Nonlinear Programming 1

- General form of nonlinear programs, constrained and unconstrained optimization, convex programs, local and global optimal solution for nonconvex feasible region and/or objective function, complexity
- Optimality conditions, smoothness, the concept of improving directions and improving feasible directions, characterizing the optimality of convex programs in terms of improving feasible directions
- Solving simple nonlinear programs using successive linear programming, the Method of Zoutendijk, finding improving feasible directions, line search problems, choosing the step size

Nonlinear Programming

- Linear programming is widely used in practice
- Sophisticated modeling frameworks, efficient solvers, build-in from the basic level (Excel!)
- Even problems of enormous size (hundreds of thousands of variables, millions of constraints) can be solved by computer-aided tools
- Unfortunately, the world is highly nonlinear: in many cases either the constraints or the objective function (or both) may be nonlinear
- In lucky cases the problem can be linearized without too much loss of precision
- In more complex cases, however, linear programming is of limited use

Nonlinear Programs

- Consider the below **nonlinear program**:

$$\begin{aligned} \min \ & f(\boldsymbol{x}) \\ \text{s.t.} \ & g_i(\boldsymbol{x}) \leq 0 \qquad\qquad i \in I = \{1, \ldots, m\} \end{aligned}$$

  where $\boldsymbol{x} \in \mathbb{R}^n$ is a real-valued column n-vector and f and g_i are continuously differentiable $\mathbb{R}^n \mapsto \mathbb{R}$ functions
- If $I = \emptyset$ then the nonlinear program is **unconstrained**, otherwise it is **constrained**
- If the feasible region $X = \left\{\boldsymbol{x} : g_i(\boldsymbol{x}) \leq 0,\ i \in \{1, \ldots, m\}\right\}$ is convex and f is convex on X, then the problem is a **convex program**
- For this, it is enough that all $g_i : i \in \{1, \ldots, m\}$ be convex (we omit the proof)

Nonlinear Programs

- The form $\min\left\{f(\boldsymbol{x}) : g_i(\boldsymbol{x}) \leq 0, i \in \{1,\ldots,m\}\right\}$ is a generalization of linear programs to the nonlinear case
- A linear program $\min\{\boldsymbol{c}^T\boldsymbol{x} : \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b}\}$ can easily be written in the general form with the choice:

$$f(\boldsymbol{x}) = \boldsymbol{c}^T\boldsymbol{x} \text{ és } \quad g_i(\boldsymbol{x}) = \boldsymbol{a}^i\boldsymbol{x} - b_i \qquad i \in \{1,\ldots,m\}$$

- Unfortunately, nonlinear programs lack the appealing simplicity of linear programs
  - local optimum $\neq$ global optimum
  - the optimum might not necessarily occur at an extreme point of the feasible region
  - not even on the boundary of the feasible region
  - the simplex method cannot be used
  - the concept of duality is less overarching

Nonconvex Feasible Region

- Minimization of a Convex objective function over a convex region: local optima correspond to global optima (recall the Fundamental Theorem of Convex Programming)
- In general this is not the case
- Consider the nonlinear program

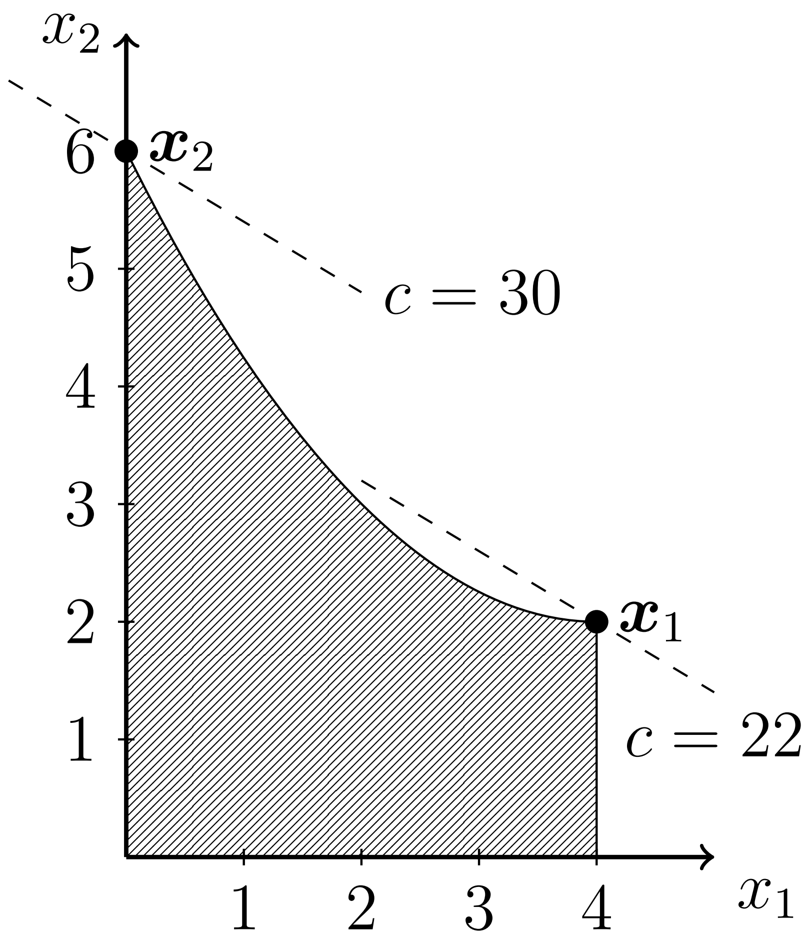
$$\begin{aligned} \max \;\; & 3x_1 + 5x_2 \\ \text{s.t.} \;\; & x_1 \leq 4 \\ & x_2 \leq 7 \\ & 4x_2 \leq (x_1 - 4)^2 + 8 \\ & x_1, x_2 \geq 0 \end{aligned}$$

- In the general form, converting to minimization:

$$f(x_1, x_2) = -3x_1 - 5x_2, \quad g_1(x_1, x_2) = x_1 - 4, \ldots$$

Nonconvex Feasible Region

- The objective function is linear (thus convex) but the feasible region is nonconvex



- The contours of the objective function are the straight lines $-3x_1 - 5x_2 = c$
- $\boldsymbol{x}_1 = [4 \quad 2]^T$ is a local optimum, objective: 22
- $\boldsymbol{x}_2 = [0 \quad 6]^T$ is both locally and globally optimal, objective: 30
- Global optimization is difficult, since a local check cannot decide whether a point is local or global optimum

Nonconvex Objective Function

- Consider the nonlinear program

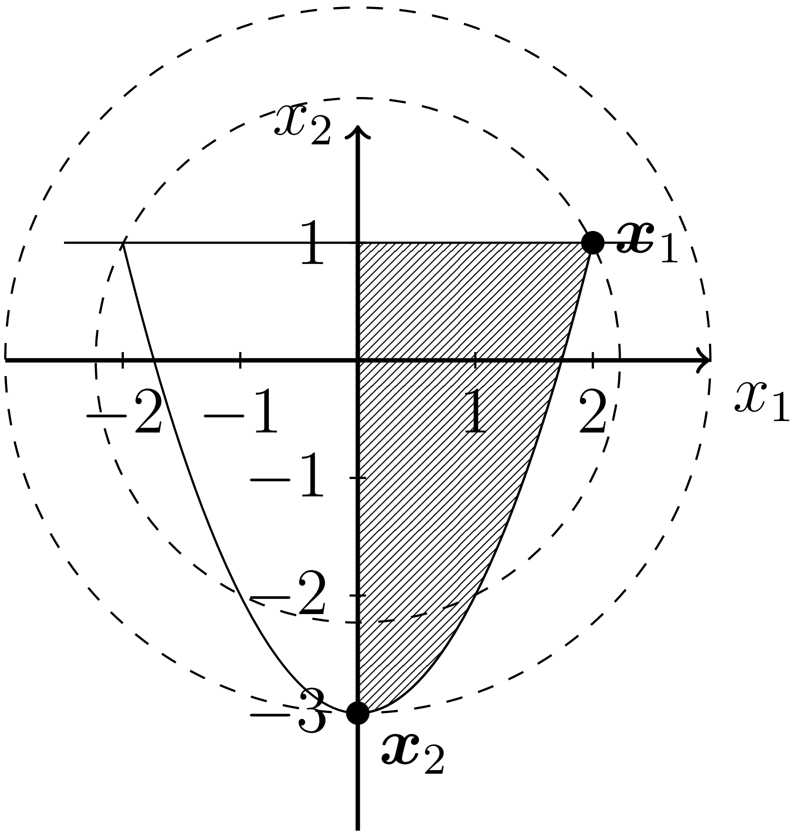
$$
\begin{aligned}
\min \quad & -x_1^2 - x_2^2 \\
\text{s.t.} \quad & x_1^2 - x_2 - 3 \leq 0 \\
& x_2 - 1 \leq 0 \\
& -x_1 \leq 0
\end{aligned}
$$

- In general form:

$$
\begin{aligned}
f(x_1, x_2) &= -x_1^2 - x_2^2 \\
g_1(x_1, x_2) &= x_1^2 - x_2 - 3 \\
g_2(x_1, x_2) &= x_2 - 1 \\
g_3(x_1, x_2) &= -x_1
\end{aligned}
$$

Nonconvex Objective Function

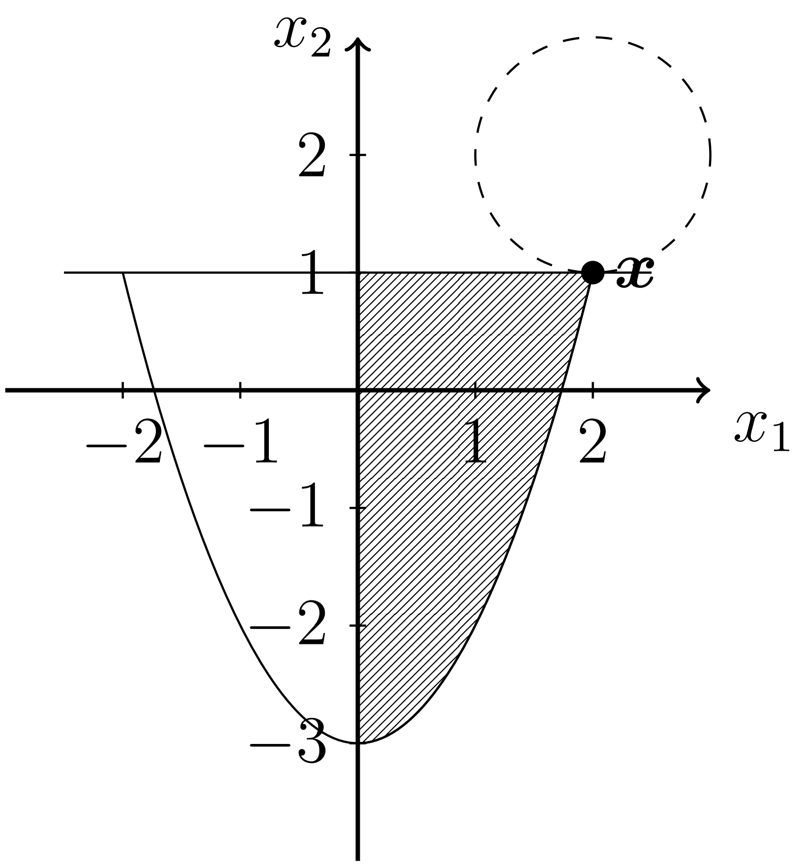
- The feasible region is convex but the objective function is not (concave)



- The contours of the objective function $-x_1^2 - x_2^2 = c$ are origin-centered cycles
- $\boldsymbol{x}_1 = [2 \quad 1]^T$ is a local optimum, objective: -5
- $\boldsymbol{x}_x = [0 \ -3]^T$ is both a local and a global optimum, objective: -9
- Again, no local check can establish this

Nonconvex Objective on Convex Region

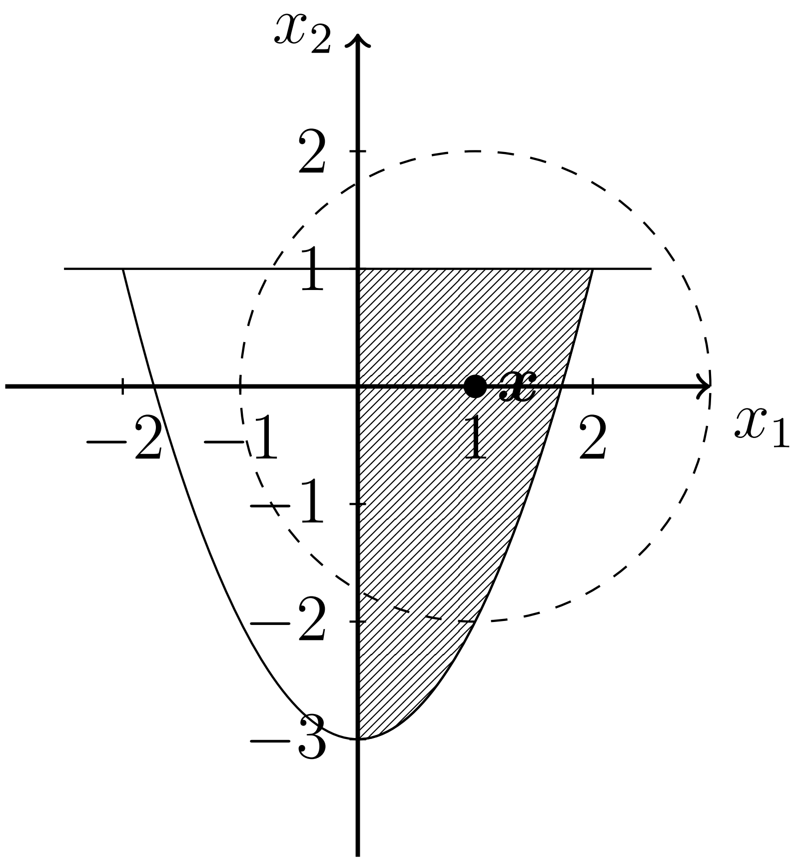
- If we change the objective function to the convex function $f(x_1, x_2) = (x_1 - 2)^2 + (x_2 - 2)^2$



- The objective contours are concentric cycles centered at $[2 \quad 2]^T$
- $\boldsymbol{x} = [2 \quad 1]^T$ is local and global optimum, objective: 1
- Convex feasible region $+$ convex objective function $+$ minimization $=$ each local optimum is global optimum
- Convex programming is much simpler than generic nonlinear programming

Convex Programs: Optima

- If the objective is the convex $f(x_1, x_2) = (x_1 - 1)^2 + x_2^2$ function



- Objective contours are concentric cycles around $\begin{bmatrix} 1 & 0 \end{bmatrix}^T$
- Convex program, but the optimum lies in the interior of the feasible region!
- Boundary point methods (like the simplex) cannot be used
- Interior point solvers are needed

Complexity

- Solving a nonconvex nonlinear program is difficult even when the feasible region is a polyhedron (i.e., constraints are linear)
- For instance, the well-known (and notoriously difficult) **Boolean 3-satisfiability** problem (3SAT) can easily be formulated in such a form
- We seek the logical variables A, B, C, and D can be set so that the below Boolean function evaluates to true

$$(A \text{ OR } \neg B \text{ OR } C) \text{ AND } (\neg A \text{ OR } C \text{ OR } \neg D) = \text{ TRUE}$$

- For instance, $C =$ TRUE is such a choice

Complexity

- 3SAT: decide whether an assignment of TRUE or FALSE values to logical variables $X_i : i \in \{1, \ldots, n\}$ exists so that the logical function

$$f(X_1, X_2, \ldots, X_n) = \bigwedge_{i=1}^{m} \Big((\neg) X_{i_1} \vee (\neg) X_{i_2} \vee (\neg) X_{i_3} \Big) = \text{TRUE}$$

  in conjunctive normal form (CNF) consisting of m clauses evaluates to TRUE
- Propositional logic:
  - $\wedge$: logical AND operation (conjunction)
  - $\vee$: logical OR operation (disjunction)
  - $(\neg)X_{i_j}$: variable j in clause i, may be negated (literal)
  - $(\neg)X_{i_1} \vee (\neg)X_{i_2} \vee (\neg)X_{i_3}$: clause

Complexity

- Define the $m \times n$ **clause matrix**:
  - $a_{ij} = -1$, if variable X_j occurs negated in clause i
  - $a_{ij} = 1$, if variable X_j occurs non-negated in clause i
  - $a_{ij} = 0$ otherwise
- Any 3SAT instance can be equivalently formulated as a nonlinear program using the clause matrix
- Let x_j be a continuous variable for logical variable X_j:

$$z = \max \sum_{j=1}^{n} x_j^2$$

$$\sum_{j=1}^{n} a_{ij} x_j \geq -1 \qquad \forall i = 1, \ldots, m$$

$$-1 \leq x_j \leq 1 \qquad \forall j = 1, \ldots, n$$

Complexity

- **Theorem:** a 3SAT instance is satisfiable, if and only the optimal objective function value of the equivalent nonlinear program is $z = n$ (n is the number of variables)
- If this is the case, then from $-1 \leq x_j \leq 1$ and $z = \max \sum_j x_j^2 = n$ it follows that x_j may either take value 1 (X_j: TRUE) or -1 (X_j: FALSE)

$$\neg X_5 \vee X_{12} \vee \neg X_{19} = \text{TRUE} \Leftrightarrow -x_5 + x_{12} - x_{19} \geq -1$$

- The constraint $\sum_j a_{ij} x_j \geq -1$ holds only if at least one literal in clause i evaluates to true
  - $X_5 = X_{12} = X_{19} =$ TRUE, then $-1 + 1 - 1 = -1$
  - $X_5 = \neg X_{12} = X_{19} =$ TRUE, then $-1 - 1 - 1 = -3$
- **Corollary:** nonconvex programming is NP-hard

Nonlinear Programming: Optimality

- In the sequel we consider the below nonlinear program:

$$\min \quad f(\boldsymbol{x}) : \boldsymbol{x} \in X$$

$$X = \Big\{ \boldsymbol{x} : g_i(\boldsymbol{x}) \le 0 \qquad i \in I = \{1, \dots, m\} \Big\}$$

  where f and g_i are **continuously differentiable** functions

- Smoothness (differentiability) is important as we want to use the gradient
- We seek conditions for given $\bar{\boldsymbol{x}} \in \mathbb{R}^n$ to be optimal
  - if $I = \emptyset$ (unconstrained problem) and $\bar{\boldsymbol{x}}$ is local minimum, then $\nabla f(\bar{\boldsymbol{x}}) = \mathbf{0}$ (proved below)
  - if f is convex and $I = \emptyset$ then this is also sufficient (next lecture)
  - if, on the other hand, $I \neq \emptyset$, then we can use the Karush-Kuhn-Tucker conditions or the below method

Improving Directions

- **Definition:** some $\boldsymbol{d} \in \mathbb{R}^n$ is an **improving direction** of function f at point $\bar{\boldsymbol{x}} \in X$ if there exists $\delta > 0$ so that

$$f(\bar{\boldsymbol{x}} + \lambda \boldsymbol{d}) < f(\bar{\boldsymbol{x}}) \qquad \forall \lambda \in (0, \delta)$$

- Moving along the improving direction we obtain better solutions
- Such directions can be characterized using the gradient of f
- **Theorem:** is f is smooth at point $\bar{\boldsymbol{x}}$ and there is $\boldsymbol{d}$ so that $\nabla f(\bar{\boldsymbol{x}})^T \boldsymbol{d} < 0$, then $\boldsymbol{d}$ is an improving direction of f in $\bar{\boldsymbol{x}}$
- **Proof:** the gradient characterizes the change in the value of function f while we move from point $\bar{\boldsymbol{x}}$ to $\bar{\boldsymbol{x}} + \lambda \boldsymbol{d}$

$$f(\bar{\boldsymbol{x}} + \lambda \boldsymbol{d}) - f(\bar{\boldsymbol{x}}) \approx \nabla f(\bar{\boldsymbol{x}})^T (\lambda \boldsymbol{d}) < 0 \qquad \square$$

Unconstrained Programs: Optimum

- **Corollary:** if f is differentiable at $\bar{\boldsymbol{x}}$ and $\bar{\boldsymbol{x}}$ is a local optimum of f, then $\nabla f(\bar{\boldsymbol{x}}) = \mathbf{0}$
- **Proof:** suppose that $\nabla f(\bar{\boldsymbol{x}}) \neq \mathbf{0}$, then the choice $\boldsymbol{d} = -\nabla f(\bar{\boldsymbol{x}})$ gives an improving direction, since

$$\nabla f(\bar{\boldsymbol{x}})^T \boldsymbol{d} = \nabla f(\bar{\boldsymbol{x}})^T (-\nabla f(\bar{\boldsymbol{x}})) = -\|\nabla f(\bar{\boldsymbol{x}})\|^2 < 0$$

- So we have found a direction $\boldsymbol{d}$ so that moving along $\boldsymbol{d}$ we obtain smaller values for f, which is a contradiction $\square$
- Cannot be used for solving nonlinear programs in general
  - since it is only a necessary condition, not sufficient
  - since the algebraic equation $\nabla f(\boldsymbol{x}) = \mathbf{0}$ usually cannot be solved directly
  - or because we may exit the feasible region along $\boldsymbol{d}$

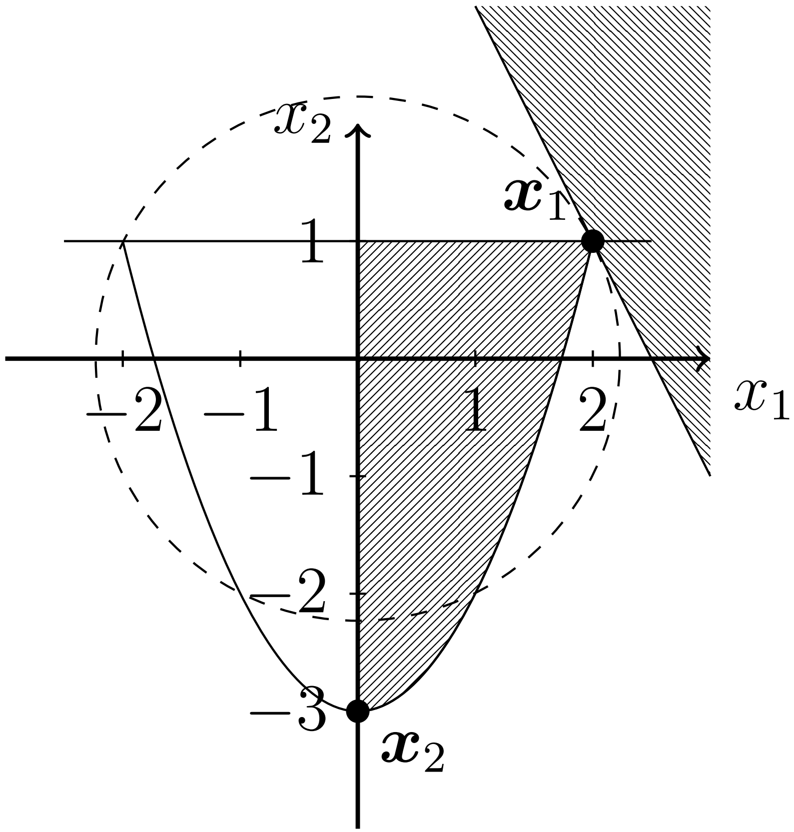
Improving Feasible Directions

- **Definition:** $\boldsymbol{d} \in \mathbb{R}^n$ is a **feasible direction** of some set X at point $\bar{\boldsymbol{x}} \in X$, if there is $\delta > 0$ so that

$$\forall \lambda \in (0, \delta) : \bar{\boldsymbol{x}} + \lambda \boldsymbol{d} \in X$$

- **Definition:** $\boldsymbol{d}$ is an **improving feasible direction** at $\boldsymbol{x}$ if it is both a feasible and an improving direction
- We can move some nonzero distance along the improving feasible direction to get all feasible solutions that improve the objective function
- **Theorem:** if $\bar{\boldsymbol{x}} \in X$ is a local minimum of the nonlinear program $\min f(\boldsymbol{x}) : \boldsymbol{x} \in X$ then there is no improving feasible direction at $\bar{\boldsymbol{x}}$
- **Proof:** suppose otherwise and obtain a contradiction $\square$
- The condition is again only necessary, not sufficient

Improving Directions: Example



- The running example:

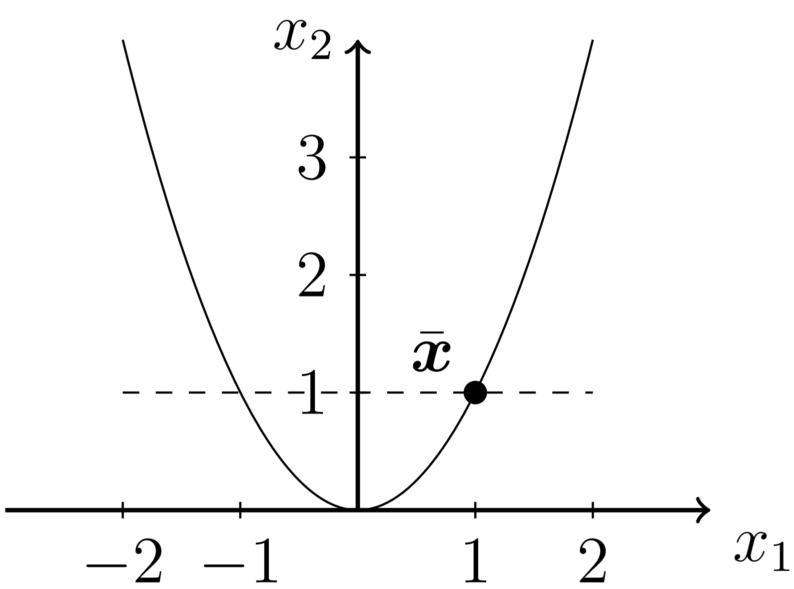
$$\min\ -x_1^2 - x_2^2$$
$$x_1^2 - x_2 - 3 \leq 0$$
$$x_2 - 1 \leq 0$$
$$-x_1 \leq 0$$

- $\nabla f(\boldsymbol{x})^T = [-2x_1 \quad -2x_2]$
- Consider the local minimum $\bar{\boldsymbol{x}}_1 = [2 \quad 1]^T$

- Improving directions lie in the open half-space $\nabla f(\bar{\boldsymbol{x}})^T \boldsymbol{d} = -4d_1 - 2d_2 < 0$, neither of which is feasible
- Similarly for local minimum $\boldsymbol{x}_2$

Improving Directions: Example

- Consider the nonlinear program $\min\{x_2 : x_2 = x_1^2\}$
- Point $\bar{\boldsymbol{x}} = [1 \quad 1]^T$ is not optimal, since x_2 decreases with decreasing x_1



- Yet there is no feasible direction at point $\bar{\boldsymbol{x}}$
- Since we cannot move along a straight line *and* still remain inside the feasible region
- So it does not follow from the lack of an improving feasible direction that a point is a local optimum
- The condition is only necessary

The Method of Feasible Directions

- We have seen that if some $\bar{\boldsymbol{x}}$ is local minimum of the nonlinear program $\min f(\boldsymbol{x}) : \boldsymbol{x} \in X$ then there is no improving feasible direction at $\bar{\boldsymbol{x}}$
- The condition is only necessary but not sufficient, since in pathological cases it can happen that there is no feasible direction at all at a point
- This cannot happen for “well-behaved” (e.g., convex) feasible regions
- The method of feasible directions due to Zoutendijk is a simple iterative algorithm to find a point where there is no improving feasible direction
- Traces back the solution of a nonlinear program to the sequential solution of simple linear programs (successive linear programming)
- Can be used for simpler convex programs

The Method of Feasible Directions

- Consider the following convex program, characterized by a linear constraint system:

$$\begin{aligned} \min \quad & f(\boldsymbol{x}) \\ \text{s.t.} \quad & \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b} \\ & \boldsymbol{Q}\boldsymbol{x} = \boldsymbol{q} \end{aligned}$$

  where $\boldsymbol{x} \in \mathbb{R}^n$, $\boldsymbol{A}$ is an $m \times n$ and $\boldsymbol{Q}$ is an $l \times n$ matrix, $\boldsymbol{b}$ is a column m-vector and $\boldsymbol{q}$ is a column l-vector, and f is a smooth convex function

- Let $\bar{\boldsymbol{x}}$ be a feasible solution and separate the constraint system to two groups:
  - denote by $\boldsymbol{A}_1\boldsymbol{x} \leq \boldsymbol{b}_1$ the **tight** (or active) constraints at $\bar{\boldsymbol{x}}$: $\boldsymbol{A}_1\bar{\boldsymbol{x}} = \boldsymbol{b}_1$
  - let $\boldsymbol{A}_2\boldsymbol{x} \leq \boldsymbol{b}_2$ be the **inactive** constraints: $\boldsymbol{A}_2\bar{\boldsymbol{x}} < \boldsymbol{b}_2$

The Method of Feasible Directions

- In what follows the term "feasible direction" will mean "improving feasible direction" where no ambiguity arises
- An algebraic characterization for feasible directions
- **Theorem:** some $\boldsymbol{d} \neq \mathbf{0}$ is a feasible direction at $\bar{\boldsymbol{x}}$ if

$$\nabla f(\bar{\boldsymbol{x}})^T \boldsymbol{d} < 0 \tag{1}$$

$$\boldsymbol{A}_1 \boldsymbol{d} \leq \mathbf{0} \text{ és } \boldsymbol{Q}\boldsymbol{d} = \mathbf{0} \tag{2}$$

- **Proof:** by (1) $\boldsymbol{d}$ is improving
- We show that it is a feasible direction as well
- Equality constraints hold for any $\boldsymbol{x} = \bar{\boldsymbol{x}} + \lambda \boldsymbol{d} : \lambda \in \mathbb{R}$, since

$$\boldsymbol{Q}\boldsymbol{x} = \boldsymbol{Q}(\bar{\boldsymbol{x}} + \lambda \boldsymbol{d}) = \boldsymbol{Q}\bar{\boldsymbol{x}} + \lambda \boldsymbol{Q}\boldsymbol{d} = \boldsymbol{q} + \lambda \mathbf{0} = \boldsymbol{q}$$

due to the assumption of the theorem that $\boldsymbol{Q}\boldsymbol{d} = \mathbf{0}$

The Method of Feasible Directions

- Tight constraints hold for any $\bar{\boldsymbol{x}} + \lambda \boldsymbol{d} : \lambda \geq 0$, since

$$\boldsymbol{A}_1(\bar{\boldsymbol{x}} + \lambda \boldsymbol{d}) = \boldsymbol{A}_1\bar{\boldsymbol{x}} + \lambda \boldsymbol{A}_1 \boldsymbol{d} = \boldsymbol{b}_1 + \lambda \boldsymbol{A}_1 \boldsymbol{d} \leq \boldsymbol{b}_1$$

  and, by assumption, $\boldsymbol{A}_1 \boldsymbol{d} \leq \boldsymbol{0}$
- To show that the inactive constraints hold as well, choose $\delta > 0$ small enough so that in the δ-neighborhood of $\bar{\boldsymbol{x}}$ the condition $\boldsymbol{A}_2 \boldsymbol{x} \leq \boldsymbol{b}_2$ is satisfied
- This can always be done
- We conclude that there is $\delta > 0$ so that $\bar{\boldsymbol{x}} + \lambda \boldsymbol{d}$ is both feasible and improving for any $\lambda \in (0, \delta)$ $\square$
- By the claim of the theorem, linear programming can be used to seek feasible directions
- The minimum can be iteratively found along such feasible directions

Finding a Feasible Direction: Example

- Consider the convex program

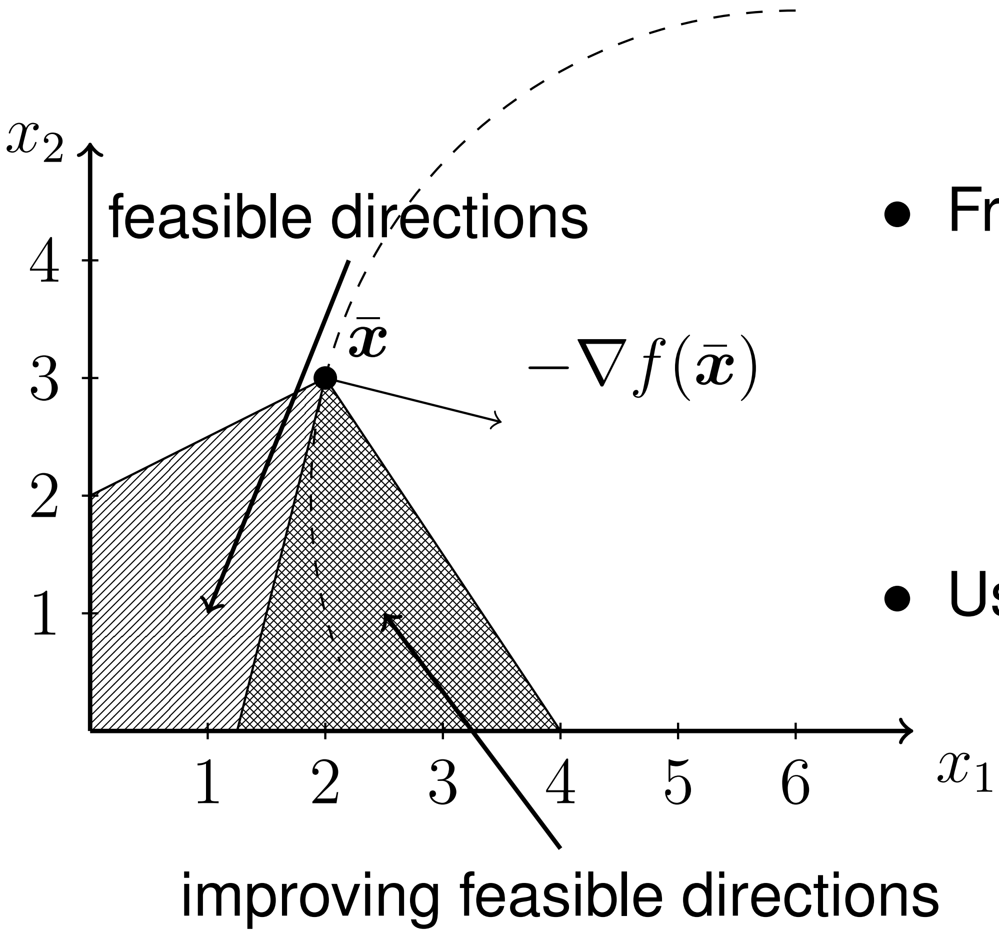
$$\begin{aligned} \min \quad & (x_1 - 6)^2 + (x_2 - 2)^2 \\ \text{s.t.} \quad & -x_1 + 2x_2 \leq 4 \\ & 3x_1 + 2x_2 \leq 12 \\ & x_1, x_2 \geq 0 \end{aligned}$$

- The tight conditions at $\bar{\boldsymbol{x}} = [2 \quad 3]^T$

$$\boldsymbol{A}_1 = \begin{bmatrix} -1 & 2 \\ 3 & 2 \end{bmatrix}, \quad \boldsymbol{b}_1 = \begin{bmatrix} 4 \\ 12 \end{bmatrix}$$

- The gradient of the objective function is $\nabla f(\bar{\boldsymbol{x}})^T = [-8 \quad 2]$ at point $\bar{\boldsymbol{x}}$, since $\nabla f(\boldsymbol{x})^T = [2(x_1 - 6) \quad 2(x_2 - 2)]$

Finding a Feasible Direction: Example



- From constraint $\boldsymbol{A}_1\boldsymbol{d} \leq \boldsymbol{0}$

$$-d_1 + 2d_2 \leq 0$$
$$3d_1 + 2d_2 \leq 0$$

- Using $\nabla f(\bar{\boldsymbol{x}})^T\boldsymbol{d} < 0$

$$-8d_1 + 2d_2 < 0$$

- Feasible directions are characterized by linear constraints
- Can be solved by linear programming

Finding a Feasible Direction

- Let $\min\{f(\boldsymbol{x}) : \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b}, \boldsymbol{Q}\boldsymbol{x} = \boldsymbol{q}\}$ be a convex program, let $\bar{\boldsymbol{x}}$ be a feasible solution, and denote by $\boldsymbol{A}_1\boldsymbol{x} \leq \boldsymbol{b}_1$ the tight and by $\boldsymbol{A}_2\boldsymbol{x} \leq \boldsymbol{b}_2$ the inactive constraints at $\bar{\boldsymbol{x}}$
- We search for $\boldsymbol{d}$ so that $\nabla f(\bar{\boldsymbol{x}})^T\boldsymbol{d} < 0$, $\boldsymbol{A}_1\boldsymbol{d} \leq \boldsymbol{0}$ and $\boldsymbol{Q}\boldsymbol{d} = \boldsymbol{0}$
- Solve the below linear program in variables $\boldsymbol{d}$:

$$z = \min \ \nabla f(\bar{\boldsymbol{x}})^T\boldsymbol{d} \tag{1}$$

$$\text{s.t. } \boldsymbol{A}_1\boldsymbol{d} \leq \boldsymbol{0} \tag{2}$$

$$\boldsymbol{Q}\boldsymbol{d} = \boldsymbol{0} \tag{3}$$

$$\nabla f(\bar{\boldsymbol{x}})^T\boldsymbol{d} \geq -1 \tag{4}$$

- Linear program because $\nabla f(\bar{\boldsymbol{x}})^T$ is a constant row-vector

Finding a Feasible Direction

- Let the optimal solution to the linear program be $\boldsymbol{d}$
- $\boldsymbol{d}$ is feasible by (2) and (3), and by (1) it is also improving
- Without (4) we would get unbounded minimum, since if for any $\boldsymbol{d}$ it holds that $\nabla f(\bar{\boldsymbol{x}})^T \boldsymbol{d} < 0$, $\boldsymbol{A}_1 \boldsymbol{d} \leq \mathbf{0}$, and $\boldsymbol{Q}\boldsymbol{d} = \mathbf{0}$, then it also holds for any $\lambda \boldsymbol{d}$, $\lambda > 0$ too
- (4) normalizes the resultant vector $\boldsymbol{d}$
- It is easy to see that $\boldsymbol{d} = \mathbf{0}$ is always feasible, therefore the optimal objective function value is guaranteed to be nonpositive
  - if $z < 0$ then it must hold that $z = -1$ by (4) and $\boldsymbol{d} \neq \mathbf{0}$ is an improving feasible direction at $\bar{\boldsymbol{x}}$
  - if $z = 0$ then there is no improving feasible direction at $\bar{\boldsymbol{x}}$ and so $\bar{\boldsymbol{x}}$ may be a local minimum

Finding a Feasible Direction: Example

- Find an improving feasible direction in the running example at point $\bar{\boldsymbol{x}} = [2 \quad 3]^T$
- Solve the below linear program:

$$
\begin{array}{lrcrcr}
\min & -8d_1 & + & 2d_2 & & \\
 & -d_1 & + & 2d_2 & \leq & 0 \\
 & 3d_1 & + & 2d_2 & \leq & 0 \\
 & -8d_1 & + & 2d_2 & \geq & -1
\end{array}
$$

- Multiply the third constraint by -1, introduce slack variables s_1, s_2, and s_3, and convert to maximization (note the eventual inversion!)
- Still not in standard form as variables d_1 and d_2 are free, whereas the simplex method requires nonegative variables

Finding a Feasible Direction: Example

- Substitute nonnegative variables according to $d_i = d'_i - d''_i$: $d'_i \geq 0$, $d''_i \geq 0$

$$
\begin{array}{lrrrrrrrcl}
\max & 8d'_1 & -8d''_1 & -2d'_2 & +2d''_2 & & & & & \\
 & -d'_1 & +d''_1 & +2d'_2 & -2d''_2 & +s_1 & & & = & 0 \\
 & 3d'_1 & -3d''_1 & +2d'_2 & -2d''_2 & & +s_2 & & = & 0 \\
 & 8d'_1 & -8d''_1 & -2d'_2 & +2d''_2 & & & +s_3 & = & 1 \\
 & d'_1, & d''_1, & d'_2, & d''_2, & s_1, & s_2, & s_3 & \geq & 0
\end{array}
$$

- Initial simplex tableau with the slack variables as basis:

	z	d'_1	d''_1	d'_2	d''_2	s_1	s_2	s_3	RHS
z	1	-8	8	2	-2	0	0	0	0
s_1	0	-1	1	2	-2	1	0	0	0
s_2	0	$\boxed{3}$	-3	2	-2	0	1	0	0
s_3	0	8	-8	-2	2	0	0	1	1

Finding a Feasible Direction: Example

	z	d_1'	d_1''	d_2'	d_2''	s_1	s_2	s_3	RHS
z	1	0	0	$\frac{22}{3}$	$-\frac{22}{3}$	0	$\frac{8}{3}$	0	0
s_1	0	0	0	$\frac{8}{3}$	$-\frac{8}{3}$	1	$\frac{1}{3}$	0	0
d_1'	0	1	-1	$\frac{2}{3}$	$-\frac{2}{3}$	0	$\frac{1}{3}$	0	0
s_3	0	0	0	$-\frac{22}{3}$	$\boxed{\frac{22}{3}}$	0	$-\frac{8}{3}$	1	1

	z	d_1'	d_1''	d_2'	d_2''	s_1	s_2	s_3	RHS
z	1	0	0	0	0	0	0	1	1
s_1	0	0	0	0	0	1	$-\frac{7}{11}$	$\frac{4}{11}$	$\frac{4}{11}$
d_1'	0	1	-1	0	0	0	$\frac{1}{11}$	$\frac{1}{11}$	$\frac{1}{11}$
d_2''	0	0	0	-1	1	0	$-\frac{4}{11}$	$\frac{3}{22}$	$\frac{3}{22}$

- We have found an improving feasible direction: $[\frac{1}{11} \quad -\frac{3}{22}]^T$

Choosing the Step Size

- Suppose that we have found a feasible direction $\boldsymbol{d}$ at some point $\bar{\boldsymbol{x}} \in X$ so that $\nabla f(\bar{\boldsymbol{x}})^T \boldsymbol{d} < 0$, $\boldsymbol{A}_1 \boldsymbol{d} \leq \boldsymbol{0}$ and $\boldsymbol{Q}\boldsymbol{d} = \boldsymbol{0}$
- Question is, how much to move along d
  - on the one hand, we must remain in the feasible region
  - on the other hand, we can go only as far as the objective function keeps on dropping and we must stop before it may start to increase again
- Thus, we need to solve the below convex program for λ:

$$
\begin{aligned}
\min \quad & f(\bar{\boldsymbol{x}} + \lambda \boldsymbol{d}) && (1)\\
\text{s.t.} \quad & \boldsymbol{A}(\bar{\boldsymbol{x}} + \lambda \boldsymbol{d}) \leq \boldsymbol{b} && (2)\\
& \boldsymbol{Q}(\bar{\boldsymbol{x}} + \lambda \boldsymbol{d}) = \boldsymbol{q} && (3)\\
& \lambda \geq 0 && (4)
\end{aligned}
$$

Choosing the Step Size

- (3) is redundant because $\boldsymbol{Q}\bar{\boldsymbol{x}} = \boldsymbol{q}$ and $\boldsymbol{Q}\boldsymbol{d} = \boldsymbol{0}$
- (1) can be replaced by the constraints $\boldsymbol{A}_1(\bar{\boldsymbol{x}} + \lambda \boldsymbol{d}) \leq \boldsymbol{b}_1$ and $\boldsymbol{A}_2(\bar{\boldsymbol{x}} + \lambda \boldsymbol{d}) \leq \boldsymbol{b}_2$
- From these $\boldsymbol{A}_1(\bar{\boldsymbol{x}} + \lambda \boldsymbol{d}) \leq \boldsymbol{b}_1$ trivially holds for each $\lambda > 0$, since $\boldsymbol{A}_1\bar{\boldsymbol{x}} = \boldsymbol{b}_1$ and $\boldsymbol{A}_1\boldsymbol{d} \leq \boldsymbol{0}$
- What remains are the inactive constraints $\boldsymbol{A}_2(\bar{\boldsymbol{x}} + \lambda \boldsymbol{d}) \leq \boldsymbol{b}_2$
- Simple convex program for λ:

$$
\begin{aligned}
\min \quad & f(\bar{\boldsymbol{x}} + \lambda \boldsymbol{d}) \\
\text{s.t.} \quad & (\boldsymbol{A}_2\boldsymbol{d})\lambda \leq \boldsymbol{b}_2 - \boldsymbol{A}_2\bar{\boldsymbol{x}} \\
& \lambda \geq 0
\end{aligned}
$$

- Here, $\boldsymbol{A}_2\boldsymbol{d}$ and $\boldsymbol{b}_2 - \boldsymbol{A}_2\bar{\boldsymbol{x}}$ are constant column vectors with as many components as there are inactive constraints at $\bar{\boldsymbol{x}}$

Choosing the Step Size

- Let $\boldsymbol{s} = \boldsymbol{A}_2\boldsymbol{d}$ and $\boldsymbol{t} = \boldsymbol{b}_2 - \boldsymbol{A}_2\bar{\boldsymbol{x}}$ be column r-vectors
- We know that $\boldsymbol{t} > \boldsymbol{0}$, as $\boldsymbol{A}_2\bar{\boldsymbol{x}} < \boldsymbol{b}_2$ is precisely the set of inactive constraints at $\bar{\boldsymbol{x}}$
- Our convex program can be written equivalently as

$$\min f(\bar{\boldsymbol{x}} + \lambda \boldsymbol{d}) : \lambda \in L$$

$$L = \{\lambda : \begin{array}{ll} s_1\lambda & \leq t_1 \\ s_2\lambda & \leq t_2 \\ \vdots & \vdots \\ s_r\lambda & \leq t_r \\ 0 & \leq \lambda \end{array} \}$$

- Convex program with a single unknown and simple structure
- Can be simplified even further

Choosing the Step Size

- The constraints are of the general form $s_i \lambda \leq t_i$, where $t_i > 0$ for each i

$$\min f(\bar{\boldsymbol{x}} + \lambda \boldsymbol{d}) : \lambda \geq 0,\ s_i \lambda \leq t_i, i = 1, \ldots, r$$

1. for $s_i < 0$ we get a redundant constraint $-|s_i|\lambda \leq t_i$, or equivalently $\lambda \geq -\frac{t_i}{|s_i|}(< 0)$ (redundant as $\lambda \geq 0$)
2. for $s_i = 0$ we get the trivial $0\lambda \leq t_i (> 0)$
3. $s_i > 0$ yields the irredundant constraint $\lambda \leq \frac{t_i}{s_i}(> 0)$

- We obtain the simplified form:

$$\min f(\bar{\boldsymbol{x}} + \lambda \boldsymbol{d}) : 0 \leq \lambda \leq \lambda_{\max}$$

$$\lambda_{\max} = \begin{cases} \min_{i \in \{1,\ldots,r\}} \left(\frac{t_i}{s_i} : s_i > 0 \right) & \text{ha } \boldsymbol{s} \not\leq \boldsymbol{0} \\ \infty & \text{ha } \boldsymbol{s} \leq \boldsymbol{0} \end{cases}$$

Choosing the Step Size

- Simple convex program with a single unknown

$$\min f(\bar{\boldsymbol{x}} + \lambda \boldsymbol{d}) : 0 \leq \lambda \leq \lambda_{\max}$$

- If $\lambda_{\max} = \infty$ then $\boldsymbol{x} + \lambda \boldsymbol{d}$ remains feasible for any λ: unconstrained line search
- Otherwise the search interval: $\lambda \in [0, \lambda_{\max}]$
- Convex program, can also be solved using the method of feasible directions
- Or use a direct line search method (next lecture)

Choosing the Step Size: Example

- Consider the running example:

$$
\begin{aligned}
\min \ & (x_1 - 6)^2 + (x_2 - 2)^2 \\
\text{s.t.} \ & -x_1 + 2x_2 \leq 4 \\
& 3x_1 + 2x_2 \leq 12 \\
& x_1, x_2 \geq 0
\end{aligned}
$$

- After solving the respective linear program, we obtained the feasible direction $\boldsymbol{d} = [\frac{1}{11} \quad -\frac{3}{22}]^T$ at $\bar{\boldsymbol{x}} = [2 \quad 3]^T$
- We need to find the best step size to move along $\boldsymbol{d}$
- It is worth scaling $\boldsymbol{d}$ in a way that the components become integer-valued (the norm of $\boldsymbol{d}$ does not matter)
- So let $\boldsymbol{d} = [2 \quad -3]^T$

Choosing the Step Size: Example

- Solve $\min \{f(\bar{\boldsymbol{x}} + \lambda \boldsymbol{d}) : s_i \lambda \leq t_i, i = 1, \ldots, r\}$ to find the optimal step size, where $\boldsymbol{s} = \boldsymbol{A}_2 \boldsymbol{d}$ and $\boldsymbol{t} = \boldsymbol{b}_2 - \boldsymbol{A}_2 \bar{\boldsymbol{x}}$ are column r-vectors and $\boldsymbol{A}_2 \boldsymbol{x} \leq \boldsymbol{b}_2$ are the inactive constraints
- At $\bar{\boldsymbol{x}} = [2 \quad 3]^T$ the first two conditions are tight, the nonnegativity constraints ($x_1 \geq 0$, $x_2 \geq 0$) are inactive
- From this $\boldsymbol{A}_2$ and $\boldsymbol{b_2}$, and $\boldsymbol{s}$ and $\boldsymbol{t}$

$$\boldsymbol{A}_2 = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}, \qquad \boldsymbol{b}_2 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\boldsymbol{s} = \boldsymbol{A}_2 \boldsymbol{d} = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} 2 \\ -3 \end{bmatrix} = \begin{bmatrix} -2 \\ 3 \end{bmatrix}$$

$$\boldsymbol{t} = \boldsymbol{b}_2 - \boldsymbol{A}_2 \bar{\boldsymbol{x}} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} - \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} 2 \\ 3 \end{bmatrix} = \begin{bmatrix} 2 \\ 3 \end{bmatrix}$$

Choosing the Step Size: Example

- Write the objective function as the function of λ:

$$\theta(\lambda) = f(\bar{\boldsymbol{x}}+\lambda \boldsymbol{d}) = ((\bar{x}_1+\lambda d_1)-6)^2+((\bar{x}_2+\lambda d_2)-2)^2 =$$
$$((2+2\lambda)-6)^2+((3-3\lambda)-2)^2 = (2\lambda-4)^2+(1-3\lambda)^2 =$$
$$13\lambda^2 - 22\lambda + 17$$

- To find the optimal step size λ, we need to solve the below convex program:

$$\begin{aligned} \min \quad & 13\lambda^2 - 22\lambda + 17 \\ \text{s.t.} \quad & -2\lambda \leq 2 \\ & 3\lambda \leq 3 \\ & \lambda \geq 0 \end{aligned}$$

Choosing the Step Size: Example

- The first constraint $\lambda \geq -1$ is redundant, there remains the second constraint:

$$\min \; 13\lambda^2 - 22\lambda + 17 : 0 \leq \lambda \leq 1$$

- The objective (a parabola) attains its minimum at $\frac{11}{13} < 1$
- So $\lambda \leq 1$ is not tight, the optimal step size is $\bar{\lambda} = \frac{11}{13}$
- Consequently, from point $[2 \quad 3]^T$ we move along direction $\boldsymbol{d} = [2 \quad -3]^T$ to a distance $\bar{\lambda} = \frac{11}{13}$ to arrive to the new point $\bar{\boldsymbol{x}} = [\frac{48}{13} \quad \frac{6}{13}]^T$
- Here it was easy to solve the line search problem directly, in the general case it may be more difficult (next lecture)
- To solve the original convex program, we now need to find a feasible direction at point $\bar{\boldsymbol{x}} = [\frac{48}{13} \quad \frac{6}{13}]^T$

Finding a Feasible Direction: Example

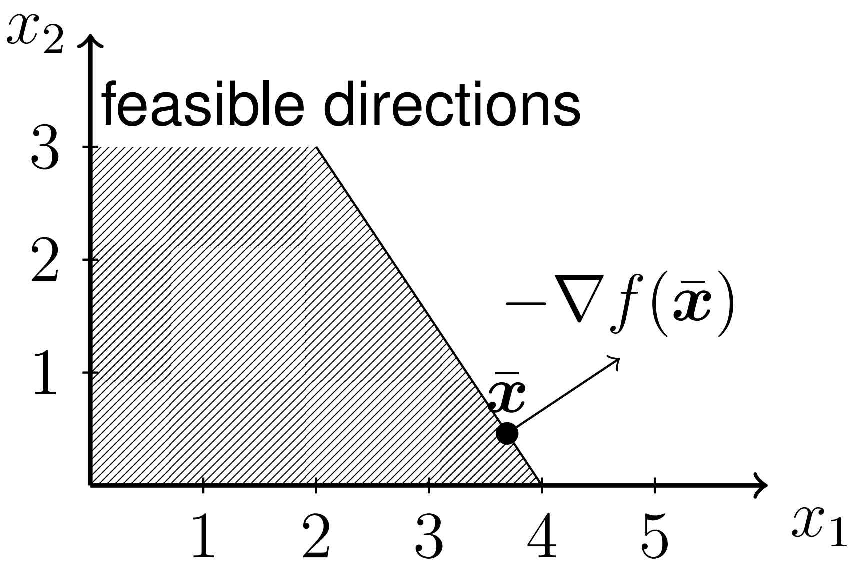
- Solve $\min\,\{\nabla f(\bar{\boldsymbol{x}})^T \boldsymbol{d} : \boldsymbol{A}_1 \boldsymbol{d} \leq \boldsymbol{0}, \nabla f(\bar{\boldsymbol{x}})^T \boldsymbol{d} \geq -1\}$ to find the new feasible direction
- At $\bar{\boldsymbol{x}} = [\frac{48}{13} \quad \frac{6}{13}]^T$ only the first constraint is tight, so $A_1 = [3 \quad 2]$
- The gradient: $\nabla f(\bar{\boldsymbol{x}}) = [-\frac{60}{13} \quad -\frac{40}{13}]^T$
- We can again scale the gradient freely, only the direction matters
- So consider the gradient $[-3 \quad -2]^T$ (multiply $\nabla f(\bar{\boldsymbol{x}})$ by $\frac{20}{13}$), from which we obtain:

$$
\begin{array}{lrcrcr}
\min & -3d_1 & - & 2d_2 & & \\
\text{s.t.} & 3d_1 & + & 2d_2 & \leq & 0 \\
 & -3d_1 & - & 2d_2 & \geq & -1
\end{array}
$$

Finding a Feasible Direction: Example

- After inverting the second constraint, as maximization:

$$\begin{array}{lcccrr} \max & 3d_1 & + & 2d_2 & & \\ \text{s.t.} & 3d_1 & + & 2d_2 & \leq & 0 \\ & 3d_1 & + & 2d_2 & \leq & 1 \end{array}$$



- Second constraint redundant, the optimum is 0
- The gradient is orthogonal to the boundary of the set of feasible directions
- No improving feasible direction at $\bar{\boldsymbol{x}} = \begin{bmatrix} \frac{48}{13} & \frac{6}{13} \end{bmatrix}^T$
- (Probably) optimal solution

Finding a Feasible Direction: Example

- Now suppose that we wish to solve the same nonlinear program, but this time starting from the point $\bar{\boldsymbol{x}} = [4 \quad 0]^T$

$$\begin{aligned} \min \;\; & (x_1 - 6)^2 + (x_2 - 2)^2 \\ \text{s.t.} \;\; & -x_1 + 2x_2 \leq 4 \\ & 3x_1 + 2x_2 \leq 12 \\ & x_1, x_2 \geq 0 \end{aligned}$$

- To find a feasible direction, we need the set of inactive constraints and the gradient at $\bar{\boldsymbol{x}}$

$$\boldsymbol{A}_1 = \begin{bmatrix} 3 & 2 \\ 0 & -1 \end{bmatrix}, \quad \nabla f(\bar{\boldsymbol{x}}) = [-4 \quad -4]^T$$

Finding a Feasible Direction: Example

- The linear program to find the feasible direction:

$$\begin{array}{lrcrcr} \min & -4d_1 & - & 4d_2 & & \\ \text{s.t.} & 3d_1 & + & 2d_2 & \leq & 0 \\ & & & -d_2 & \leq & 0 \\ & -4d_1 & - & 4d_2 & \geq & -1 \end{array}$$

- From the second constraint $d_2 \geq 0$, no need to substitute
- Inverting the third condition, substituting $d_1 = d_1' - d_1''$: $d_1' \geq 0$, $d_1'' \geq 0$, introducing slack variables, and converting to maximization:

$$\begin{array}{lrcrcrcrcrcr} \max & 4d_1' & - & 4d_1'' & + & 4d_2 & & & & & & \\ \text{s.t.} & 3d_1' & - & 3d_1'' & + & 2d_2 & + & s_1 & & & = & 0 \\ & 4d_1' & - & 4d_1'' & + & 4d_2 & & & + & s_2 & = & 1 \\ & d_1', & & d_1'', & & d_2, & & s_1, & & s_2 & \geq & 0 \end{array}$$

Finding a Feasible Direction: Example

- Initial simplex tableau (primal feasible):

	z	d_1'	d_1''	d_2	s_1	s_2	RHS
z	1	-4	4	-4	0	0	0
s_1	0	$\boxed{3}$	-3	2	1	0	0
s_2	0	4	-4	4	0	1	1

- The optimal tableau:

	z	d_1'	d_1''	d_2	s_1	s_2	RHS
z	1	0	0	0	0	1	1
d_2	0	0	0	1	-1	$\frac{3}{4}$	$\frac{3}{4}$
d_1''	0	-1	1	0	-1	$\frac{1}{2}$	$\frac{1}{2}$

- The improving feasible direction: $\boldsymbol{d} = [-\frac{1}{2} \quad \frac{3}{4}]^T$

Finding a Feasible Direction: Example

- Rescaling yields $\boldsymbol{d} = [-2 \quad 3]^T$
- Parameters for computing the optimal step size:

$$\boldsymbol{A}_2 = \begin{bmatrix} -1 & 2 \\ -1 & 0 \end{bmatrix}, \qquad \boldsymbol{b}_2 = \begin{bmatrix} 4 \\ 0 \end{bmatrix}$$

$$\boldsymbol{s} = \boldsymbol{A}_2\boldsymbol{d} = \begin{bmatrix} -1 & 2 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} -2 \\ 3 \end{bmatrix} = \begin{bmatrix} 8 \\ 2 \end{bmatrix}$$

$$\boldsymbol{t} = \boldsymbol{b}_2 - \boldsymbol{A}_2\bar{\boldsymbol{x}} = \begin{bmatrix} 4 \\ 0 \end{bmatrix} - \begin{bmatrix} -1 & 2 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} 4 \\ 0 \end{bmatrix} = \begin{bmatrix} 8 \\ 4 \end{bmatrix}$$

$$\theta(\lambda) = f(\bar{\boldsymbol{x}} + \lambda\boldsymbol{d}) = ((4 - 2\lambda) - 6)^2 + ((0 + 3\lambda) - 2)^2 =$$

$$= (2\lambda - 4)^2 + (1 - 3\lambda)^2 = 13\lambda^2 - 4\lambda + 8$$

Finding a Feasible Direction: Example

- The optimal step size:

$$\begin{aligned} \min \ \ & 13\lambda^2 - 4\lambda + 8 \\ \text{s.t.} \ \ & 8\lambda \leq 2 \\ & 8\lambda \leq 4 \\ & \lambda \geq 0 \end{aligned}$$

- Simplified: $\min \ 13\lambda^2 - 4\lambda + 8 : 0 \leq \lambda \leq \frac{1}{4}$
- Solution at the minimum of the parabola: $\bar{\lambda} = \frac{2}{13} < \frac{1}{4}$

$$\bar{\boldsymbol{x}} + \bar{\lambda}\boldsymbol{d} = \begin{bmatrix} 4 \\ 0 \end{bmatrix} + \frac{2}{13} \begin{bmatrix} -2 \\ 3 \end{bmatrix} = \begin{bmatrix} \frac{48}{13} \\ \frac{6}{13} \end{bmatrix}$$

- We returned to the same optimal point $[\frac{48}{13} \quad \frac{6}{13}]^T$ as before

Method of Feasible Directions: Summary

- Given a convex program $\min f(\boldsymbol{x}) : \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b}, \boldsymbol{Q}\boldsymbol{x} = \boldsymbol{q}$, an initial feasible solution $\boldsymbol{x}_1$, and let $k = 1$

1. Find an improving feasible direction at $\boldsymbol{x}_k$: decompose $\boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b}$ to tight $\boldsymbol{A}_1\boldsymbol{x}_k = \boldsymbol{b}_1$ and inactive $\boldsymbol{A}_2\boldsymbol{x}_k < \boldsymbol{b}_2$ constraints at $\boldsymbol{x}_k$

- Let $\boldsymbol{d}_k$ be the optimal solution to the below linear program:

$$
\begin{aligned}
z = \min \quad & \nabla f(\boldsymbol{x}_k)^T \boldsymbol{d} \\
\text{s.t.} \quad & \boldsymbol{A}_1 \boldsymbol{d} \leq \boldsymbol{0} \\
& \boldsymbol{Q}\boldsymbol{d} = \boldsymbol{0} \\
& \nabla f(\boldsymbol{x}_k)^T \boldsymbol{d} \geq -1
\end{aligned}
$$

- If the optimal objective $z_{\text{opt}} = 0$ then halt, $\boldsymbol{x}_k$ is optimal
- Otherwise, $z_{\text{opt}} = -1$ and $\boldsymbol{d}_k \neq \boldsymbol{0}$: line search along $\boldsymbol{d}_k$

Method of Feasible Directions: Summary

2. Let $\boldsymbol{s} = \boldsymbol{A}_2\boldsymbol{d}$ and $\boldsymbol{t} = \boldsymbol{b}_2 - \boldsymbol{A}_2\boldsymbol{x}_k$

- Let λ_k be the optimal solution to the below convex program:

$$\min f(\boldsymbol{x}_k + \lambda\boldsymbol{d}) : 0 \leq \lambda \leq \lambda_{\max}$$

$$\lambda_{\max} = \begin{cases} \min_{i\in\{1,\ldots,r\}}\left(\frac{t_i}{s_i} : s_i > 0\right) & \text{ha } \boldsymbol{s} \not\leq \boldsymbol{0} \\ \infty & \text{ha } \boldsymbol{s} \leq \boldsymbol{0} \end{cases}$$

- Let $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \lambda_k\boldsymbol{d}_k$, $k = k+1$ and go to (1)
- Can also be modified for convex programs over nonlinear constraints, and also for nonconvex programs
- Globally non-convergent (zig-zagging), for convergence we'd need a second-order method
- Uses line search as a subroutine