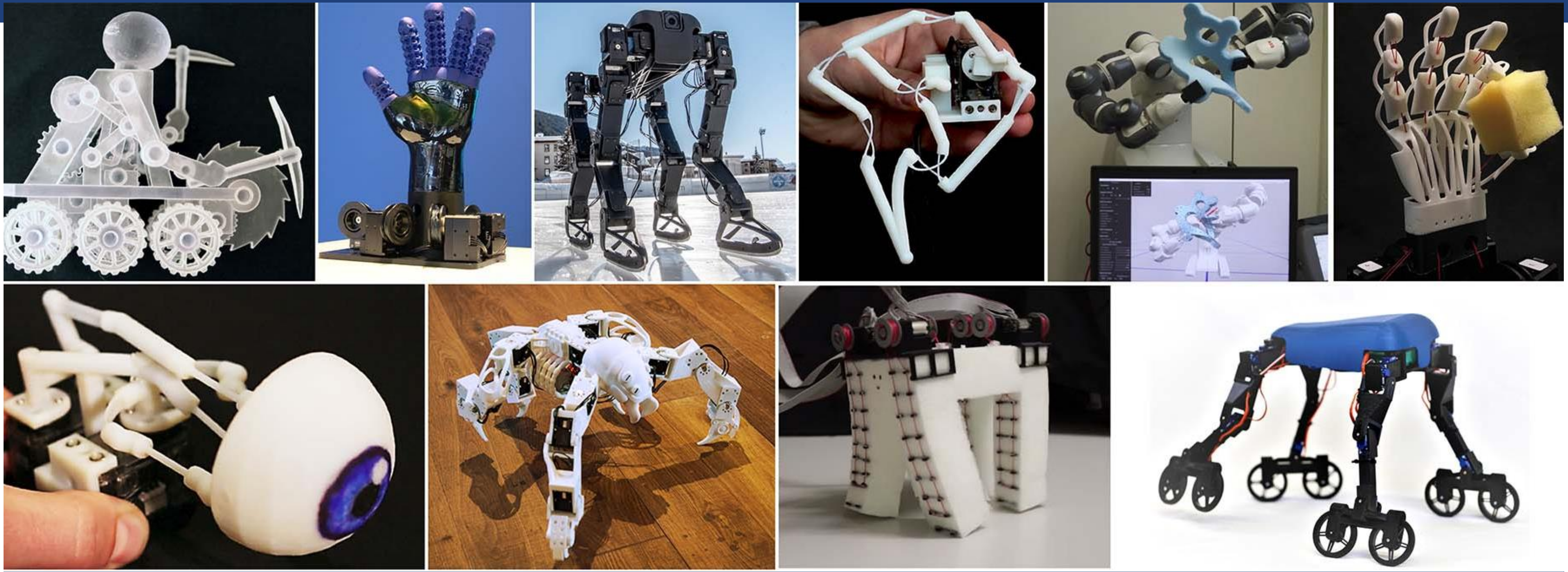
ETH zürich



Computational Models of Motion

Constrained Optimization & Sensitivity Analysis

CRL

Unconstrained Minimization

Minimization problem: find minimum of objective function $f(\boldsymbol{x}): \boldsymbol{R}^n \rightarrow \boldsymbol{R}$

- Solve inverse kinematics problem
- Find static equilibrium state
- Solve discrete equations of motion (time integration)
- …

However, for many problems

- we have to find compromise between multiple objectives, $f = \sum f_i$

  $\rightarrow$ minimizer of f is not a minimizer of any individual objective f_i in general
- there are conditions that must be satisfied

  $\rightarrow$ constraints

Constraints in Simulation

Constraints model conditions that **must** *be satisfied* (*no deviation allowed)*

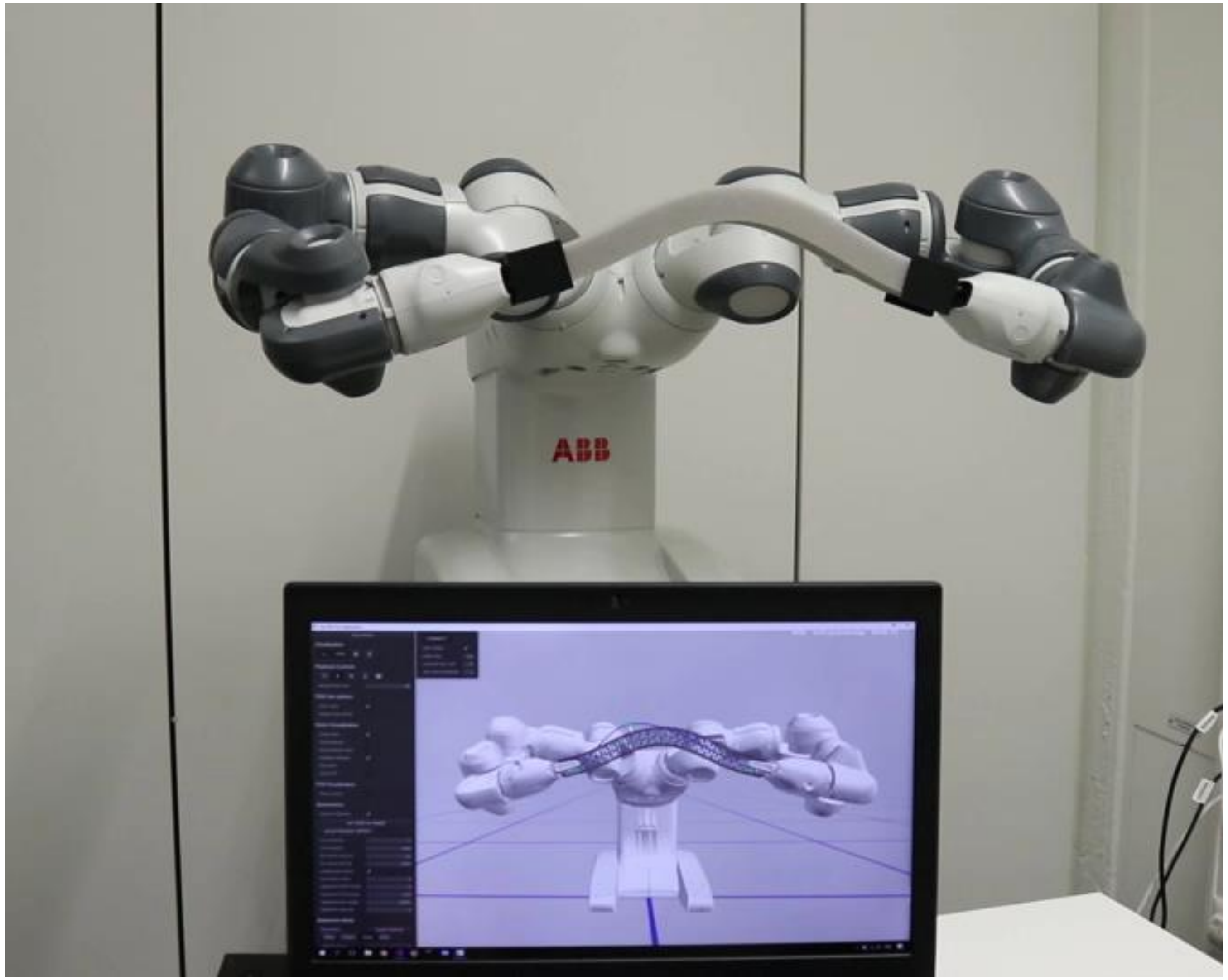
- Parts of mechanical systems can have fixed position/orientation
- Joint limits
- Motor velocity or torque limits
- Non-interpenetration constraints
- Contact force magnitude and direction (frictional contact)
- …

Constraints in Design, Manipulation, and Motion Planning

Many applications in robotics and animation give rise to formulations that model physical feasibility of the result using constraints.

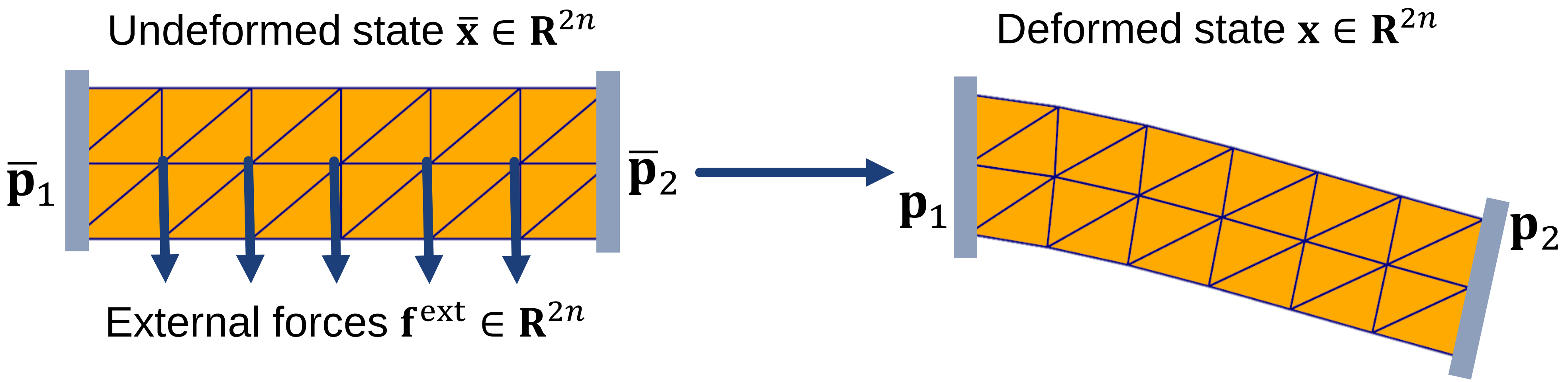
- Find rest shape such that resulting equilibrium state is as close as possible to target shape
- Find end-effector trajectories that achieve given manipulation goals
- Find control forces such that resulting motion is as close as possible to target motion
- …

Robotic Manipulation of Soft Materials



[CRL, IROS '18]

Robotic Manipulation: 2D Elastic Bar



Variables:

- Parameters $\mathbf{p}_1 = (x_1, y_1, \theta_1)$ and $\mathbf{p}_2 = (x_2, y_2, \theta_2)$ describing the positions and orientations of the two end-effectors (boundaries)
- Equilibrium positions $\boldsymbol{x} = \boldsymbol{R}^{2n}$ for the n nodes

Forward problem: given $\mathbf{p}$, compute equilibrium configuration $\mathbf{x}$ by solving

$$\mathbf{f}^{\text{ext}} + \mathbf{f}^{\text{int}}(\bar{\mathbf{x}}, \mathbf{x}) = \mathbf{0} \qquad \text{for all internal nodes}$$

$$\mathbf{x} = \mathbf{p}(\bar{\mathbf{x}}) \qquad \text{for all boundary nodes}$$

Robotic Manipulation: 2D Elastic Bar

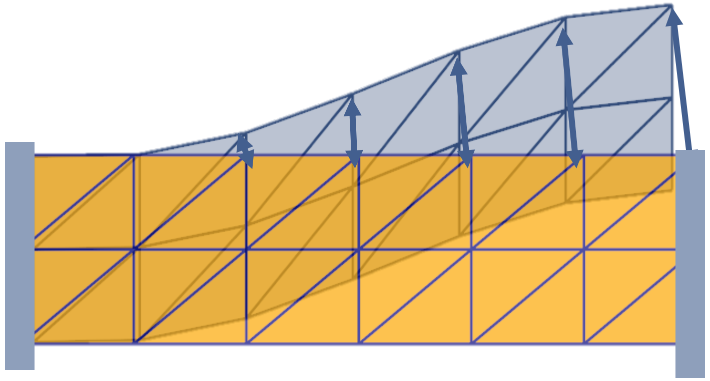
Observation: changing the end-effector parameters $\mathbf{p}$ changes the equilibrium state $\mathbf{x}$

How can we determine parameters that lead to a desired equilibrium state?

Objective

- Introduce *objective* that quantifies distance to target

$$T(\mathbf{x}) = \frac{1}{2}\,\|\mathbf{x} - \mathbf{x}^{\mathrm{tar}}\|^2$$



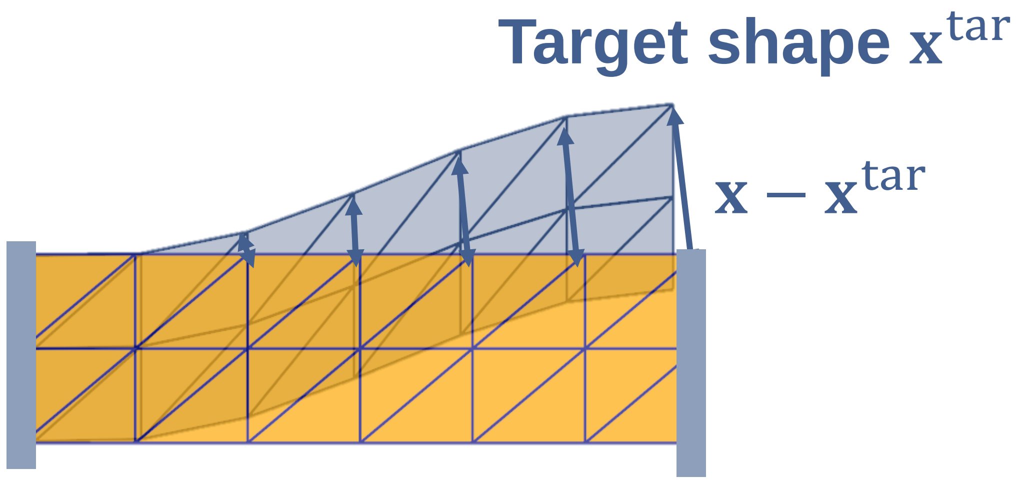
- **Goal**: find $\mathbf{x}$ and $\mathbf{p}$ such that $\mathbf{x}$ minimizes distance to target
- **Constraints**: $\mathbf{x}$ has to be an equilibrium state for $\mathbf{p}$, i.e.,

$$\mathbf{C}(\mathbf{x},\mathbf{p}) = \mathbf{0} \quad \begin{cases} \mathbf{f}^{\mathrm{ext}} + \mathbf{f}^{\mathrm{int}}(\bar{\mathbf{x}},\mathbf{x}) = \mathbf{0} & \text{for all internal nodes} \\ \mathbf{x} = \mathbf{p}(\bar{\mathbf{x}}) & \text{for all boundary nodes} \end{cases}$$

Objective

- Introduce *objective* that quantifies distance to target

$$T(\mathbf{x}) = \frac{1}{2}\,\|\mathbf{x} - \mathbf{x}^{\mathrm{tar}}\|^2$$



Plain language description:

from all possible equilibrium states $\mathbf{x}$,

i.e., those $\mathbf{x}$ for which there exists $\mathbf{p}$ such that $\mathbf{C}(\mathbf{x}, \mathbf{p}) = \mathbf{0}$,

find the one that minimizes $T(\mathbf{x})$.

Constrained Optimization Basics

Optimization

- Optimization problem = minimization problem + constraints
- Generic form

$$\min_{\boldsymbol{x}} f(\boldsymbol{x}) \qquad \text{s.t.} \quad \boldsymbol{C}(\boldsymbol{x}) = 0$$

  - Objective function $f(\boldsymbol{x}): \mathbf{R}^n \rightarrow \mathbf{R}$
  - Unknowns $\boldsymbol{x} \in \mathbf{R}^n$
  - Constraints $\boldsymbol{C}(\boldsymbol{x}): \mathbf{R}^n \rightarrow \mathbf{R}^m$

How can we solve such an optimization problem?

What characterizes solutions to optimization problems?

Optimization – First Naïve Approach

Generic optimization problem

$$\min_{\boldsymbol{x}} f(\boldsymbol{x}) \qquad \text{s.t.} \quad \boldsymbol{C}(\boldsymbol{x}) = 0$$

- ~~Objective should be at minimum~~ $\cancel{\rightarrow \nabla f(\boldsymbol{x}) = \boldsymbol{0}}$
- Constraints should be satisfied → $\boldsymbol{C}(\boldsymbol{x}) = \boldsymbol{0}$
- ~~Formulate as~~ $\cancel{\boldsymbol{g}(\boldsymbol{x}) = \left(\nabla f(\boldsymbol{x})^T, \boldsymbol{C}(\boldsymbol{x})^T\right)^T = \boldsymbol{0}}$ ~~and solve with Newton~~

- **Problem**: in general, there is no $\boldsymbol{x}$ such that $\boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{0}$.

Counter Examples

Example 1: $\min_{x} x^2$ s.t. $x = 1$

- Solution is $x = 1$, but $\nabla f(1) = 2$

Example 2: $\min_{x,y} x^2 + y^2$ s.t. $x + y = 1$

- Solution is $(x, y) = (\frac{1}{2}, \frac{1}{2})$, but $\nabla f\left(\frac{1}{2}, \frac{1}{2}\right) = (1,1)$

A Single Equality Constraint

- Assume that for given $\boldsymbol{x}$ the constraint is satisfied, $C(\boldsymbol{x}) = \boldsymbol{0}$, but $\boldsymbol{x}$ is not the optimum.
- Then $\exists \boldsymbol{dx}$ such that $C(\boldsymbol{x} + \boldsymbol{dx}) = \boldsymbol{0}$ and $f(\boldsymbol{x} + \boldsymbol{dx}) < f(\boldsymbol{x})$
- For linear constraints $C(\boldsymbol{x} + \boldsymbol{dx}) = C(\boldsymbol{x}) + \nabla C^T \boldsymbol{dx}$
- For small $|\boldsymbol{dx}|$, $f(\boldsymbol{x} + \boldsymbol{dx}) < f(\boldsymbol{x})$ implies that $\nabla f(\boldsymbol{x})^T \boldsymbol{dx} < \boldsymbol{0}$
- In summary, we have

$$\nabla C^T \boldsymbol{dx} = \boldsymbol{0} \qquad \text{and} \qquad \nabla f(\boldsymbol{x})^T \boldsymbol{dx} < \boldsymbol{0}$$

- Therefore, we can take a step into the direction $\boldsymbol{dx}$ to improve the objective without violating the constraint.

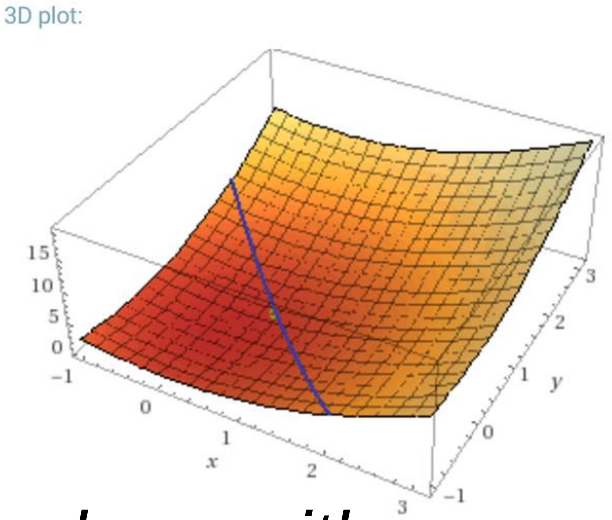
A Single Equality Constraint

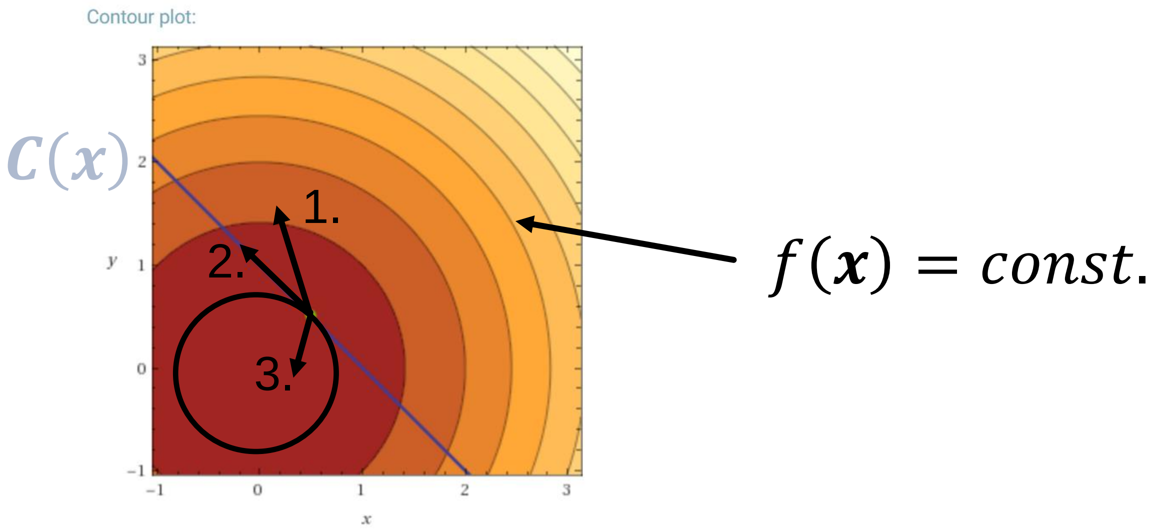
- Now assume that $\boldsymbol{x}$ is the optimum
- Then $\nexists \boldsymbol{dx}$ such that $C(\boldsymbol{x} + \boldsymbol{dx}) = \mathbf{0}$ and $f(\boldsymbol{x} + \boldsymbol{dx}) < f(\boldsymbol{x})$

A Single Equality Constraint

$$\nexists \boldsymbol{dx} \text{ such that } C(\boldsymbol{x}+\boldsymbol{dx}) = 0 \text{ and } f(\boldsymbol{x}+\boldsymbol{dx}) < f(\boldsymbol{x})$$

What does that mean?





For all dx, we have either

1. $f(\boldsymbol{x}+\boldsymbol{dx}) \geq f(\boldsymbol{x})$ and $C(\boldsymbol{x}+\boldsymbol{dx}) \neq \boldsymbol{0}$
2. $C(\boldsymbol{x}+\boldsymbol{dx}) = \boldsymbol{0}$ but $f(\boldsymbol{x}+\boldsymbol{dx}) \geq f(\boldsymbol{x})$
3. $f(\boldsymbol{x}+\boldsymbol{dx}) < f(\boldsymbol{x})$ but $C(\boldsymbol{x}+\boldsymbol{dx}) \neq \boldsymbol{0}$

A Single Equality Constraint

- Now assume that $\boldsymbol{x}$ is the optimum
- Then $\nexists \boldsymbol{dx}$ such that $C(\boldsymbol{x} + \boldsymbol{dx}) = 0$ and $f(\boldsymbol{x} + \boldsymbol{dx}) < f(\boldsymbol{x})$
- Equivalently, we can say that there $\nexists \boldsymbol{dx}$ such that both

$$\nabla C^T \boldsymbol{dx} = \boldsymbol{0} \qquad \text{and} \qquad \nabla f(\boldsymbol{x})^T \boldsymbol{dx} < \boldsymbol{0}$$

- We can therefore deduce that the constraint gradient and the objective gradient are collinear, i.e.

$$\nabla f = -\lambda \nabla C \quad \text{for some } \lambda > 0$$

At the optimum, the constraint gradient and the objective gradient have to be collinear!

Multiple Equality Constraints

- Assume that for given $\boldsymbol{x}$ the constraints are satisfied, $\boldsymbol{C}(\boldsymbol{x}) = \mathbf{0}$, but $\boldsymbol{x}$ is not the optimum.
- Then $\exists \boldsymbol{dx}$ such that $\boldsymbol{C}(\boldsymbol{x}+\boldsymbol{dx}) = \mathbf{0}$ and $f(\boldsymbol{x}+\boldsymbol{dx}) < f(\boldsymbol{x})$
- For linear constraints $\boldsymbol{C}(\boldsymbol{x}+\boldsymbol{dx}) = \boldsymbol{C}(\boldsymbol{x}) + \nabla\boldsymbol{C}(\boldsymbol{x})\boldsymbol{dx}$
- For small $|\boldsymbol{dx}|$, $f(\boldsymbol{x}+\boldsymbol{dx}) < f(\boldsymbol{x})$ implies that $\nabla f(\boldsymbol{x})^T\boldsymbol{dx} < \mathbf{0}$
- In summary, we have

$$\nabla\boldsymbol{C}(\boldsymbol{x})\boldsymbol{dx} = \mathbf{0} \qquad \text{and} \qquad \nabla f(\boldsymbol{x})^T\boldsymbol{dx} < \mathbf{0}$$

or equivalently

$$\begin{bmatrix}\nabla f(\boldsymbol{x})^T \\ \nabla\boldsymbol{C}(\boldsymbol{x})\end{bmatrix}\boldsymbol{dx} = \begin{bmatrix}\boxed{-1} \\ \mathbf{0}\end{bmatrix}$$

the amount of decrease in objective is arbitrary

Multiple Equality Constraints

- Now assume that $\boldsymbol{x}$ is the optimum
- Then $\nexists \boldsymbol{dx}$ such that $\boldsymbol{C}(\boldsymbol{x}+\boldsymbol{dx}) = \mathbf{0}$ and $f(\boldsymbol{x}+\boldsymbol{dx}) < f(\boldsymbol{x})$
- Equivalently, the linear system

$$\begin{bmatrix} \nabla f(\boldsymbol{x})^T \\ \nabla \mathbf{C}(\boldsymbol{x}) \end{bmatrix} \boldsymbol{dx} = \begin{bmatrix} -1 \\ \mathbf{0} \end{bmatrix}$$

has no solution.

- Then the first row must be a linear combination of the other rows,

$$\nabla f(\boldsymbol{x}) = \nabla \boldsymbol{C}(\boldsymbol{x})^T \boldsymbol{\lambda}$$

for some $\boldsymbol{\lambda} \in \boldsymbol{R}^m$

At the optimum, the objective gradient must be in the linear span of the constraint gradients!

Example

$$\min_{x,y} \; x^2 + y^2 \quad \text{s.t.} \quad C(x,y) = x + y - 1 = 0$$

$$\nabla f = \begin{bmatrix} 2x \\ 2y \end{bmatrix} \qquad \nabla C = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \qquad \begin{bmatrix} \nabla f^T \\ \nabla C^T \end{bmatrix} = \begin{bmatrix} 2x & 2y \\ 1 & 1 \end{bmatrix}$$

Let $x = 0.5, y = 0.5.$

$$\begin{bmatrix} \nabla f^T \\ \nabla C^T \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$$ $\leftarrow \nabla f = \nabla C\lambda$ *with* $\lambda = 1$

$$\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} dx \\ dy \end{bmatrix} = \begin{bmatrix} -1 \\ 0 \end{bmatrix}$$ $\leftarrow$ *No solution!*

$(x, y) = (0.5, 0.5)$ is the solution (*)

(*) it satisfies the first-order optimality conditions

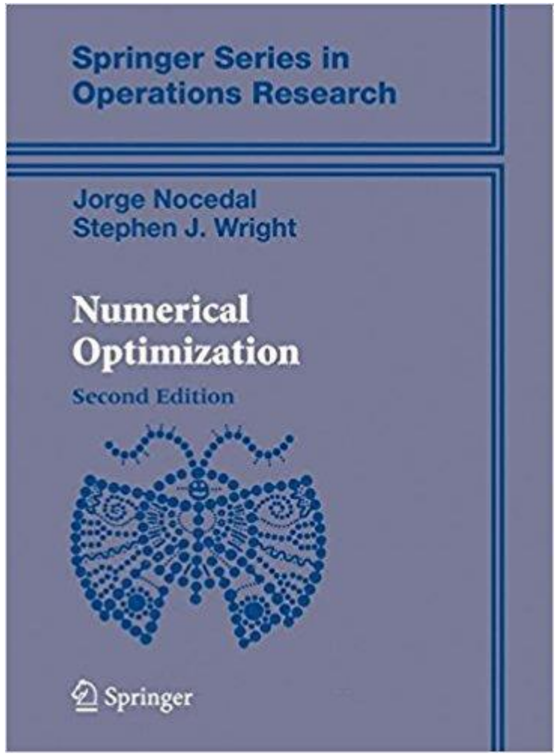
First Order Optimality Conditions

- First Order Optimality Conditions

$$\nabla f(\boldsymbol{x}) + \nabla \boldsymbol{C}(\boldsymbol{x})^T \boldsymbol{\lambda} = \mathbf{0} \text{ , and}$$
$$\boldsymbol{C}(\boldsymbol{x}) = \mathbf{0}$$

- **Note 1**: these conditions are also known as Karush-Kuhn-Tucker (KKT) conditions.
- **Note 2**: the KKT conditions are necessary, but not sufficient in general for $(\boldsymbol{x}, \boldsymbol{\lambda})$ to be a (strict local) solution to the optimization problem.

References



Available through ETH account at
https://link.springer.com/content/pdf/10.1007%2F978-0-387-40065-5.pdf

Towards Second Order Optimality – Assumptions

- The objective function $f(\boldsymbol{x})$ is quadratic, i.e.,
$$f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T\boldsymbol{H}\boldsymbol{x} + \boldsymbol{g}^T\boldsymbol{x} + \boldsymbol{a}$$
- All m constraints are linear equality constraints, i.e.,
$$\boldsymbol{C}(\boldsymbol{x}) = \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}$$
- Aside: an optimization problem with quadratic f and linear $\boldsymbol{C}$ is called a **Quadratic Program** (QP).
- Constraint gradients are linearly independent, i.e., $\text{rank}(\boldsymbol{A}) = m$
  → no redundant constraints, the optimal $\boldsymbol{\lambda}$ are unique
  → no inconsistent/conflicting constraints, the problem is feasible

Towards Second Order Optimality – Definitions

- A point $\boldsymbol{x}$ is **feasible** if $C_i(\boldsymbol{x}) = 0\ \forall i$, i.e., $\boldsymbol{C}(\boldsymbol{x}) = 0$.
- The set of all feasible points is the feasible set $\Omega = \{\boldsymbol{x} | \boldsymbol{C}(\boldsymbol{x}) = \boldsymbol{0}\}$
- For a given feasible $\boldsymbol{x} \in \Omega$, the set of **first-order feasible directions**

$$F(\boldsymbol{x}) = \{\boldsymbol{d} | \nabla \boldsymbol{C}(\boldsymbol{x})\boldsymbol{d} = \boldsymbol{0}\}$$

contains all directions that are orthogonal to all constraint gradients.

Towards Second Order Optimality – Undecided Directions

- Let $\boldsymbol{x}^*$ denote a point that satisfies the first-order optimality conditions
- For $\boldsymbol{x}^*$ to be a **strict** local minimizer of f, we require that

$$f(\boldsymbol{x}^* + \boldsymbol{w}) > f(\boldsymbol{x}^*) \ \ \forall \boldsymbol{w} \in F(\boldsymbol{x}^*)$$

- However, for all feasible directions $\boldsymbol{w} \in F(\boldsymbol{x}^*)$ we have (to first order),

$$\nabla f^T \boldsymbol{w} = 0 \ \rightarrow f(\boldsymbol{x} + \epsilon \boldsymbol{w}) = f(\boldsymbol{x}) + \epsilon \nabla f^T \boldsymbol{w} + O(\epsilon^2) = f(\boldsymbol{x})$$

- When using only first-order information, we cannot decide whether

$$f(\boldsymbol{x}^* + \boldsymbol{w}) \geq f(\boldsymbol{x}^*) \qquad \text{or} \qquad f(\boldsymbol{x}^* + \boldsymbol{w}) < f(\boldsymbol{x}^*) .$$

- For these *undediced* directions, we need higher order information to verify the strictness of a first order optimal solution.

Towards Second Order Optimality – Undecided Directions

- Taylor expansion

$$f(\boldsymbol{x}^* + \boldsymbol{w}) = f(\boldsymbol{x}^*) + \nabla f(\boldsymbol{x}^*)^T \boldsymbol{w} + \frac{1}{2}\boldsymbol{w}^T \boldsymbol{H} \boldsymbol{w} = f(\boldsymbol{x}^*) + \frac{1}{2}\boldsymbol{w}^T \boldsymbol{H} \boldsymbol{w}$$

- For a strict optimum we must have $f(\boldsymbol{x}^* + \boldsymbol{w}) \geq f(\boldsymbol{x}^*)\ \forall \boldsymbol{w} \in F$, hence

$$\rightarrow \boldsymbol{w}^T \boldsymbol{H} \boldsymbol{w} > 0 \forall \boldsymbol{w} \in F$$

How can we ensure this condition?

- Asking for $\boldsymbol{H}$ to be positive definite is too strong (sufficient but not necessary), since infeasible directions do not matter
- Idea: consider definiteness of Hessian on space of feasible directions only

Second Order Sufficient Conditions

- We want

$$\boldsymbol{w}^T \boldsymbol{H} \boldsymbol{w} > 0 \ \forall \boldsymbol{w} \in F$$

- Let $\boldsymbol{Z} \in \mathbf{R}^{n \times (n-m)}$ be a basis for the **null-space** of F such that

$$F = \{\boldsymbol{Z}\boldsymbol{u} | \boldsymbol{u} \in \mathbf{R}^{n-m}\}$$

- Using the null-space basis $\boldsymbol{Z}$, we can alternatively write

$$\boldsymbol{w}^T \boldsymbol{H} \boldsymbol{w} > 0 \ \ \forall \boldsymbol{w} \in F \quad \leftrightarrow \quad \boldsymbol{u}^T \boldsymbol{Z}^T \boldsymbol{H} \boldsymbol{Z} \boldsymbol{u} > 0 \forall \ \boldsymbol{u} \in \mathbf{R}^{n-m},$$

which is the case if $\boldsymbol{Z}^T \boldsymbol{H} \boldsymbol{Z}$ is positive definite.

Second order sufficient conditions: $\boldsymbol{x}^*$ is a strict local solution to the optimization problem if

1. $\boldsymbol{x}^*$ satisfies the KKT conditions, and
2. $\boldsymbol{Z}^T \boldsymbol{H} \boldsymbol{Z}$ is positive definite

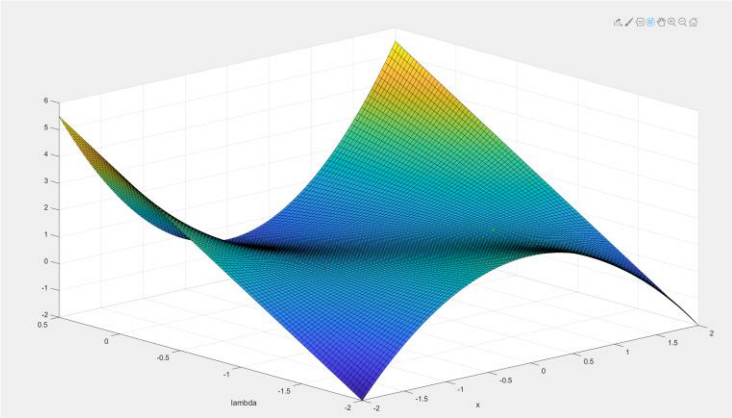
The Lagrangian – A Reformulation

- Generic optimization

$$\min_{\boldsymbol{x}} f(\boldsymbol{x}) \qquad \text{s.t.} \quad \boldsymbol{C}(\boldsymbol{x}) = 0$$

- Define the Lagrangian

$$L(\boldsymbol{x}, \boldsymbol{\lambda}) = f(\boldsymbol{x}) + \boldsymbol{\lambda}^T \boldsymbol{C}(\boldsymbol{x})$$



Example: $L(x, \lambda) = x^2 + \lambda(x^2 - 1)$

- Consider the gradient of $L(\boldsymbol{x}, \boldsymbol{\lambda})$, i.e.,

$$\nabla_{\boldsymbol{x}} L = \nabla f + \nabla \boldsymbol{C}^T \lambda \qquad \text{and} \qquad \nabla_\lambda L = \boldsymbol{C}(\boldsymbol{x})$$

- **Observations**:
  - the first-order optimality conditions correspond to $\nabla L = 0$
  - Solving the optimality conditions means solving for a stationary point of L
  - Since $\boldsymbol{C}$ is not bounded from below or above, $\nabla L = 0$ must be a saddle point

Solving a Quadratic Program*

- For a quadratic program, we have

$$f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T\boldsymbol{H}\boldsymbol{x} + \boldsymbol{g}^T\boldsymbol{x} + \boldsymbol{a} \quad \text{and} \quad \boldsymbol{C}(\boldsymbol{x}) = \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}$$

- Therefore

$$L(\boldsymbol{x}, \boldsymbol{\lambda}) = \frac{1}{2}\boldsymbol{x}^T\boldsymbol{H}\boldsymbol{x} + \boldsymbol{g}^T\boldsymbol{x} + \boldsymbol{a} + \boldsymbol{\lambda}^T(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b})$$

$$\nabla_{\boldsymbol{x}} L = \boldsymbol{H}\boldsymbol{x} + \boldsymbol{g} + \boldsymbol{A}^T\boldsymbol{\lambda} \qquad \text{and} \qquad \nabla_{\boldsymbol{\lambda}} L = \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}$$

- For first-order optimality, we need

$$\begin{bmatrix} \nabla_{\boldsymbol{x}} L \\ \nabla_{\boldsymbol{\lambda}} L \end{bmatrix} = \boldsymbol{0} \quad \Rightarrow \quad \begin{bmatrix} \boldsymbol{H} & \boldsymbol{A}^T \\ \boldsymbol{A} & \boldsymbol{0} \end{bmatrix} \cdot \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} -\boldsymbol{g} \\ \boldsymbol{b} \end{bmatrix}$$

*with only equality constraints

Solving a Quadratic Program

$$\begin{bmatrix} \boldsymbol{H} & \boldsymbol{A}^T \\ \boldsymbol{A} & \boldsymbol{0} \end{bmatrix} \cdot \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} -\boldsymbol{g} \\ \boldsymbol{b} \end{bmatrix}$$

- KKT-Matrix is
  - Symmetric
  - Indefinite (n positive eigenvalues, m negative eigenvalues)
- If $\boldsymbol{A}$ is full-rank and $\boldsymbol{H}$ is positive-definite on the subspace orthogonal to $\boldsymbol{A}$, then the QP is **convex** and has a unique solution

How do we solve the KKT system?

Solving the KKT System

- Direct indefinite solvers
  - Cannot use Cholesky since the KKT matrix is indefinite
  - Can use LU, but ignores symmetry
  - Pardiso (Parallel symmetric indefinite factorization, i.e., $PAP^T = LDL^T$)
- Iterative solvers such as the Uzawa algorithm
- Alternatively: use QP solver
  - Mosek
  - ..

Example – NW E16.2

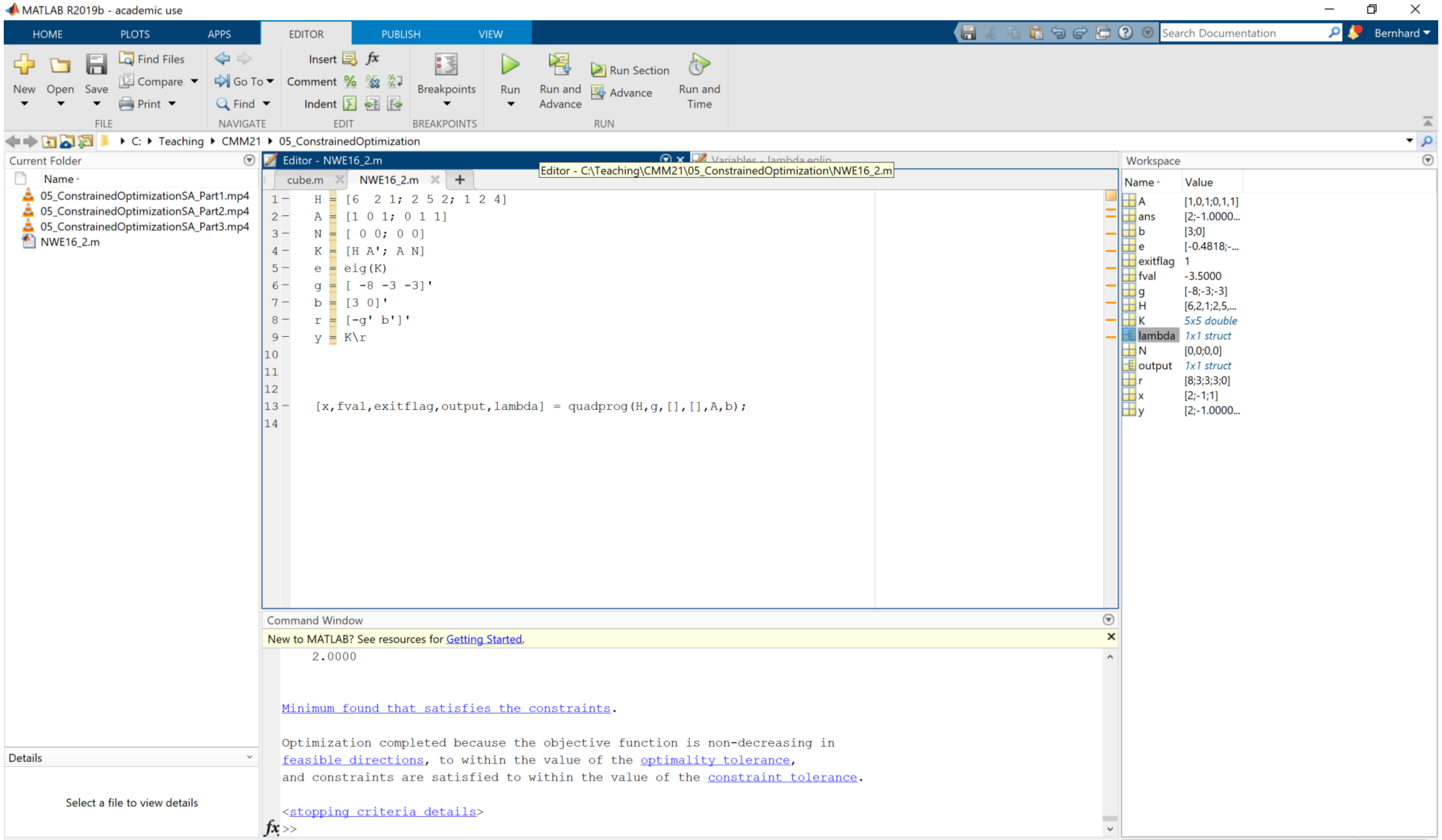
$$f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T \boldsymbol{H}\boldsymbol{x} + \boldsymbol{g}^T \boldsymbol{x} + a$$
$$\boldsymbol{C}(\boldsymbol{x}) = \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}$$

$$\begin{bmatrix} \boldsymbol{H} & \boldsymbol{A}^T \\ \boldsymbol{A} & \boldsymbol{0} \end{bmatrix} \cdot \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} -\boldsymbol{g} \\ \boldsymbol{b} \end{bmatrix}$$

$$\min_{\boldsymbol{x}} f(\boldsymbol{x}) = 3x_1^2 + 2x_1x_2 + x_1x_3 + 2.5x_2^2 + 2x_2x_3 + 2x_3^2 - 8x_1 - 3x_2 - 3x_3,$$
$$\text{subject to} \quad x_1 + x_3 = 3, \quad x_2 + x_3 = 0.$$

$$\boldsymbol{H} = \begin{bmatrix} 6 & 2 & 1 \\ 2 & 5 & 2 \\ 1 & 2 & 4 \end{bmatrix} \quad \boldsymbol{g} = \begin{bmatrix} -8 \\ -3 \\ -3 \end{bmatrix} \quad \boldsymbol{A} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \quad \boldsymbol{b} = \begin{bmatrix} 3 \\ 0 \end{bmatrix} \quad a = 0$$

Example – NW E16.2



Example – NW E16.2

$$f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T\boldsymbol{H}\boldsymbol{x} + \boldsymbol{g}^T\boldsymbol{x} + a$$
$$\boldsymbol{C}(\boldsymbol{x}) = \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}$$

$$\begin{bmatrix} \boldsymbol{H} & \boldsymbol{A}^T \\ \boldsymbol{A} & \boldsymbol{0} \end{bmatrix} \cdot \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} -\boldsymbol{g} \\ \boldsymbol{b} \end{bmatrix}$$

$$\min_{\boldsymbol{x}} f(\boldsymbol{x}) = 3x_1^2 + 2x_1x_2 + x_1x_3 + 2.5x_2^2 + 2x_2x_3 + 2x_3^2 - 8x_1 - 3x_2 - 3x_3,$$
$$\text{subject to} \quad x_1 + x_3 = 3, \qquad x_2 + x_3 = 0.$$

$$\boldsymbol{H} = \begin{bmatrix} 6 & 2 & 1 \\ 2 & 5 & 2 \\ 1 & 2 & 4 \end{bmatrix} \quad \boldsymbol{g} = \begin{bmatrix} -8 \\ -3 \\ -3 \end{bmatrix} \quad \boldsymbol{A} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \quad \boldsymbol{b} = \begin{bmatrix} 3 \\ 0 \end{bmatrix} \quad a = 0$$

The solution is $\boldsymbol{x}^*(2, -1, 1)^T$ and the optimal Lagrange multiplier are $\boldsymbol{\lambda}^* = (-3, 2)^T$

A null-space basis matrix is $\boldsymbol{Z} = (-1, -1, 1)^T$ and $\boldsymbol{Z}^T\boldsymbol{H}\boldsymbol{Z} = 13$

CRL

Inequality Constraints

- Inequality constraints occur naturally in optimization problems
  - Positivity on variables, $x_i \geq 0$
  - Limited resources available, but not all have to be used, $\sum_i x_i \leq M$
- Let $\mathcal{E}$ denote the index set of all equality constraints. Then

$$c_i(\boldsymbol{x}) = 0 \;\; \forall i \in \mathcal{E}$$

- Let $\mathcal{I}$ denote the index set of all inequality constraints. Then

$$c_i(\boldsymbol{x}) \geq 0 \;\; \forall i \in \mathcal{I}$$

- Lagrangian

$$\mathcal{L}(x, \lambda) = f(x) + \sum_{i \in \mathcal{E} \cup \mathcal{I}} \lambda_i c_i(x)$$

Inequality Constrained Problems

- Lagrangian $$\mathcal{L}(x, \lambda) = f(x) + \sum_{i \in \mathcal{E} \cup \mathcal{I}} \lambda_i c_i(x)$$
- First-order optimality (KKT) conditions

$$\begin{aligned}
\nabla_x \mathcal{L}(x^*, \lambda^*) &= 0, && \\
c_i(x^*) &= 0, && \text{for all } i \in \mathcal{E}, \\
c_i(x^*) &\geq 0, && \text{for all } i \in \mathcal{I}, \\
\lambda_i^* &\leq 0, && \text{for all } i \in \mathcal{I}, \\
\lambda_i^* c_i(x^*) &= 0, && \text{for all } i \in \mathcal{E} \cup \mathcal{I}.
\end{aligned}$$

Feasibility: Inequality constraints have to be satisfied

One-sidedness: Inequality constraints can only push, not pull

Complementary slackness: Either constraint is active, or its LM is zero

Active Set

- For a feasible point $\boldsymbol{x}$, the inequality constraint c_i is

  active if $c_i(\boldsymbol{x}) = 0$ and *inactive* if $c_i(\boldsymbol{x}) > 0$.
- For any feasible point $\boldsymbol{x}$, the active set $\mathcal{A}(\boldsymbol{x})$ is defined as

$$\mathcal{A}(x) = \mathcal{E} \cup \{i \in \mathcal{I} \mid c_i(x) = 0\}$$

- If we knew the active set, then we could just solve an equality-constrained QP with only the active IC present
- However, we generally do not know the active set in advance
- **Idea**: instead of explicitly enforcing complementarity conditions, build active set iteratively by *guessing* active constraints and solving QPs until optimality conditions are satisfied.

Nonlinear Programming

- What changes if f and $\boldsymbol{C}$ are no longer quadratic/linear?
  - KKT-conditions are still necessary
  - The second-order optimality conditions are still sufficient
  - But no guarantee that local optimum is global optimum
- One strategy: solve non-linear KKT conditions to find local optimum

Nonlinear KKT Conditions

- Lagrangian $L(\boldsymbol{x}, \boldsymbol{\lambda}) = f(\boldsymbol{x}) + \boldsymbol{\lambda}^T \boldsymbol{C}(\boldsymbol{x})$
- Given $\boldsymbol{s} = (\boldsymbol{x}^t, \boldsymbol{\lambda}^t)$ for which $\nabla_{\boldsymbol{s}} L \neq \boldsymbol{0}$, find $\Delta \boldsymbol{s}$ such that
$$\nabla_{\boldsymbol{s}} L(\boldsymbol{s} + \Delta \boldsymbol{s}) = \boldsymbol{0}$$
- Expand the gradient around $\boldsymbol{s}$
$$\nabla_{\boldsymbol{s}} L(\boldsymbol{s} + \Delta \boldsymbol{s}) = \nabla_{\boldsymbol{s}} L(\boldsymbol{s}) + \nabla_{\boldsymbol{ss}} L(\boldsymbol{s}) \Delta \boldsymbol{s} + O(\Delta \boldsymbol{s}^2)$$
- First-order approximation

$$\nabla_{ss} L = \begin{bmatrix} \frac{\partial^2 L}{\partial \boldsymbol{x}^2} & \frac{\partial}{\partial \boldsymbol{\lambda}} \frac{\partial L}{\partial \boldsymbol{x}} \\ \frac{\partial}{\partial \boldsymbol{x}} \frac{\partial L}{\partial \boldsymbol{\lambda}} & \frac{\partial^2 L}{\partial \boldsymbol{\lambda}^2} \end{bmatrix}$$

$$\nabla_{\boldsymbol{ss}} L \cdot \Delta \boldsymbol{s} = -\nabla_{\boldsymbol{s}} L$$

$$\begin{bmatrix} \nabla_{\boldsymbol{xx}} L(\boldsymbol{x}) & \nabla \boldsymbol{C}(\boldsymbol{x})^T \\ \nabla \boldsymbol{C}(\boldsymbol{x}) & \boldsymbol{0} \end{bmatrix} \cdot \begin{bmatrix} \Delta \boldsymbol{x} \\ \Delta \boldsymbol{\lambda} \end{bmatrix} = -\begin{bmatrix} \nabla_{\boldsymbol{x}} L \\ \nabla_{\boldsymbol{\lambda}} L \end{bmatrix}$$

- Note: the Hessian of the Lagrangian now involves 2nd derivatives of the constraints, $\nabla_{\boldsymbol{xx}} L = \nabla_{\boldsymbol{xx}} f + \nabla_{\boldsymbol{xx}} \boldsymbol{C}^T \boldsymbol{\lambda}$

Nonlinear Programming

$$\begin{bmatrix} \nabla_{\boldsymbol{xx}} L(\boldsymbol{x}) & \nabla \boldsymbol{C}(\boldsymbol{x})^T \\ \nabla \boldsymbol{C}(\boldsymbol{x}) & \mathbf{0} \end{bmatrix} \cdot \begin{bmatrix} \Delta \boldsymbol{x} \\ \Delta \boldsymbol{\lambda} \end{bmatrix} = - \begin{bmatrix} \nabla_{\boldsymbol{x}} L \\ \nabla_{\boldsymbol{\lambda}} L \end{bmatrix}$$

- Since the KKT system is based on a first-order approximation,
  $\nabla_{\boldsymbol{s}} L(\boldsymbol{s} + \Delta \boldsymbol{s}) \neq \mathbf{0}$ in general.
- Solution: *iterate!*

Sequential Quadratic Programming (SQP)

Until convergence
- solve $\nabla_{\mathbf{ss}} L \cdot \Delta \mathbf{s} = -\nabla_{\mathbf{s}} L$
- line search $\alpha = \text{line\_search}(\mathbf{s}, \Delta \mathbf{s})$
- update $\mathbf{s} = \mathbf{s} + \alpha \Delta \mathbf{s}$

end

Remarks and Pointers

- The second derivatives of the constraints (e.g., forces) can introduce indefiniteness into the system
- Alternative: instead of using analytical Hessian, use approximation

  → Quasi-Newton methods (e.g., BFGS [NW 6.1] and variants)
- Line search [NW 3.1] requires careful balancing of progress in objectives vs. constraint violations (→ merit functions [NW 15.4])
- For inequality-constrained problems, interior point methods [NW 16.6] are often superior to active set methods [NW 16.5]

Why not just Penalties?

- Instead of using Lagrange multipliers, enforce constraints $C(\boldsymbol{x})$ through penalty function $f_P(\boldsymbol{x}) = k_p C(\boldsymbol{x})^2$ [NW 17.1]
- Advantages:
  - Unconstrained minimization problem
  - No additional DOFs
  - Can work well for non-stiff constraints
- Disadvantages
  - May need large k_P for sufficient constraint satisfaction
    $\rightarrow$ numerical problems

Optimization Methods

- Randomized Search (Simulated Annealing, CMA-ES, Genetic Algorithms, …)
- Sequential Quadratic Programming
- Sensitivity Analysis
- Interior Point Methods
- Augmented Lagrangian Method (penalty method done properly, [NW 17.3])

- …

Sensitivity Analysis for Equilibrium-Constrained Problems

Equilibrium-Constrained Problems Revisited

- **Observation**: when we set parameters $\boldsymbol{p}$, we observe the equilibrium state $\boldsymbol{x}$ as the result of simulation.
- Although $\boldsymbol{x}$ are problem variables, they are not real DOFs – they are functions of the parameters, i.e.,

$$\boldsymbol{x} = \boldsymbol{x}(\boldsymbol{p})$$

- Map from parameters to state is

$$\boldsymbol{x} = \text{simulate}(\boldsymbol{p})$$

- For design, we need derivatives of $\boldsymbol{x}(\boldsymbol{p})$,

$$\frac{\partial T}{\partial \boldsymbol{p}} = \left(\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{p}}\right)^t \frac{\partial T}{\partial \boldsymbol{x}}$$

- But how to compute these derivatives,

$$\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{p}} = \frac{\partial \text{simulate}}{\partial \boldsymbol{p}}\,?$$

Differentiating the Map

- Although we can evaluate the map $\boldsymbol{x} \to \boldsymbol{x}(\boldsymbol{p})$, this map is not available in closed-form (i.e., *analytically*)
- $\boldsymbol{x} \to \boldsymbol{x}(\boldsymbol{p})$ requires minimizing a function, i.e., solving a system of nonlinear equations.
- In general, it is impractical to compute derivatives of the minimization process.
- But even though $\boldsymbol{x} \to \boldsymbol{x}(\boldsymbol{p})$ is not given *explicitly*, the constraints

$$\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{p}) = \boldsymbol{0}$$

provide this map *implicitly*.

Differentiating the Map

- Suppose that ($\boldsymbol{x}$,$\boldsymbol{p}$) is a feasible pair, i.e., $\boldsymbol{f}(\boldsymbol{x},\boldsymbol{p}) = \mathbf{0}$. In other words, $\boldsymbol{x}$ is an equilibrium configuration for $\boldsymbol{p}$.
- If we apply a parameter perturbation $\Delta\boldsymbol{p}$, the system will undergo displacements $\Delta\boldsymbol{x}$ such that it is again in equilibrium,

$$\boldsymbol{f}(\boldsymbol{x}+\Delta\boldsymbol{x},\boldsymbol{p}+\Delta\boldsymbol{p}) = \mathbf{0}\ .$$

- Since this has to hold for arbitrary parameter variations, we have

$$\frac{d\boldsymbol{f}}{d\boldsymbol{p}} = \frac{\partial\boldsymbol{f}}{\partial\boldsymbol{x}}\frac{\partial\boldsymbol{x}}{\partial\boldsymbol{p}} + \frac{\partial\boldsymbol{f}}{\partial\boldsymbol{p}} = \mathbf{0}\ .$$

- If the Jacobian $\nabla_{\boldsymbol{x}}\boldsymbol{f}$ is square and non-singular, we have

$$\frac{\partial\boldsymbol{x}}{\partial\boldsymbol{p}} = -\frac{\partial\boldsymbol{f}}{\partial\boldsymbol{x}}^{-1}\frac{\partial\boldsymbol{f}}{\partial\boldsymbol{p}}\ .$$

Implicit Function Theorem

- **Implicit function theorem (IFT)**: let $f\colon \mathbf{R}^{n+p} \to \mathbf{R}^n$ be a continuously differentiable function. If for given $\boldsymbol{x}_0 \in \mathbf{R}^n$ and $\boldsymbol{p}_0 \in \mathbf{R}^p$ we have $\boldsymbol{f}(\boldsymbol{x}_0, \boldsymbol{p}_0) = \mathbf{0}$ and if $\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}|_{\boldsymbol{x}_0}$ is invertible, then there exists a unique, continuously differentiable function $\boldsymbol{y}$ such that $\boldsymbol{f}(\boldsymbol{y}(\boldsymbol{p}), \boldsymbol{p}) = \mathbf{0}$ for all $\boldsymbol{p}$ in a Neighborhood $\mathcal{N}$ around $\boldsymbol{p}$.

- The IFT is applicable to equilibrium-constrained optimization problems
- The IFT asserts the existence and (local) uniqueness of the map $\boldsymbol{x} = \boldsymbol{y}(\boldsymbol{p})$ between parameters and state as well as its derivative(s).
- The fact that $\boldsymbol{y}$ exists does not imply that there is a closed-form expression for it (or its derivatives) $\to$ compute $\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{p}}$ numerically

Computing the Sensitivity Matrix

$$\boldsymbol{S} := \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{p}} = -\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}^{-1} \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{p}}$$

- The **Sensitivity Matrix** $\boldsymbol{S}$ maps infinitesimal changes in parameters to infinitesimal changes in equilibrium state, i.e.,

$$\Delta \boldsymbol{x} = \boldsymbol{S} \Delta \boldsymbol{p}$$

- $\boldsymbol{S}$ can be computed numerically by solving systems of linear equations,

$$\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}} \boldsymbol{S} = -\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{p}} \quad \rightarrow \quad \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}} \boldsymbol{S}[i,.] = -\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{p}} \text{ for all rows } i \text{ of } \boldsymbol{S}$$

- For direct solvers, the $(\boldsymbol{L}\boldsymbol{L}^T)$ factorization for $\boldsymbol{S}$ needs to be computed only once, can be reused for all rows.

Sensitivity Analysis for Equilibrium-Constrained Problems

- How can we use $\boldsymbol{S}$ for solving equilibrium-constrained problems?
- We can compute the gradient of the objective T wrt. the parameters,

$$\frac{\partial T}{\partial \boldsymbol{p}} = \left(\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{p}}\right)^T \frac{\partial T}{\partial \boldsymbol{x}} \quad \Rightarrow$$

Sensitivity Analysis Steepest Descent - SASD

Until convergence

$$\begin{aligned}
\mathbf{S} &= -\nabla_{\mathbf{x}}\mathbf{f}^{-1}\nabla_{\mathbf{p}}\mathbf{f} \\
\Delta\mathbf{p} &= -\mathbf{S}^T\nabla_{\mathbf{x}}T \\
\alpha &= \text{line_search}(\Delta\mathbf{p}) \\
\mathbf{p} &= \mathbf{p} + \alpha\Delta\mathbf{p}; \\
\mathbf{x} &= \text{simulate}(\mathbf{x}, \mathbf{p})
\end{aligned}$$

end

SASD – Cost per Iteration

- In each iteration of SASD we have to
  - Compute sensitivity matrix → solve $\dim(\boldsymbol{p})$ linear systems
  - Line search → each step requires simulation with new $\boldsymbol{p}$
  - Steepest descent converges slowly

- Acceleration
  - Use adjoint sensitivities instead of computing entire $\boldsymbol{S}$
  - Use Quasi-Newton method for better convergence