Analysis and Control of Numerical Uncertainty in PDE–Constrained Optimization

SAND2007-6826P

Denis Ridzal

Optimization and Uncertainty Quantification
Sandia National Laboratories, Albuquerque, New Mexico
dridzal@sandia.gov

October 18, 2007

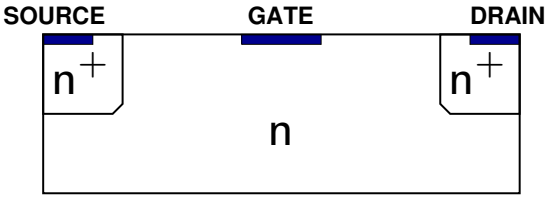
Texas Tech University
Department of Mathematics and Statistics
Lubbock, TX

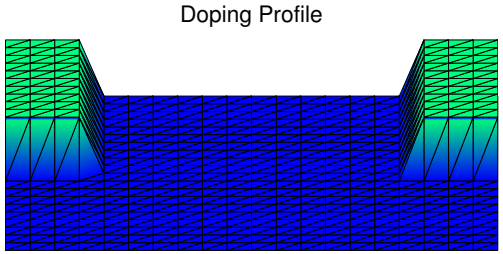
Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under Contract DE-AC04-94-AL85000.

Motivation

Applications of PDE–Constrained Optimization

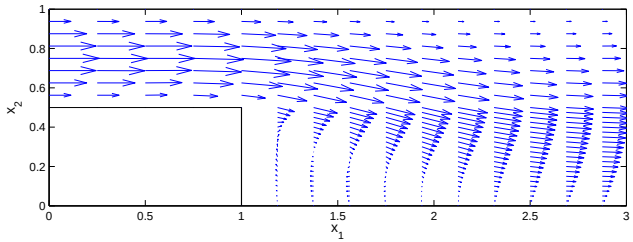
- Solution of parameter estimation, optimal design, and inverse problems arising in the modeling and design of semiconductor devices; collaboration with the *Charon* project at Sandia National Labs.

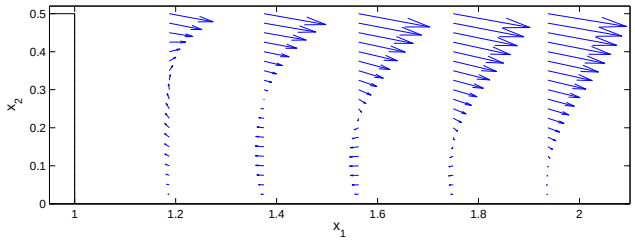




  - increase the current flow over a contact by tweaking the doping profile
  - determine the doping profile based on a profile measurement and the corresponding (experimental) current data

- Solution of optimal control, shape optimization, and inverse problems in computational fluid dynamics.

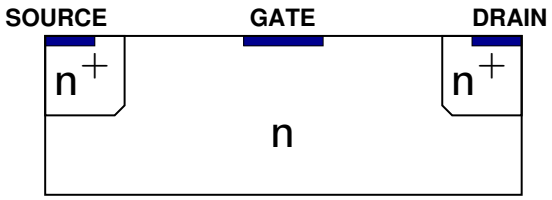


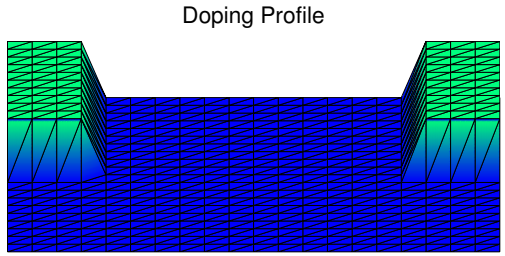


  - minimize vorticity in a region of incompressible flow via boundary controls
  - control temperature in an HVAC system by dynamically directing the flow

Motivation

Optimal Control of Drift–Diffusion Semiconductor Equations





$$\text{Minimize} \quad \mathcal{J} = \frac{1}{2}\|J(x)\cdot\nu - \widehat{J}(x)\cdot\nu\|^2_{-1/2,\Gamma_o} + \frac{\alpha}{2}\|(u(x)-\widehat{u}(x)\|^2_{0,\Omega}$$

subject to

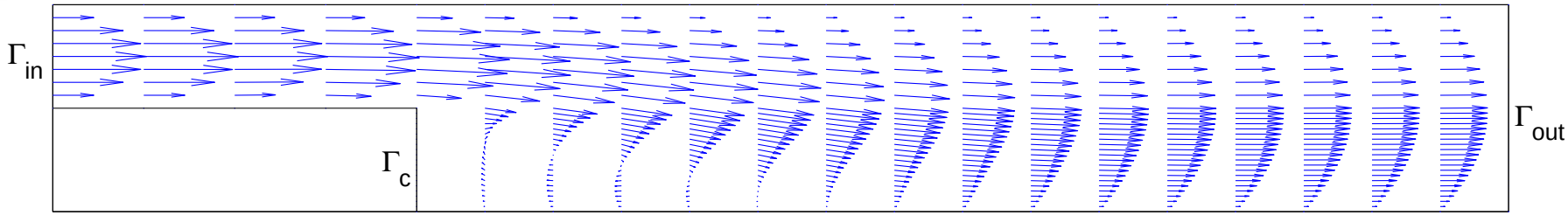
$$\left.\begin{aligned} J_n(x) &= \mu_n(\nabla n(x) + n(x)\nabla y(x)) \\ J_p(x) &= \mu_p(\nabla p(x) - p(x)\nabla y(x)) \\ \nabla\cdot J_n(x) &= 0 \\ \nabla\cdot J_p(x) &= 0 \\ -\nabla\cdot(k(x)\nabla y(x)) &= n(x) - p(x) - u(x), \end{aligned}\right\} \text{DRIFT–DIFFUSION}$$

where y is the electrostatic potential, n and p are electron and hole densities, u is the doping profile, μ_n and μ_p are electron and hole mobilities, k is the permittivity, and the total current density is given by

$$J(x) = J_n(x) + J_p(x).$$

Motivation

Boundary Control of the Incompressible Navier–Stokes Flow



$$\text{Minimize} \quad \frac{1}{2}\int_D (\partial_{x_1}\mathbf{y}_2 - \partial_{x_2}\mathbf{y}_1)^2 d\mathbf{x} + \frac{\alpha}{2}\int_{\Gamma_c} |\mathbf{u}|^2 d\mathbf{x}$$

subject to

$$\begin{aligned}
-\nu\Delta\mathbf{y} + (\mathbf{y}\cdot\nabla)\mathbf{y} + \nabla p &= \mathbf{f} && \text{in} \quad \Omega,\\
\nabla\cdot\mathbf{y} &= 0 && \text{in} \quad \Omega,\\
(\nu\nabla\mathbf{y} - pI)\mathbf{n} &= \mathbf{0} && \text{on} \quad \Gamma_{out},\\
\mathbf{y} &= \mathbf{u} && \text{on} \quad \Gamma_c,\\
\mathbf{y} &= \mathbf{b} && \text{on} \quad \partial\Omega \setminus (\Gamma_c \cup \Gamma_{out}),
\end{aligned}$$

where $\mathbf{y}$ is the velocity field, p denotes the pressure, ν is the inverse Reynolds number, and $\mathbf{u}$ are the boundary controls.

General Formulation and Solution

PDE–constrained optimization problems fit the *mathematical model*

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & x \in \mathcal{S}, \end{aligned}$$

where the spaces $\mathcal{S}$ are subsets of *function spaces*, defined by PDE and possibly other (inequality, integer, etc.) constraints.

Two–Step Solution Process

1. the mathematical model is translated into its *algebraic form*
2. the finite–dimensional algebraic problem is solved numerically

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & x \in \mathcal{S} \end{aligned} \longrightarrow \begin{aligned} \min \quad & f^h(x^h) \\ \text{s.t.} \quad & x^h \in \mathcal{S}^h \end{aligned} \longrightarrow x^h_{\min}$$

General Formulation and Solution

PDE–constrained optimization problems fit the *mathematical model*

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & x \in \mathcal{S}, \end{aligned}$$

where the spaces $\mathcal{S}$ are subsets of *function spaces*, defined by PDE and possibly other (inequality, integer, etc.) constraints.

Two–Step Solution Process

1. the mathematical model is translated into its *algebraic form*
2. the finite–dimensional algebraic problem is solved numerically

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & x \in \mathcal{S} \end{aligned} \longrightarrow \begin{aligned} \min \quad & f^h(x^h) \\ \text{s.t.} \quad & x^h \in \mathcal{S}^h \end{aligned} \longrightarrow x^h_{\min}$$

General Formulation and Solution

PDE–constrained optimization problems fit the *mathematical model*

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & x \in \mathcal{S}, \end{aligned}$$

where the spaces $\mathcal{S}$ are subsets of *function spaces*, defined by PDE and possibly other (inequality, integer, etc.) constraints.

Two–Step Solution Process

1. the mathematical model is translated into its *algebraic form*
2. the finite–dimensional algebraic problem is solved numerically

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & x \in \mathcal{S} \end{aligned} \longrightarrow \begin{aligned} \min \quad & f^h(x^h) \\ \text{s.t.} \quad & x^h \in \mathcal{S}^h \end{aligned} \longrightarrow x^h_{\min}$$

General Formulation and Solution

PDE–constrained optimization problems fit the *mathematical model*

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & x \in \mathcal{S}, \end{aligned}$$

where the spaces $\mathcal{S}$ are subsets of *function spaces*, defined by PDE and possibly other (inequality, integer, etc.) constraints.

Two–Step Solution Process

1. the mathematical model is translated into its *algebraic form*
2. the finite–dimensional algebraic problem is solved numerically

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & x \in \mathcal{S} \end{aligned} \longrightarrow \begin{aligned} \min \quad & f^h(x^h) \\ \text{s.t.} \quad & x^h \in \mathcal{S}^h \end{aligned} \longrightarrow x^h_{\min}$$

- both steps can be very challenging (function spaces, problem size)
- a frequently ignored, yet critical challenge, is the handling of the so–called **numerical uncertainty**

Numerical Uncertainty in PDE–Constrained Optimization

$$\begin{array}{ll} \min & f(x) \\ \text{s.t.} & x \in \mathcal{S} \end{array} \longrightarrow \begin{array}{ll} \min & f^h(x^h) \\ \text{s.t.} & x^h \in \mathcal{S}^h \end{array}$$

1. *Loss of information associated with the reduction of the infinite–dimensional mathematical model to its algebraic form.*
   - choice of spatial discretization (FE: Galerkin, Mixed *or* FV *or* FD)
   - choice of temporal discretization (explicit & type *or* implicit & type)
   - size of discretization (spatial resolution, time step), etc.

$$\begin{array}{ll} \min & f^h(x^h) \\ \text{s.t.} & x^h \in \mathcal{S}^h \end{array} \longrightarrow x^h_{\min}$$

2. *Loss of information associated with the use of a particular numerical algorithm in solving the finite–dimensional algebraic problem.*
   - classical truncation or round-off error (well–studied)
   - for large–scale problems, inexactness in the iterative solution of linear systems, which are the core component of the algebraic form, and therefore of critical importance for the behavior of optimization algorithms (currently ignored in practice)

Numerical Uncertainty in PDE–Constrained Optimization

$$\begin{array}{ll} \min & f(x) \\ \text{s.t.} & x \in \mathcal{S} \end{array} \longrightarrow \begin{array}{ll} \min & f^h(x^h) \\ \text{s.t.} & x^h \in \mathcal{S}^h \end{array}$$

1. *Loss of information associated with the reduction of the infinite–dimensional mathematical model to its algebraic form.*
   - choice of spatial discretization (FE: Galerkin, Mixed *or* FV *or* FD)
   - choice of temporal discretization (explicit & type *or* implicit & type)
   - size of discretization (spatial resolution, time step), etc.

$$\begin{array}{ll} \min & f^h(x^h) \\ \text{s.t.} & x^h \in \mathcal{S}^h \end{array} \longrightarrow x^h_{\min}$$

2. *Loss of information associated with the use of a particular numerical algorithm in solving the finite–dimensional algebraic problem.*
   - classical truncation or round-off error (well–studied)
   - for large–scale problems, inexactness in the iterative solution of linear systems, which are the core component of the algebraic form, and therefore of critical importance for the behavior of optimization algorithms (currently ignored in practice)

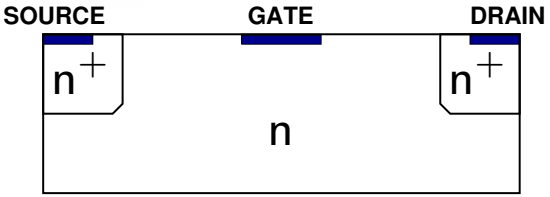
Part I

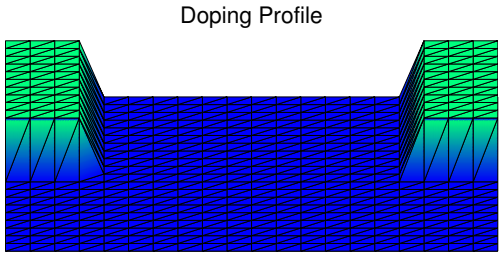
Mathematical Model ⟶ Algebraic Representation

How does the choice of the spatial discretization affect
the solution of PDE–constrained optimization problems?

- Semiconductor Modeling: Survey of Discretization Techniques
- The Discrete Optimization Problem: Galerkin vs. Mixed Galerkin
- Demonstration of Numerical Failure

Optimal Control of the Drift–Diffusion Equations





$$\text{Minimize} \quad \mathcal{J} = \frac{1}{2}\|J(x)\cdot\nu - \widehat{J}(x)\cdot\nu\|^2_{-1/2,\Gamma_o} + \frac{\alpha}{2}\|(u(x)-\widehat{u}(x)\|^2_{0,\Omega}$$

subject to

$$\left.\begin{aligned} J_n(x) &= \mu_n(\nabla n(x) + n(x)\nabla y(x)) \\ J_p(x) &= \mu_p(\nabla p(x) - p(x)\nabla y(x)) \\ \nabla\cdot J_n(x) &= 0 \\ \nabla\cdot J_p(x) &= 0 \\ -\nabla\cdot(k(x)\nabla y(x)) &= n(x) - p(x) - u(x), \end{aligned}\right\} \text{DRIFT–DIFFUSION}$$

where y is the electrostatic potential, n and p are electron and hole densities, u is the doping profile, μ_n and μ_p are electron and hole mobilities, k is the permittivity, and the total current density is given by

$$J(x) = J_n(x) + J_p(x).$$

Survey

Discretization of the Drift–Diffusion Equations

- primal Galerkin FE schemes with streamline or flux upwinding (SUPG – Hughes, Brooks; FUPG – Carey, Sharma)
- mixed and hybrid FE methods with exponential fitting (Brezzi, Marini, Pietra; Holst, Jüngel, Pietra)
- exponentially fitted triangular and tetrahedral FE methods (Wang, Miller, Angermann)
- finite volume / covolume methods, e.g. the *box method* with Scharfetter–Gummel upwinding (McCartin, Bank et al., Mock)

Optimization

- the impact of the spatial discretization on the solution of PDE–constrained optimization problems is not well-studied
- one example: study of the SUPG method in discretize-then-optimize vs. optimize-then-discretize (Collis and Heinkenschloss)
- comparative study of Galerkin versus mixed Galerkin discretizations

Simplified Model Problem

minimize $$\mathcal{J} = \frac{1}{2}\|J(x)\cdot\nu - \widehat{J}(x)\cdot\nu\|^2_{-1/2,\Gamma_o} + \frac{\alpha}{2}\|(u(x)-\widehat{u}(x)\|^2_{0,\Omega}$$

subject to

$$\begin{aligned} J_n(x) &= \mu_n(\nabla n(x) + n(x)\nabla y(x)) \\ J_p(x) &= \mu_p(\nabla p(x) - p(x)\nabla y(x)) \\ \nabla\cdot J_n(x) &= 0 \\ \nabla\cdot J_p(x) &= 0 \\ -\nabla\cdot(k(x)\nabla y(x)) &= n(x) - p(x) - u(x) \end{aligned}$$

Simplified Model Problem

minimize $\mathcal{J} = \frac{1}{2}\|\nabla y(x)\cdot\nu - \nabla\widehat{y}(x)\cdot\nu\|^2_{-1/2,\Gamma_o} + \frac{\alpha}{2}\|(u(x)-\widehat{u}(x)\|^2_{0,\Omega}$

subject to

$$J_n(x) = \mu_n(\nabla n(x) + n(x)\nabla y(x))$$
$$J_p(x) = \mu_p(\nabla p(x) - p(x)\nabla y(x))$$
$$\nabla\cdot J_n(x) = 0$$
$$\nabla\cdot J_p(x) = 0$$
$$-\nabla\cdot(k(x)\nabla y(x)) = n(x) - p(x) - u(x)$$

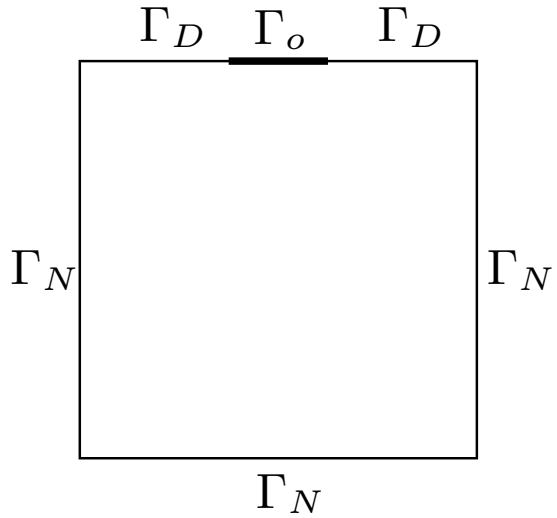
$$J \rightarrow \nabla y$$

Simplified Model Problem

minimize $\mathcal{J} = \frac{1}{2}\|\nabla y(x)\cdot\nu - \nabla\widehat{y}(x)\cdot\nu\|^2_{-1/2,\Gamma_o} + \frac{\alpha}{2}\|(u(x)-\widehat{u}(x)\|^2_{0,\Omega}$

subject to

$$\begin{aligned} -\nabla\cdot(k(x)\nabla y(x)) &= f(x)+u(x) && \text{in } \Omega \\ y(x) &= y_D(x) && \text{on } \Gamma_D \\ (k(x)\nabla y(x))\cdot\nu &= g(x) && \text{on } \Gamma_N \end{aligned}$$



Galerkin Discretization

- state and control spaces

$$Y = \left\{ y \in H^1(\Omega) \ : \ y = y_D \text{ on } \Gamma_D \right\}, \quad U = L^2(\Omega)$$

- test function space

$$V = \left\{ v \in H^1(\Omega) \ : \ v = 0 \text{ on } \Gamma_D \right\}$$

- (bi)linear forms

$$a(y,v) = \int_\Omega k \nabla y \cdot \nabla v \, dx, \qquad b(u,v) = -\int_\Omega uv \, dx,$$
$$\langle f, v \rangle = \int_\Omega fv \, dx, \qquad \langle g, v \rangle_{\Gamma_N} = \int_{\Gamma_N} gv \, dx$$

- **Weak form**: Find $y \in Y, u \in U$, which solve, *for all* $v \in V$

minimize $\frac{1}{2} \| \nabla y(x) \cdot \nu - \nabla \widehat{y}(x) \cdot \nu \|^2_{-1/2,\Gamma_o} + \frac{\alpha}{2} \| (u(x) - \widehat{u}(x) \|^2_{0,\Omega}$

subject to

$$a(y,v) + b(u,v) = \langle f, v \rangle + \langle g, v \rangle_{\Gamma_N}.$$

Galerkin Discretization

- state and control spaces

$$Y = \left\{ y \in H^1(\Omega) \ : \ y = y_D \text{ on } \Gamma_D \right\}, \quad U = L^2(\Omega)$$

- test function space

$$V = \left\{ v \in H^1(\Omega) \ : \ v = 0 \text{ on } \Gamma_D \right\}$$

- (bi)linear forms

$$a(y,v) = \int_\Omega k \nabla y \cdot \nabla v \, dx, \qquad b(u,v) = -\int_\Omega uv \, dx,$$
$$\langle f, v \rangle = \int_\Omega fv \, dx, \qquad \langle g, v \rangle_{\Gamma_N} = \int_{\Gamma_N} gv \, dx$$

- **Weak form**: Find $y \in Y, u \in U$, which solve, *for all* $v \in V$

minimize $\frac{1}{2} \| \nabla y(x) \cdot \nu - \nabla \widehat{y}(x) \cdot \nu \|^2_{-1/2,\Gamma_o} + \frac{\alpha}{2} \| (u(x) - \widehat{u}(x) \|^2_{0,\Omega}$

subject to

$$a(y,v) + b(u,v) = \langle f, v \rangle + \langle g, v \rangle_{\Gamma_N}.$$

Galerkin Discretization

Flux Term: $\|\nabla y(x) \cdot \nu - \nabla \widehat{y}(x) \cdot \nu\|^2_{-1/2,\Gamma_o}$

- **Standard approach.** Restrict the states to a finite element subspace Y_h, compute terms $\nabla y_h \cdot \nu$ directly, and use a weighted L^2–norm to approximate the norm in $H^{-1/2}(\Gamma_o)$:

$$\|\nabla y \cdot \nu - \nabla \widehat{y} \cdot \nu\|^2_{-1/2,\Gamma_o} \approx h\|\nabla y_h \cdot \nu - \nabla \widehat{y}_h \cdot \nu\|^2_{0,\Gamma_o}$$

- **Better choice: Variational Flux Approximation (VFA).** Replace flux $\nabla y_h \cdot \nu$ by a more accurate, C^0 approximation λ_h, obtained by solving the equation

$$\int_{\Gamma_0} \lambda_h v_h \, dl = k^{-1} \left(a(y_h, v_h) + b(u_h, v_h) - (f, v_h) - \int_{\Gamma \setminus \Gamma_o} k \nabla y_h \cdot \nu v_h \, dl \right)$$

then approximate the flux term as follows:

$$\|\nabla y \cdot \nu - \nabla \widehat{y} \cdot \nu\|^2_{-1/2,\Gamma_o} \approx h\|\lambda_h - \widehat{\lambda}_h\|^2_{0,\Gamma_o}$$

Galerkin Discretization

Variational Flux Approximation

$$\int_{\Gamma_0} \lambda_h v_h \, dl = k^{-1} \left(a(y_h, v_h) + b(u_h, v_h) - (f, v_h) - \int_{\Gamma \setminus \Gamma_o} k \nabla y_h \cdot \nu v_h \, dl \right)$$

- PDEs: Wheeler, Babuška, Brezzi, Hughes, etc.
  When used to postprocess a given finite element solution (y_h, u_h), the right hand side above involves only known quantities.
- Optimal control (to date): Berggren, et al. (Thanks!)
  VFA is used in an already defined optimality system to improve the accuracy of the solution.
- Our case, NEW use of VFA:
  VFA *changes* the optimization problem, because the discretization

$$\|\nabla y \cdot \nu - \nabla \widehat{y} \cdot \nu\|^2_{-1/2,\Gamma_o} \approx h \|\lambda_h - \widehat{\lambda}_h\|^2_{0,\Gamma_o},$$

  where λ_h is given above, is a function of *both* the unknown states y_h and the unknown controls u_h!

Mixed Galerkin Discretization

- consider the Poisson equation as a first–order system:
  Given $u \in L^2(\Omega)$, $f \in L^2(\Omega)$, $g \in L^2(\Gamma_N)$, and $y_d \in H^1(\Omega) \cap C(\overline{\Omega})$, seek $y \in L^2(\Omega)$ and $p \in \left[L^2(\Omega)\right]^2$, with $\nabla \cdot p \in L^2(\Omega)$, satisfying

$$\begin{aligned} \nabla \cdot p + u &= -f && \text{in } \Omega \\ k^{-1}\, p - \nabla y &= 0 && \text{in } \Omega \\ y &= y_D && \text{on } \Gamma_D \\ (k \nabla y) \cdot \nu &= g && \text{on } \Gamma_N. \end{aligned}$$

- for the weak form, we introduce the spaces

$$\begin{aligned} H(\mathrm{div}, \Omega) &= \left\{ q \in \left[L^2(\Omega)\right]^2 : \nabla \cdot q \in L^2(\Omega) \right\}, \\ H_{0,N}(\mathrm{div}, \Omega) &= \left\{ q \in H(\mathrm{div}, \Omega) : q \cdot \nu = 0 \text{ on } \Gamma_N \right\}, \\ H_{g,N}(\mathrm{div}, \Omega) &= \left\{ q \in H(\mathrm{div}, \Omega) : (k\, q) \cdot \nu = g \text{ on } \Gamma_N \right\}. \end{aligned}$$

Mixed Galerkin Discretization

- state spaces $Y = L^2(\Omega)$ and $P = H_{g,N}(\text{div}, \Omega)$
- control space $U = L^2(\Omega)$
- test function spaces $V = L^2(\Omega)$ and $Q = H_{0,N}(\text{div}, \Omega)$
- (bi)linear forms

$$a(p,q) = \int_\Omega \frac{1}{k}\, p \cdot q \; dx, \qquad b(q,y) = \int_\Omega (\nabla \cdot q) y \; dx, \qquad c(u,v) = \int_\Omega uv \; dx,$$

$$\langle f, v \rangle = \int_\Omega fv \; dx, \qquad \langle y_d, q \cdot \nu \rangle_{\Gamma_D} = \int_{\Gamma_D} y_D \; q \cdot \nu \; dx$$

- **Weak form**: Find $(y,p) \in Y \times P$, and $u \in U$, which solve, *for all* $q \in Q$ *and all* $v \in V$, the problem

$$\text{minimize} \quad \frac{1}{2}\|k^{-2}(p \cdot \nu - \widehat{p} \cdot \nu)\|^2_{-1/2,\Gamma_o} + \frac{\alpha}{2}\|u - \widehat{u}\|^2_{0,\Omega}$$

subject to

$$a(p,q) + b(q,y) = \langle y_d, q \cdot \nu \rangle_{\Gamma_D}$$
$$b(p,v) + c(u,v) = -\langle f, v \rangle .$$

Mixed Galerkin Discretization

Flux Term: $\|k^{-2}(p\cdot\nu-\widehat{p}\cdot\nu)\|^2_{-1/2,\Gamma_o}$

- **Direct approximation.** The flux is approximated directly by p_h:

$$\|k^{-2}(p\cdot\nu-\widehat{p}\cdot\nu)\|^2_{-1/2,\Gamma_o}\approx h\|k^{-2}(p_h\cdot\nu-\widehat{p}_h\cdot\nu\|^2_{0,\Gamma_o}\,.$$

- A more natural, "compatible" flux approximation!

Preliminary Comparison

$$\text{minimize} \quad \frac{1}{2}\|J(x)\cdot\nu - \widehat{J}(x)\cdot\nu\|^2_{-1/2,\Gamma_o} + \frac{\alpha}{2}\|(u(x)-\widehat{u}(x)\|^2_{0,\Omega}$$

subject to

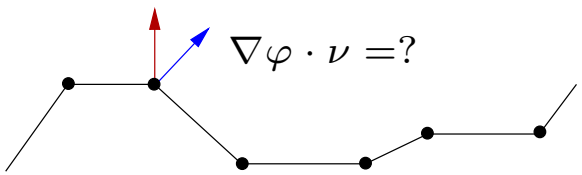
$$\begin{aligned} -\nabla\cdot(k(x)\nabla y(x)) &= f(x)+u(x) && \text{in } \Omega \\ y(x) &= y_D(x) && \text{on } \Gamma_D \\ (k(x)\nabla y(x))\cdot\nu &= g(x) && \text{on } \Gamma_N \end{aligned}$$

GALERKIN

$\int_{\Gamma_o} \nabla\varphi_i\cdot\nu\ \nabla\varphi_j\cdot\nu\ dx$

$\varphi_{i,j}$ nodal basis functions

VFA



⊖ STD: inaccurate flux computations

⊖ VFA: cumbersome implementation!

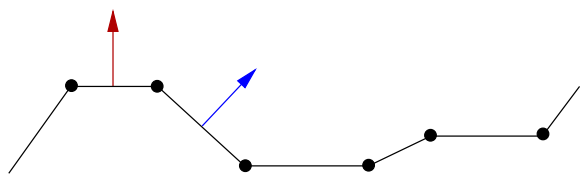
$$a(y,v)+b(u,v) = \langle f,v\rangle + \langle g,v\rangle_{\Gamma_N}$$

⊕ less expensive solution of the PDE

MIXED GALERKIN

$\int_{\Gamma_o} k^{-2}\ \psi_i\cdot\nu\ \psi_j\cdot\nu\ dx$

$\psi_{i,j}$ "face" basis functions



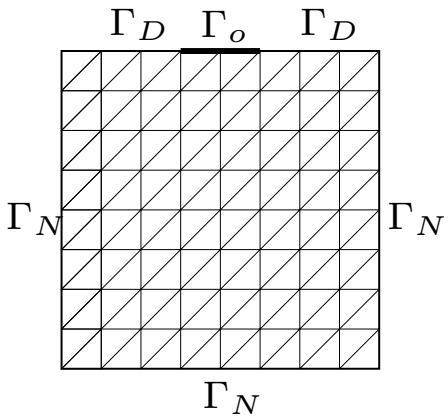
⊕ more natural choice for flux objectives

$$a(p,q)+b(q,y) = \langle y_d, q\cdot\nu\rangle_{\Gamma_D}$$

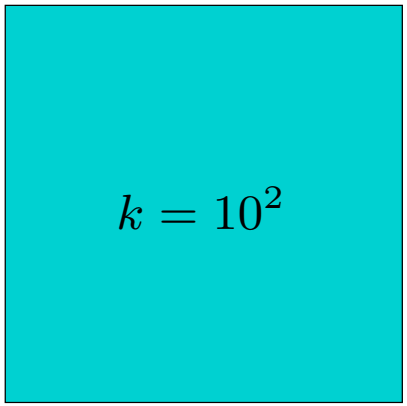
$$b(p,v)+c(u,v) = -\langle f,v\rangle$$

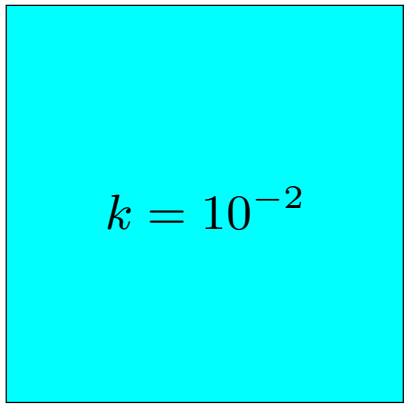
⊖ more expensive solution of the PDE

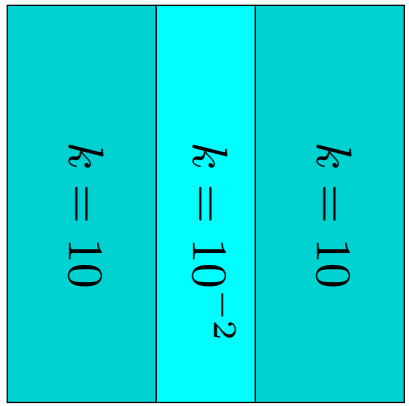
Experimental Setup



- Galerkin: first–order nodal elements ($\mathcal{P}^1$)
- Mixed: lowest–order Raviart–Thomas elements (RT_0)
- $32 \times 32 \times 2$ triangular mesh
- forcing term $f(x) = 0$, $y_D = 0$ on Γ_D
- target doping $\widehat{u} = 1$, reg. param. $\alpha = 6.25 \cdot 10^{-4}$
- $g = 0$ on left, right Γ_N; $g(x) = -k(x)$ on bottom Γ_N
- **parameter study**: (a) diffusivity profile k
  (b) target flux $\nabla \widehat{y} \cdot \nu$







Experimental Setup ... cont'd

Example 1. The desired flux is $\nabla\widehat{y}\cdot\nu = 1$ and $k(x) = 10^2$ in Ω.

Example 2. The desired flux is $\nabla\widehat{y}\cdot\nu = 1$ and $k(x) = 10^{-2}$ in Ω.

Example 3. The desired flux is $\nabla\widehat{y}\cdot\nu = 1$ and

$$k(x) = \begin{cases} 10 & \text{in } [-1,-0.25]\times[-1,1] \\ 10^{-2} & \text{in } [-0.25,0.25]\times[-1,1] \\ 10 & \text{in } [0.25,1]\times[-1,1], \end{cases}$$

Example 4. The desired flux is $\nabla\widehat{y}\cdot\nu = 100$ and $k(x)$ is as in Ex. 3.

Objective Functional

	Example 1			Example 2		
	GM	Mixed	GM-VFA	GM	Mixed	GM-VFA
$\mathcal{J}_F$	1.99e-06	1.88e-06	1.95e-04	6.42e-09	2.51e-09	2.70e-09
$\mathcal{J}_u$	1.10e-08	1.10e-08	3.81e-07	1.12e-03	1.08e-03	1.11e-03
$\mathcal{J}$	2.00e-06	1.89e-06	1.95e-04	1.12e-03	1.08e-03	1.11e-03
	Example 3			Example 4		
	GM	Mixed	GM-VFA	GM	Mixed	GM-VFA
$\mathcal{J}_F$	6.07e-05	8.10e-10	2.83e-07	1.06e+01	2.56e-07	6.29e-07
$\mathcal{J}_u$	7.17e-05	4.62e-05	4.50e-03	3.63e+00	4.57e-03	7.19e-03
$\mathcal{J}$	1.32e-04	4.62e-05	4.50e-03	1.42e+01	4.57e-03	7.19e-03

Table: $\mathcal{J}_F$, $\mathcal{J}_u$ and $\mathcal{J}$ denote the values of the flux term, the control term and their sum (the total value of the objective functional).

Example 3

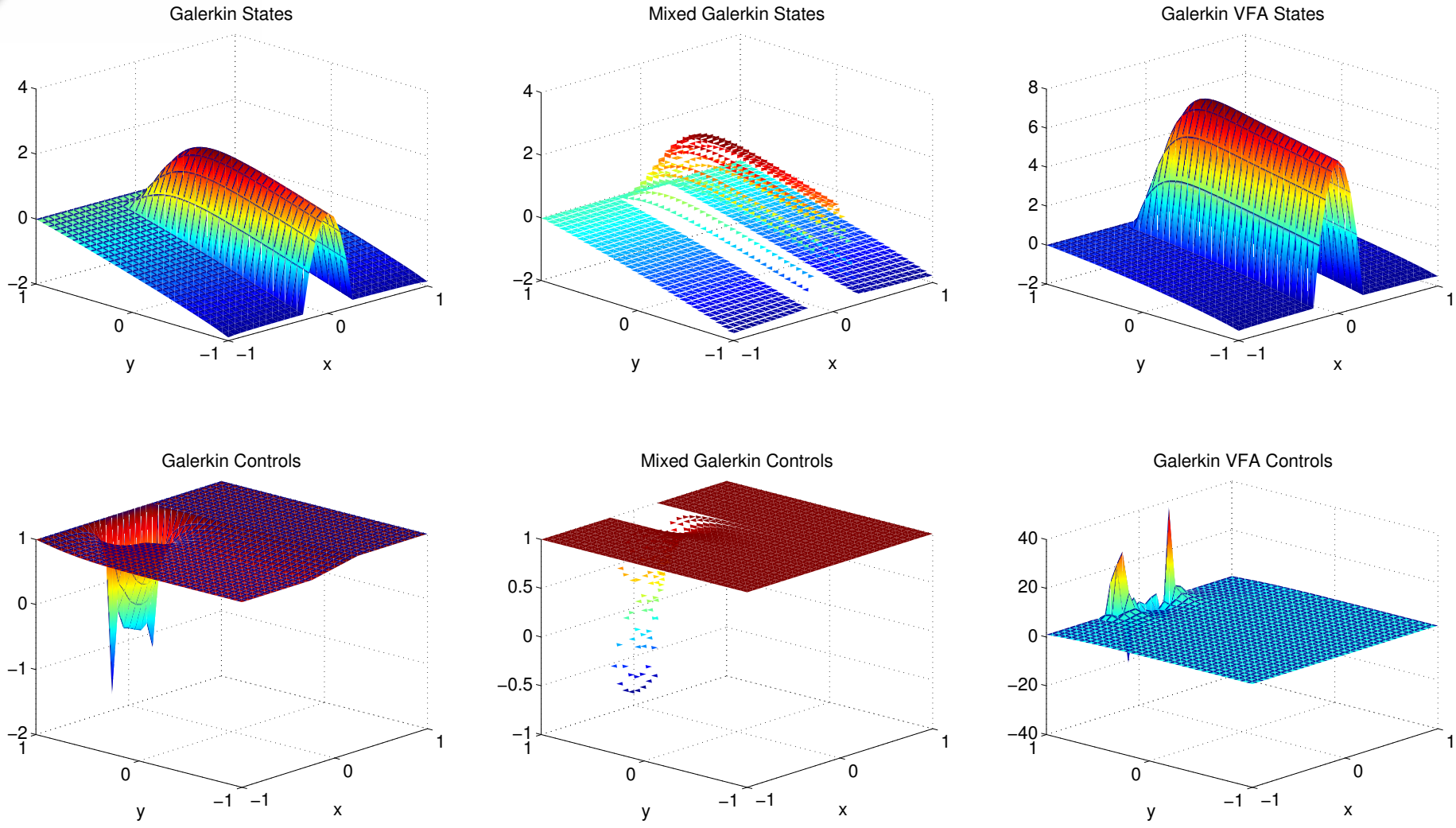


Figure: Galerkin, mixed Galerkin, and Galerkin VFA optimal states (top row) and optimal controls (bottom row) for Example 3.

Example 4

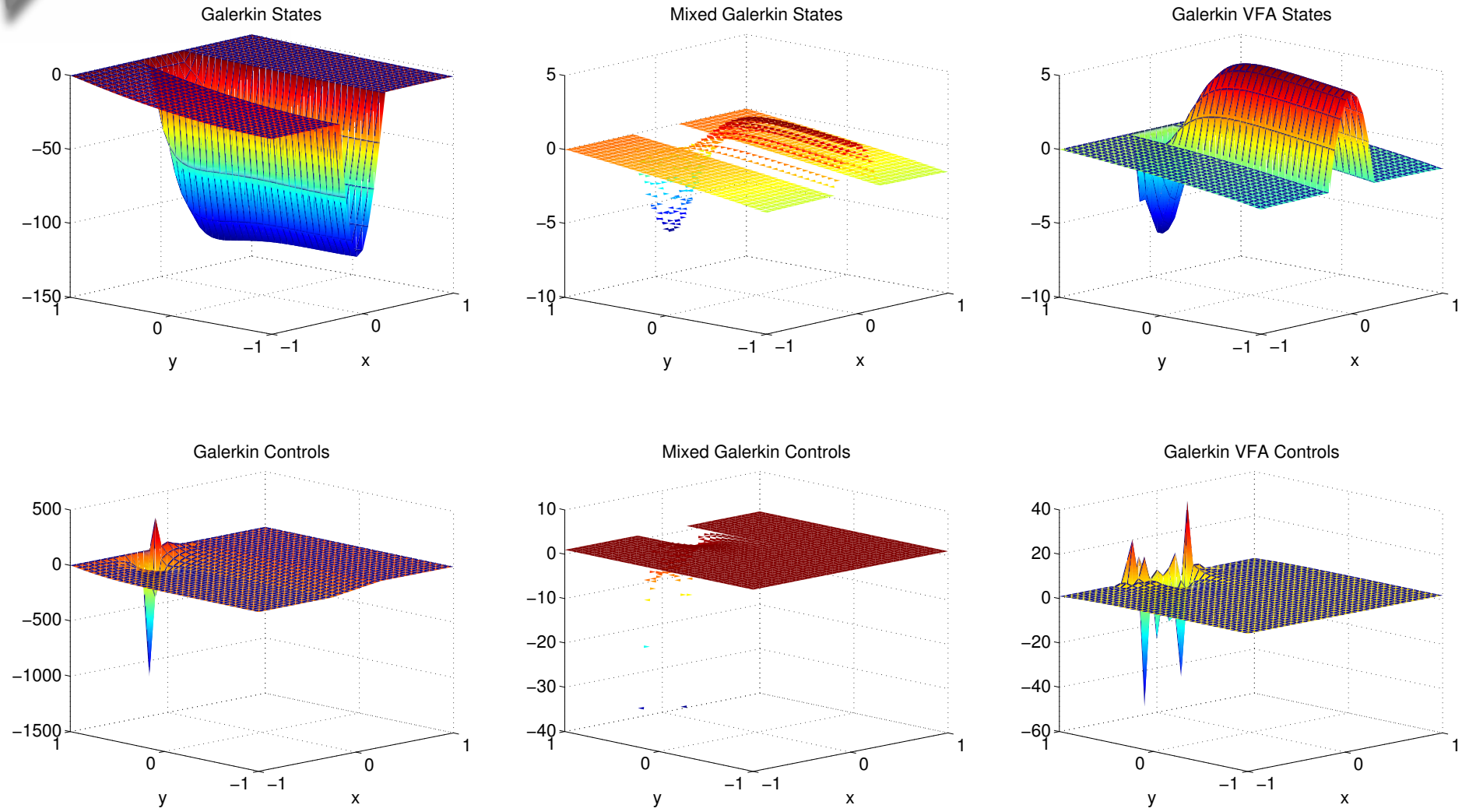


Figure: Galerkin, mixed Galerkin, and Galerkin VFA optimal states (top row) and optimal controls (bottom row) for Example 4.

Summary

- performed study of Galerkin and mixed Galerkin discretizations used for the numerical solution of PDE–constrained optimization problems with applications to semiconductor design
- *unique problem feature*: objective functionals involve flux terms, which have fundamentally different discrete representations depending on the type of FE discretization
- for problems with heterogeneous material properties the mixed Galerkin method offers the most robust performance and the most accurate results
- the worst performer is the standard Galerkin method (not recommended!), which may yield state and control approximations that are many orders of magnitude less accurate than those computed by the mixed method
- if, for whatever reason, the use of the mixed method is not feasible, then the Galerkin discretization of the state equations should be combined with the VFA approach in order to improve robustness and accuracy

Summary

- performed study of Galerkin and mixed Galerkin discretizations used for the numerical solution of PDE–constrained optimization problems with applications to semiconductor design
- *unique problem feature*: objective functionals involve flux terms, which have fundamentally different discrete representations depending on the type of FE discretization
- for problems with heterogeneous material properties the mixed Galerkin method offers the most robust performance and the most accurate results
- the worst performer is the standard Galerkin method (not recommended!), which may yield state and control approximations that are many orders of magnitude less accurate than those computed by the mixed method
- if, for whatever reason, the use of the mixed method is not feasible, then the Galerkin discretization of the state equations should be combined with the VFA approach in order to improve robustness and accuracy
- Compatibility of a spatial discretization with respect to a PDE may not be enough to ensure stable and accurate solution of an optimization problem governed by that PDE.

Part II

Algebraic Form ⟶ Numerical Solution

How does inexactness in the solution of linear systems affect the development of robust large–scale optimization algorithms?

- PDE–Constrained Optimization and Inexactness
- An Inexact Sequential Quadratic Programming Algorithm
- Numerical Results: Global and Local Convergence, Robustness

PDE–Constrained Optimization Problems

- can be solved as constrained, nonconvex nonlinear programming problems (NLPs) using *all-at-once* "intrusive" techniques
- due to constraint regularity, fast Newton–type algorithms are often applicable, e.g. sequential quadratic programming (SQP)
- each iteration in an SQP algorithm requires the solution of several linear systems involving the linearized constraints
- in large–scale applications, linear systems are solved using iterative solvers

$\Rightarrow$ the optimization algorithm must be responsible for dynamically managing stopping tolerances for linear solvers!

Inexactness

- unconstrained optimization: Eisenstat, Steihaug, Dennis, Walker, Carter
- reduced–space line–search SQP: Jäger et. al (1997), Biros et. al (2002)
  - dependence on Lipschitz constants and derivative bounds
- reduced–space TR SQP: Heinkenschloss / Vicente (2001)
  - provide generic theoretical framework, but no concrete algorithm
- **first usable algorithms**: Heinkenschloss / R. 2006 (full–space TR SQP), Byrd / Curtis / Nocedal 2007 (full–space line–search SQP)

Review of Trust-Region SQP

Solve NLP:

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & c(x) = 0 \end{aligned}$$

where $f : \mathcal{X} \to \mathbb{R}$ and $c : \mathcal{X} \to \mathcal{C}$, for some Hilbert spaces $\mathcal{X}$ and $\mathcal{C}$, and f and c are twice continuously Fréchet differentiable.

- define Lagrangian functional $\mathcal{L} : \mathcal{X} \times \mathcal{C} \to \mathbb{R}$:
$$\mathcal{L}(x, \lambda) = f(x) + \langle \lambda, c(x) \rangle_{\mathcal{C}}$$
- if *regular* point x_* is a local solution of the NLP, then there exists a $\lambda_* \in \mathcal{C}$ satisfying the *1st order necessary optimality conditions*:
$$\begin{aligned} \nabla_x f(x_*) + c_x(x_*)^* \lambda_* &= 0 \\ c(x_*) &= 0 \end{aligned}$$

- Newton's method applied to the 1st order optimality conditions:

$$\begin{pmatrix} \nabla_{xx}\mathcal{L}(x_k,\lambda_k) & c_x(x_k)^* \\ c_x(x_k) & 0 \end{pmatrix} \begin{pmatrix} s_k^x \\ s_k^\lambda \end{pmatrix} = - \begin{pmatrix} \nabla_x f(x_k) + c_x(x_k)^*\lambda_k \\ c(x_k) \end{pmatrix}$$

- If $\nabla_{xx}\mathcal{L}(x_k,\lambda_k)$ is positive definite on the null space of $c_x(x_k)$, the above *KKT system* is necessary and sufficient for the solution of the quadratic programming problem (QP):

$$\begin{aligned} \text{min} \quad & \frac{1}{2}\langle \nabla_{xx}\mathcal{L}(x_k,\lambda_k)s_k^x, s_k^x\rangle_{\mathcal{X}} + \langle \nabla_x\mathcal{L}(x_k,\lambda_k), s_k^x\rangle_{\mathcal{X}} \\ \text{s.t.} \quad & c_x(x_k)s_k^x + c(x_k) = 0 \end{aligned}$$

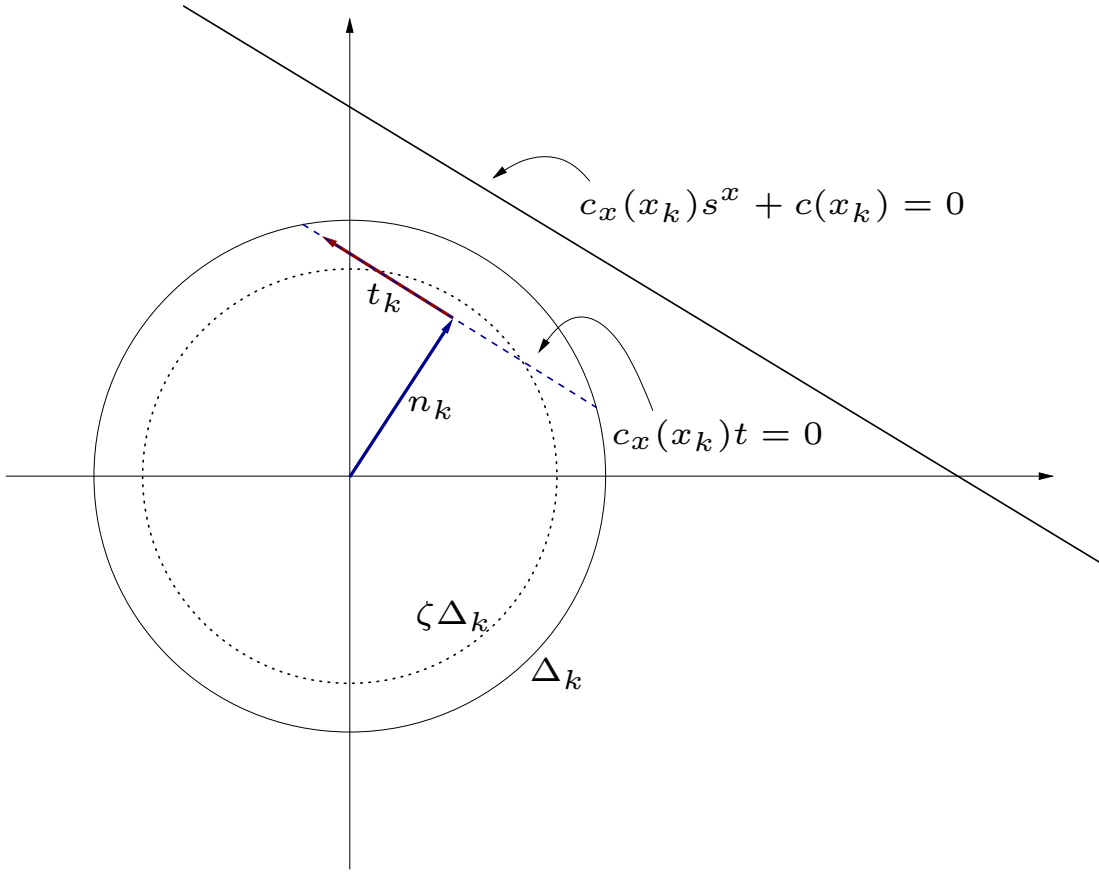
- To globalize the convergence, we add a trust–region constraint:

$$\begin{aligned} \text{min} \quad & \frac{1}{2}\langle H_k s_k^x, s_k^x\rangle_{\mathcal{X}} + \langle \nabla_x\mathcal{L}_k, s_k^x\rangle_{\mathcal{X}} \\ \text{s.t.} \quad & c_x(x_k)s_k^x + c(x_k) = 0 \\ & \|s_k^x\|_{\mathcal{X}} \le \Delta_k. \end{aligned}$$

Possible incompatibility of constraints: Composite–Step Approach.

Composite–Step Approach for the Solution of the Quadratic Subproblem

- **TR SQP step**: $s_k = n_k + t_k$
- **quasi-normal step** n_k: moves toward feasibility
- **tangential step** t_k: moves toward optimality while staying in the null space of the linearized constraints



e.g. Omojokun [1989],
Byrd, Hribar, Nocedal [1997],
Dennis, El–Alem, Maciel [1997],
Dennis, Heinkenschloss, Vicente [1998], Conn, Gould, Toint [2000]

Acceptance of the Step

- Merit function:

$$\phi(x, \lambda; \rho) = f(x) + \langle \lambda, c(x) \rangle_{\mathcal{C}} + \rho \|c(x)\|_{\mathcal{C}}^2 = \mathcal{L}(x, \lambda) + \rho \|c(x)\|_{\mathcal{C}}^2.$$

- *Actual reduction* at step k:

$$ared(s_k^x; \rho_k) = \phi(x_k, \lambda_k; \rho_k) - \phi(x_k + s_k, \lambda_{k+1}; \rho_k)$$

- *Predicted reduction* at step k:

$$pred(s_k^x; \rho_k) = \phi(x_k, \lambda_k; \rho_k) - \Big[\mathcal{L}(x_k, \lambda_k) + \langle g_k, s_k \rangle_{\mathcal{X}} + \frac{1}{2} \langle H_k s_k^x, s_k^x \rangle_{\mathcal{X}}$$
$$+ \langle \lambda_{k+1} - \lambda_k, c_x(x_k) s_k^x + c(x_k) \rangle_{\mathcal{C}} + \rho_k \|c_x(x_k) s_k^x + c(x_k)\|_{\mathcal{C}}^2 \Big].$$

Composite–Step Trust–Region SQP Algorithm

1. Compute quasi–normal step n_k.
2. Compute tangential step t_k.
3. Compute new Lagrange multiplier estimate λ_{k+1}.
4. Update penalty parameter ρ_k.
5. Compute $ared_k$, $pred_k$.
6. Decide whether to accept the new iterate $x_{k+1} = x_k + n_k + t_k$, and update Δ_{k+1} from Δ_k, based on $\frac{ared_k}{pred_k}$.

Composite–Step Trust–Region SQP Algorithm

1. Compute quasi–normal step n_k.
   → One linear system involving $c_x(x_k)$. Control inexactness!
2. Compute tangential step t_k.

3. Compute new Lagrange multiplier estimate λ_{k+1}.
   → One linear system involving $c_x(x_k)$. Control inexactness!
4. Update penalty parameter ρ_k.

5. Compute $ared_k$, $pred_k$.

6. Decide whether to accept the new iterate $x_{k+1} = x_k + n_k + t_k$, and update Δ_{k+1} from Δ_k, based on $\frac{ared_k}{pred_k}$.

Composite–Step Trust–Region SQP Algorithm

1. Compute quasi–normal step n_k.
   → One linear system involving $c_x(x_k)$. Control inexactness!
2. Compute tangential step t_k.
   → Multiple linear systems involving $c_x(x_k)$. Control inexactness!
3. Compute new Lagrange multiplier estimate λ_{k+1}.
   → One linear system involving $c_x(x_k)$. Control inexactness!
4. Update penalty parameter ρ_k.

5. Compute $ared_k$, $pred_k$.

6. Decide whether to accept the new iterate $x_{k+1} = x_k + n_k + t_k$, and update Δ_{k+1} from Δ_k, based on $\frac{ared_k}{pred_k}$.

Composite–Step Trust–Region SQP Algorithm

1. Compute quasi–normal step n_k.

   → One linear system involving $c_x(x_k)$. Control inexactness!
2. Compute tangential step t_k.

   → Multiple linear systems involving $c_x(x_k)$. Control inexactness!
3. Compute new Lagrange multiplier estimate λ_{k+1}.

   → One linear system involving $c_x(x_k)$. Control inexactness!
4. Update penalty parameter ρ_k.

   → Need to modify penalty parameter update!
5. Compute $ared_k$, $pred_k$.

   → Need to modify the definition of $pred_k$!
6. Decide whether to accept the new iterate $x_{k+1} = x_k + n_k + t_k$, and update Δ_{k+1} from Δ_k, based on $\frac{ared_k}{pred_k}$.

TRSQP Algorithm: Overview of Inexactness

- Iterative linear system solves arise in the computation of:
  (1) Lagrange multipliers, (2) quasi-normal step, (3) tangential step.
- Global convergence theory for TR/SQP methods provides generic conditions that must be satisfied by (1)–(3).
- Our algorithm ties:
  generic conditions $\longleftrightarrow$ *inexactness specific to linear system solves*

- The devised stopping criteria for linear system solves
  - are dynamically adjusted by the SQP algorithm, based on its current progress toward a KKT point,
  - trade gains in feasibility for gains in optimality and vice versa,
  - can be easily implemented and are sufficient to guarantee first–order global convergence of the algorithm,
  - allow for a rigorous integration of preconditioners for KKT systems
  - give a mechanism for matching a prescribed local convergence rate

Lagrange Multipliers

GLOBAL CONVERGENCE CONDITION
The sequence of Lagrange multiplier estimates $\{\lambda_k\}_{k\in\mathbb{N}}$ is bounded.

- 1st–order necessary condition: $\nabla_x f(x_k) + c_x(x_k)^*\lambda = 0$.
  Least-squares estimate $\lambda_k = -\left(c_x(x_k)c_x(x_k)^*\right)^{-1} c_x(x_k)\nabla_x f(x_k)$,
  computed by solving the *augmented system*:

$$\begin{pmatrix} I & c_x(x_k)^* \\ c_x(x_k) & 0 \end{pmatrix}\begin{pmatrix} z \\ \lambda_k \end{pmatrix} = \begin{pmatrix} -\nabla_x f(x_k) \\ 0 \end{pmatrix}$$

- **With inexactness**, $\lambda_k = \lambda_{k-1} + \Delta\lambda$, where $\Delta\lambda$ solves

$$\begin{pmatrix} I & c_x(x_k)^* \\ c_x(x_k) & 0 \end{pmatrix}\begin{pmatrix} z \\ \Delta\lambda \end{pmatrix} = \begin{pmatrix} -\nabla_x f(x_k) - c_x(x_k)^*\lambda_{k-1} + e_k^1 \\ e_k^2 \end{pmatrix}$$

Theorem. Let the sequence $\{\lambda_k\}_{k\in\mathbb{N}}$ of multiplier estimates be generated by $\lambda_k = \lambda_{k-1} + \Delta\lambda$, where $\|e_k^1\|_{\mathcal{X}} + \|e_k^2\|_{\mathcal{C}} \leq \epsilon$ for some $\epsilon > 0$ independent of k. Then $\{\lambda_k\}_{k\in\mathbb{N}}$ is bounded.

REMARK: *relative* stopping tolerance → direct link to $\|\nabla_x f(x_k) + c_x(x_k)^*\lambda_{k-1}\|$

Quasi–Normal Step

GLOBAL CONVERGENCE CONDITIONS

The quasi–normal step n_k must satisfy the boundedness condition

$$\|n_k\|_{\mathcal{X}} \leq \kappa_1 \|c(x_k)\|_{\mathcal{C}},$$

and the fraction of Cauchy decrease (FCD) condition

$$\|c(x_k)\|_{\mathcal{C}}^2 - \|c_x(x_k)n_k + c(x_k)\|_{\mathcal{C}}^2 \geq \kappa_2 \|c(x_k)\|_{\mathcal{C}} \min\left\{\kappa_3 \|c(x_k)\|_{\mathcal{C}}, \Delta_k\right\},$$

where $\kappa_1, \kappa_2, \kappa_3 > 0$ are independent of k.

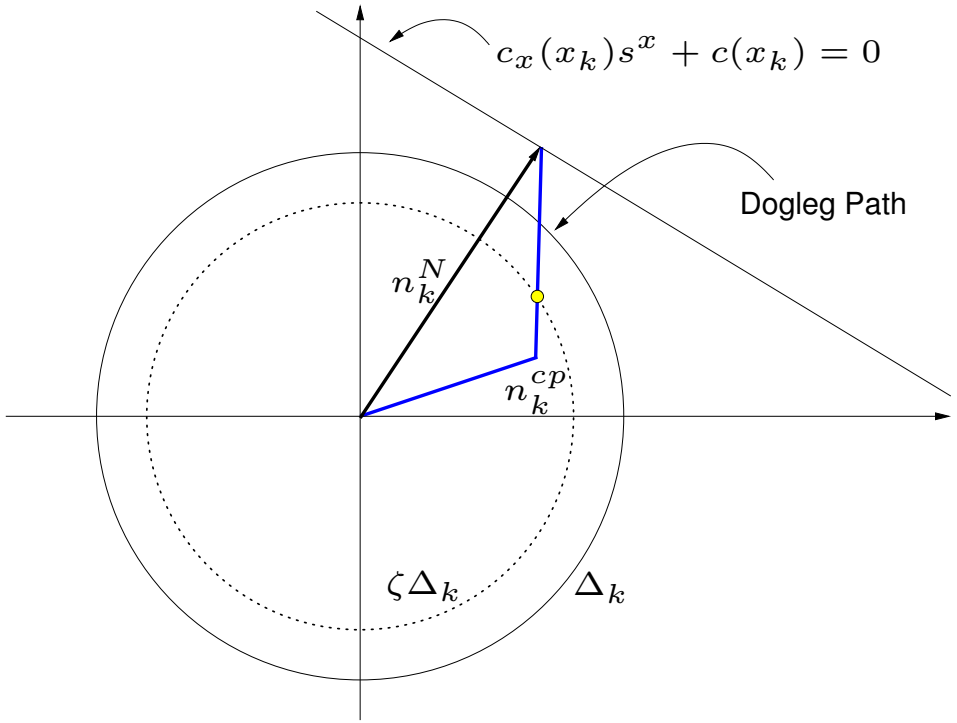
- Let n_k approximately solve the problem:

$$\begin{aligned} \min \quad & \|c_x(x_k)n + c(x_k)\|_{\mathcal{Y}}^2 \\ \text{s.t.} \quad & \|n\|_{\mathcal{X}} \leq \zeta \Delta_k. \end{aligned}$$

- A practical approach: Powell's dogleg method.

Quasi–Normal Step

$$n_k^{cp} = \min_{\alpha \geq 0} \quad \|c_x(x_k)n + c(x_k)\|_C^2$$
$$\text{s.t.} \quad n = -\alpha c_x(x_k)^* c(x_k)$$



n_k^N = minimum norm solution of

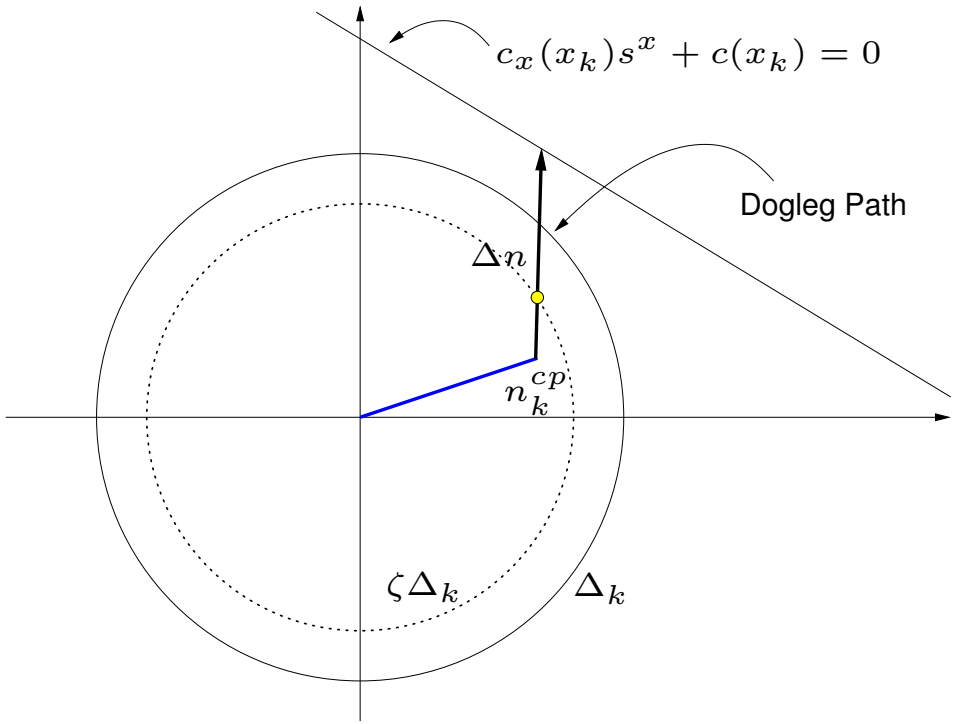
$$\min \|c_x(x_k)n + c(x_k)\|_C^2$$

The minimum norm solution n_k^N can be computed by solving:

$$\begin{pmatrix} I & c_x(x_k)^* \\ c_x(x_k) & 0 \end{pmatrix} \begin{pmatrix} n_k^N \\ y \end{pmatrix} = \begin{pmatrix} 0 \\ -c(x_k) \end{pmatrix}.$$

Quasi–Normal Step

$$n_k^{cp} = \min_{\alpha \geq 0} \quad \|c_x(x_k)n + c(x_k)\|_C^2$$
$$\text{s.t.} \quad n = -\alpha c_x(x_k)^* c(x_k)$$



n_k^N = minimum norm solution of

$$\min \|c_x(x_k)n + c(x_k)\|_C^2$$

With inexactness, we solve for $\Delta n = n_k^N - n_k^{cp}$:

$$\begin{pmatrix} I & c_x(x_k)^* \\ c_x(x_k) & 0 \end{pmatrix} \begin{pmatrix} \Delta n \\ y \end{pmatrix} = \begin{pmatrix} -n_k^{cp} + e_k^1 \\ -c_x(x_k)n_k^{cp} - c(x_k) + e_k^2 \end{pmatrix}.$$

Theorem. If $\|e_k^1\|_C^2 + \|e_k^2\|_C^2 \leq \|c_x(x_k)n_k^{cp} + c(x_k)\|_C^2$, then the inexact quasi-normal step n_k satisfies boundedness and FCD conditions.

Skip Tangential Step

Tangential Step

- The exact model requires that t_k approximately solve the problem:

$$\begin{aligned} \min \quad & \frac{1}{2}\left\langle H_k(t+n_k), t+n_k\right\rangle_{\mathcal{X}} + \left\langle \nabla_x \mathcal{L}_k, t+n_k\right\rangle_{\mathcal{X}} \\ \text{s.t.} \quad & c_x(x_k)t = 0 \\ & \|t+n_k\|_{\mathcal{X}} \leq \Delta_k. \end{aligned}$$

- Assume that there exists a bounded linear operator $W_k : \mathcal{Z} \to \mathcal{X}$, where $\mathcal{Z}$ is a Hilbert space, such that $\text{Range}(W_k) = \text{Null}(c_x(x_k))$.
  $\rightarrow$ Covers all existing implementations for handling $c_x(x_k)t = 0$.
- Drop constant term from the QP, ignore n_k in the trust–region constraint, set $g_k = H_k n_k + \nabla_x \mathcal{L}_k$.
- Use a full space approach, in which the CG operator is H_k (exact), and the inexactness is moved into an inexact projector $\mathcal{W}_k$.

$$\begin{aligned} \min \quad & \frac{1}{2}\left\langle H_k t, t\right\rangle_{\mathcal{X}} + \left\langle g_k, t\right\rangle_{\mathcal{X}} \\ \text{s.t.} \quad & t \in \text{Range}(\mathcal{W}_k),\ \|t\|_{\mathcal{X}} \leq \Delta_k. \end{aligned}$$

- The application of $\mathcal{W}_k$ requires linear system solves.
  Example: $\mathcal{W}_k$ is an orthogonal projector onto $\text{Null}(c_x(x_k))$.

Tangential Step

Steihaug–Toint Flexible Full–Space CG

0. Let $t_0 = 0 \in \mathcal{X}$. Let $r_0 = g_k$. Given tol > 0.
1. For $i = 0, 1, 2, \ldots$
   - 1.1 Compute $z_i = \mathcal{W}_k(r_i)$. If $\|z_i\|_{\mathcal{X}} <$ tol, stop and return t_i.
   - 1.2 $p_i = -z_i + \sum_{j=0}^{i-1} \frac{\langle z_i, H_k p_j \rangle_{\mathcal{X}}}{\langle p_j, H_k p_j \rangle_{\mathcal{X}}} p_j$
   - 1.3 If $\langle r_i, p_i \rangle_{\mathcal{X}} \neq 0$ and $\langle p_i, H p_i \rangle_{\mathcal{X}} \leq 0$, compute θ such that $\text{sign}(\theta) = \text{sign}(-\langle r_i, p_i \rangle_{\mathcal{X}})$ and $\|t_i + \theta p_i\|_{\mathcal{X}} = \Delta_k$, and return $t_i + \theta p_i$.

     If $\langle r_i, p_i \rangle_{\mathcal{X}} = 0$ and $\langle p_i, H p_i \rangle_{\mathcal{X}} < 0$, compute θ such that $\|t_i + \theta p_i\|_{\mathcal{X}} = \Delta_k$, and return $t_i + \theta p_i$.
   - 1.4 If $\langle r_i, p_i \rangle_{\mathcal{X}} = 0$, stop and return t_i.
   - 1.5 $\alpha_i = -\langle r_i, p_i \rangle_{\mathcal{X}} / \langle p_i, H_k p_i \rangle_{\mathcal{X}}$
   - 1.6 $t_{i+1} = t_i + \alpha_i p_i$
   - 1.7 If $\|t_{i+1}\|_{\mathcal{X}} \geq \Delta_k$, compute θ such that $\text{sign}(\theta) = \text{sign}(-\langle r_i, p_i \rangle_{\mathcal{X}})$ and $\|t_i + \theta p_i\|_{\mathcal{X}} = \Delta_k$, and return $t_i + \theta p_i$.
   - 1.8 $r_{i+1} = r_i + \alpha_i H_k p_i$

Tangential Step

Steihaug–Toint Flexible Full–Space CG

0. Let $t_0 = 0 \in \mathcal{X}$. Let $r_0 = g_k$. Given tol > 0.
1. For $i = 0, 1, 2, \ldots$
   - 1.1 Compute $z_i = \mathcal{W}_k(r_i)$. If $\|z_i\|_{\mathcal{X}} <$ tol, stop and return t_i.
   - 1.2 $p_i = -z_i + \sum_{j=0}^{i-1} \frac{\langle z_i, H_k p_j \rangle_{\mathcal{X}}}{\langle p_j, H_k p_j \rangle_{\mathcal{X}}} p_j$
   - 1.3 If $\langle r_i, p_i \rangle_{\mathcal{X}} \neq 0$ and $\langle p_i, H p_i \rangle_{\mathcal{X}} \leq 0$, compute θ such that $\text{sign}(\theta) = \text{sign}(-\langle r_i, p_i \rangle_{\mathcal{X}})$ and $\|t_i + \theta p_i\|_{\mathcal{X}} = \Delta_k$, and return $t_i + \theta p_i$.

     If $\langle r_i, p_i \rangle_{\mathcal{X}} = 0$ and $\langle p_i, H p_i \rangle_{\mathcal{X}} < 0$, compute θ such that $\|t_i + \theta p_i\|_{\mathcal{X}} = \Delta_k$, and return $t_i + \theta p_i$.
   - 1.4 If $\langle r_i, p_i \rangle_{\mathcal{X}} = 0$, stop and return t_i.
   - 1.5 $\alpha_i = -\langle r_i, p_i \rangle_{\mathcal{X}} / \langle p_i, H_k p_i \rangle_{\mathcal{X}}$
   - 1.6 $t_{i+1} = t_i + \alpha_i p_i$
   - 1.7 If $\|t_{i+1}\|_{\mathcal{X}} \geq \Delta_k$, compute θ such that $\text{sign}(\theta) = \text{sign}(-\langle r_i, p_i \rangle_{\mathcal{X}})$ and $\|t_i + \theta p_i\|_{\mathcal{X}} = \Delta_k$, and return $t_i + \theta p_i$.
   - 1.8 $r_{i+1} = r_i + \alpha_i H_k p_i$

Tangential Step

Steihaug–Toint Flexible Full–Space CG

0. Let $t_0 = 0 \in \mathcal{X}$. Let $r_0 = g_k$. Given tol > 0.
1. For $i = 0, 1, 2, \ldots$
   - 1.1 Compute $z_i = \mathcal{W}_k(r_i)$. If $\|z_i\|_{\mathcal{X}} <$ tol, stop and return t_i.
   - 1.2 $p_i = -z_i + \sum_{j=0}^{i-1} \frac{\langle z_i, H_k p_j \rangle_{\mathcal{X}}}{\langle p_j, H_k p_j \rangle_{\mathcal{X}}} p_j$
   - 1.3 If $\langle r_i, p_i \rangle_{\mathcal{X}} \neq 0$ and $\langle p_i, H p_i \rangle_{\mathcal{X}} \leq 0$, compute θ such that $\text{sign}(\theta) = \text{sign}(-\langle r_i, p_i \rangle_{\mathcal{X}})$ and $\|t_i + \theta p_i\|_{\mathcal{X}} = \Delta_k$, and return $t_i + \theta p_i$.

     If $\langle r_i, p_i \rangle_{\mathcal{X}} = 0$ and $\langle p_i, H p_i \rangle_{\mathcal{X}} < 0$, compute θ such that $\|t_i + \theta p_i\|_{\mathcal{X}} = \Delta_k$, and return $t_i + \theta p_i$.
   - 1.4 If $\langle r_i, p_i \rangle_{\mathcal{X}} = 0$, stop and return t_i.
   - 1.5 $\alpha_i = -\langle r_i, p_i \rangle_{\mathcal{X}} / \langle p_i, H_k p_i \rangle_{\mathcal{X}}$
   - 1.6 $t_{i+1} = t_i + \alpha_i p_i$
   - 1.7 If $\|t_{i+1}\|_{\mathcal{X}} \geq \Delta_k$, compute θ such that $\text{sign}(\theta) = \text{sign}(-\langle r_i, p_i \rangle_{\mathcal{X}})$ and $\|t_i + \theta p_i\|_{\mathcal{X}} = \Delta_k$, and return $t_i + \theta p_i$.
   - 1.8 $r_{i+1} = r_i + \alpha_i H_k p_i$

Tangential Step

Steihaug–Toint Flexible Full–Space CG

0. Let $t_0 = 0 \in \mathcal{X}$. Let $r_0 = g_k$. Given tol > 0.
1. For $i = 0, 1, 2, \ldots$
   - 1.1 Compute $z_i = \mathcal{W}_k(r_i)$. If $\|z_i\|_{\mathcal{X}} <$ tol, stop and return t_i.
   - 1.2 $p_i = -z_i + \sum_{j=0}^{i-1} \frac{\langle z_i, H_k p_j \rangle_{\mathcal{X}}}{\langle p_j, H_k p_j \rangle_{\mathcal{X}}} p_j$
   - 1.3 If $\langle r_i, p_i \rangle_{\mathcal{X}} \neq 0$ and $\langle p_i, H p_i \rangle_{\mathcal{X}} \leq 0$, compute θ such that $\text{sign}(\theta) = \text{sign}(-\langle r_i, p_i \rangle_{\mathcal{X}})$ and $\|t_i + \theta p_i\|_{\mathcal{X}} = \Delta_k$, and return $t_i + \theta p_i$.

     If $\langle r_i, p_i \rangle_{\mathcal{X}} = 0$ and $\langle p_i, H p_i \rangle_{\mathcal{X}} < 0$, compute θ such that $\|t_i + \theta p_i\|_{\mathcal{X}} = \Delta_k$, and return $t_i + \theta p_i$.
   - 1.4 If $\langle r_i, p_i \rangle_{\mathcal{X}} = 0$, stop and return t_i.
   - 1.5 $\alpha_i = -\langle r_i, p_i \rangle_{\mathcal{X}} / \langle p_i, H_k p_i \rangle_{\mathcal{X}}$
   - 1.6 $t_{i+1} = t_i + \alpha_i p_i$
   - 1.7 If $\|t_{i+1}\|_{\mathcal{X}} \geq \Delta_k$, compute θ such that $\text{sign}(\theta) = \text{sign}(-\langle r_i, p_i \rangle_{\mathcal{X}})$ and $\|t_i + \theta p_i\|_{\mathcal{X}} = \Delta_k$, and return $t_i + \theta p_i$.
   - 1.8 $r_{i+1} = r_i + \alpha_i H_k p_i$

Tangential Step

Global Convergence Requirements

Assume that $\mathcal{W}_k(r_i) = \widetilde{W}_k r_i$ for every iteration i of the flexible STCG algorithm (*proof skipped!*). The inexact reduced–space tangential step w_k and the inexact projection operator $\widetilde{W}_k$ must satisfy

(C1) $$\|\widetilde{W}_k{}^* g_k - W_k^* g_k\|_{\mathcal{X}} \leq \kappa_1 \min\left(\|\widetilde{W}_k{}^* g_k\|_{\mathcal{X}}, \Delta_k\right),$$

(C2) $$\left\langle \widetilde{W}_k{}^* H_k \widetilde{W}_k w_k, w_k \right\rangle_{\mathcal{X}} \leq \kappa_2 \|w_k\|_{\mathcal{X}}^2,$$

(C3) $$-\frac{1}{2}\left\langle \widetilde{W}_k{}^* H_k \widetilde{W}_k w_k, w_k \right\rangle_{\mathcal{X}} - \left\langle \widetilde{W}_k{}^* g_k, w_k \right\rangle_{\mathcal{X}} \geq \kappa_3 \|\widetilde{W}_k{}^* g_k\|_{\mathcal{X}} \min\left\{\kappa_4 \|\widetilde{W}_k{}^* g_k\|_{\mathcal{X}}, \kappa_5 \Delta_k\right\},$$

for positive constants $\kappa_1, ..., \kappa_5$ independent of k.

Tangential Step

Application of the Inexact Operator $\widetilde{W}_k$

Recall:

(i) At every iteration k of the SQP algorithm, flexible STCG is called.

(ii) At every STCG iteration i, we compute iteratively an inexact projected residual $z_i = \mathcal{W}_k(r_i) = \widetilde{W}_k r_i$ such that

$$\begin{pmatrix} I & c_x(x_k)^* \\ c_x(x_k) & 0 \end{pmatrix} \begin{pmatrix} z_i \\ y \end{pmatrix} = \begin{pmatrix} r_i \\ 0 \end{pmatrix} + \begin{pmatrix} e_i^1 \\ e_i^2 \end{pmatrix}.$$

Tangential Step

Application of the Inexact Operator $\widetilde{W}_k$

Recall:

(i) At every iteration k of the SQP algorithm, flexible STCG is called.

(ii) At every STCG iteration i, we compute iteratively an inexact projected residual $z_i = \mathcal{W}_k(r_i) = \widetilde{W}_k r_i$ such that

$$\begin{pmatrix} I & c_x(x_k)^* \\ c_x(x_k) & 0 \end{pmatrix} \begin{pmatrix} z_i \\ y \end{pmatrix} = \begin{pmatrix} r_i \\ 0 \end{pmatrix} + \begin{pmatrix} e_i^1 \\ e_i^2 \end{pmatrix}.$$

Theorem. If the augmented system solver is stopped at iteration m satisfying

$$\|e_i^{(m)}\| \leq \underbrace{\min\left\{ \frac{\|\widetilde{W}_k g_k\|}{\|g_k\|}, \ \frac{\Delta_k}{\|g_k\|}, \ \beta \right\}}_{\gamma} \|z_i^{(m)}\|,$$

then the global SQP convergence requirements (C1)–(C3) are satisfied.†

Simplified Model Problem

$$\text{minimize} \quad \mathcal{J} = \frac{1}{2}\int_{\Gamma_o}(J(x)\cdot\nu - \widehat{J}(x)\cdot\nu)^2 dx \; + \; \frac{\alpha}{2}\int_{\Omega}(u(x)-\widehat{u}(x))^2 dx$$

subject to

$$\begin{aligned} J_n(x) &= \mu_n(\nabla n(x) + n(x)\nabla y(x)) \\ J_p(x) &= \mu_p(\nabla p(x) - p(x)\nabla y(x)) \\ \nabla\cdot J_n(x) &= 0 \\ \nabla\cdot J_p(x) &= 0 \\ -\nabla\cdot(k(x)\nabla y(x)) &= n(x) - p(x) - u(x). \end{aligned}$$

Simplified Model Problem

$$\text{minimize} \quad \mathcal{J} = \frac{1}{2}\int_{\Gamma_o}(J(x)\cdot\nu - \widehat{J}(x)\cdot\nu)^2 dx \; + \; \frac{\alpha}{2}\int_{\Omega}(u(x)-\widehat{u}(x))^2 dx$$

subject to

$$\begin{aligned} J_n(x) &= \mu_n(\nabla n(x) + n(x)\nabla y(x)) \\ J_p(x) &= \mu_p(\nabla p(x) - p(x)\nabla y(x)) \\ \nabla\cdot J_n(x) &= 0 \\ \nabla\cdot J_p(x) &= 0 \\ -\nabla\cdot(k(x)\nabla y(x)) &= e^{y(x)} - e^{-y(x)} - u(x). \end{aligned}$$

$$\underbrace{n \to e^{y}, \quad p \to e^{-y}}$$

"Slotboom variables"

Simplified Model Problem

$$\text{minimize} \quad \mathcal{J} = \frac{1}{2}\int_{\Gamma_o} (\nabla y(x)\cdot\nu - \nabla\widehat{y}(x)\cdot\nu)^2 dx + \frac{\alpha}{2}\int_{\Omega}(u(x)-\widehat{u}(x))^2 dx$$

subject to

$$J_n(x) = \mu_n(\nabla n(x) + n(x)\nabla y(x))$$
$$J_p(x) = \mu_p(\nabla p(x) - p(x)\nabla y(x))$$
$$\nabla \cdot J_n(x) = 0$$
$$\nabla \cdot J_p(x) = 0$$
$$-\nabla\cdot(k(x)\nabla y(x)) = e^{y(x)} - e^{-y(x)} - u(x).$$

$$\underbrace{n \to e^{y}, \quad p \to e^{-y}}$$

"Slotboom variables"

$$J \to \nabla y$$

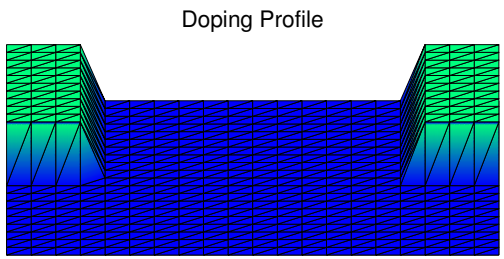
Experimental Setup

$$\text{minimize} \quad \mathcal{J} = \frac{1}{2}\int_{\Gamma_o} (\nabla y(x)\cdot\nu - \nabla\widehat{y}(x)\cdot\nu)^2 dx + \frac{\alpha}{2}\int_{\Omega}(u(x)-\widehat{u}(x))^2 dx$$

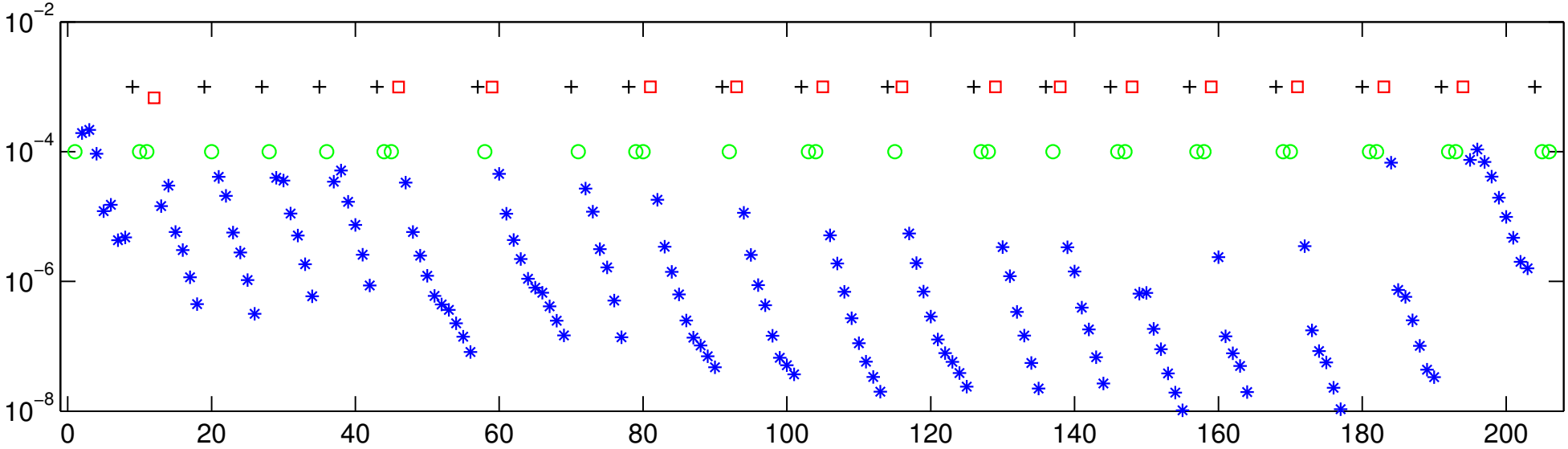
subject to

$$\begin{aligned} -\nabla\cdot(k(x)\nabla y(x)) &= e^{y(x)} - e^{-y(x)} - u(x) && \text{in } \Omega \\ y(x) &= y_D(x) && \text{on } \Gamma_D \\ (k(x)\nabla y(x))\cdot\nu &= 0 && \text{on } \Gamma_N. \end{aligned}$$

- Galerkin FE discretization with linear elements
- use a simple fixed target potential $\widehat{y}$
- several target doping profiles $\widehat{u}$:
  1/1, 1e-1/1e1, 1e-2/1e2, 1e-3/1e3, ...
- linear solver: GMRES with ILU preconditioner
- SQP convergence criteria:
  $\|c(x_k)\| < 10^{-6}$, $\|\nabla_x \mathcal{L}(x_k,\lambda_k)\| < 10^{-6}$.



Inexactness Control: 1e-3/1e3 Doping



□ Lagrange multiplier tolerances.

* Tangential step tolerances.

○ Quasi–normal step tolerances.

\+ Additional projection tolerances.

		fixed relative tol's				
	inx. ctrl	1e-10	1e-8	1e-6	1e-4	1e-2
converges	YES	YES	YES	NO	NO	NO
GMRES iter's	1384	7847	6177	—	—	—
SQP iter's	19	20	20	—	—	—

Inexactness Control: Robustness

doping	inx. ctrl	1e-10	1e-8	1e-6	1e-4	1e-2
1 / 1	4	4	4	F	F	F
1e1 / 1e-1	8	8	8	F	F	F
1e2 / 1e-2	10	11	11	F	F	F
1e3 / 1e-3	19	20	20	F	F	F
1e4 / 1e-4	23	24	24	F	F	F
1e5 / 1e-5	26	29	29	F	F	F

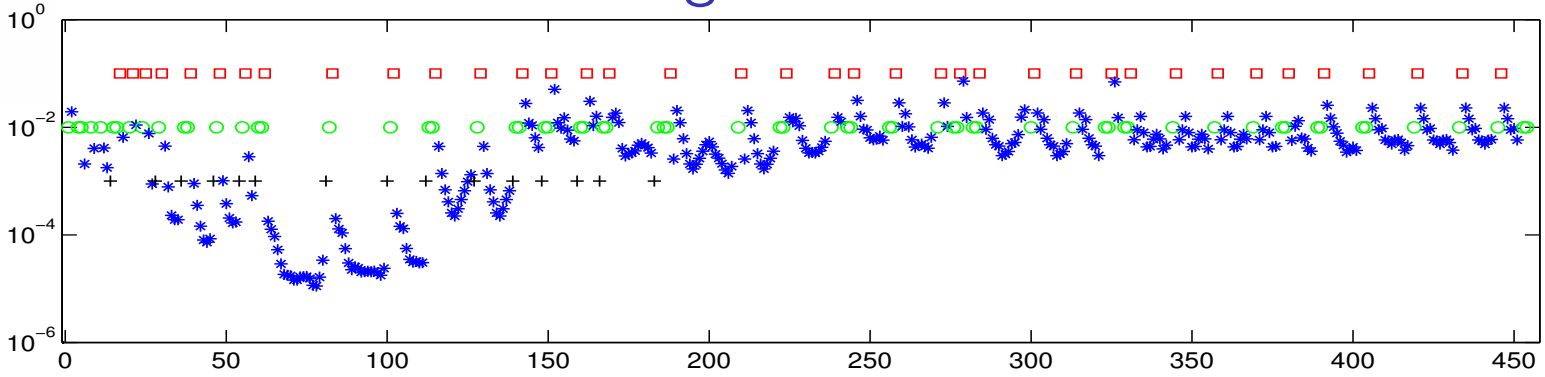
Table: Number of SQP iterations with respect to changes in the doping profile. Inexactness control vs. fixed stopping tolerances. 'F' indicates failure.

doping	inx. ctrl	1e-10	1e-8	1e-6	1e-4	1e-2
1 / 1	683	3105	2564	–	–	–
1e1 / 1e-1	534	3058	2417	–	–	–
1e2 / 1e-2	845	3574	2753	–	–	–
1e3 / 1e-3	1384	7847	6177	–	–	–
1e4 / 1e-4	2678	9107	6988	–	–	–
1e5 / 1e-5	2113	9542	7564	–	–	–

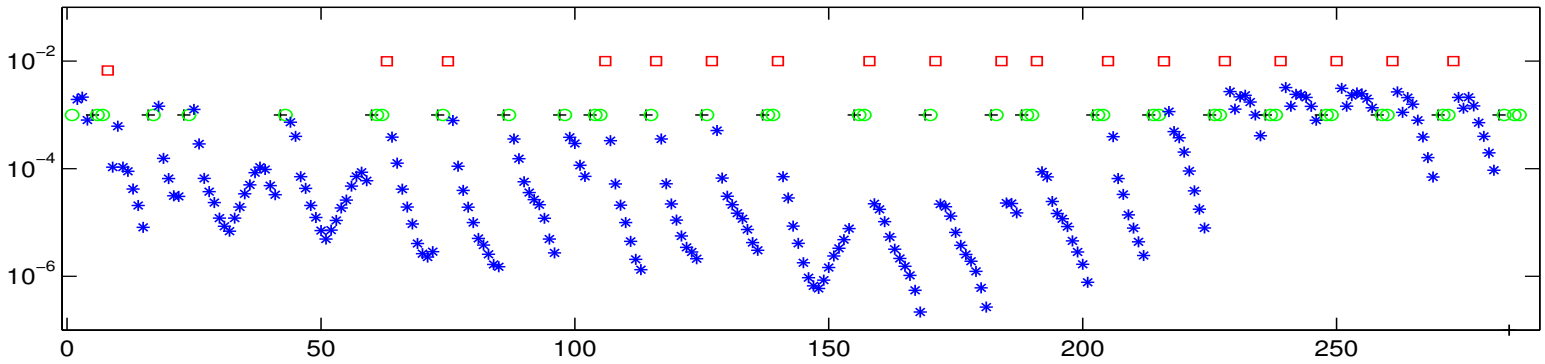
Table: Total number of GMRES iterations.

The Local Convergence "Knob"

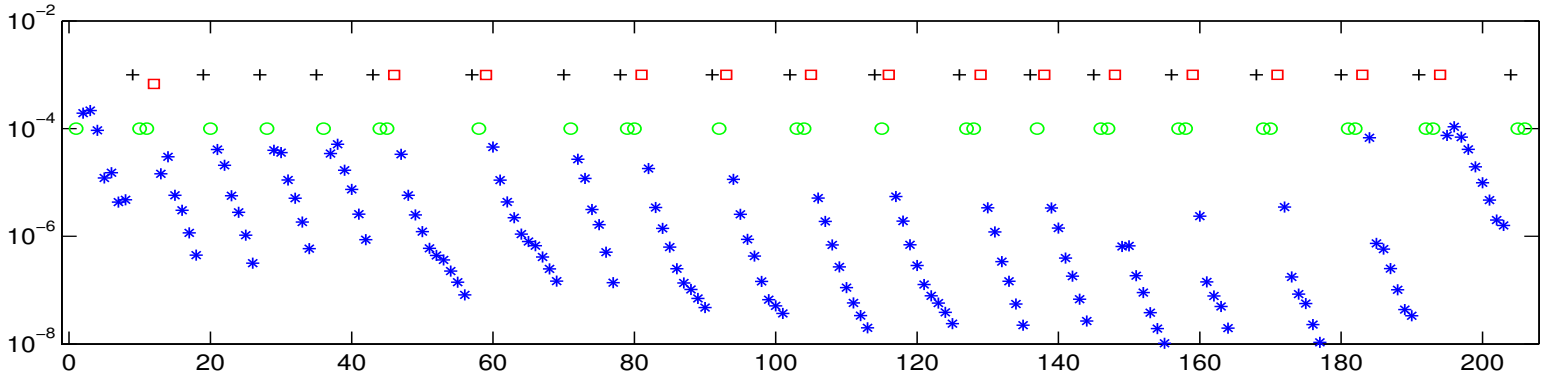
SLOW LOCAL
3.5



MODERATE LOCAL
5.7



FAST LOCAL
6.7



FIXED RELATIVE TOLERANCE OF 1e-12 → 20.1 !!

Summary

- Large–scale optimization algorithms *must* control stopping conditions for all underlying iterative linear system solves.
- For trust–region SQP algorithms, global convergence can be guaranteed through a mechanism of inexpensive and easily implementable stopping conditions for linear system solvers.
- Completely eliminated the need to "guess" fixed solver tolerances; the optimization algorithm automatically controls inexactness based on its progress toward a KKT point.
- Dynamic stopping conditions significantly reduce oversolves and allow the user to match a prescribed local convergence rate.