PDE Constrained Optimization

Matthias Heinkenschloss

Department of Computational and Applied Mathematics
Rice University, Houston, Texas
heinken@rice.edu

July 18, 2008

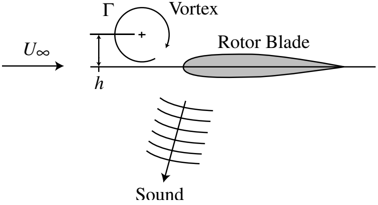
London Mathematical Society Durham Symposium
Computational Linear Algebra for Partial Differential Equations
Monday 14th July - Thursday 24th July 2008

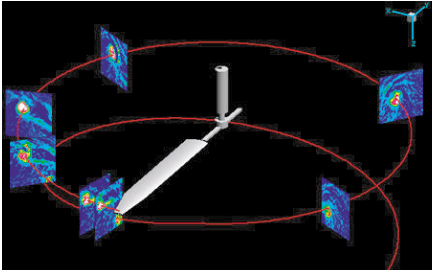
Outline

Blade Vortex Interaction (BVI)

with S.S. Collis (Sandia), K. Ghayour (Adv. Scientific)

- Trailing vortex from preceeding blade interacts with following blade, generating unsteady lift and dipole sound source
- Severe, impulsive sound radiated toward ground

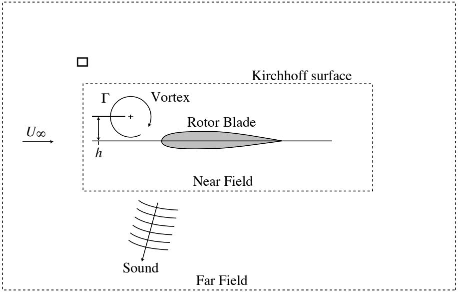




PIV measurements of a hovering rotor (P.B. Martin, Univ. Maryland)

On-Blade Control of BVI Noise

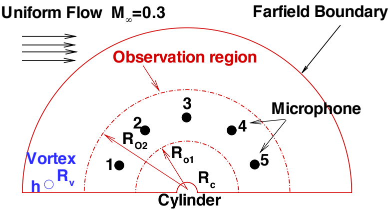
Can on-blade actuators be used to reduce BVI generated noise?
Formulate as optimization problem which couples to complex flow simulation.

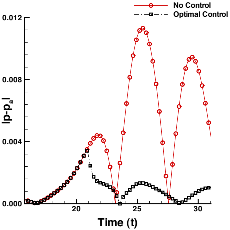


Minimize pressure fluctuations in a far field region through suction/blowing on the rotor blade.

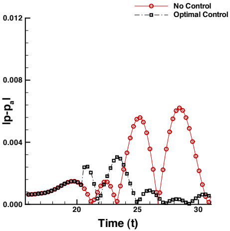
Pressure is computed by solving the unsteady compressible Navier Stokes equations with boundary data for velocities given by suction/blowing control.

Results for a 2D model problem

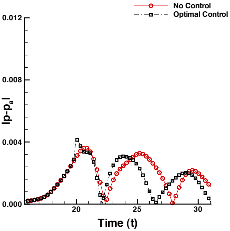




Mic 1



Mic 2

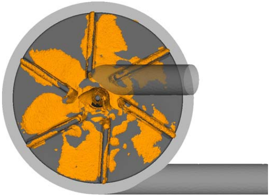


Mic 3

Shape Optimization

with M. Behr (RWTH Aachen), F. Abraham (GlaxoSmithKline)

Shear-stress distribution in PI 710 Centrifugal Pump



Cannula Shape

Minimize $J(\mathbf{u}, p, \alpha)$,

subject to

$$
\begin{aligned}
\mathbf{u} \cdot \boldsymbol{\nabla} \mathbf{u} - \boldsymbol{\nabla} \cdot \boldsymbol{\sigma}(\mathbf{u}, p) &= \mathbf{0} \quad \text{on } \Omega(\alpha), \\
\boldsymbol{\nabla} \cdot \mathbf{u} &= 0 \quad \text{on } \Omega(\alpha), \\
\mathbf{n} \cdot \boldsymbol{\sigma}(\mathbf{u}, p) &= \mathbf{h} \quad \text{on } \Gamma_1(\alpha), \\
\mathbf{u} &= \mathbf{g} \quad \text{on } \Gamma_2(\alpha),
\end{aligned}
$$

where

$$
\begin{aligned}
\boldsymbol{\sigma}(\mathbf{u}, p) &= -p\mathbf{I} + 2\mu\boldsymbol{\varepsilon}(\mathbf{u}), \\
\boldsymbol{\varepsilon}(\mathbf{u}) &= \tfrac{1}{2}\left(\boldsymbol{\nabla}\mathbf{u} + \boldsymbol{\nabla}\mathbf{u}^T\right), \\
\mu(\dot\gamma) &= \mu_\infty + \frac{\mu_0 - \mu_\infty}{(1 + (\lambda\dot\gamma)^b)^a}, \\
\dot\gamma &= \sqrt{2\boldsymbol{\varepsilon}(\mathbf{u}) : \boldsymbol{\varepsilon}(\mathbf{u})}.
\end{aligned}
$$

(Oil) Reservoir Management

with A. El Bakry and K. D. Wiegand (ExxonMobil)

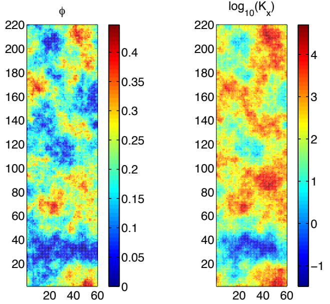
We consider an incompressible oil-water system described by a coupled system of nonlinear, time-dependent partial differential equations (PDEs)

$$-\nabla \cdot \Big(K(x) \lambda_t \big(s_w(x,t) \big) \nabla p(x,t) \Big) = q_o(x,t) + q_w(x,t),$$

$$\phi(x) \frac{d}{dt} s_w(x,t) - \nabla \cdot \Big(K(x) \lambda_w (s_w(x,t)) \nabla p(x,t) \Big) = q_w(x,t),$$

$x \in \Omega, t \in (0,T)$, for the pressure p and the water saturation s_w, combined with boundary and initial conds.

s_w :	Water Saturation
p :	Pressure
K :	Absolute Permeability
λ_t, λ_w :	Phase Mobilities
ϕ :	Rock Porosity
q_w, q_o :	Well Sources/Sinks for Water/Oil

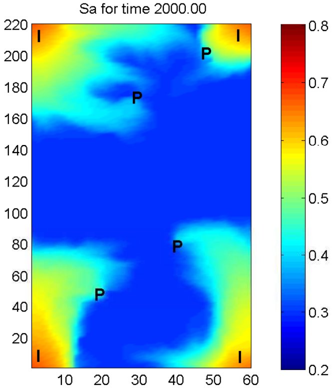


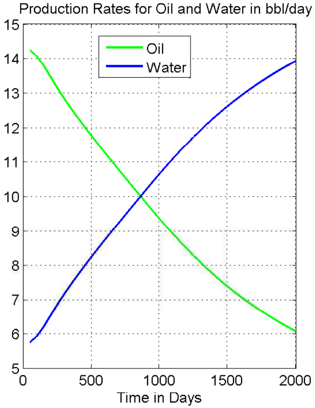
Porositiy Permeability

(top layer of the SPE10 model 2)

Simulation Result

Four injection wells and four production wells.
Water is injected into all four injection wells at a constant rate.



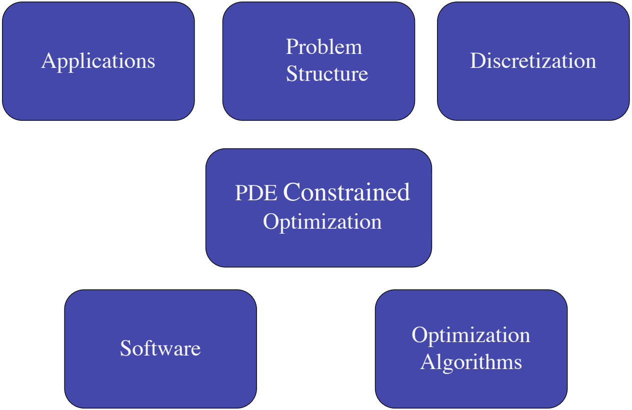


Much of the oil gets trapped!
Need optimization to determine injection/production rates.

Characteristics of PDE Constrained Optimization Problems

- All problems are PDE constrained optimization problems - there are many, many more.
- Evaluation of objective function and constraint functions involves expensive simulations (in the previous examples solution of partial differential equations (PDEs)).
- THE optimization problem does not exist. Instead each problem leads to a family of optimization problems which are closely linked.
- The robust and efficient solution of such optimization problems requires the integration of application specific structure, numerical simulation and optimization algorithms.

Need to look at the big picture, not only at one component



Optimization Approach

- Selection of a suitable optimization algorithm depends on the properties of the optimization problem, properties of the PDE simulator, ...

Optimization Approach

- Selection of a suitable optimization algorithm depends on the properties of the optimization problem, properties of the PDE simulator, ...
- I will discuss (a few) issues in PDE constrained optimization that arise from the presence of the PDE (in contrast to 'just' large-scale finite dimensional constraints). I will focus on CLAPDE and interfaces, but not on optimization algorithms.

Optimization Approach

- Selection of a suitable optimization algorithm depends on the properties of the optimization problem, properties of the PDE simulator, ...
- I will discuss (a few) issues in PDE constrained optimization that arise from the presence of the PDE (in contrast to 'just' large-scale finite dimensional constraints). I will focus on CLAPDE and interfaces, but not on optimization algorithms.
- My research focusses on derivative based, Newton-type algorithms for PDE constrained optimization problems.

Optimization Approach

- Selection of a suitable optimization algorithm depends on the properties of the optimization problem, properties of the PDE simulator, ...
- I will discuss (a few) issues in PDE constrained optimization that arise from the presence of the PDE (in contrast to 'just' large-scale finite dimensional constraints). I will focus on CLAPDE and interfaces, but not on optimization algorithms.
- My research focusses on derivative based, Newton-type algorithms for PDE constrained optimization problems.
  - + many control variables/parameters u,
  - + fast convergence,
  - + often mesh independent convergence behavior,
  - + efficiency from integration of optimization and simulation,

Optimization Approach

- Selection of a suitable optimization algorithm depends on the properties of the optimization problem, properties of the PDE simulator, ...
- I will discuss (a few) issues in PDE constrained optimization that arise from the presence of the PDE (in contrast to 'just' large-scale finite dimensional constraints). I will focus on CLAPDE and interfaces, but not on optimization algorithms.
- My research focusses on derivative based, Newton-type algorithms for PDE constrained optimization problems.
  - \+ many control variables/parameters u,
  - \+ fast convergence,
  - \+ often mesh independent convergence behavior,
  - \+ efficiency from integration of optimization and simulation,
  - − require insight into simulator.

Outline

Abstract Optimization Problem

$$
\begin{aligned}
\min\ & J(y,u) \\
\text{s.t. } & c(y,u) = 0, && \text{(the governing PDE)} \\
& g(y,u) = 0, && \text{(additional equality constr.)} \\
& h(y,u) \in -K && \text{(additional inequality constr.)} \\
& y \in \mathcal{Y}_{ad}, u \in \mathcal{U}_{ad}.
\end{aligned}
$$

where

- $(\mathcal{Y}, \|\cdot\|_{\mathcal{Y}})$, $(\mathcal{U}, \|\cdot\|_{\mathcal{U}})$, $(\mathcal{C}, \|\cdot\|_{\mathcal{C}})$ are real Banach spaces,
- $(\mathcal{H}, \|\cdot\|_{\mathcal{H}})$ is a real normed space,
- $\mathcal{Y}_{ad} \subset \mathcal{Y}$, $\mathcal{U}_{ad} \subset \mathcal{U}$ are nonempty, closed convex sets,
- $K \subset \mathcal{H}$ is a nonempty, closed convex cone,
- $J : \mathcal{Y} \times \mathcal{U} \to \mathbb{R}$, $c : \mathcal{Y} \times \mathcal{U} \to \mathcal{C}$, $h : \mathcal{Y} \times \mathcal{U} \to \mathcal{H}$ are smooth mappings.

Notation:
y: states, $\mathcal{Y}$: state space, u: controls, $\mathcal{U}$: control space,
$c(y,u) = 0$ state equation.

Problem Formulation

$$\begin{aligned} \min \quad & J(y,u) \\ \text{s.t.} \quad & c(y,u) = 0, \\ & g(y,u) = 0, \\ & h(y,u) \in -K \end{aligned}$$

$$\Downarrow$$

$y(u)$ is the unique solution of $c(y,u) = 0$

$$\Downarrow$$

$$\left.\begin{aligned} \min \quad & \widehat{J}(u) \\ \text{s.t.} \quad & \widehat{g}(u) = 0, \\ & \widehat{h}(u) \in -K, \end{aligned}\right\} \text{reduced problem}$$

where $\widehat{J}(u) \stackrel{\text{def}}{=} J(y(u),u)$, $\widehat{g}(u) \stackrel{\text{def}}{=} g(y(u),u)$, $\widehat{h}(u) \stackrel{\text{def}}{=} h(y(u),u)$.

- The full and the reduced order problems are closely related.

$$\begin{array}{ll} \min_{y,u} & J(y,u) \\ \text{s.t.} & c(y,u) = 0. \end{array}$$

$$\Downarrow$$

$$\min_u \ \widehat{J}(u) \stackrel{\text{def}}{=} J(y(u), u).$$

For example, the gradient of the reduced functional is given by

$$\nabla \widehat{J}(u) = \nabla_u L(y, u, \lambda)\big|_{y=y(u), \lambda=\lambda(u)},$$

where

$$L(y, u, \lambda) = J(y, u) + \langle \lambda, c(y, u) \rangle$$

is the Lagrangian of the constrained problem and $\lambda(u)$ is the solution of the adjoint equation (see later).

- The reduced problem formulation is often used, but it is not always clear that it can be used.

  For example, the problem

$$
\begin{aligned}
\text{minimize} \quad & \frac{1}{2}\int_\Omega (y(x) - \hat{y}(x))^2 dx + \frac{\alpha}{2}\int_{\partial\Omega} u^2(x) ds, \\
\text{subject to} \quad & -\Delta y(x) = l(x) \text{ in } \Omega, \\
& \frac{\partial}{\partial n} y(x) = u(x) \text{ on } \partial\Omega
\end{aligned}
$$

  is well-posed and has a unique solution, but for given u the state equation does not have a solution or it has infinitely many solutions.

- The reduced problem formulation is often used, but it is not always clear that it can be used.

  For example, the problem

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2}\int_\Omega (y(x)-\hat{y}(x))^2 dx + \frac{\alpha}{2}\int_{\partial\Omega} u^2(x) ds, \\ \text{subject to} \quad & -\Delta y(x) = l(x) \text{ in } \Omega, \\ & \frac{\partial}{\partial n} y(x) = u(x) \text{ on } \partial\Omega \end{aligned}$$

  is well-posed and has a unique solution, but for given u the state equation does not have a solution or it has infinitely many solutions.

- In practice the equation $c(y,u) = 0$ cannot be solved exactly. Only an approximation $y_\epsilon(u)$ of $y(u)$ can be computed such that, e.g., $\|c(y_\epsilon(u),u)\| < \epsilon$ for some user determined parameter ϵ.

  Thus the functions $\widehat{J}(u) \stackrel{\text{def}}{=} J(y(u),u)$, $\widehat{g}(u) \stackrel{\text{def}}{=} g(y(u),u)$, $\widehat{h}(u) \stackrel{\text{def}}{=} h(y(u),u)$ (and their derivatives) are never available. (More on this later).

Optimality Conditions

- Recall the optimization problem in Banach spaces

$$
\begin{aligned}
\min\ & J(y,u) \\
\text{s.t. } & c(y,u) = 0, && \text{(the governing PDE)} \\
& h(y,u) \in -K && \text{(additional inequality constr.)} \\
& y \in \mathcal{Y}_{ad}, u \in \mathcal{U}_{ad}.
\end{aligned}
$$

Optimality Conditions

- Recall the optimization problem in Banach spaces

$$\begin{aligned} \min\ & J(y,u) \\ \text{s.t. } & c(y,u) = 0, && \text{(the governing PDE)} \\ & h(y,u) \in -K && \text{(additional inequality constr.)} \\ & y \in \mathcal{Y}_{ad}, u \in \mathcal{U}_{ad}. \end{aligned}$$

- This is a generalization of well-known (finite dimensional) nonlinear programs of the type

$$\begin{aligned} \min\ & J(y,u) \\ \text{s.t. } & c(y,u) = 0, \\ & h(y,u) \le 0 \\ & y \in [y_{\text{low}}, y_{\text{up}}]^{n_y}, u \in [u_{\text{low}}, u_{\text{up}}]^{n_u}. \end{aligned}$$

Optimality Conditions

- Recall the optimization problem in Banach spaces

$$\begin{aligned} &\min J(y,u) \\ &\text{s.t. } c(y,u) = 0, && \text{(the governing PDE)} \\ &\qquad h(y,u) \in -K && \text{(additional inequality constr.)} \\ &\qquad y \in \mathcal{Y}_{ad}, u \in \mathcal{U}_{ad}. \end{aligned}$$

- This is a generalization of well-known (finite dimensional) nonlinear programs of the type

$$\begin{aligned} &\min J(y,u) \\ &\text{s.t. } c(y,u) = 0, \\ &\qquad h(y,u) \le 0 \\ &\qquad y \in [y_{\text{low}}, y_{\text{up}}]^{n_y}, u \in [u_{\text{low}}, u_{\text{up}}]^{n_u}. \end{aligned}$$

- One can derive Karush-Kuhn-Tucker (KKT) type optimality conditions (see, e.g., Zowe/Kurcyusz (1979) and the books by J. Jahn (1996), J. Werner (1984), D. G. Luenberger (1969)) ...

Karush-Kuhn-Tucker Theorem in Banach Spaces

- ... but the Lagrange multipliers are not vectors in $\mathbb{R}^m$, but functionals.

Karush-Kuhn-Tucker Theorem in Banach Spaces

- ... but the Lagrange multipliers are not vectors in $\mathbb{R}^m$, but functionals.
- Lagrangian

$$L(y, u, \lambda, \mu) = J(y, u) + \lambda \circ c(y, u) + \mu \circ h(y, u).$$

Karush-Kuhn-Tucker Theorem in Banach Spaces

- ... but the Lagrange multipliers are not vectors in $\mathbb{R}^m$, but functionals.
- Lagrangian

$$L(y, u, \lambda, \mu) = J(y, u) + \lambda \circ c(y, u) + \mu \circ h(y, u).$$

- If (y_*, u_*) is a local minimizer and if a regularity condition (CQ) holds, then there exist continuous linear functionals (Lagrange multipliers) $\lambda_* \in \mathcal{C}^*$,

$$\mu_* \in K^* \equiv \{\ell \in \mathcal{H}^* : \ell(v) \geq 0 \text{ for all } v \in K\}$$

  such that

$$\begin{aligned}\Big(D_y J(y_*, u_*) + \lambda_* \circ D_y c(y_*, u_*) + \mu_* \circ D_y h(y_*, u_*)\Big)(y - y_*) &\geq 0,\\ \Big(D_u J(y_*, u_*) + \lambda_* \circ D_u c(y_*, u_*) + \mu_* \circ D_u h(y_*, u_*)\Big)(u - u_*) &\geq 0,\\ \mu_*(h(y_*, u_*)) &= 0\end{aligned}$$

  for all $y \in \mathcal{Y}_{ad}$, $u \in \mathcal{U}_{ad}$.

- For finite dimensional nonlinear programs this reduces to

$$L(y,u,\lambda,\mu) = J(y,u) + \lambda^T c(y,u) + \mu^T h(y,u).$$

If (y_*, u_*) is a local minimizer and if a regularity condition (CQ) holds, then there exist $\lambda_* \in \mathbb{R}^m$, $\mu_* \in \mathbb{R}^k$, $\mu_* \geq 0$,

$$\begin{aligned}
\Big(\nabla_y J(y_*,u_*) + D_y c(y_*,u_*)^T \lambda_* + D_y h(y_*,u_*)^T \lambda_*\Big)^T (y - y_*) &\geq 0, \\
\Big(\nabla_u J(y_*,u_*) + D_y c(y_*,u_*)^T \lambda_* + D_u h(y_*,u_*)^T \lambda_*\Big)^T (u - u_*) &\geq 0, \\
\mu_*^T h(y_*,u_*) &= 0
\end{aligned}$$

for all $y \in [y_{\rm low}, y_{\rm up}]^{n_y}$, $u \in [u_{\rm low}, u_{\rm up}]^{n_u}$.

- For finite dimensional nonlinear programs this reduces to

$$L(y, u, \lambda, \mu) = J(y, u) + \lambda^T c(y, u) + \mu^T h(y, u).$$

If (y_*, u_*) is a local minimizer and if a regularity condition (CQ) holds, then there exist $\lambda_* \in \mathbb{R}^m$, $\mu_* \in \mathbb{R}^k$, $\mu_* \geq 0$,

$$\begin{aligned} \Big(\nabla_y J(y_*, u_*) + D_y c(y_*, u_*)^T \lambda_* + D_y h(y_*, u_*)^T \lambda_*\Big)^T (y - y_*) &\geq 0, \\ \Big(\nabla_u J(y_*, u_*) + D_y c(y_*, u_*)^T \lambda_* + D_u h(y_*, u_*)^T \lambda_*\Big)^T (u - u_*) &\geq 0, \\ \mu_*^T h(y_*, u_*) &= 0 \end{aligned}$$

for all $y \in [y_{\text{low}}, y_{\text{up}}]^{n_y}$, $u \in [u_{\text{low}}, u_{\text{up}}]^{n_u}$.

- The KKT Theorem is a good guideline, but applying it to PDE constrained optimization is difficult.
  - The choice of state and control spaces are important.
  - Precise characterization of Lagrange multipliers is important for design and analysis of optimization algorithms.
  - Precise characterization of Lagrange mult. requires (a lot of) work.
  - Optimality conditions for optimal control problems with control and state constraints have been derived by Casas, Bonnans, Kunisch, Bergounioux, Raymond, Tröltzsch,.....

Example 1 (Only PDE Constraint)

$$\begin{array}{ll} \text{Minimize} & \dfrac{1}{2}\displaystyle\int_\Omega (y(x)-\hat{y}(x))^2 dx + \dfrac{\alpha}{2}\int_\Omega u^2(x)dx, \\ \text{subject to} & -\Delta y(x) = u(x) + l(x),\ x \in \Omega, \quad y(x) = 0\ x \in \partial\Omega. \end{array}$$

Example 1 (Only PDE Constraint)

$$\text{Minimize} \quad \frac{1}{2}\int_\Omega (y(x)-\hat{y}(x))^2 dx + \frac{\alpha}{2}\int_\Omega u^2(x)dx,$$
$$\text{subject to} \quad -\Delta y(x) = u(x) + l(x),\ x \in \Omega, \quad y(x) = 0\ x \in \partial\Omega.$$

- $\mathcal{Y} = H_0^1(\Omega)$, $\mathcal{U} = L^2(\Omega)$.
- $J(y,u) = \frac{1}{2}\int_\Omega (y(x)-\hat{y}(x))^2 dx + \frac{\alpha}{2}\int_\Omega u^2(x)dx$,
- $c : H_0^1(\Omega) \times L^2(\Omega) \to H^{-1}(\Omega)$, where

$$\langle c(y,u), \phi\rangle_{\mathcal{Y}',\mathcal{Y}} = \int_\Omega \nabla y \nabla \phi dx - \int_\Omega u\phi dx - \int_\Omega l\phi dx.$$

- $L(y,u,\lambda) = \frac{1}{2}\int_\Omega (y-\hat{y})^2 dx + \frac{\alpha}{2}\int_\Omega u^2 dx + \int_\Omega \nabla y \nabla \lambda dx - \int_\Omega u\lambda + l\lambda dx.$

Example 1 (Only PDE Constraint)

$$\begin{aligned} \text{Minimize} \quad & \frac{1}{2}\int_\Omega (y(x)-\hat{y}(x))^2 dx + \frac{\alpha}{2}\int_\Omega u^2(x)dx, \\ \text{subject to} \quad & -\Delta y(x) = u(x) + l(x),\ x\in\Omega, \quad y(x) = 0\ x \in \partial\Omega. \end{aligned}$$

- $\mathcal{Y} = H_0^1(\Omega)$, $\mathcal{U} = L^2(\Omega)$.
- $J(y,u) = \frac{1}{2}\int_\Omega (y(x)-\hat{y}(x))^2 dx + \frac{\alpha}{2}\int_\Omega u^2(x)dx$,
- $c : H_0^1(\Omega)\times L^2(\Omega) \to H^{-1}(\Omega)$, where

$$\langle c(y,u),\phi\rangle_{\mathcal{Y}',\mathcal{Y}} = \int_\Omega \nabla y \nabla \phi dx - \int_\Omega u\phi dx - \int_\Omega l\phi dx.$$

- $L(y,u,\lambda) = \frac{1}{2}\int_\Omega (y-\hat{y})^2 dx + \frac{\alpha}{2}\int_\Omega u^2 dx + \int_\Omega \nabla y \nabla \lambda dx - \int_\Omega u\lambda + l\lambda dx.$

If $(y_*,u_*) \in H_0^1 \times L^2$ is a local min., there exists $\lambda_* \in H_0^1$ such that

$$\begin{aligned} -\Delta\lambda_*(x) &= -(y_*(x)-\widehat{y}(x)), && x\in\Omega, \\ \lambda_*(x) &= 0 && x \in \partial\Omega, \\ \alpha u_*(x) - \lambda_*(x) &= 0 && \text{a.e. in } \Omega. \end{aligned}$$

Example 1 (Only PDE Constraint)

$$\text{Minimize} \quad \frac{1}{2}\int_\Omega (y(x)-\hat{y}(x))^2 dx + \frac{\alpha}{2}\int_\Omega u^2(x)dx,$$
$$\text{subject to} \quad -\Delta y(x) = u(x) + l(x),\ x\in\Omega, \quad y(x)=0\ x\in\partial\Omega.$$

- $\mathcal{Y} = H_0^1(\Omega)$, $\mathcal{U} = L^2(\Omega)$.
- $J(y,u) = \frac{1}{2}\int_\Omega (y(x)-\hat{y}(x))^2 dx + \frac{\alpha}{2}\int_\Omega u^2(x)dx$,
- $c: H_0^1(\Omega)\times L^2(\Omega)\to H^{-1}(\Omega)$, where

$$\langle c(y,u),\phi\rangle_{\mathcal{Y}',\mathcal{Y}} = \int_\Omega \nabla y\nabla\phi dx - \int_\Omega u\phi dx - \int_\Omega l\phi dx.$$

- $L(y,u,\lambda) = \frac{1}{2}\int_\Omega (y-\hat{y})^2 dx + \frac{\alpha}{2}\int_\Omega u^2 dx + \int_\Omega \nabla y\nabla\lambda dx - \int_\Omega u\lambda + l\lambda dx.$

If $(y_*, u_*) \in H_0^1\times L^2$ is a local min., there exists $\lambda_*\in H_0^1$ such that

$$\begin{aligned} -\Delta\lambda_*(x) &= -(y_*(x)-\widehat{y}(x)), && x\in\Omega,\\ \lambda_*(x) &= 0 && x\in\partial\Omega,\\ \alpha u_*(x) - \lambda_*(x) &= 0 && \text{a.e. in } \Omega. \end{aligned}$$

Optimality conditions involve another linear PDE, the adjoint PDE.

Example 2 (Pointwise Control Constraints)

$$\begin{aligned}
\text{Minimize} \quad & \frac{1}{2}\int_\Omega (y(x)-\hat{y}(x))^2 dx + \frac{\alpha}{2}\int_\Omega u^2(x)dx, \\
\text{subject to} \quad & -\Delta y(x) = u(x) + l(x),\ x \in \Omega, \quad y(x) = 0\ x \in \partial\Omega, \\
& u_{\text{low}}(x) \le u(x) \le u_{\text{up}}(x) \text{ a.e. in } \Omega.
\end{aligned}$$

If $(y_*, u_*) \in H_0^1 \times L^2$ is a local minimizer, then there exist $\lambda_* \in H_0^1$ and $\mu_{low,*}, \mu_{up,*} \in L^2$, with $\mu_{low,*}, \mu_{up,*} \ge 0$ a.e. in Ω such that

$$\begin{aligned}
-\Delta\lambda_*(x) &= -(y_*(x) - \widehat{y}(x)), && x \in \Omega, \\
\lambda_*(x) &= 0 && x \in \partial\Omega, \\
\alpha u_*(x) - \lambda_*(x) - \mu_{low,*}(x) + \mu_{up,*}(x) &= 0, && \text{a.e. in } \Omega, \\
\int_\Omega (u_{low,*} - u_*)\mu_{low,*} dx &= \int_\Omega (u_* - u_{up,*})\mu_{up,*} dx = 0.
\end{aligned}$$

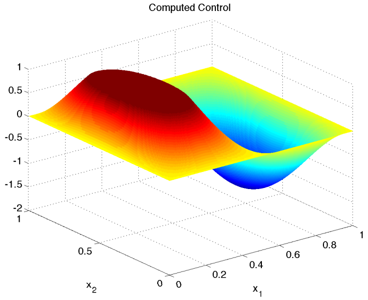
Lagrange multipliers corresponding to pointwise control constraints are L^2 functions.

Example 2 (Pointwise Control Constraints)

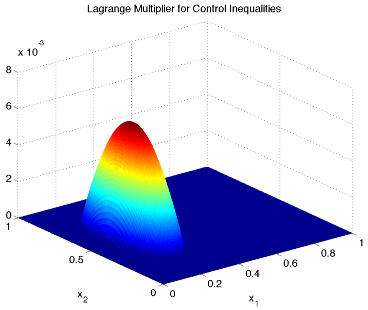
$$\text{Minimize} \quad \frac{1}{2}\int_\Omega (y(x)-\hat{y}(x))^2 dx + \frac{\alpha}{2}\int_\Omega u^2(x)dx,$$

$$\text{subject to} \quad -\Delta y(x) = u(x),\ x\in\Omega, \quad y(x)=0\ x\in\partial\Omega,$$

$$u(x)\le 1 \text{ a.e. in } \Omega.$$



u_*



$\mu_{up,*}$

Example 3 (Pointwise State Constraints)

$$\begin{aligned} \text{Minimize} \quad & \frac{1}{2}\int_\Omega (y(x)-\hat{y}(x))^2 dx + \frac{\alpha}{2}\int_\Omega u^2(x)dx, \\ \text{subject to} \quad & -\Delta y(x) = u(x) + l(x),\ x \in \Omega, \quad y(x) = 0\ x \in \partial\Omega, \\ & y_{\text{low}}(x) \le y(x) \le y_{\text{up}}(x) \text{ a.e. in } \Omega. \end{aligned}$$

- Need more regular states y to make sense out of $y_{\text{low}}(x) \le y(x) \le y_{\text{up}}(x)$ a.e. in Ω. Require $y \in C(\overline{\Omega})$.
- Lagrange multipliers $\nu_{low,*}, \nu_{up,*}$ are in $C(\overline{\Omega})^*$, i.e., are measures.
- Optimality conditions

$$\begin{aligned} -\Delta\lambda_* &= -(y_* - \widehat{y}) + \nu_{up,*} - \nu_{low,*}, && x \in \Omega, \\ \lambda_* &= 0 && x \in \partial\Omega, \\ \alpha u_* - \lambda_* &= 0, && \text{a.e. in } \Omega. \end{aligned}$$

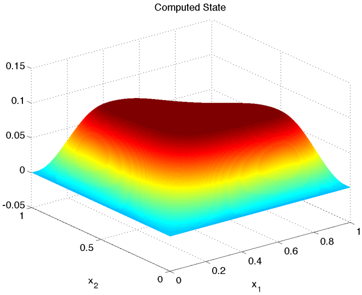
$$\int_\Omega (y_{low,*} - y_*)d\nu_{low,*} = \int_\Omega (y_* - y_{up,*})d\nu_{up,*} = 0.$$

Adjoint equation involves measures on the right hand side.

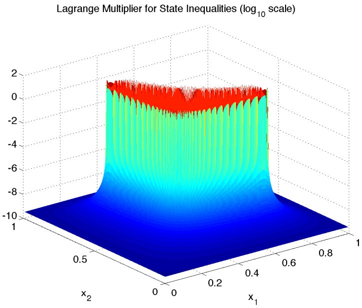
- Often, more can be said about the structure of $\nu_{up,*}$, $\nu_{low,*}$.

Example 3 (Pointwise State Constraints)

$$\begin{aligned} &\text{Minimize} \quad \frac{1}{2}\int_\Omega (y(x) - \sin(2\pi x_1 x_2))^2 dx + \frac{\alpha}{2}\int_\Omega u^2(x)dx, \\ &\text{subject to} \quad -\Delta y(x) = u(x),\ x \in \Omega, \quad y(x) = 0\ x \in \partial\Omega, \\ &\qquad\qquad\quad y(x) \le 0.1 \text{ a.e. in } \Omega. \end{aligned}$$



y_*



$\nu_{up,*}$

Optimization Algorithms

- Handling pointwise control and especially state constraints is difficult.
- PDE constrained optimization problems have motivated many algorithms or modifications of algorithms (semismooth Newton methods, interior point methods, primal-dual active set methods, regularization methods for state constrained problems). Convergence analyses are available for infinite dimensional problems, but often only for small classes of problems (especially when state constraints are present).
- Interior-point methods for large-scale finite dimensional problems also work well (almost mesh independent behavior), but there is no supporting theory.
- Most of the computing time in these algorithms is spent on the solution of KKT (optimality saddle point) systems.
  - Need matrix free KKT system solvers.
  - These are used in optimization context (detection of negative curvature).
  - Solvers need to be insensitive to penalty/regularization/barrier parameters, as well as to mesh size.

Discretization

- We want to solve

$$
\begin{aligned}
&\min J(y,u) \\
&\text{s.t. } c(y,u) = 0, && \text{(the governing PDE)} \\
&\qquad g(y,u) = 0, && \text{(additional equality constr.)}
\end{aligned}
$$

where $\mathcal{Y}$,$\mathcal{U}$,$\mathcal{C}$,$\mathcal{G}$,$\mathcal{H}$ are Banach spaces, $K \subset \mathcal{H}$ is a cone, and

$$
\begin{aligned}
J &: \mathcal{Y} \times \mathcal{U} \to \mathbb{R}, & c &: \mathcal{Y} \times \mathcal{U} \to \mathcal{C}, \\
g &: \mathcal{Y} \times \mathcal{U} \to \mathcal{G}, & h &: \mathcal{Y} \times \mathcal{U} \to \mathcal{H},
\end{aligned}
$$

Discretization

- We want to solve

$$
\begin{aligned}
&\min J(y,u) \\
&\text{s.t. } c(y,u) = 0, && \text{(the governing PDE)} \\
&\qquad g(y,u) = 0, && \text{(additional equality constr.)}
\end{aligned}
$$

where $\mathcal{Y}, \mathcal{U}, \mathcal{C}, \mathcal{G}, \mathcal{H}$ are Banach spaces, $K \subset \mathcal{H}$ is a cone, and

$$
\begin{aligned}
J &: \mathcal{Y} \times \mathcal{U} \to \mathbb{R}, & c &: \mathcal{Y} \times \mathcal{U} \to \mathcal{C}, \\
g &: \mathcal{Y} \times \mathcal{U} \to \mathcal{G}, & h &: \mathcal{Y} \times \mathcal{U} \to \mathcal{H},
\end{aligned}
$$

- but we can only solve a discretization

$$
\begin{aligned}
\min \quad & J_h(y_h, u_h) \\
\text{s.t.} \quad & c_h(y_h, u_h) = 0, \\
& g_h(y_h, u_h) = 0,
\end{aligned}
$$

where $\mathcal{Y}_h, \mathcal{U}_h, \mathcal{C}_h, \mathcal{G}_h, \mathcal{H}_h$ are finite dimensional Banach spaces,

$$
\begin{aligned}
J_h &: \mathcal{Y}_h \times \mathcal{U}_h \to \mathbb{R}, & c_h &: \mathcal{Y}_h \times \mathcal{U}_h \to \mathcal{C}_h, \\
g_h &: \mathcal{Y}_h \times \mathcal{U}_h \to \mathcal{G}_h, & h_h &: \mathcal{Y}_h \times \mathcal{U}_h \to \mathcal{H}_h.
\end{aligned}
$$

Discretization

- We want to solve

$$
\begin{aligned}
&\min\ J(y,u)\\
&\text{s.t. } c(y,u)=0, && \text{(the governing PDE)}\\
&\qquad g(y,u)=0, && \text{(additional equality constr.)}
\end{aligned}
$$

where $\mathcal{Y}$,$\mathcal{U}$,$\mathcal{C}$,$\mathcal{G}$,$\mathcal{H}$ are Banach spaces, $K \subset \mathcal{H}$ is a cone, and

$$
\begin{aligned}
J &: \mathcal{Y}\times\mathcal{U}\to\mathbb{R}, & c &: \mathcal{Y}\times\mathcal{U}\to\mathcal{C},\\
g &: \mathcal{Y}\times\mathcal{U}\to\mathcal{G}, & h &: \mathcal{Y}\times\mathcal{U}\to\mathcal{H},
\end{aligned}
$$

- but we can only solve a discretization

$$
\begin{aligned}
\min\quad & J_h(y_h,u_h)\\
\text{s.t.}\quad & c_h(y_h,u_h)=0,\\
& g_h(y_h,u_h)=0,
\end{aligned}
$$

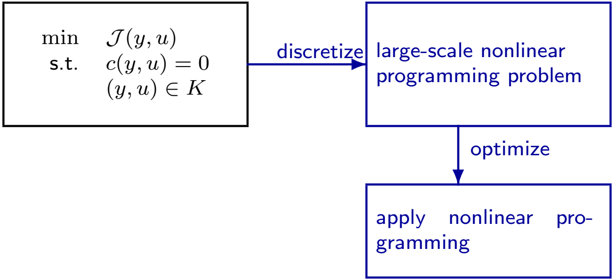
where $\mathcal{Y}_h$,$\mathcal{U}_h$,$\mathcal{C}_h$,$\mathcal{G}_h$,$\mathcal{H}_h$ are finite dimensional Banach spaces,

$$
\begin{aligned}
J_h &: \mathcal{Y}_h\times\mathcal{U}_h\to\mathbb{R}, & c_h &: \mathcal{Y}_h\times\mathcal{U}_h\to\mathcal{C}_h,\\
g_h &: \mathcal{Y}_h\times\mathcal{U}_h\to\mathcal{G}_h, & h_h &: \mathcal{Y}_h\times\mathcal{U}_h\to\mathcal{H}_h.
\end{aligned}
$$

- Does the solution (u_h, y_h) of the discretized problem converge to the solution (u, y) of the original problem? How fast?

Standard Approach

Discretize-then-optimze



Example (W.W. Hager, 2000)

Optimal Control Problem

$$\text{Minimize } \tfrac{1}{2}\int_0^1 u^2(t) + 2y^2(t)dt$$

where

$$\begin{aligned} \dot{y}(t) &= \tfrac{1}{2}y(t) + u(t),\ t \in [0,1], \\ y(0) &= 1. \end{aligned}$$

Solution

$$y^*(t) = \frac{2e^{3t} + e^3}{e^{3t/2}(2+e^3)},$$

$$u^*(t) = \frac{2(e^{3t} - e^3)}{e^{3t/2}(2+e^3)}.$$

Example (W.W. Hager, 2000)

Optimal Control Problem

$$\text{Minimize } \tfrac{1}{2}\int_0^1 u^2(t) + 2y^2(t)dt$$

where

$$\begin{aligned} \dot{y}(t) &= \tfrac{1}{2}y(t) + u(t), \ t \in [0,1], \\ y(0) &= 1. \end{aligned}$$

Solution

$$y^*(t) = \frac{2e^{3t} + e^3}{e^{3t/2}(2 + e^3)},$$

$$u^*(t) = \frac{2(e^{3t} - e^3)}{e^{3t/2}(2 + e^3)}.$$

Discretization using a 2nd order Runge Kutta method

$$\text{Minimize } \frac{h}{2}\sum_{k=0}^{K-1} u_{k+1/2}^2 + 2y_{k+1/2}^2$$

where

$$\begin{aligned} y_{k+1/2} &= y_k + \frac{h}{2}(\tfrac{1}{2}y_k + u_k), \\ y_{k+1} &= y_k + h(\tfrac{1}{2}y_{k+1/2} + u_{k+1/2}), \end{aligned}$$

$k = 0, \dots, K.$

Example (W.W. Hager, 2000)

Optimal Control Problem

$$\text{Minimize } \tfrac{1}{2}\int_0^1 u^2(t) + 2y^2(t)dt$$

where

$$\begin{aligned} \dot{y}(t) &= \tfrac{1}{2}y(t) + u(t),\ t \in [0,1], \\ y(0) &= 1. \end{aligned}$$

Solution

$$y^*(t) = \frac{2e^{3t} + e^3}{e^{3t/2}(2+e^3)},$$
$$u^*(t) = \frac{2(e^{3t} - e^3)}{e^{3t/2}(2+e^3)}.$$

DOES NOT CONVERGE! WHY?

Discretization using a 2nd order Runge Kutta method

$$\text{Minimize } \frac{h}{2}\sum_{k=0}^{K-1} u_{k+1/2}^2 + 2y_{k+1/2}^2$$

where

$$\begin{aligned} y_{k+1/2} &= y_k + \frac{h}{2}(\tfrac{1}{2}y_k + u_k), \\ y_{k+1} &= y_k + h(\tfrac{1}{2}y_{k+1/2} + u_{k+1/2}), \end{aligned}$$

$k = 0, \ldots, K$.

Solution of the discretized problem:

$$y_k = 1, \quad y_{k+1/2} = 0,$$
$$u_k = -\frac{4+h}{2h}, \quad u_{k+1/2} = 0,$$

$k = 0, \ldots, K$.

Discretization of state equation and objective function implies a discretization for the adjoint equation, which may have different convergence properties than the discretization scheme applied to state equation and objective function.

Discretization of state equation and objective function implies a discretization for the adjoint equation, which may have different convergence properties than the discretization scheme applied to state equation and objective function.

For the example problem

$$\dot{y}(t) = \tfrac{1}{2}y(t) + u(t), \qquad y_{k+1/2} = y_k + \frac{\Delta t}{2}(\tfrac{1}{2}y_k + u_k),$$
$$y(0) = 1, \qquad y_{k+1} = y_k + \Delta t(\tfrac{1}{2}y_{k+1/2} + u_{k+1/2}),$$

$$\dot{\lambda}(t) = -\tfrac{1}{2}\lambda(t) + 2y(t), \qquad \lambda_{k+1/2} = \Delta t(\tfrac{1}{2}\lambda_{k+1} - 2y_{k+1/2}),$$
$$\lambda(1) = 0, \qquad \lambda_k = \lambda_{k+1} + (1 + \Delta t/4)\lambda_{k+1/2},$$

$$u(t) - \lambda(t) = 0. \qquad -\lambda_{k+1/2} = 0,$$
$$u_{k+1/2} - \lambda_{k+1} = 0.$$

Discretization of state equation and objective function implies a discretization for the adjoint equation, which may have different convergence properties than the discretization scheme applied to state equation and objective function.

For the example problem

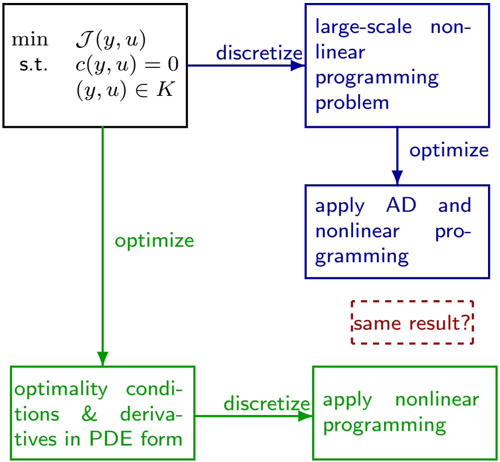
$$\dot y(t) = \tfrac{1}{2} y(t) + u(t), \qquad y_{k+1/2} = y_k + \frac{\Delta t}{2}(\tfrac{1}{2} y_k + u_k),$$
$$y(0) = 1, \qquad y_{k+1} = y_k + \Delta t(\tfrac{1}{2} y_{k+1/2} + u_{k+1/2}),$$

$$\dot\lambda(t) = -\tfrac{1}{2}\lambda(t) + 2y(t), \qquad \lambda_{k+1/2} = \Delta t(\tfrac{1}{2}\lambda_{k+1} - 2y_{k+1/2}),$$
$$\lambda(1) = 0, \qquad \lambda_k = \lambda_{k+1} + (1 + \Delta t/4)\lambda_{k+1/2},$$

$$u(t) - \lambda(t) = 0. \qquad -\lambda_{k+1/2} = 0,$$
$$u_{k+1/2} - \lambda_{k+1} = 0.$$

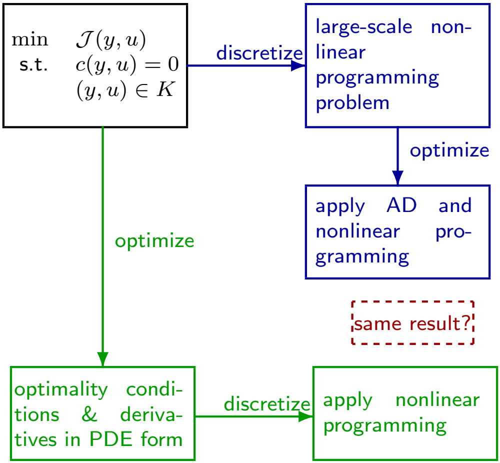
Note, this is a discretization issue, not an issue of how the discretized optimization problem is solved!

Discretize-then-optimize



Optimize-then-discretize

Discretize-then-optimize



Optimize-then-discretize

Both approaches are different, in general. One is not better than the other. It is important to look at the whole picture.

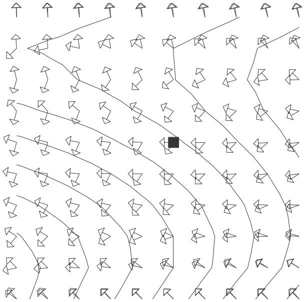
- For nonlinear problems, the optimize-then-discretize may lead to inexact gradients:

$$(\nabla \widehat{J}(u_h))_h \neq \nabla \widehat{J}_h(u_h).$$

But, usually one can show $\|(\nabla \widehat{J}(u_h))_h - \nabla \widehat{J}_h(u_h)\| \to 0$.

- For nonlinear problems, the optimize-then-discretize may lead to inexact gradients:
$$(\nabla \widehat{J}(u_h))_h \neq \nabla \widehat{J}_h(u_h).$$
But, usually one can show $\|(\nabla \widehat{J}(u_h))_h - \nabla \widehat{J}_h(u_h)\| \to 0$.
- Need to use optimization carefully! At a fixed discretization the (gradient based) optimization algorithm will get stuck if the stopping tolerance is too fine relative to the accuracy in the computed gradient $(\nabla \widehat{J}(u_h))_h$.

- For nonlinear problems, the optimize-then-discretize may lead to inexact gradients:
$$(\nabla \widehat{J}(u_h))_h \neq \nabla \widehat{J}_h(u_h).$$
But, usually one can show $\|(\nabla \widehat{J}(u_h))_h - \nabla \widehat{J}_h(u_h)\| \to 0$.
- Need to use optimization carefully! At a fixed discretization the (gradient based) optimization algorithm will get stuck if the stopping tolerance is too fine relative to the accuracy in the computed gradient $(\nabla \widehat{J}(u_h))_h$.



$(\nabla \widehat{J}(u_h))_h$ and $\nabla \widehat{J}_h(u_h)$ for a shape design problem from Burkardt, Gunzburger, Peterson (2002).

- Need to investigate the discretization scheme for the optimal control problem.
- Approaches to coordinate choice of discretization level and optimization.
  - Consistent approximations (Polak (1997)):
    How accurately does one solve the discretized optimization problem before increasing the discretization level? Requires only asymptotic error estimates.
  - Trust-region based model management approaches (see Sec. 10.6 in Conn, Gould, Toint (2000) for an overview):
    At a given iterate y_k, u_k select an approximate problem based on function and derivative information for the original problem. Can go back to approximate model. Requires error estimates.
  - Adaptive mesh refinement for elliptic/parabolic optimal control problems
    Becker/Rannacher (2001,...), Liu et. al. (2003,...), Günther/Hinze (2008), Hintermüller/Hoppe (2005,..), S. Ulbrich (2008), Vexler (2005,...). Applies mostly to linear-quadratic or convex optimal control problems.

From an optimization point of view this is an issue of managing inexactness in function evaluations.

KKT Solver

- Newton-type or Sequential Quadratic Programming (SQP)-type methods require the solution of

$$\begin{array}{ll} \min & \frac{1}{2}\begin{pmatrix} y \\ u \end{pmatrix}^T \begin{pmatrix} H_{yy} & H_{yu} \\ H_{uy} & H_{uu} \end{pmatrix} \begin{pmatrix} y \\ u \end{pmatrix} + \begin{pmatrix} c \\ d \end{pmatrix}^T \begin{pmatrix} y \\ u \end{pmatrix}, \\ \text{s.t.} & Ay + Bu + b = 0 \qquad \text{(discretized PDE)} \end{array}$$

- If $A \in \mathbb{R}^{n_y \times n_y}$ is invertible the QP is equivalent to

$$\min \tfrac{1}{2} u^T Z^T H Z u + u^T Z^T (H x^c + g) + \tfrac{1}{2}(x^c)^T H x^c,$$

where

$$Z = \begin{pmatrix} -A^{-1}B \\ I \end{pmatrix}, \quad x^c = \begin{pmatrix} -A^{-1}b \\ 0 \end{pmatrix}, \quad g = \begin{pmatrix} c \\ d \end{pmatrix}.$$

- Necessary optimality condition

$$\begin{pmatrix} H_{yy} & H_{yu} & A^T \\ H_{uy} & H_{uu} & B^T \\ A & B & 0 \end{pmatrix} \begin{pmatrix} y \\ u \\ \lambda \end{pmatrix} = - \begin{pmatrix} c \\ d \\ b \end{pmatrix}.$$

- H may not be spd on null space of constraints

$$\begin{pmatrix} H_{yy} & H_{yu} & A^T \\ H_{uy} & H_{uu} & B^T \\ A & B & 0 \end{pmatrix} \begin{pmatrix} y \\ u \\ \lambda \end{pmatrix} = - \begin{pmatrix} c \\ d \\ b \end{pmatrix}. \tag{1}$$

- Symmetric permutation of (1) gives

$$\left(\begin{array}{cc|c} H_{yy} & A^T & H_{yu} \\ A & 0 & B \\ \hline H_{uy} & B^T & H_{uu} \end{array}\right) \left(\begin{array}{c} y \\ \lambda \\ \hline u \end{array}\right) = - \left(\begin{array}{c} c \\ b \\ \hline d \end{array}\right). \tag{2}$$

  If $A \in \mathbb{R}^{n_y \times n_y}$ is invertible, (1,1)-block is invertible.

- Schur complement

$$S = H_{uu} - (H_{uy} \mid B^T) \begin{pmatrix} H_{yy} & A^T \\ A & 0 \end{pmatrix}^{-1} \begin{pmatrix} H_{yu} \\ B \end{pmatrix} = Z^T H Z$$

  is equal to the reduced Hessian.

- If good preconditioners for the reduced Hessian $Z^T H Z$ and for the discretized PDE A and adjoint A^T are known, block preconditioners tend to work well (no theory).

- Reduced Hessian $Z^T H Z$ can be very complicated, especially for QP subproblems arising in optimization algorithms for nonlinear problems or for problems with inequality constraints.
- For PDE constrained optimization we need matrix-free preconditioners.
- For some (simple?) applications, optimization based multigrid or domain decomposition methods work well, but they need to be extended case by case to other problems. KKT systems arising in PDE constrained optimization can be very different than saddle point systems arising in PDE.

Personal View of PDE Constrained Optimization in an 'Industry' Setting

- Simulation:

- Optimization:

Personal View of PDE Constrained Optimization in an 'Industry' Setting

- Simulation:
  - Simulators are very complex (complex physics, legacy codes,...) and are often developed without optimization in mind. For example, it can be difficult to extract derivatives even though some may be used inside.
  - Simulations are done with high fidelity, but a low fidelity simulator can be very useful in the optimization context. (It is easier to use and to interface with a gradient based optimization algorithm; can be used in a model management strategy - think 'preconditioning').
  - Simulation tools are used by many - their use cannot be disrupted.
  - Simulator calls optimizer. (Simulator controls the optimizer.)
  - Is it worth the time and money to add complicated optimization capability? Optimization needs evolve; choice of optimizer determined by first need.
  - Optimization problem evolves; simulator may not cover all physics.
  - Improve rather than optimize?
- Optimization:

- Optimization:
    - Give me (exact) function values and derivatives.
    - Optimizer calls subroutines for function and derivative evaluation. (Optimizer controls simulator).
    - Optimize, not only improve.
    - Optimizers are developed for ‘clean’ test problems.

- Optimization:
  - Give me (exact) function values and derivatives.
  - Optimizer calls subroutines for function and derivative evaluation. (Optimizer controls simulator).
  - Optimize, not only improve.
  - Optimizers are developed for 'clean' test problems.
- PDE Constrained Optimization:
  - PDE constrained optimization problems arise in more and more applications.
  - Fast, high fidelity optimization becomes more important.
  - Optimization and simulation need to work together. (What is needed? What can be done? ...)
  - PDE constrained optimization for a small class of important problems, rather than 'general purpose' PDE constrained optimization?
  - PDE constrained optimization poses many challenging theoretical, algorithmic, software engineering questions.

- Optimization:
  - Give me (exact) function values and derivatives.
  - Optimizer calls subroutines for function and derivative evaluation. (Optimizer controls simulator).
  - Optimize, not only improve.
  - Optimizers are developed for 'clean' test problems.
- PDE Constrained Optimization:
  - PDE constrained optimization problems arise in more and more applications.
  - Fast, high fidelity optimization becomes more important.
  - Optimization and simulation need to work together. (What is needed? What can be done? ...)
  - PDE constrained optimization for a small class of important problems, rather than 'general purpose' PDE constrained optimization?
  - PDE constrained optimization poses many challenging theoretical, algorithmic, software engineering questions.
- PDE constrained optimization is difficult:
  If we could solve every PDE constrained optimization problem, we could solve every PDE.