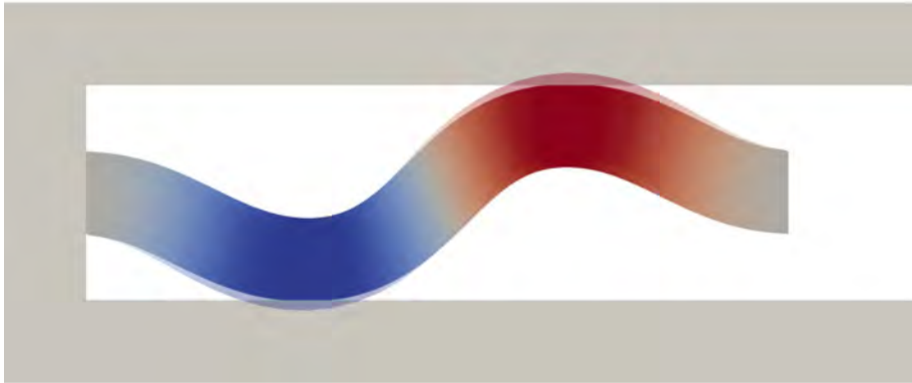
Multiple solutions of the beam with contact constraints.



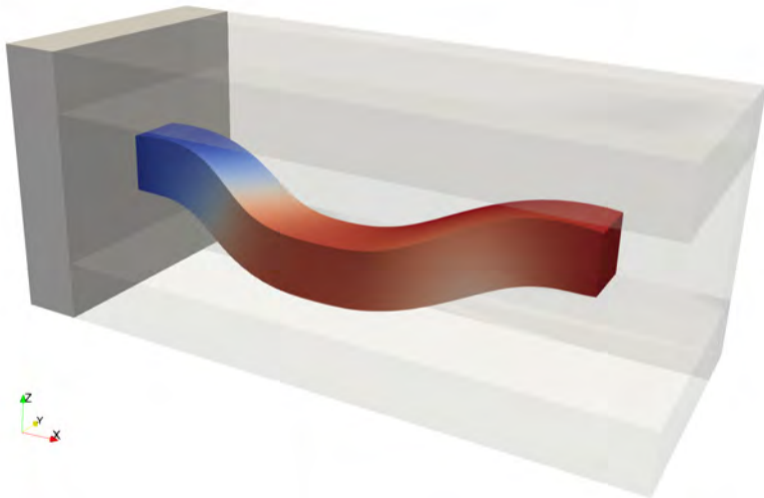
Multiple solutions of the beam with contact constraints.



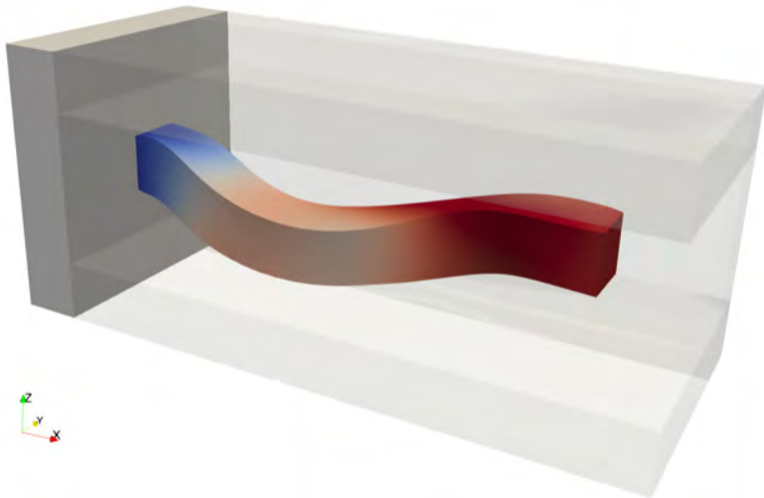
Multiple solutions of the beam with contact constraints.



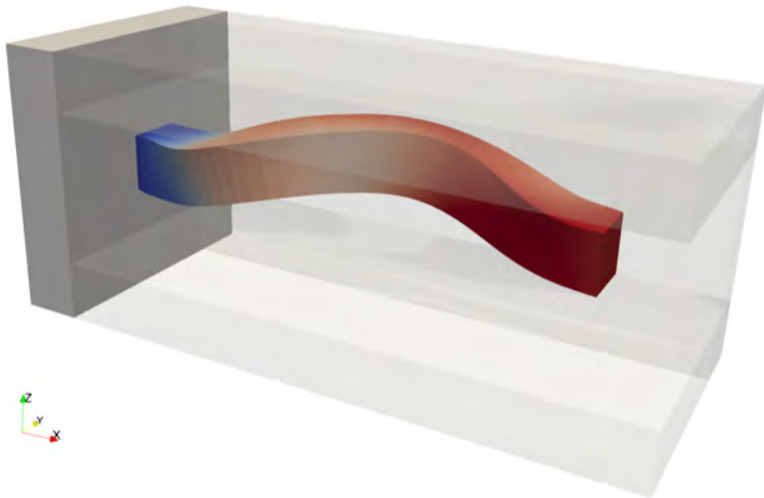
Multiple solutions of the beam with contact constraints.



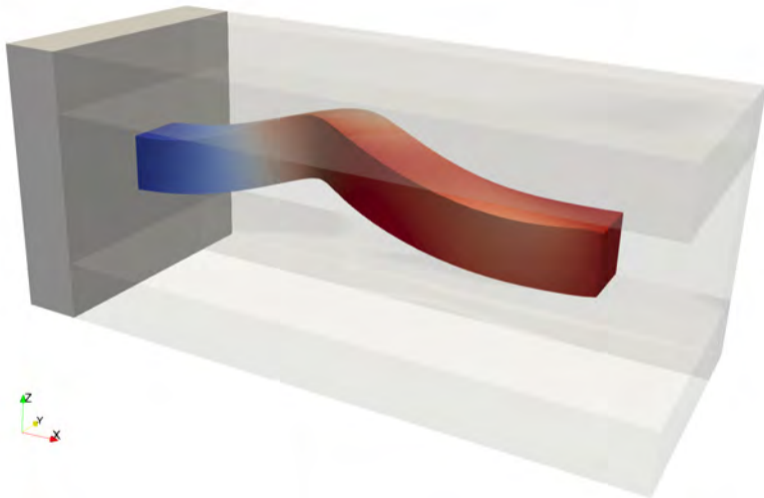
Multiple solutions of the beam with contact constraints.



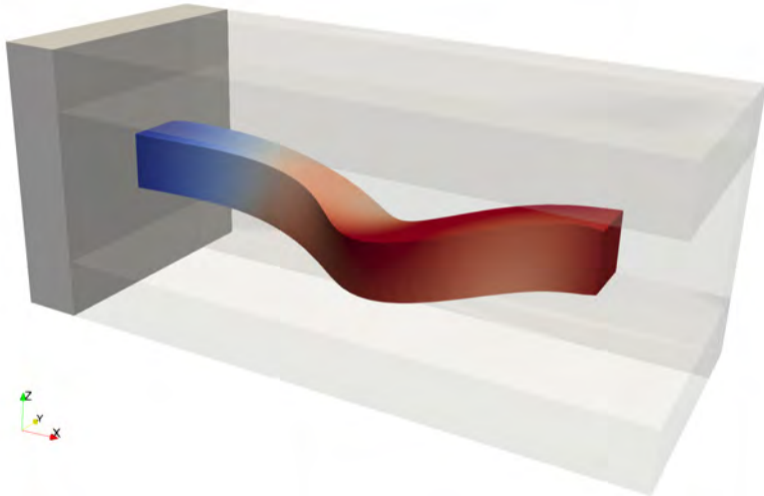
Multiple solutions of the beam with contact constraints.



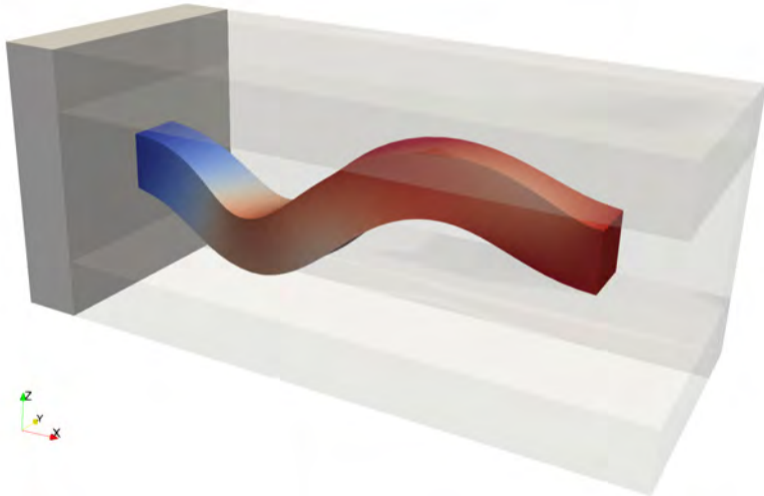
Multiple solutions of the beam with contact constraints.



Multiple solutions of the beam with contact constraints.



Multiple solutions of the beam with contact constraints.



Multiple solutions of the beam with contact constraints.

Conclusions!

Conclusions!

Observation

Nonlinear problems often have multiple solutions.

Conclusions!

Observation

Nonlinear problems often have multiple solutions.

Main message

Deflation is a powerful and useful technique for computing multiple solutions.

Conclusions!

Observation

Nonlinear problems often have multiple solutions.

Main message

Deflation is a powerful and useful technique for computing multiple solutions.

Question for the audience

How should deflation + continuation be combined with adaptive discretisations?