Adjoint–Based Optimization in Fluid Mechanics: Theory, Computations and Industrial Applications

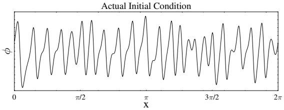
Bartosz Protas

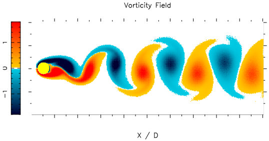
Department of Mathmatics & Statistics
McMaster University
Hamilton, ON, Canada
URL: http://www.math.mcmaster.ca/bprotas

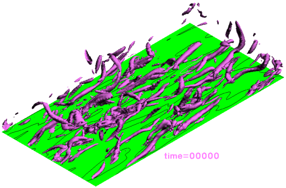
Collaborators: O. Volkov, W .Liao, D. Glander

Funded by NSERC–Discovery, NSERC–CRD, McMaster Centre for Advanced Manufacturing and Materials (MCAMM), Ontario Centres of Excellence (OCE) & General Motors of Canada

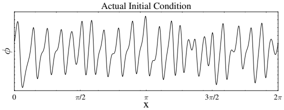
Motivation & Model Problems

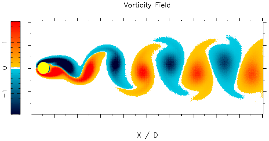


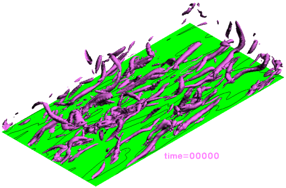




Motivation & Model Problems

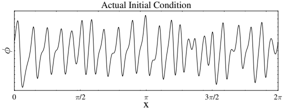


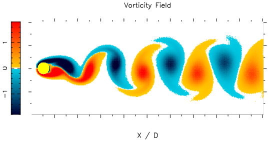


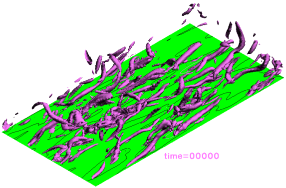


- Objectives:
  - Control fluid flow with the least amount of energy possible
  - Estimate flow based on incomplete and/or noisy measurements

Motivation & Model Problems



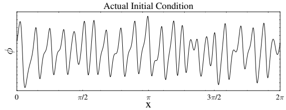


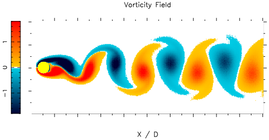


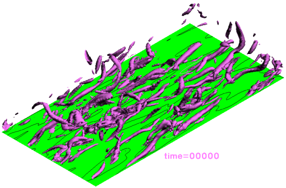
- Objectives:
  - Control fluid flow with the least amount of energy possible
  - Estimate flow based on incomplete and/or noisy measurements
- The Navier–Stokes system

$$\begin{cases} \dfrac{\partial \mathbf{v}}{\partial t} + (\mathbf{v}\cdot\boldsymbol{\nabla})\mathbf{v} + \boldsymbol{\nabla} p - \mu\Delta\mathbf{v} = \phi, & \text{in } \Omega\times(0,T) \\ \boldsymbol{\nabla}\cdot\mathbf{v} = 0, & \text{in } \Omega\times(0,T) \\ \text{Initial condition} & \text{on } \Gamma\times(0,T) \\ \text{Boundary condition} & \text{in } \Omega \text{ at } t=0 \end{cases}$$

Motivation & Model Problems







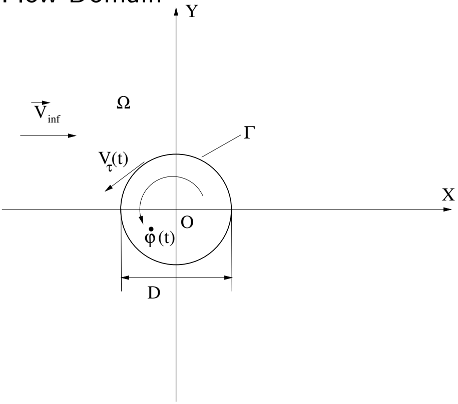
- Objectives:
  - Control fluid flow with the least amount of energy possible
  - Estimate flow based on incomplete and/or noisy measurements
- The Navier–Stokes system

$$\begin{cases} \dfrac{\partial \mathbf{v}}{\partial t} + (\mathbf{v}\cdot\boldsymbol{\nabla})\mathbf{v} + \boldsymbol{\nabla} p - \mu\Delta\mathbf{v} = \phi, & \text{in } \Omega\times(0,T) \\ \boldsymbol{\nabla}\cdot\mathbf{v} = 0, & \text{in } \Omega\times(0,T) \\ \text{Initial condition} & \text{on } \Gamma\times(0,T) \\ \text{Boundary condition} & \text{in } \Omega \text{ at } t=0 \end{cases}$$

- Inverse problems

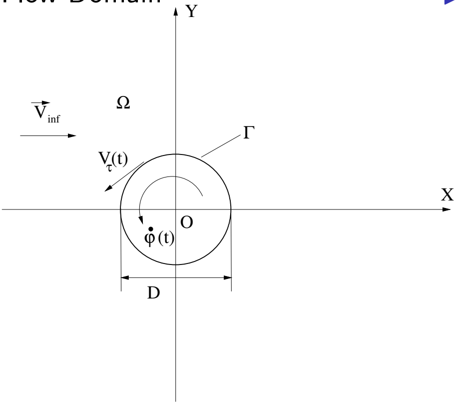
Statement of the Problem I

- Flow Domain



Statement of the Problem I

- Flow Domain



- Assumptions:
  - viscous, incompressible flow
  - plane, infinite domain
  - $Re = 150$

Statement of the Problem II

- Find $\dot{\varphi}_{opt} = \operatorname{argmin}_{\dot{\varphi}} \mathcal{J}(\dot{\varphi})$, where

$$\mathcal{J}(\dot{\varphi}) = \frac{1}{2}\int_0^T \left\{ \left[\begin{array}{c} \text{power related to} \\ \text{the drag force} \end{array} \right] + \left[\begin{array}{c} \text{power needed to} \\ \text{control the flow} \end{array} \right] \right\} dt$$

$$= \frac{1}{2}\int_0^T \oint_{\Gamma_0} \{[p(\dot{\varphi})\mathbf{n} - \mu\mathbf{n}\cdot\mathbf{D}(\mathbf{v}(\dot{\varphi}))]\cdot[\dot{\varphi}\,(\mathbf{e}_z \times \mathbf{r}) + \mathbf{v}_\infty]\}\, d\sigma dt$$

Statement of the Problem II

- Find $\dot{\varphi}_{opt} = \operatorname{argmin}_{\dot{\varphi}} \mathcal{J}(\dot{\varphi})$, where

$$\mathcal{J}(\dot{\varphi}) = \frac{1}{2}\int_0^T \left\{ \left[\begin{array}{c} \text{power related to} \\ \text{the drag force} \end{array} \right] + \left[\begin{array}{c} \text{power needed to} \\ \text{control the flow} \end{array} \right] \right\} dt$$
$$= \frac{1}{2}\int_0^T \oint_{\Gamma_0} \{[p(\dot{\varphi})\mathbf{n} - \mu\mathbf{n}\cdot\mathbf{D}(\mathbf{v}(\dot{\varphi}))]\cdot[\dot{\varphi}\,(\mathbf{e}_z \times \mathbf{r}) + \mathbf{v}_\infty]\}\, d\sigma dt$$

- Subject to:

$$\begin{cases} \left[\begin{array}{c} \frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v}\cdot\boldsymbol{\nabla})\mathbf{v} - \mu\Delta\mathbf{v} + \boldsymbol{\nabla} p \\ \boldsymbol{\nabla}\cdot\mathbf{v} \end{array} \right] = \left[\begin{array}{c} 0 \\ 0 \end{array} \right] & \text{in } \Omega \times (0, T), \\ \mathbf{v} = 0 & \text{at } t = 0, \\ \mathbf{v} = \dot{\varphi}_{opt}\tau & \text{on } \Gamma \end{cases}$$

Abstract Framework I

- Constrained optimization problem

$$\begin{cases} \min_{(x,\varphi)} \tilde{\mathcal{J}}(x,\varphi) \\ S(x(\varphi),\varphi) = 0 \end{cases}$$

Abstract Framework I

- Constrained optimization problem

$$\begin{cases} \min_{(x,\varphi)} \tilde{\mathcal{J}}(x,\varphi) \\ S(x(\varphi),\varphi) = 0 \end{cases}$$

- Equivalent UNCONSTRAINED optimization problem (note that $x = x(\varphi)$)

$$\min_{\varphi} \tilde{\mathcal{J}}(x(\varphi),\varphi) = \min_{\varphi} \mathcal{J}(\varphi)$$

Abstract Framework I

- Constrained optimization problem

$$\begin{cases} \min_{(x,\varphi)} \tilde{\mathcal{J}}(x,\varphi) \\ S(x(\varphi),\varphi) = 0 \end{cases}$$

- Equivalent UNCONSTRAINED optimization problem (note that $x = x(\varphi)$)

$$\min_{\varphi} \tilde{\mathcal{J}}(x(\varphi),\varphi) = \min_{\varphi} \mathcal{J}(\varphi)$$

- First–Order OPTIMALITY CONDITIONS ($\mathcal{U}$ - Hilbert space of controls)

$$\forall_{\varphi' \in \mathcal{U}} \ \ \mathcal{J}'(\varphi;\varphi') = (\boldsymbol{\nabla}\mathcal{J},\varphi')_{\mathcal{U}} = 0,$$

with the GÂTEAUX DIFFERENTIAL
$\mathcal{J}'(\varphi;\varphi') = \lim_{\epsilon\to 0} \frac{1}{\epsilon}[\mathcal{J}(\varphi+\epsilon\varphi') - \mathcal{J}(\varphi)]$.

Abstract Framework II

- Minimization of $\mathcal{J}(\varphi)$ with a DESCENT ALGORITHM in $\mathcal{U}$
  $\Longrightarrow$ solution to a STEADY STATE of the ODE in $\mathcal{U}$

$$\begin{cases} \dfrac{d\varphi}{d\tau} = -\mathcal{Q}\nabla_{\varphi}\mathcal{J}(\varphi) & \text{on } \tau \in (0,\infty) \ \ \text{(pseudo–time)}, \\ \varphi = \varphi_0 & \text{at } \tau = 0. \end{cases}$$

Abstract Framework II

- Minimization of $\mathcal{J}(\varphi)$ with a DESCENT ALGORITHM in $\mathcal{U}$
  $\Longrightarrow$ solution to a STEADY STATE of the ODE in $\mathcal{U}$

$$\begin{cases} \dfrac{d\varphi}{d\tau} = -\mathcal{Q}\nabla_{\varphi}\mathcal{J}(\varphi) & \text{on } \tau \in (0,\infty) \quad \text{(pseudo–time)}, \\ \varphi = \varphi_0 & \text{at } \tau = 0. \end{cases}$$

- Typically well–behaved (quadratic) cost functionals

Abstract Framework II

- Minimization of $\mathcal{J}(\varphi)$ with a DESCENT ALGORITHM in $\mathcal{U}$
  $\Longrightarrow$ solution to a STEADY STATE of the ODE in $\mathcal{U}$

$$\begin{cases} \dfrac{d\varphi}{d\tau} = -\mathcal{Q}\nabla_{\varphi}\mathcal{J}(\varphi) & \text{on } \tau \in (0,\infty) \ \text{(pseudo–time)}, \\ \varphi = \varphi_0 & \text{at } \tau = 0. \end{cases}$$

- Typically well–behaved (quadratic) cost functionals
- Typically ill–behaved constraints: THE NAVIER–STOKES SYSTEM
  - nonlinear, nonlocal, multiscale, evolutionary PDE,

Abstract Framework II

- Minimization of $\mathcal{J}(\varphi)$ with a DESCENT ALGORITHM in $\mathcal{U}$
  $\Longrightarrow$ solution to a STEADY STATE of the ODE in $\mathcal{U}$

$$\begin{cases} \dfrac{d\varphi}{d\tau} = -\mathcal{Q}\nabla_{\varphi}\mathcal{J}(\varphi) & \text{on } \tau \in (0,\infty) \ \text{(pseudo–time)}, \\ \varphi = \varphi_0 & \text{at } \tau = 0. \end{cases}$$

- Typically well–behaved (quadratic) cost functionals
- Typically ill–behaved constraints: THE NAVIER–STOKES SYSTEM
  - nonlinear, nonlocal, multiscale, evolutionary PDE,
- Dimensions:
  - state: 10^6 — 10^7 DoF $\times$ 10^2 — 10^3 time levels
  - control: 10^4 — 10^5 DoF $\times$ 10^2 — 10^3 time levels

Abstract Framework II

- Minimization of $\mathcal{J}(\varphi)$ with a DESCENT ALGORITHM in $\mathcal{U}$ $\Longrightarrow$ solution to a STEADY STATE of the ODE in $\mathcal{U}$

$$\begin{cases} \dfrac{d\varphi}{d\tau} = -\mathcal{Q}\nabla_{\varphi}\mathcal{J}(\varphi) & \text{on } \tau \in (0,\infty) \ \text{(pseudo–time)}, \\ \varphi = \varphi_0 & \text{at } \tau = 0. \end{cases}$$

- Typically well–behaved (quadratic) cost functionals
- Typically ill–behaved constraints: THE NAVIER–STOKES SYSTEM
  - nonlinear, nonlocal, multiscale, evolutionary PDE,
- Dimensions:
  - state: 10^6 — 10^7 DoF $\times$ 10^2 — 10^3 time levels
  - control: 10^4 — 10^5 DoF $\times$ 10^2 — 10^3 time levels
- No hope of using "matrix" formulation ...

Abstract Framework II

- Minimization of $\mathcal{J}(\varphi)$ with a DESCENT ALGORITHM in $\mathcal{U}$
  $\Longrightarrow$ solution to a STEADY STATE of the ODE in $\mathcal{U}$

$$\begin{cases} \dfrac{d\varphi}{d\tau} = -\mathcal{Q}\nabla_{\varphi}\mathcal{J}(\varphi) & \text{on } \tau \in (0,\infty) \ \text{(pseudo–time)}, \\ \varphi = \varphi_0 & \text{at } \tau = 0. \end{cases}$$

- Typically well–behaved (quadratic) cost functionals
- Typically ill–behaved constraints: THE NAVIER–STOKES SYSTEM
  - nonlinear, nonlocal, multiscale, evolutionary PDE,
- Dimensions:
  - state: 10^6 — 10^7 DoF $\times$ 10^2 — 10^3 time levels
  - control: 10^4 — 10^5 DoF $\times$ 10^2 — 10^3 time levels
- No hope of using "matrix" formulation ...
- Formulation equivalent to Lagrange Multipliers

Differential of the Cost Functional

- The cost functional:

$$\mathcal{J}(\dot\varphi) = \frac{1}{2}\int_0^T \left\{ \left[\begin{array}{c}\text{power related to} \\ \text{the drag force}\end{array} \right] + \left[\begin{array}{c}\text{power needed to} \\ \text{control the flow}\end{array} \right] \right\} dt$$
$$= \frac{1}{2}\int_0^T \oint_{\Gamma_0} \{[p(\dot\varphi)\mathbf{n} - \mu\mathbf{n}\cdot\mathbf{D}(\mathbf{v}(\dot\varphi))]\cdot[\dot\varphi\,(\mathbf{e}_z\times\mathbf{r}) + \mathbf{v}_\infty]\}\, d\sigma\, dt,$$

Differential of the Cost Functional

- The cost functional:

$$\mathcal{J}(\dot\varphi) = \frac{1}{2}\int_0^T \left\{ \left[\begin{array}{c} \text{power related to} \\ \text{the drag force} \end{array} \right] + \left[\begin{array}{c} \text{power needed to} \\ \text{control the flow} \end{array} \right] \right\} dt$$

$$= \frac{1}{2}\int_0^T \oint_{\Gamma_0} \{ [p(\dot\varphi)\mathbf{n} - \mu\mathbf{n}\cdot\mathbf{D}(\mathbf{v}(\dot\varphi))]\cdot[\dot\varphi\,(\mathbf{e}_z\times\mathbf{r}) + \mathbf{v}_\infty]\}\, d\sigma dt,$$

- Expression for the Gâteaux differential:

$$\mathcal{J}'(\dot\varphi; h) = \frac{1}{2}\int_0^T \oint_{\Gamma_0} \Big\{ [p'(h)\mathbf{n} - \mu\mathbf{n}\cdot\mathbf{D}\,(\mathbf{v}'(h))]\cdot[\dot\varphi\,(\mathbf{e}_z\times\mathbf{r}) + \mathbf{v}_\infty] +$$

$$[p(\dot\varphi)\mathbf{n} - \mu\mathbf{n}\cdot\mathbf{D}(\mathbf{v}(\dot\varphi))]\cdot(\mathbf{e}_z\times\mathbf{r})\, h\Big\}\, d\sigma\, dt = B_1$$

$$= (\nabla\mathcal{J}(t), h)_{L_2([0,T])}$$

The fields $\{\mathbf{v}'(h), p'(h)\}$ solve the linearized perturbation system.

Differential of the Cost Functional

- The cost functional:

$$\mathcal{J}(\dot\varphi) = \frac{1}{2}\int_0^T \left\{ \left[\begin{array}{c}\text{power related to}\\ \text{the drag force}\end{array} \right] + \left[\begin{array}{c}\text{power needed to}\\ \text{control the flow}\end{array} \right] \right\} dt$$

$$= \frac{1}{2}\int_0^T \oint_{\Gamma_0} \{ [p(\dot\varphi)\mathbf{n} - \mu\mathbf{n}\cdot\mathbf{D}(\mathbf{v}(\dot\varphi))] \cdot [\dot\varphi\,(\mathbf{e}_z\times\mathbf{r}) + \mathbf{v}_\infty] \}\, d\sigma\, dt,$$

- Expression for the Gâteaux differential:

$$\mathcal{J}'(\dot\varphi; h) = \frac{1}{2}\int_0^T \oint_{\Gamma_0} \Big\{ [p'(h)\mathbf{n} - \mu\mathbf{n}\cdot\mathbf{D}(\mathbf{v}'(h))] \cdot [\dot\varphi\,(\mathbf{e}_z\times\mathbf{r}) + \mathbf{v}_\infty] + [p(\dot\varphi)\mathbf{n} - \mu\mathbf{n}\cdot\mathbf{D}(\mathbf{v}(\dot\varphi))] \cdot (\mathbf{e}_z\times\mathbf{r})\, h \Big\}\, d\sigma\, dt = B_1$$

$$= (\nabla\mathcal{J}(t), h)_{L_2([0,T])}$$

The fields $\{\mathbf{v}'(h), p'(h)\}$ solve the linearized perturbation system.

- How to calculate the GRADIENT $\nabla\mathcal{J}$?

Sensitivities and Adjoint States

- The linearized perturbation system

$$\begin{cases} \mathcal{N}\begin{bmatrix} \mathbf{v}' \\ p' \end{bmatrix} = \begin{bmatrix} \frac{\partial \mathbf{v}'}{\partial t} + (\mathbf{v}\cdot\boldsymbol{\nabla})\mathbf{v}' + (\mathbf{v}'\cdot\boldsymbol{\nabla})\mathbf{v} - \mu\Delta\mathbf{v}' + \boldsymbol{\nabla} p' \\ -\boldsymbol{\nabla}\cdot\mathbf{v}' \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} & \text{in } \Omega\times(0,T), \\ \mathbf{v}' = 0 & \text{at } t=0, \\ \mathbf{v}' = h\tau & \text{on } \Gamma\times(0,T) \end{cases}$$

Sensitivities and Adjoint States

- The linearized perturbation system

$$\begin{cases} \mathcal{N}\begin{bmatrix} \mathbf{v}' \\ p' \end{bmatrix} = \begin{bmatrix} \frac{\partial \mathbf{v}'}{\partial t} + (\mathbf{v}\cdot\boldsymbol{\nabla})\mathbf{v}' + (\mathbf{v}'\cdot\boldsymbol{\nabla})\mathbf{v} - \mu\Delta\mathbf{v}' + \boldsymbol{\nabla}p' \\ -\boldsymbol{\nabla}\cdot\mathbf{v}' \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} & \text{in } \Omega\times(0,T), \\ \mathbf{v}' = 0 & \text{at } t=0, \\ \mathbf{v}' = h\tau & \text{on } \Gamma\times(0,T) \end{cases}$$

- Duality pairing defining the adjoint operator

$$\left\langle \mathcal{N}\begin{bmatrix} \mathbf{v}' \\ p' \end{bmatrix}, \begin{bmatrix} \mathbf{v}^* \\ p^* \end{bmatrix} \right\rangle_{L_2(0,T;L_2(\Omega))} = \left\langle \begin{bmatrix} \mathbf{v}' \\ p' \end{bmatrix}, \mathcal{N}^*\begin{bmatrix} \mathbf{v}^* \\ p^* \end{bmatrix} \right\rangle_{L_2(0,T;L_2(\Omega))} + B_1 + B_2$$

Sensitivities and Adjoint States

- The linearized perturbation system

$$\begin{cases} \mathcal{N}\begin{bmatrix} \mathbf{v}' \\ p' \end{bmatrix} = \begin{bmatrix} \frac{\partial \mathbf{v}'}{\partial t} + (\mathbf{v}\cdot\boldsymbol{\nabla})\mathbf{v}' + (\mathbf{v}'\cdot\boldsymbol{\nabla})\mathbf{v} - \mu\Delta\mathbf{v}' + \boldsymbol{\nabla} p' \\ -\boldsymbol{\nabla}\cdot\mathbf{v}' \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} & \text{in } \Omega\times(0,T), \\ \mathbf{v}' = 0 & \text{at } t=0, \\ \mathbf{v}' = h\tau & \text{on } \Gamma\times(0,T) \end{cases}$$

- Duality pairing defining the adjoint operator

$$\left\langle \mathcal{N}\begin{bmatrix} \mathbf{v}' \\ p' \end{bmatrix}, \begin{bmatrix} \mathbf{v}^* \\ p^* \end{bmatrix} \right\rangle_{L_2(0,T;L_2(\Omega))} = \left\langle \begin{bmatrix} \mathbf{v}' \\ p' \end{bmatrix}, \mathcal{N}^*\begin{bmatrix} \mathbf{v}^* \\ p^* \end{bmatrix} \right\rangle_{L_2(0,T;L_2(\Omega))} + B_1 + B_2$$

- The adjoint system (TERMINAL VALUE PROBLEM !!)

$$\begin{cases} \mathcal{N}^*\begin{bmatrix} \mathbf{v}^* \\ p^* \end{bmatrix} = \begin{bmatrix} -\frac{\partial \mathbf{v}^*}{\partial t} - \mathbf{v}\cdot\left[\boldsymbol{\nabla}\mathbf{v}^* + (\boldsymbol{\nabla}\mathbf{v}^*)^T\right] - \mu\Delta\mathbf{v}^* + \boldsymbol{\nabla} p^* \\ -\boldsymbol{\nabla}\cdot\mathbf{v}^* \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} & \text{in } \Omega\times(0,T), \\ \mathbf{v}^* = 0 & \text{at } t=T, \\ \mathbf{v}^* = \mathbf{r}\times(\dot{\varphi}\mathbf{e}_z) + \mathbf{v}_\infty & \text{on } \Gamma\times(0,T) \end{cases}$$

Cost Functional Gradient

- The ADJOINT STATE and DUALITY PAIRING can now be used to re–express the cost functional differential as:

$$\mathcal{J}'(\dot{\varphi}; h) = \frac{1}{2}\int_0^T \oint_\Gamma \{\mu R \mathbf{n} \cdot \mathbf{D}(\mathbf{v}^*) \cdot \tau + \mu \mathbf{n} \cdot \mathbf{D}(\mathbf{v}(\dot{\varphi})) \cdot (\mathbf{e}_z \times \mathbf{r})\} \; h \, d\sigma \, dt$$

Cost Functional Gradient

- The ADJOINT STATE and DUALITY PAIRING can now be used to re–express the cost functional differential as:

$$\mathcal{J}'(\dot{\varphi}; h) = \frac{1}{2}\int_0^T \oint_\Gamma \{\mu R \mathbf{n} \cdot \mathbf{D}(\mathbf{v}^*) \cdot \tau + \mu \mathbf{n} \cdot \mathbf{D}(\mathbf{v}(\dot{\varphi})) \cdot (\mathbf{e}_z \times \mathbf{r})\} \, h \, d\sigma \, dt$$

- Identification of the COST FUNCTIONAL GRADIENT

$$\mathcal{J}'(\dot{\varphi}; h) = (\nabla\mathcal{J}(t), h)_{L_2([0,T])} = \int_0^T \nabla\mathcal{J}(t)\, h \, dt$$

$$\boxed{\nabla\mathcal{J}(t) = \frac{1}{2}\oint_\Gamma \{\mu R \mathbf{n} \cdot \mathbf{D}(\mathbf{v}^*) \cdot \tau + \mu \mathbf{n} \cdot \mathbf{D}(\mathbf{v}(\dot{\varphi})) \cdot (\mathbf{e}_z \times \mathbf{r})\} \, d\sigma}$$

Optimality (KKT) system

- Complete optimality system for $\dot{\varphi}_{opt}$, $[\mathbf{v}_{opt}, p_{opt}]$, and $[\mathbf{v}^*, p^*]$

$$
\begin{cases}
\frac{1}{2}\oint_\Gamma \{\mu R \mathbf{n}\cdot \mathbf{D}(\mathbf{v}^*)\cdot \tau + \mu \mathbf{n}\cdot \mathbf{D}(\mathbf{v}(\dot{\varphi}_{opt}))\cdot(\mathbf{e}_z \times \mathbf{r})\}\, d\sigma = 0 \\
\begin{cases}
\begin{bmatrix} \frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v}\cdot\boldsymbol{\nabla})\mathbf{v} - \mu\Delta\mathbf{v} + \boldsymbol{\nabla} p \\ \boldsymbol{\nabla}\cdot\mathbf{v} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} & \text{in } \Omega\times(0,T), \\
\mathbf{v} = 0 & \text{at } t = 0, \\
\mathbf{v} = \dot{\varphi}_{opt}\tau & \text{on } \Gamma
\end{cases} \\
\begin{cases}
\mathcal{N}^* \begin{bmatrix} \mathbf{v}^* \\ p^* \end{bmatrix} = \begin{bmatrix} -\frac{\partial \mathbf{v}^*}{\partial t} - \mathbf{v}\cdot\left[\boldsymbol{\nabla}\mathbf{v}^* + (\boldsymbol{\nabla}\mathbf{v}^*)^T\right] - \mu\Delta\mathbf{v}^* + \boldsymbol{\nabla} p^* \\ -\boldsymbol{\nabla}\cdot\mathbf{v}^* \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} & \text{in } \Omega\times(0,T), \\
\mathbf{v}^* = 0 & \text{at } t = T, \\
\mathbf{v}^* = \mathbf{r}\times(\dot{\varphi}_{opt}\mathbf{e}_z) + \mathbf{v}_\infty & \text{on } \Gamma
\end{cases}
\end{cases}
$$

Optimality (KKT) system

- Complete optimality system for $\dot{\varphi}_{opt}$, $[\mathbf{v}_{opt}, p_{opt}]$, and $[\mathbf{v}^*, p^*]$

$$
\begin{cases}
\frac{1}{2}\oint_\Gamma \{\mu R \mathbf{n}\cdot\mathbf{D}(\mathbf{v}^*)\cdot\tau + \mu\mathbf{n}\cdot\mathbf{D}(\mathbf{v}(\dot{\varphi}_{opt}))\cdot(\mathbf{e}_z\times\mathbf{r})\}\,d\sigma = 0 \\
\begin{cases}
\begin{bmatrix} \frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v}\cdot\boldsymbol{\nabla})\mathbf{v} - \mu\Delta\mathbf{v} + \boldsymbol{\nabla} p \\ \boldsymbol{\nabla}\cdot\mathbf{v} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} & \text{in } \Omega\times(0,T), \\
\mathbf{v} = 0 & \text{at } t = 0, \\
\mathbf{v} = \dot{\varphi}_{opt}\tau & \text{on } \Gamma
\end{cases} \\
\begin{cases}
\mathcal{N}^*\begin{bmatrix} \mathbf{v}^* \\ p^* \end{bmatrix} = \begin{bmatrix} -\frac{\partial \mathbf{v}^*}{\partial t} - \mathbf{v}\cdot\left[\boldsymbol{\nabla}\mathbf{v}^* + (\boldsymbol{\nabla}\mathbf{v}^*)^T\right] - \mu\Delta\mathbf{v}^* + \boldsymbol{\nabla} p^* \\ -\boldsymbol{\nabla}\cdot\mathbf{v}^* \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} & \text{in } \Omega\times(0,T), \\
\mathbf{v}^* = 0 & \text{at } t = T, \\
\mathbf{v}^* = \mathbf{r}\times(\dot{\varphi}_{opt}\mathbf{e}_z) + \mathbf{v}_\infty & \text{on } \Gamma
\end{cases}
\end{cases}
$$

- A counterpart of the Euler–Lagrange equation

Optimality (KKT) system

- Complete optimality system for $\dot\varphi_{opt}$, $[\mathbf{v}_{opt}, p_{opt}]$, and $[\mathbf{v}^*, p^*]$

$$
\begin{cases}
\frac{1}{2}\oint_\Gamma \left\{\mu R \mathbf{n}\cdot\mathbf{D}(\mathbf{v}^*)\cdot\tau + \mu\mathbf{n}\cdot\mathbf{D}(\mathbf{v}(\dot\varphi_{opt}))\cdot(\mathbf{e}_z\times\mathbf{r})\right\} d\sigma = 0 \\
\begin{cases}
\begin{bmatrix} \frac{\partial\mathbf{v}}{\partial t} + (\mathbf{v}\cdot\boldsymbol{\nabla})\mathbf{v} - \mu\Delta\mathbf{v} + \boldsymbol{\nabla}p \\ \boldsymbol{\nabla}\cdot\mathbf{v} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} & \text{in } \Omega\times(0,T), \\
\mathbf{v} = 0 & \text{at } t = 0, \\
\mathbf{v} = \dot\varphi_{opt}\tau & \text{on } \Gamma
\end{cases} \\
\begin{cases}
\mathcal{N}^*\begin{bmatrix} \mathbf{v}^* \\ p^* \end{bmatrix} = \begin{bmatrix} -\frac{\partial\mathbf{v}^*}{\partial t} - \mathbf{v}\cdot\left[\boldsymbol{\nabla}\mathbf{v}^* + (\boldsymbol{\nabla}\mathbf{v}^*)^T\right] - \mu\Delta\mathbf{v}^* + \boldsymbol{\nabla}p^* \\ -\boldsymbol{\nabla}\cdot\mathbf{v}^* \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} & \text{in } \Omega\times(0,T), \\
\mathbf{v}^* = 0 & \text{at } t = T, \\
\mathbf{v}^* = \mathbf{r}\times(\dot\varphi_{opt}\mathbf{e}_z) + \mathbf{v}_\infty & \text{on } \Gamma
\end{cases}
\end{cases}
$$

- A counterpart of the Euler–Lagrange equation
- Solved with an iterative Gradient Algorithm (e.g., Conjugate Gradients, quasi–Newton, etc.)

An Iterative Optimization Procedure

0. provide initial guess $\dot{\varphi}^0$

An Iterative Optimization Procedure

0. provide initial guess $\dot{\varphi}^0$
1. Solve for $\left\{\mathbf{v}(\dot{\varphi}^i); p(\dot{\varphi}^i)\right\}$ on $[0, T]$

An Iterative Optimization Procedure

0. provide initial guess $\dot{\varphi}^0$
1. Solve for $\{\mathbf{v}(\dot{\varphi}^i); p(\dot{\varphi}^i)\}$ on $[0,T]$
2. Solve for $\{\mathbf{v}^*(\dot{\varphi}^i); p^*(\dot{\varphi}^i)\}$ on $[0,T]$

An Iterative Optimization Procedure

0. provide initial guess $\dot{\varphi}^0$
1. Solve for $\left\{\mathbf{v}(\dot{\varphi}^i); p(\dot{\varphi}^i)\right\}$ on $[0, T]$
2. Solve for $\left\{\mathbf{v}^*(\dot{\varphi}^i); p^*(\dot{\varphi}^i)\right\}$ on $[0, T]$
3. Use $\left\{\mathbf{v}(\dot{\varphi}^i); p(\dot{\varphi}^i)\right\}$ and $\left\{\mathbf{v}^*(\dot{\varphi}^i); p^*(\dot{\varphi}^i)\right\}$ to compute $\nabla\mathcal{J}^i(t)$ on $[0, T]$

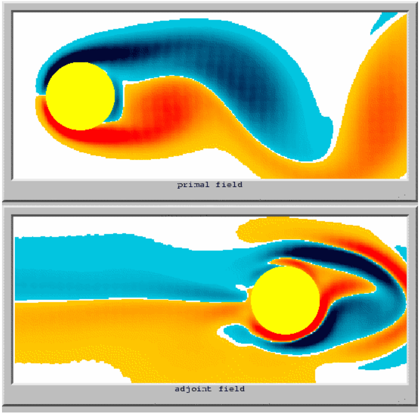
An Iterative Optimization Procedure

0. provide initial guess $\dot{\varphi}^0$
1. Solve for $\left\{ \mathbf{v}(\dot{\varphi}^i); p(\dot{\varphi}^i) \right\}$ on $[0, T]$
2. Solve for $\left\{ \mathbf{v}^*(\dot{\varphi}^i); p^*(\dot{\varphi}^i) \right\}$ on $[0, T]$
3. Use $\left\{ \mathbf{v}(\dot{\varphi}^i); p(\dot{\varphi}^i) \right\}$ and $\left\{ \mathbf{v}^*(\dot{\varphi}^i); p^*(\dot{\varphi}^i) \right\}$ to compute $\nabla \mathcal{J}^i(t)$ on $[0, T]$
4. update control according to $\dot{\varphi}^{i+1}(t) = \dot{\varphi}^i(t) - \alpha_i \gamma_i \left(\nabla \mathcal{J}(t) \right)$

An Iterative Optimization Procedure

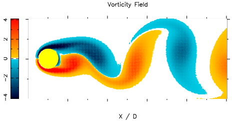
0. provide initial guess $\dot{\varphi}^0$
1. Solve for $\left\{ \mathbf{v}(\dot{\varphi}^i); p(\dot{\varphi}^i) \right\}$ on $[0, T]$
2. Solve for $\left\{ \mathbf{v}^*(\dot{\varphi}^i); p^*(\dot{\varphi}^i) \right\}$ on $[0, T]$
3. Use $\left\{ \mathbf{v}(\dot{\varphi}^i); p(\dot{\varphi}^i) \right\}$ and $\left\{ \mathbf{v}^*(\dot{\varphi}^i); p^*(\dot{\varphi}^i) \right\}$ to compute $\nabla \mathcal{J}^i(t)$ on $[0, T]$
4. update control according to $\dot{\varphi}^{i+1}(t) = \dot{\varphi}^i(t) - \alpha_i \gamma_i \left(\nabla \mathcal{J}(t)\right)$
5. iterate 1. through 4. until convergence, i.e. until $\boldsymbol{\nabla} J^i(t) \simeq 0$

Primal and Adjoint Simulations for Cylinder Rotation as Control



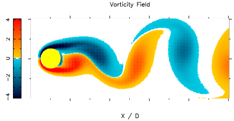
Results

- No Control

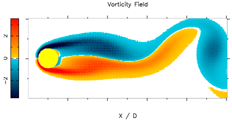


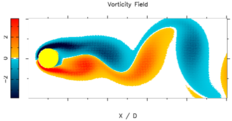
Results

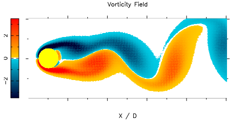
- No Control



- Flow Pattern Modifications due to Control ($T = 6$)

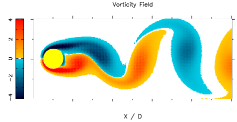




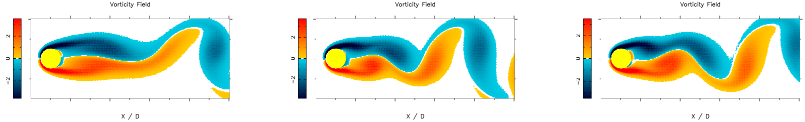


Results

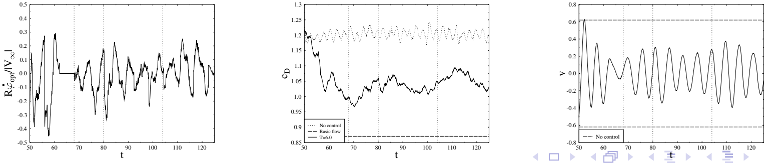
- No Control



- Flow Pattern Modifications due to Control ($T = 6$)



- Optimal Control $\dot{\varphi}_{opt}$, drag coefficient c_D, transverse velocity v



Conditioning and Preconditioning

- Rate of convergence in a NLP depends on CONDITIONING OF THE PROBLEM

Conditioning and Preconditioning

- Rate of convergence in a NLP depends on CONDITIONING OF THE PROBLEM
- Conditioning determined by the reduced Hessian

$$\nabla^2_{(\varphi)} F = L_{\varphi\varphi} + L_{\varphi u}\left(\frac{du}{d\varphi}\right) + \left(\frac{du}{d\varphi}\right)^* L_{u\varphi} + \left(\frac{du}{d\varphi}\right)^* L_{uu}\left(\frac{du}{d\varphi}\right)$$

where (via the implicit function theorem)

$$L(u,\varphi,\lambda) = f(u,\varphi) + \langle \lambda, S(u(\varphi),\varphi)\rangle \qquad \frac{du}{d\varphi} = -S_u^{-1} S_\varphi$$

Conditioning and Preconditioning

- Rate of convergence in a NLP depends on CONDITIONING OF THE PROBLEM
- Conditioning determined by the reduced Hessian

$$\nabla^2_{(\varphi)} F = L_{\varphi\varphi} + L_{\varphi u}\left(\frac{du}{d\varphi}\right) + \left(\frac{du}{d\varphi}\right)^* L_{u\varphi} + \left(\frac{du}{d\varphi}\right)^* L_{uu}\left(\frac{du}{d\varphi}\right)$$

where (via the implicit function theorem)

$$L(u,\varphi,\lambda) = f(u,\varphi) + \langle \lambda, S(u(\varphi),\varphi)\rangle \qquad \frac{du}{d\varphi} = -S_u^{-1} S_\varphi$$

- Preconditioning via "Sobolev Gradients" [Neuberger (1997)]

$$\begin{cases} \dfrac{d\varphi}{d\tau} = -\boldsymbol{\nabla}^{\mathcal{Q}} \mathcal{J}(\varphi) & \text{on } \tau \in (0,\infty) \ \text{(pseudo–time)}, \\ \varphi = \varphi_0 & \text{at } \tau = 0. \end{cases}$$

where $(\boldsymbol{\nabla}^{\mathcal{Q}}\mathcal{J}, \varphi')_{\mathcal{Q}} = (\boldsymbol{\nabla}^{L_2}\mathcal{J}, \varphi')_{L_2}$ and $\mathcal{Q} \subset \mathcal{U}$

Conditioning and Preconditioning

- Rate of convergence in a NLP depends on CONDITIONING OF THE PROBLEM
- Conditioning determined by the reduced Hessian

$$\nabla^2_{(\varphi)} F = L_{\varphi\varphi} + L_{\varphi u}\left(\frac{du}{d\varphi}\right) + \left(\frac{du}{d\varphi}\right)^* L_{u\varphi} + \left(\frac{du}{d\varphi}\right)^* L_{uu}\left(\frac{du}{d\varphi}\right)$$

where (via the implicit function theorem)

$$L(u,\varphi,\lambda) = f(u,\varphi) + \langle \lambda, S(u(\varphi),\varphi)\rangle \qquad \frac{du}{d\varphi} = -S_u^{-1} S_\varphi$$

- Preconditioning via "Sobolev Gradients" [Neuberger (1997)]

$$\begin{cases} \frac{d\varphi}{d\tau} = -\boldsymbol{\nabla}^{\mathcal{Q}}\mathcal{J}(\varphi) & \text{on } \tau \in (0,\infty) \text{ (pseudo–time)}, \\ \varphi = \varphi_0 & \text{at } \tau = 0. \end{cases}$$

where $(\boldsymbol{\nabla}^{\mathcal{Q}}\mathcal{J}, \varphi')_{\mathcal{Q}} = (\boldsymbol{\nabla}^{L_2}\mathcal{J}, \varphi')_{L_2}$ and $\mathcal{Q} \subset \mathcal{U}$

- Variable Preconditioning: $\mathcal{Q} = \mathcal{Q}(\tau)$

An Illuminating Example: Ritz–Galerkin Method for the Poisson equation I

- Solve

$$\begin{cases} \Delta u = g, & u \in H^1_{per}(\Omega), \;\; g \in H^{-1}_{per}(\Omega) \\ u\big|_x = u\big|_{x+2\pi}, & \Delta \; : \; H^1_{per}(\Omega) \;\rightarrow\; H^{-1}_{per}(\Omega) \end{cases}$$

by minimizing the functional $\mathcal{J} \, : H^1_{per}(\Omega) \rightarrow \mathbb{R}$,

$$\mathcal{J}(\Phi) = \int_\Omega [(1/2)(\nabla\Phi)^2 + g\Phi] \, d\Omega$$

An Illuminating Example: Ritz–Galerkin Method for the Poisson equation I

- Solve

$$\begin{cases} \Delta u = g, & u \in H^1_{per}(\Omega), \ \ g \in H^{-1}_{per}(\Omega) \\ u\big|_x = u\big|_{x+2\pi}, & \Delta \ : \ H^1_{per}(\Omega) \ \rightarrow \ H^{-1}_{per}(\Omega) \end{cases}$$

by minimizing the functional $\mathcal{J} \ : H^1_{per}(\Omega) \ \rightarrow \ \mathbb{R}$,

$$\mathcal{J}(\Phi) = \int_\Omega [(1/2)(\boldsymbol{\nabla}\Phi)^2 + g\Phi] \, d\Omega$$

- The Gâteaux differential (the optimality condition)

$$\mathcal{J}(\Phi; \Phi') = \int_\Omega [-\Delta\Phi + g]\Phi' \, d\Omega = 0$$

An Illuminating Example: Ritz–Galerkin Method for the Poisson equation II

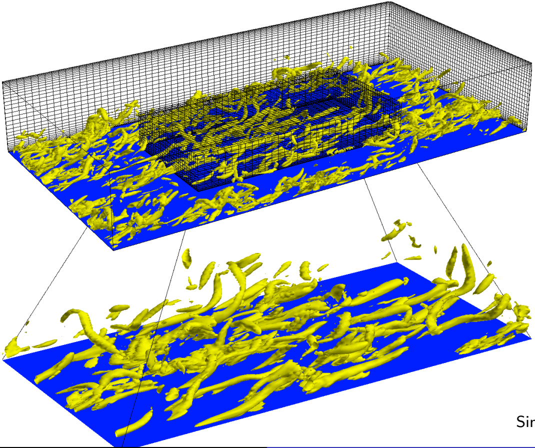
- Gradient in $L_2(\Omega)$: $\nabla^{L_2}\mathcal{J} = -\Delta\Phi + g \in L_2(\Omega)$
  - Hessian eigenvalues (Fourier space): $\{k_1^2, k_2^2, \dots, k_N^2\}$
  - The condition number: $\kappa = \frac{k_N^2}{k_1^2} \to \infty$ for $k_N \to \infty$

An Illuminating Example: Ritz–Galerkin Method for the Poisson equation II

- Gradient in $L_2(\Omega)$: $\nabla^{L_2}\mathcal{J} = -\Delta\Phi + g \in L_2(\Omega)$
  - Hessian eigenvalues (Fourier space): $\{k_1^2, k_2^2, \dots, k_N^2\}$
  - The condition number: $\kappa = \frac{k_N^2}{k_1^2} \to \infty$ for $k_N \to \infty$
- Gradient in $H_0^1(\Omega)$: $\nabla^{H^1}\mathcal{J} = -\Delta_0^{-1}[\Delta\Phi - g] \in H^1(\Omega)$
  - Hessian eigenvalues (Fourier space): $\{1, 1, \dots, 1\}$
  - The condition number: $\kappa = 1$ independent of k_N

Reconstruction of a Turbulent Channel Flow I

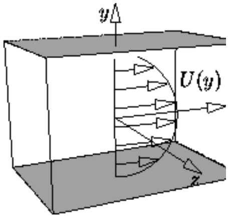
WALL SHEAR — the "footprint" of streaky structures in the boundary layer



Simulation: T. Bewley

Reconstruction of a Turbulent Channel Flow II

- Flow domain

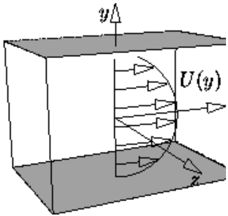


- Navier–Stokes system:

$$\begin{cases} \mathcal{N}(\mathbf{q}) = \begin{pmatrix} \frac{\partial u_j}{\partial x_j} \\ \frac{\partial u_i}{\partial t} + \frac{\partial u_j u_i}{\partial x_j} - \nu \frac{\partial u_i}{\partial x_j} + \frac{\partial p}{\partial x_i} \end{pmatrix} \\ \mathbf{u}|_{t=0} = \mathbf{\Phi} \ \text{ in } \Omega, \\ \mathbf{u}(0,y,z) = \mathbf{u}(2\pi L_x, y, z); \\ \mathbf{u}(x,y,0) = \mathbf{u}(x,y,2\pi L_z) \\ \mathbf{u}(x,\pm 1, x) = 0 \end{cases}$$

Reconstruction of a Turbulent Channel Flow II

- Flow domain



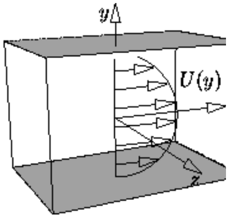
- Navier–Stokes system:

$$\begin{cases} \mathcal{N}(\mathbf{q}) = \begin{pmatrix} \frac{\partial u_j}{\partial x_j} \\ \frac{\partial u_i}{\partial t} + \frac{\partial u_j u_i}{\partial x_j} - \nu \frac{\partial u_i}{\partial x_j} + \frac{\partial p}{\partial x_i} \end{pmatrix} \\ \mathbf{u}|_{t=0} = \mathbf{\Phi} \ \text{ in } \Omega, \\ \mathbf{u}(0,y,z) = \mathbf{u}(2\pi L_x, y, z); \\ \mathbf{u}(x,y,0) = \mathbf{u}(x,y,2\pi L_z) \\ \mathbf{u}(x,\pm 1, x) = 0 \end{cases}$$

- constant mass flux
- turbulent flow at $Re_\tau = 100$

Reconstruction of a Turbulent Channel Flow II

- Flow domain



- wall shear and wall pressure measurements

- Navier–Stokes system:

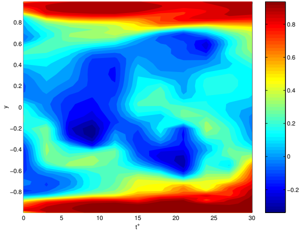
$$\begin{cases} \mathcal{N}(\mathbf{q}) = \begin{pmatrix} \frac{\partial u_j}{\partial x_j} \\ \frac{\partial u_i}{\partial t} + \frac{\partial u_j u_i}{\partial x_j} - \nu \frac{\partial u_i}{\partial x_j} + \frac{\partial p}{\partial x_i} \end{pmatrix} \\ \mathbf{u}|_{t=0} = \mathbf{\Phi} \quad \text{in } \Omega, \\ \mathbf{u}(0, y, z) = \mathbf{u}(2\pi L_x, y, z); \\ \mathbf{u}(x, y, 0) = \mathbf{u}(x, y, 2\pi L_z) \\ \mathbf{u}(x, \pm 1, x) = 0 \end{cases}$$

- constant mass flux
- turbulent flow at $Re_\tau = 100$

$$\mathcal{J}(\Phi) = \frac{1}{2}\int_0^T \left[\alpha_1 \left\| \frac{\partial u_1}{\partial x_2} - m_1 \right\|^2_{\Gamma_2^{\pm}} + \alpha_2 \left\| p - m_2 \right\|^2_{\Gamma_2^{\pm}} + \alpha_3 \left\| \frac{\partial u_3}{\partial x_2} - m_3 \right\|^2_{\Gamma_2^{\pm}} \right] dt,$$

Different strategies for gradient extraction

- L_2 Gradient Extraction

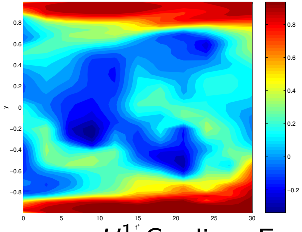


$$\mathcal{J}'(\boldsymbol{\Phi};\mathbf{h}) = -\int_\Omega \mathbf{u}^*\Big|_{t=0} \cdot \mathbf{h}\, d\Omega = \left(\boldsymbol{\nabla}\mathcal{J}^{L_2(\Omega)}, \mathbf{h}\right)_{L_2(\Omega)}$$

$$\Longrightarrow \boldsymbol{\nabla}\mathcal{J}^{L_2(\Omega)} = -\mathbf{u}^*\Big|_{t=0}$$

Different strategies for gradient extraction

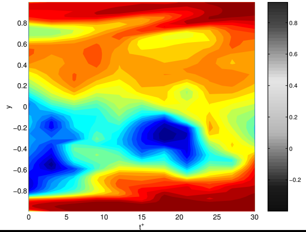
- L_2 Gradient Extraction



$$\mathcal{J}'(\boldsymbol{\Phi};\mathbf{h}) = -\int_\Omega \mathbf{u}^*\Big|_{t=0}\cdot\mathbf{h}\,d\Omega = \left(\boldsymbol{\nabla}\mathcal{J}^{L_2(\Omega)},\,\mathbf{h}\right)_{L_2(\Omega)}$$

$$\Longrightarrow \boldsymbol{\nabla}\mathcal{J}^{L_2(\Omega)} = -\mathbf{u}^*\Big|_{t=0}$$

- H^1 Gradient Extraction



$$\mathcal{J}'(\boldsymbol{\Phi};\mathbf{h}) = \frac{1}{1+l_1^2}\int_\Omega\left(\boldsymbol{\nabla}\mathcal{J}^{H^1}\cdot\mathbf{h} + l_1^2\,\partial_{\mathbf{x}}\boldsymbol{\nabla}\mathcal{J}^{H^1}\cdot\partial_{\mathbf{x}}\mathbf{h}\right)d\Omega$$

$$\Longrightarrow \begin{cases} \overbrace{\frac{1}{1+l_1^2}\left[1+l_1^2\,\Delta\right]}^{\text{Helmholtz operator}}\boldsymbol{\nabla}\mathcal{J}^{H^1} = -\mathbf{u}^*\Big|_{t=0} \\ \boldsymbol{\nabla}\mathcal{J}^{H^1}\Big|_\Gamma = 0 \end{cases}$$

Gradient Extraction in Banach Spaces — Theory (I)

- Consider a BANACH space $\mathbf{X}$ (without HILBERT structure!)

$$\mathcal{J}'(\varphi;\varphi') = \int_0^{2\pi} \varphi' \, v^*\big|_{t=0} \, dx = \langle \nabla^{\mathbf{X}} \mathcal{J}, \varphi' \rangle_{\mathbf{X}^* \times \mathbf{X}}, \quad \Longrightarrow \quad \boxed{\nabla^{\mathbf{X}} \mathcal{J} \in \mathbf{X}^*}$$

Note that $\mathbf{X}^*$ (the dual space) is usually "bigger" than $\mathbf{X}$
Hence $\nabla^{\mathbf{X}} \mathcal{J} \notin \mathbf{X}$ is not an acceptable descent direction !!!
No Riesz Theorem in Banach spaces ...

Gradient Extraction in Banach Spaces — Theory (I)

- Consider a BANACH space $\mathbf{X}$ (without HILBERT structure!)

$$\mathcal{J}'(\varphi;\varphi') = \int_0^{2\pi} \varphi'\, v^*\big|_{t=0}\, dx = \langle \nabla^{\mathbf{X}}\mathcal{J}, \varphi' \rangle_{\mathbf{X}^*\times\mathbf{X}}, \quad \Longrightarrow \quad \boxed{\nabla^{\mathbf{X}}\mathcal{J} \in \mathbf{X}^*}$$

  Note that $\mathbf{X}^*$ (the dual space) is usually "bigger" than $\mathbf{X}$
  Hence $\nabla^{\mathbf{X}}\mathcal{J} \notin \mathbf{X}$ is not an acceptable descent direction !!!
  No Riesz Theorem in Banach spaces ...

- Alternative definition of the descent direction g ensuring that $g \in \mathbf{X}$

$$g = \operatorname{argmax}_{\|\varphi'\|_{\mathbf{X}}=1} \langle \nabla^{\mathbf{X}}\mathcal{J}, \varphi' \rangle_{\mathbf{X}^*\times\mathbf{X}} = \operatorname{argmax}_{\varphi'} \left[\int_0^{2\pi} \varphi'\, v^*\big|_{t=0}\, dx + \mu \|\varphi'\|_{\mathbf{X}} \right]$$

Gradient Extraction in Banach Spaces — Theory (I)

- Consider a BANACH space $\mathbf{X}$ (without HILBERT structure!)

$$\mathcal{J}'(\varphi;\varphi') = \int_0^{2\pi} \varphi'\, v^*\big|_{t=0}\, dx = \langle \nabla^{\mathbf{X}}\mathcal{J}, \varphi' \rangle_{\mathbf{X}^*\times\mathbf{X}}, \quad \Longrightarrow \quad \boxed{\nabla^{\mathbf{X}}\mathcal{J} \in \mathbf{X}^*}$$

  Note that $\mathbf{X}^*$ (the dual space) is usually "bigger" than $\mathbf{X}$
  Hence $\nabla^{\mathbf{X}}\mathcal{J} \notin \mathbf{X}$ is not an acceptable descent direction !!!
  No Riesz Theorem in Banach spaces ...

- Alternative definition of the descent direction g ensuring that $g \in \mathbf{X}$

$$g = \operatorname{argmax}_{\|\varphi'\|_{\mathbf{X}}=1} \langle \nabla^{\mathbf{X}}\mathcal{J}, \varphi' \rangle_{\mathbf{X}^*\times\mathbf{X}} = \operatorname{argmax}_{\varphi'} \left[\int_0^{2\pi} \varphi'\, v^*\big|_{t=0}\, dx + \mu\|\varphi'\|_{\mathbf{X}} \right]$$

- For instance, when $\mathbf{X} = W^{p,q}(\Omega)$ with $\|z\|_{W^{p,q}} = \int_0^{2\pi} |z|^q + l_p^q|\partial_x^p u|^q\, dx$,

$$\begin{cases} p|g|^{(p-2)}g + p\partial_x^q\left(|\partial_x^q g|^{(p-2)}\partial_x^q g\right) = -v^*\big|_{t=0} \\ \partial_x^m g\big|_{x=0} = \partial_x^m g\big|_{x=2\pi} = 0 \end{cases}$$

p–LAPLACE equation

Gradient Extraction in Banach Spaces — Theory (II)

- $\varphi \in \mathbf{X}$, but due to "Bumpy Landscape" in $\mathbf{X}$, descent may be slow

Gradient Extraction in Banach Spaces — Theory (II)

- $\varphi \in \mathbf{X}$, but due to "Bumpy Landscape" in $\mathbf{X}$, descent may be slow
- Try to smooth out the landscape via a change of variables (topology), but one change of variables may not work well for all iterates

Gradient Extraction in Banach Spaces — Theory (II)

- $\varphi \in \mathbf{X}$, but due to "Bumpy Landscape" in $\mathbf{X}$, descent may be slow
- Try to smooth out the landscape via a change of variables (topology), but one change of variables may not work well for all iterates
- Shorten the path $\varphi(\tau)$ by confining it to a Family of NESTED SPACES

$$\mathcal{Q}^{(0)} \subseteq \mathcal{Q}^{(1)} \subseteq \cdots \subseteq \mathcal{Q}^{(k)} \subseteq \cdots \subseteq \mathcal{U}.$$

Gradient Extraction in Banach Spaces — Theory (II)

- $\varphi \in \mathbf{X}$, but due to "Bumpy Landscape" in $\mathbf{X}$, descent may be slow
- Try to smooth out the landscape via a change of variables (topology), but one change of variables may not work well for all iterates
- Shorten the path $\varphi(\tau)$ by confining it to a Family of Nested Spaces

$$\mathcal{Q}^{(0)} \subseteq \mathcal{Q}^{(1)} \subseteq \cdots \subseteq \mathcal{Q}^{(k)} \subseteq \cdots \subseteq \mathcal{U}.$$

- Preconditioned Gradients ensure that $\varphi^k \in \mathcal{Q}^k$

Gradient Extraction in Banach Spaces — Theory (II)

- $\varphi \in \mathbf{X}$, but due to "Bumpy Landscape" in $\mathbf{X}$, descent may be slow
- Try to smooth out the landscape via a change of variables (topology), but one change of variables may not work well for all iterates
- Shorten the path $\varphi(\tau)$ by confining it to a Family of NESTED SPACES

$$\mathcal{Q}^{(0)} \subseteq \mathcal{Q}^{(1)} \subseteq \cdots \subseteq \mathcal{Q}^{(k)} \subseteq \cdots \subseteq \mathcal{U}.$$

- PRECONDITIONED GRADIENTS ensure that $\varphi^k \in \mathcal{Q}^k$
- Example choice of nested spaced (Lebesgue spaces):

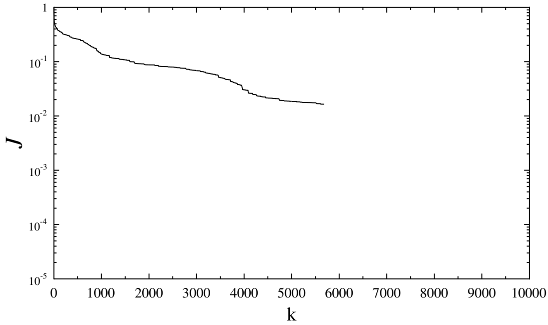
$$L_{p_1} \subseteq L_{p_2} \subseteq \cdots \subseteq L_{p_k} \subseteq \cdots \subseteq L_2,$$

where $p_1 > p_2 > \cdots > p_k > \cdots > 2$.

Gradient Extraction in Banach Spaces — Results

Results for the Kuramoto–Sivashinsky Equation: tough problem with very long optimization horizon

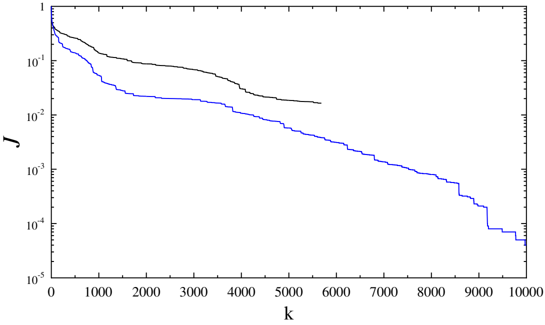
- $\nabla^{L_2} \mathcal{J}$ (classical gradients)



Gradient Extraction in Banach Spaces — Results

Results for the Kuramoto–Sivashinsky Equation: tough problem with very long optimization horizon

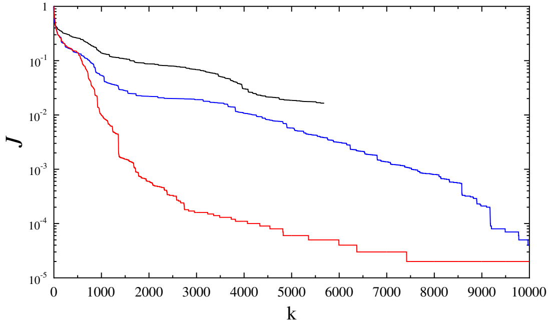
- $\nabla^{L_2} \mathcal{J}$ (classical gradients)
- $\nabla^{L_{10}} \mathcal{J} \underset{k \to \infty}{\longrightarrow} \nabla^{L_2} \mathcal{J}$ (Lebesgue gradient)



Gradient Extraction in Banach Spaces — Results

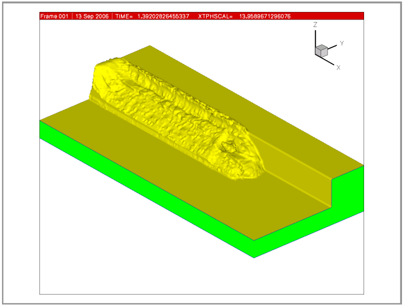
Results for the Kuramoto–Sivashinsky Equation: tough problem with very long optimization horizon

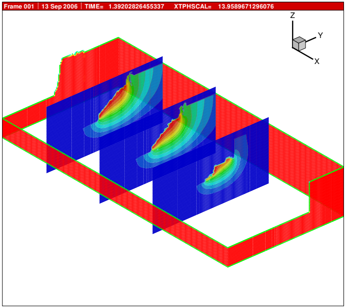
- $\nabla^{L_2}\mathcal{J}$ (classical gradients)
- $\nabla^{L_{10}}\mathcal{J} \xrightarrow[k\to\infty]{} \nabla^{L_2}\mathcal{J}$ (Lebesgue gradient)
- $\nabla^{B^s_{p,q}}\mathcal{J} \xrightarrow[k\to\infty]{} \nabla^{L_2}\mathcal{J}$ (Besov gradients)



Free–Surface Flows in a Weld Pool (I)

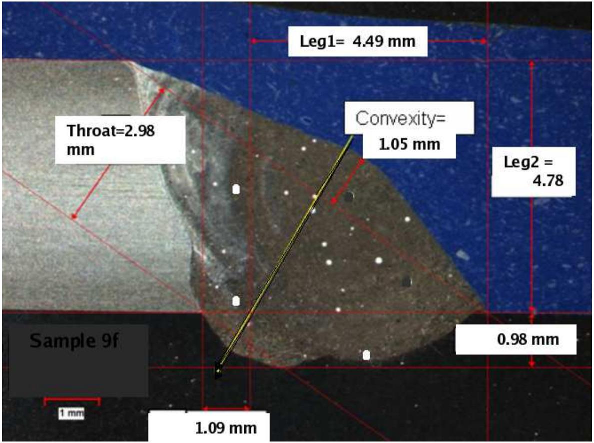
- Motivation: Welding in Automotive Industry





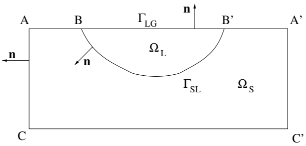
Free–Surface Flows in a Weld Pool (II)

- Goal: Optimize Shape of Free Surface During Solidification



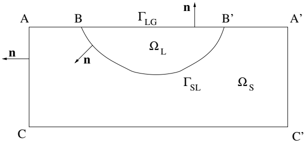
Stefan Problem in the Presence of Contact Points (I)

- Domain



Stefan Problem in the Presence of Contact Points (I)

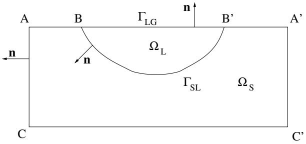
- Domain



- Governing Equations

$$-\nabla \cdot (k_S \nabla T) = 0 \qquad \text{in } \Omega_S,$$
$$-\nabla \cdot (k_L \nabla T) = 0 \qquad \text{in } \Omega_L.$$

Stefan Problem in the Presence of Contact Points (I)



- Domain
- Governing Equations

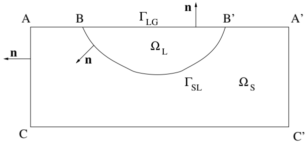
$$-\nabla \cdot (k_S \nabla T) = 0 \qquad \text{in } \Omega_S,$$
$$-\nabla \cdot (k_L \nabla T) = 0 \qquad \text{in } \Omega_L.$$

- Interface Conditions
  - (conservation of energy)

$$\left[k \frac{\partial T}{\partial n} \right]_S^L = 0 \qquad \text{on } \Gamma_{SL},$$

Stefan Problem in the Presence of Contact Points (I)

- Domain



- Governing Equations

$$-\nabla\cdot(k_S\nabla T)=0 \qquad \text{in } \Omega_S,$$
$$-\nabla\cdot(k_L\nabla T)=0 \qquad \text{in } \Omega_L.$$

- Interface Conditions
  - (conservation of energy)

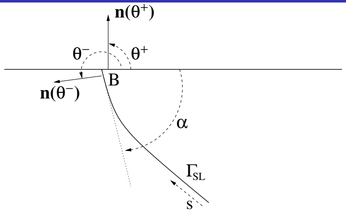
$$\left[k\frac{\partial T}{\partial n}\right]_S^L = 0 \qquad \text{on } \Gamma_{SL},$$

  - (second principle of thermodynamics)

$$T = T_m \qquad \text{on } \Gamma_{SL}.$$

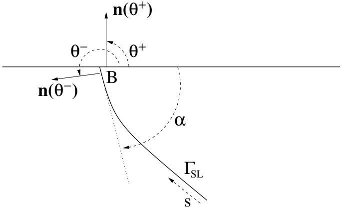
Stefan Problem in the Presence of Contact Points (II)

- Domains with "corners"



Stefan Problem in the Presence of Contact Points (II)

- Domains with "corners"



- The temperature interface condition corresponds to an inequality, and hence is nonunique

$$L\frac{T - T_m}{T_m} = \varkappa\left[f(\theta) + \frac{d^2 f(\theta)}{d\theta^2}\right] \quad \text{on the smooth part of } \Gamma_{SL},$$

$$\mathbf{C}(\theta^+) = \mathbf{C}(\theta^-) \quad \text{at the contact points } B \text{ and } B',$$

The interfacial free energy $f(\theta)$ and capillary force $\mathbf{C}(\theta)$ determined at the microscopic level and not available ...

Stefan Problem in the Presence of Contact Points (III)

- Stefan Problem as an PDE Optimization (inverse) problem

$$\min_{\Gamma_{SL}} \mathcal{J}(\Gamma_{SL}), \qquad \text{where}$$

$$\mathcal{J}(\Gamma_{SL}) \triangleq \frac{1}{2}\int_{\Gamma_{SL}} \left[T(\Gamma_{SL}) - T_m\right]^2 \, ds + \frac{\ell}{2}\left[\cos(\alpha(\Gamma_{SL})) - \cos(\alpha_m)\right]^2\Big|_{B,B'}$$

The contact angle α_m is a constitutive property of the material.

Stefan Problem in the Presence of Contact Points (III)

- Stefan Problem as an PDE Optimization (inverse) problem

$$\min_{\Gamma_{SL}} \mathcal{J}(\Gamma_{SL}), \qquad \text{where}$$

$$\mathcal{J}(\Gamma_{SL}) \triangleq \frac{1}{2}\int_{\Gamma_{SL}} \left[T(\Gamma_{SL}) - T_m\right]^2 \, ds + \frac{\ell}{2}\left[\cos(\alpha(\Gamma_{SL})) - \cos(\alpha_m)\right]^2\Big|_{B,B'}$$

  The contact angle α_m is a constitutive property of the material.
- Shape Optimization problem

Stefan Problem in the Presence of Contact Points (III)

- Stefan Problem as an PDE Optimization (inverse) problem

$$\min_{\Gamma_{SL}} \mathcal{J}(\Gamma_{SL}), \qquad \text{where}$$

$$\mathcal{J}(\Gamma_{SL}) \triangleq \frac{1}{2}\int_{\Gamma_{SL}} [T(\Gamma_{SL}) - T_m]^2 \, ds + \frac{\ell}{2}\left[\cos(\alpha(\Gamma_{SL})) - \cos(\alpha_m)\right]^2\Big|_{B,B'}$$

  The contact angle α_m is a constitutive property of the material.
- Shape Optimization problem
- Parametrization of geometry

$$\mathbf{x}(t,\mathbf{Z}) = \mathbf{x} + t\mathbf{Z} \qquad \text{for } \mathbf{x} \in \Gamma_{SL}(0),$$

  where $\mathbf{Z} : \Omega_{SL} \rightarrow \mathbb{R}^2$ is the perturbation "velocity" field.

Stefan Problem in the Presence of Contact Points (III)

- Gâteaux shape differential

$$\mathcal{J}'(\Gamma_{SL}(0);\mathbf{Z}) \triangleq \lim_{t\longrightarrow 0} \frac{\mathcal{J}(\Gamma_{SL}(t,\mathbf{Z})) - \mathcal{J}(\Gamma_{SL}(0))}{t}.$$

Stefan Problem in the Presence of Contact Points (III)

- Gâteaux shape differential

$$\mathcal{J}'(\Gamma_{SL}(0);\mathbf{Z}) \triangleq \lim_{t\rightarrow 0} \frac{\mathcal{J}(\Gamma_{SL}(t,\mathbf{Z})) - \mathcal{J}(\Gamma_{SL}(0))}{t}.$$

- L_2 gradient $\boldsymbol{\nabla}^{L_2}\mathcal{J}$ not smooth enough

$$\begin{aligned}\boldsymbol{\nabla}^{L^2}\mathcal{J} = & \left[\left[k\frac{\partial T}{\partial s}\frac{\partial T^*}{\partial s}\right]_S^L - \left[k\frac{\partial T}{\partial n}\frac{\partial T^*}{\partial n}\right]_S^L + \varkappa\frac{(T-T_m)^2}{2}\right]\mathbf{n}+ \\ & \left[T^*\left(\varphi_{LG}-\varphi_{SG}\right)\mathbf{e}_x + \frac{(T-T_m)^2}{2}\tau + \right. \\ & \left. + \varkappa\,\ell\,[\cos(\alpha)-\cos(\alpha_m)]\,\sin(\alpha)\,\tau\right][\delta(s-s_{B'})-\delta(s-s_B)]+ \\ & \ell\,[\cos(\alpha)-\cos(\alpha_m)]\,\sin(\alpha)\left[\dot{\delta}(s-s_{B'})-\dot{\delta}(s-s_B)\right]\mathbf{n} \quad \text{on } \Gamma_{SL},\end{aligned}$$

Stefan Problem in the Presence of Contact Points (III)

- Gâteaux shape differential

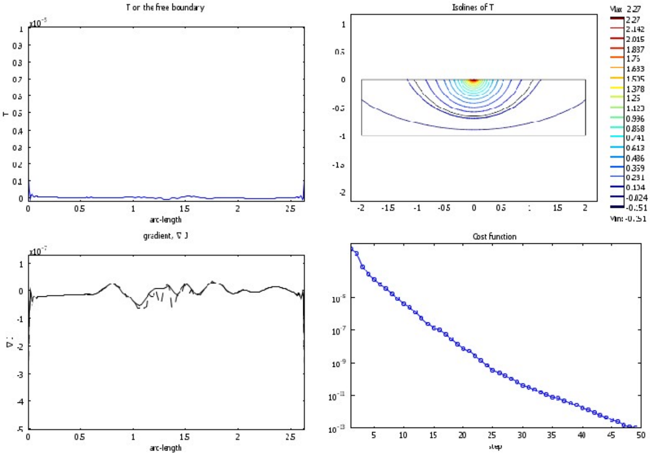
$$\mathcal{J}'(\Gamma_{SL}(0);\mathbf{Z}) \triangleq \lim_{t\rightarrow 0} \frac{\mathcal{J}(\Gamma_{SL}(t,\mathbf{Z})) - \mathcal{J}(\Gamma_{SL}(0))}{t}.$$

- L_2 gradient $\boldsymbol{\nabla}^{L_2}\mathcal{J}$ not smooth enough

$$\begin{aligned}\boldsymbol{\nabla}^{L^2}\mathcal{J} = & \left[\left[k\frac{\partial T}{\partial s}\frac{\partial T^*}{\partial s}\right]_S^L - \left[k\frac{\partial T}{\partial n}\frac{\partial T^*}{\partial n}\right]_S^L + \varkappa\frac{(T-T_m)^2}{2}\right]\mathbf{n} + \\ & \left[T^*\left(\varphi_{LG}-\varphi_{SG}\right)\mathbf{e}_x + \frac{(T-T_m)^2}{2}\tau + \right. \\ & \left. + \varkappa\,\ell\,[\cos(\alpha)-\cos(\alpha_m)]\,\sin(\alpha)\,\tau\right][\delta(s-s_{B'})-\delta(s-s_B)] + \\ & \ell\,[\cos(\alpha)-\cos(\alpha_m)]\,\sin(\alpha)\left[\dot\delta(s-s_{B'})-\dot\delta(s-s_B)\right]\mathbf{n} \quad \text{on } \Gamma_{SL},\end{aligned}$$

- Must work with smoother H^1/H^2 gradients.

Stefan Problem in the Presence of Contact Points (IV)



Simple Model Problem

- $$\begin{cases} \partial_t u - \Delta u = 0 & \text{in} \quad \Omega(\phi) = [a(\phi), b(\phi)], \\ \partial_x u\big|_{a(\phi)} = \phi, \quad \partial_x u\big|_{b(\phi)} = 0, \\ u\big|_{a(\phi)} = u\big|_{b(\phi)} = u_b, \\ + \text{Initial Condition} \end{cases}$$

  where:

Simple Model Problem

- $$\begin{cases} \partial_t u - \Delta u = 0 & \text{in} \quad \Omega(\phi) = [a(\phi), b(\phi)], \\ \partial_x u\big|_{a(\phi)} = \phi, \quad \partial_x u\big|_{b(\phi)} = 0, \\ u\big|_{a(\phi)} = u\big|_{b(\phi)} = u_b, \\ + \text{Initial Condition} \end{cases}$$

  where:

  - the control variable: $\phi = \phi(t)$

Simple Model Problem

- $$\begin{cases} \partial_t u - \Delta u = 0 & \text{in} \quad \Omega(\phi) = [a(\phi), b(\phi)], \\ \partial_x u\big|_{a(\phi)} = \phi, \quad \partial_x u\big|_{b(\phi)} = 0, \\ u\big|_{a(\phi)} = u\big|_{b(\phi)} = u_b, \\ + \text{Initial Condition} \end{cases}$$

  where:

  - the control variable: $\phi = \phi(t)$
  - the cost functional: $\mathcal{J}(\phi) = \int_0^T [b(\phi) - \overline{b}]^2 \, dt$

Simple Model Problem

- $$\begin{cases} \partial_t u - \Delta u = 0 & \text{in} \quad \Omega(\phi) = [a(\phi), b(\phi)], \\ \partial_x u\big|_{a(\phi)} = \phi, \quad \partial_x u\big|_{b(\phi)} = 0, \\ u\big|_{a(\phi)} = u\big|_{b(\phi)} = u_b, \\ + \text{INITIAL CONDITION} \end{cases}$$

  where:

  - the control variable: $\phi = \phi(t)$
  - the cost functional: $\mathcal{J}(\phi) = \int_0^T [b(\phi) - \overline{b}]^2 \, dt$
  - solution: $u = u(\Omega(\phi))$

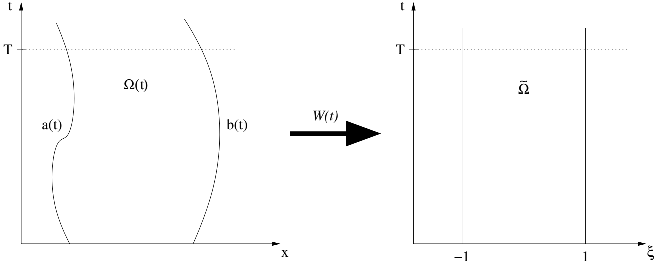
Simple Model Problem

- $$\begin{cases} \partial_t u - \Delta u = 0 & \text{in } \Omega(\phi) = [a(\phi), b(\phi)], \\ \partial_x u\big|_{a(\phi)} = \phi, \quad \partial_x u\big|_{b(\phi)} = 0, \\ u\big|_{a(\phi)} = u\big|_{b(\phi)} = u_b, \\ + \text{INITIAL CONDITION} \end{cases}$$

  where:
  - the control variable: $\phi = \phi(t)$
  - the cost functional: $\mathcal{J}(\phi) = \int_0^T [b(\phi) - \overline{b}]^2 \, dt$
  - solution: $u = u(\Omega(\phi))$
- Note that the model problem is GEOMETRICALLY NONLINEAR

Two Options

1. Optimization after transformation to a FIXED DOMAIN

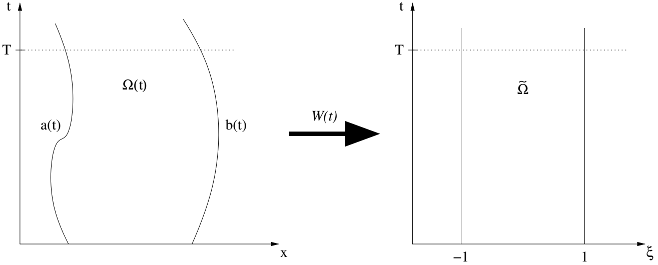


$$L(t) \triangleq b(t) - a(t), \qquad x_0(t) \triangleq \frac{a(t)+b(t)}{2},$$

$$x = x(t,\xi) = \frac{L(t)}{2}\xi + x_0(t), \qquad \tilde{u}(t,\xi) = u(t, x(t,\xi))$$

Two Options

1. Optimization after transformation to a FIXED DOMAIN



$$L(t) \triangleq b(t) - a(t), \qquad x_0(t) \triangleq \frac{a(t)+b(t)}{2},$$

$$x = x(t,\xi) = \frac{L(t)}{2}\xi + x_0(t), \qquad \tilde{u}(t,\xi) = u(t, x(t,\xi))$$

2. Optimization in a VARIABLE DOMAIN

Optimization in Fixed Domains (I)

- Geometric vs. Algebraic nonlinearity

Optimization in Fixed Domains (I)

- Geometric vs. Algebraic nonlinearity
- The Governing System $\{\tilde{u}, L, x_0\}$

$$
\begin{aligned}
&\frac{\partial \tilde{u}}{\partial t} - \frac{\partial \tilde{u}}{\partial \xi}\frac{2\dot{x}_0 + \xi \dot{L}}{L} - \frac{4\nu}{L^2}\frac{\partial^2 \tilde{u}}{\partial \xi^2} = 0 && \text{in} \quad (0,T] \times [-1,1],\\
&\left.\frac{\partial \tilde{u}}{\partial \xi}\right|_{-1} = \frac{L}{2}\phi, \quad \left.\frac{\partial \tilde{u}}{\partial \xi}\right|_{1} = \frac{L}{2} w && \text{in} \quad (0,T],\\
&\tilde{u}\big|_{-1} = \tilde{u}\big|_{1} = u_b && \text{in} \quad (0,T],\\
&\tilde{u}\big|_{t=0} = \tilde{u}_0 && \text{in} \quad [-1,1],
\end{aligned}
$$

Optimization in Fixed Domains (I)

- Geometric vs. Algebraic nonlinearity
- The Governing System $\{\tilde{u}, L, x_0\}$

$$
\begin{aligned}
&\frac{\partial \tilde{u}}{\partial t} - \frac{\partial \tilde{u}}{\partial \xi}\frac{2\dot{x}_0 + \xi \dot{L}}{L} - \frac{4\nu}{L^2}\frac{\partial^2 \tilde{u}}{\partial \xi^2} = 0 && \text{in } \ (0,T] \times [-1,1], \\
&\left.\frac{\partial \tilde{u}}{\partial \xi}\right|_{-1} = \frac{L}{2}\phi, \quad \left.\frac{\partial \tilde{u}}{\partial \xi}\right|_{1} = \frac{L}{2} w && \text{in } \ (0,T], \\
&\tilde{u}\big|_{-1} = \tilde{u}\big|_{1} = u_b && \text{in } \ (0,T], \\
&\tilde{u}\big|_{t=0} = \tilde{u}_0 && \text{in } \ [-1,1],
\end{aligned}
$$

- The Cost Functional

$$
\mathcal{J}(\phi) = \frac{1}{2}\int_0^T \left[x_0(t) + \frac{L(t)}{2} - \overline{b}(t) \right]^2 dt.
$$

Optimization in Fixed Domains (II)

- Adjoint System for The Model Problem $\{\tilde{u}^*, \tilde{a}^*, \tilde{b}^*\}$

$$-\frac{\partial \tilde{u}^*}{\partial t} + \frac{\dot{L}}{L}\tilde{u}^* + \frac{\partial \tilde{u}^*}{\partial \xi}\frac{2\dot{x}_0 + \xi\dot{L}}{L} - \frac{4\nu}{L^2}\frac{\partial^2 \tilde{u}^*}{\partial \xi^2} = 0 \qquad \text{in } (0, T] \times [-1, 1],$$

$$\tilde{u}^*\big|_{-1} = -\frac{L^2}{4\nu}\tilde{a}^*, \quad \tilde{u}^*\big|_{1} = -\frac{L^2}{4\nu}\tilde{b}^* \qquad \text{in } (0, T],$$

$$\int_{-1}^{1}\left[\frac{d}{dt}\left(\frac{\xi}{L}\frac{\partial \tilde{u}}{\partial \xi}\tilde{u}^*\right) + \frac{2\dot{x}_0 + \xi\dot{L}}{L^2}\frac{\partial \tilde{u}}{\partial \xi}\tilde{u}^* + \frac{8\nu}{L^3}\frac{\partial^2 \tilde{u}}{\partial \xi^2}\tilde{u}^*\right] d\xi - \frac{\phi}{2}\tilde{a}^* = = \frac{1}{2}\left(x_0 + \frac{L}{2} - \overline{b}\right) \qquad \text{in } (0, T],$$

$$\int_{-1}^{1}\frac{d}{dt}\left(\frac{2}{L}\frac{\partial \tilde{u}}{\partial \xi}\tilde{u}^*\right) d\xi = x_0 + \frac{L}{2} - \overline{b} \qquad \text{in } (0, T],$$

$$\tilde{u}^*\big|_{t=T} = 0 \qquad \text{in } [-1, 1],$$

$$\tilde{a}^*\big|_{t=T} = 0, \quad \tilde{b}^*\big|_{t=T} = 0$$

Note the presence of NONLOCAL CONSTRAINT

Optimization in Fixed Domains (II)

- Adjoint System for The Model Problem $\{\tilde{u}^*, \tilde{a}^*, \tilde{b}^*\}$

$$-\frac{\partial \tilde{u}^*}{\partial t} + \frac{\dot{L}}{L}\tilde{u}^* + \frac{\partial \tilde{u}^*}{\partial \xi}\frac{2\dot{x}_0 + \xi \dot{L}}{L} - \frac{4\nu}{L^2}\frac{\partial^2 \tilde{u}^*}{\partial \xi^2} = 0 \qquad \text{in } (0, T] \times [-1, 1],$$

$$\tilde{u}^*\big|_{-1} = -\frac{L^2}{4\nu}\tilde{a}^*, \quad \tilde{u}^*\big|_{1} = -\frac{L^2}{4\nu}\tilde{b}^* \qquad \text{in } (0, T],$$

$$\int_{-1}^{1}\left[\frac{d}{dt}\left(\frac{\xi}{L}\frac{\partial \tilde{u}}{\partial \xi}\tilde{u}^*\right) + \frac{2\dot{x}_0 + \xi \dot{L}}{L^2}\frac{\partial \tilde{u}}{\partial \xi}\tilde{u}^* + \frac{8\nu}{L^3}\frac{\partial^2 \tilde{u}}{\partial \xi^2}\tilde{u}^*\right] d\xi - \frac{\phi}{2}\tilde{a}^* = = \frac{1}{2}\left(x_0 + \frac{L}{2} - \overline{b}\right) \qquad \text{in } (0, T],$$

$$\int_{-1}^{1}\frac{d}{dt}\left(\frac{2}{L}\frac{\partial \tilde{u}}{\partial \xi}\tilde{u}^*\right) d\xi = x_0 + \frac{L}{2} - \overline{b} \qquad \text{in } (0, T],$$

$$\tilde{u}^*\big|_{t=T} = 0 \qquad \text{in } [-1, 1],$$

$$\tilde{a}^*\big|_{t=T} = 0, \quad \tilde{b}^*\big|_{t=T} = 0$$

Note the presence of NONLOCAL CONSTRAINT

- Cost Functional Gradient for the Model Problem

$$\nabla \mathcal{J} = \frac{L}{2}\tilde{a}^* \quad \text{in } [0, T]$$

Optimization in Variable Domains (I)

- Space–Time Tube: $Q \triangleq \bigcup_{t\in[0,T]}\{t\} \times \Omega(t)$
- Flow Map $\mathcal{T}(t)$ characterizes domain evolution $\Omega(t) = \mathcal{T}(t)\Omega_0$

Optimization in Variable Domains (I)

- Space–Time Tube: $Q \triangleq \bigcup_{t\in[0,T]} \{t\} \times \Omega(t)$
- Flow Map $\mathcal{T}(t)$ characterizes domain evolution $\Omega(t) = \mathcal{T}(t)\Omega_0$
- Parameterize the Flow Map using Velocity Field V

$$\begin{cases} \dfrac{\partial \mathcal{T}(t,x)}{\partial t} = V(t, \mathcal{T}(t,x)), & t \in (0, T], \\ \mathcal{T}(0,x) = x, & \text{in } \overline{\Omega}(0). \end{cases}$$

Optimization in Variable Domains (I)

- Space–Time Tube: $Q \triangleq \bigcup_{t\in[0,T]}\{t\}\times\Omega(t)$
- Flow Map $\mathcal{T}(t)$ characterizes domain evolution $\Omega(t) = \mathcal{T}(t)\Omega_0$
- Parameterize the Flow Map using Velocity Field V

$$\begin{cases} \dfrac{\partial \mathcal{T}(t,x)}{\partial t} = V(t,\mathcal{T}(t,x)), & t\in(0,T], \\ \mathcal{T}(0,x) = x, & \text{in } \overline{\Omega}(0). \end{cases}$$

- Differentiation of functions w. r. t. to evolution of the domain parameterized by velocity V, i.e., $\Omega = \Omega(V(t))$ in the direction of velocity $W(t)$

$$u'(V;W) \triangleq \overbrace{\frac{d}{d\epsilon}[u(V+\rho W)\circ\mathcal{T}_\rho]\Big|_{\rho=0}}^{\dot{u}(V;W)} - (\nabla u)Z(W), \quad \text{where}$$

$\mathcal{T}_\rho$ — the transverse map, $Z(W)$ — the transverse variable

Optimization in Variable Domains (II)

- Adjoint System for the Model Problem

$$
\begin{cases}
-\partial_t u^* - \Delta u^* = 0 & \text{in} \quad \Omega(\phi) = [a(\phi), b(\phi)], \\
\partial_x u^* = 0 & \text{at} \quad x = a(\phi), \\
\partial_x u^* = \dfrac{[b(\phi) - \overline{b}]}{\partial_x u} & \text{at} \quad x = b(\phi), \\
u^* = 0 & \text{at} \quad t = T
\end{cases}
$$

- Cost Functional Gradient for the Model Problem

$$
\boxed{\nabla \mathcal{J}(\phi) = u^*|_a}
$$

Optimization in Variable Domains (II)

- Adjoint System for the Model Problem

$$\begin{cases} -\partial_t u^* - \Delta u^* = 0 & \text{in} \quad \Omega(\phi) = [a(\phi), b(\phi)], \\ \partial_x u^* = 0 & \text{at} \quad x = a(\phi), \\ \partial_x u^* = \dfrac{[b(\phi) - \overline{b}]}{\partial_x u} & \text{at} \quad x = b(\phi), \\ u^* = 0 & \text{at} \quad t = T \end{cases}$$

- Cost Functional Gradient for the Model Problem

$$\boxed{\nabla \mathcal{J}(\phi) = u^*|_a}$$

- Remarks
  - the same gradient direction, but a different expression, as for the adjoint obtained in a fixed domain

Optimization in Variable Domains (II)

- Adjoint System for the Model Problem

$$
\begin{cases}
-\partial_t u^* - \Delta u^* = 0 & \text{in} \quad \Omega(\phi) = [a(\phi), b(\phi)], \\
\partial_x u^* = 0 & \text{at} \quad x = a(\phi), \\
\partial_x u^* = \dfrac{[b(\phi) - \overline{b}]}{\partial_x u} & \text{at} \quad x = b(\phi), \\
u^* = 0 & \text{at} \quad t = T
\end{cases}
$$

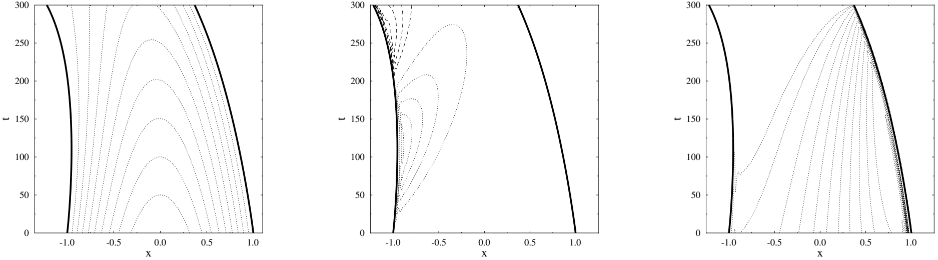
- Cost Functional Gradient for the Model Problem

$$
\boxed{\nabla \mathcal{J}(\phi) = u^*|_a}
$$

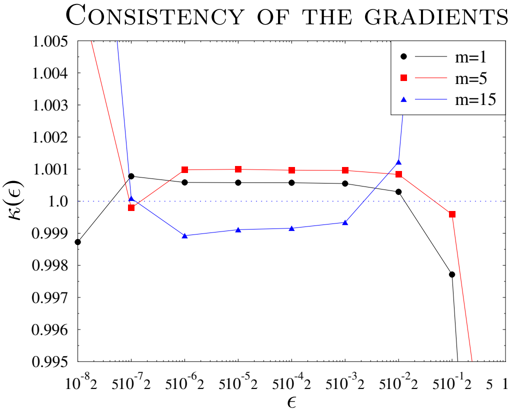
- Remarks
  - the same gradient direction, but a different expression, as for the adjoint obtained in a fixed domain
  - transformation to a fixed domain and derivation of the adjoint system do not commute

Optimization in Variable Domains (III) — Results

Reference u, perturbation u' and adjoint u^* fields
(Computations performed with the variable–domain adjoint)



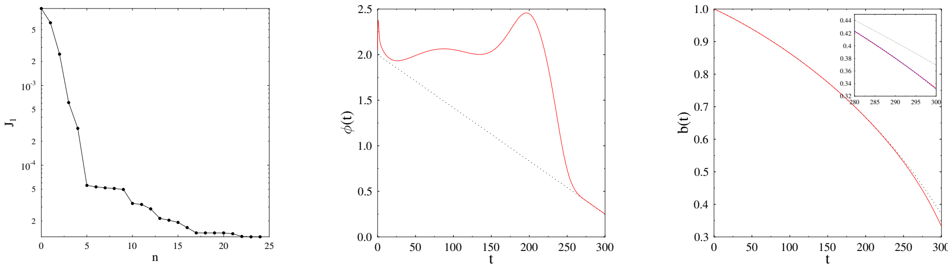
Optimization in Variable Domains (IV) — Results



$$\kappa(\epsilon) \triangleq \frac{\mathcal{J}(\phi + \epsilon\phi') - \mathcal{J}(\phi)}{(\nabla\mathcal{J}, \phi')}, \quad \text{where} \quad \phi'(t) = \sin\left(m 2\pi \frac{t}{T}\right)$$

Optimization in Variable Domains (V) — Results

Cost Functional $\mathcal{J}$, Control ϕ and Output $b(\phi)$



Conclusions

- Formulation of PDE control and estimation problems as constrained optimization

Conclusions

- Formulation of PDE control and estimation problems as constrained optimization
- PDE–constrained gradients via Adjoint Equations
  - Preconditioning: linear and nonlinear

Conclusions

- Formulation of PDE control and estimation problems as constrained optimization
- PDE–constrained gradients via Adjoint Equations
  - Preconditioning: linear and nonlinear
- Optimization of PDEs in Variable Domains
  - Free–boundary (Stefan) problem as an optimization problems
  - Time–Dependent variable domains

Conclusions

- Formulation of PDE control and estimation problems as constrained optimization
- PDE–constrained gradients via Adjoint Equations
  - Preconditioning: linear and nonlinear
- Optimization of PDEs in Variable Domains
  - Free–boundary (Stefan) problem as an optimization problems
  - Time–Dependent variable domains

References

- O. Volkov and B. Protas, "An inverse model for a free–boundary problem with a contact line: steady case", (submitted), 2008.
- B. Protas, "Adjoint–Based Optimization of PDE Systems with Alternative Gradients" *Journal of Computational Physics* (in press), 2008.
- B. Protas and W. Liao, "Adjoint–Based Optimization of PDEs in Moving Domains" *Journal of Computational Physics* **227** 2707–2723, 2008.
- B. Protas, W. Liao and D. Glander, "A Framework for Gradient–Based Optimization of Welding Processes" (submitted), 2007.
- T. R. Bewley and B. Protas, "Skin friction and pressure: the "footprints" of turbulence", *Physica D* **196**(1-2), 28-44, 2004.
- B. Protas, T. R. Bewley and G. Hagen, "A comprehensive framework for the regularization of adjoint analysis in multiscale PDE systems", *Journal of Computational Physics* **195**(1), 49-89, 2004.
- B. Protas, "On the "Vorticity" Formulation of the Adjoint Equations and its Solution Using Vortex Method", *Journal of Turbulence* **3**, 048, 2002.
- B. Protas and A. Styczek, "Optimal Rotary Control of the Cylinder Wake in the Laminar Regime", *Physics of Fluids* **14**(7), 2073–2087, 2002.