# Constrained Optimization Algorithms

**Antonio Frangioni**

Department of Computer Science
University of Pisa
https://www.di.unipi.it/~frangio
mailto:frangio@di.unipi.it

Computational Mathematics for Learning and Data Analysis
Master in Computer Science – University of Pisa

A.Y. 2024/25

## Outline

- Algorithms for $(P)$ $\min\{ f(x) : G(x) \leq 0, H(x) = 0 \}$
- Never ever nonconvex $X$: very nasty + not used in learning
  $\Longrightarrow$ only linear equalities $Ax = b$ (almost dealt with already)
- Almost only linear inequalities $Ax \leq b$, very convenient: i) always convex, ii) satisfy (LinI), iii) numerically stable, iv) cheap to compute ...
  a few hints to nonlinear convex case when things easily extend
- Usually ignore $H(\cdot)$ (implementation details), just $G(x) \leq 0 \equiv Ax \leq b$
- Important notation: sub-system (a relaxation). $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$
  $B \subseteq \{1, 2, \ldots, m\}$ subset of row('s indice)s $\equiv$ constraints
  sub-matrix / -vector / -system: $A_B = [A_i]_{i \in B}$ / $b_B = [b_i]_{i \in B}$ / $A_B x \leq b_B$
- Crucial polyhedral cone: $T_X(x) = F_X(x) = \{ d \in \mathbb{R}^n : A_{\mathcal{A}(x)} d \leq 0 \}$
- Important point: exploiting special structures in the constraints
  (only a few hints given, there is a lot more of that)

## Outline

- Equality-constrained QP: $(P)\ \ \min\left\{ \frac{1}{2}x^T Qx + qx \ :\ Ax = b \right\}$
  $A \in \mathbb{R}^{m\times n}$, w.l.o.g. $\operatorname{rank}(A) = m < n \ \equiv$ rows of $A$ linearly independent
- Usually $(P)$ convex $\ \equiv\ Q \succeq 0$ (otherwise $v(P) = -\infty$ likely)
- Minimum / saddle point: just solve the KKT system / normal equations

$$\begin{matrix}(a)\\(b)\end{matrix}\quad \begin{bmatrix} Q & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} x \\ \mu \end{bmatrix} = \begin{bmatrix} -q \\ b \end{bmatrix} \quad \text{“only linear algebra”}$$

  (symmetric but indefinite, lots of 0 eigenvalues)
- Basic step in many $\neq$ cases $\implies$ have to do that efficiently
- Clearly Federico's playground, let's just hint at some possible ways
- Just go and solve it by direct or iterative methods:
  - indefinite factorization of the matrix (may reduce sparsity)
  - Krylov-type iterative methods (GMRES, ...)
- Or try to exploit the large-scale structure (saddle-point system)

## Mathematically speaking: Reduced KKT [8, p. 455]

- $Q$ nonsingular: multiply (a) by $AQ^{-1}$ + (b) $\Longrightarrow$

$$[\, AQ^{-1}A^T \,]\mu = -b - AQ^{-1}q \quad \wedge \quad x = -Q^{-1}(\, A^T\mu + q \,)$$

- Schur Complement $M = AQ^{-1}A^T \succeq 0$ if $Q \succeq 0$
- $M \in \mathbb{R}^{m\times m}$ "small" $(m < n < m + n)$
- $M$ can be very dense even if $A$, $Q$ sparse ...
- Heuristics to permute rows to improve sparsity
- Iterative methods to solve the systems without forming $M$
  (Preconditioned Conjugate Method, appropriate preconditioners ...)

## Mathematically speaking: Null Space methods [8, p. 457]

- $A = [\, A_B \,,\, A_N \,]$, $x = [\, x_B \,,\, x_N \,]$, $det(\, A_B \,) \neq 0 \implies$

  (b) $\equiv$ $x_B = A_B^{-1}(\, b - A_N x_N \,) \implies x = Dx_N + d$ with

  $$d = \begin{bmatrix} b \\ 0 \end{bmatrix} \;,\; D = \begin{bmatrix} -A_B^{-1} A_N \\ I \end{bmatrix} \in \mathbb{R}^{m \times n-m}$$

  basis of null space of $A$ $\equiv AD = 0$

- Multiply (a) by $D^T \implies D^T Q x - D^T A^T \mu =$

  $D^T Q(\, Dx_N + d \,) = -D^T q \implies [\, D^T Q D \,] x_N = -D^T(\, Qd + q \,)$

- Reduced Hessian $H = D^T Q D \in \mathbb{R}^{n-m \times n-m}$ "small", $\succeq 0$ if $Q$ is

- Can be generalized to any basis of null space of $A$

- $Q$ does not need to be nonsingular

- $H$ can be very dense even if $A$, $Q$ sparse, proper choices of $D$ ...

- Iterative methods to solve the systems without forming $H$

  (Preconditioned Conjugate Method, appropriate preconditioners ...)

## Outline

- QP with linear constraints: $(P)\ \ \min\{\, f(x) = \frac{1}{2} x^T Q x + qx \ :\ Ax \le b \,\}$
- If one knew $\mathcal{A}(x_*)$, then it would be "just linear algebra"
- "If you don't know it estimate it, but be ready to revise your estimate": exploit dual prices to help revising it

**procedure** $x = ASMQP\,(\,Q\,,\,q\,,\,A\,,\,b\,,\,x\,)$ // precondition: $x$ s.t. $Ax \le b$
  **for**( $B \leftarrow \mathcal{A}(x)$ ; ; )
    solve $(\mathrm{P}_B)\ \ \min\{\, f(x) \ :\ A_B x = b_B \,\}$; // $(\bar{x}\,,\,\bar{\mu}_B\,)$ s.t. $-\nabla f(\bar{x}) = \bar{\mu}_B A_B$
    **if**( $A_i \bar{x} \le b_i\ \forall\, i \notin B$ ) **then**
      **if**( $\bar{\mu}_B \ge 0$ ) **then return**;
      $h \leftarrow \min\{\, i \in B \ :\ \bar{\mu}_i < 0 \,\}$; $B \leftarrow B \setminus \{\, h \,\}$; **continue**;
    $d \leftarrow \bar{x} - x$; $\bar{\alpha} \leftarrow \min\{\, \alpha_i = (\, b_i - A_i x \,) \,/\, A_i d \ :\ A_i d > 0\ ,\ i \notin B \,\}$;
    $x \leftarrow x + \bar{\alpha} d$; $B \leftarrow \mathcal{A}(x)$;

- $B =$ "active set", current estimate of $\mathcal{A}(x_*)$
- Can compute feasible initial $x$ (if any) autonomously (**check**) (nontrivial)
- $\bar{\alpha} = \max\{\, \alpha \ :\ A_i(\, x + \alpha d \,) \le b_i \,\} < \infty$ (**check**)

**Exercise:** the code has two glaring omissions if $Q \not\succ 0$: find and fix them

- $A\bar{x} \not\leq b \implies d = \bar{x} - x$ descent direction,
  $\bar{\alpha} \in \operatorname{argmin}\{ f(x + \alpha d) : x + \alpha d \in X \} < 1$, $\mathcal{A}(x + \bar{\alpha} d) \supsetneqq B$
- $A\bar{x} \leq b$ and $\bar{\mu}_B \geq 0 \implies \mu = \lambda\,(\geq 0) \equiv \bar{x}$ optimal (**check**)
- (LinI)-type condition: $A_{\mathcal{A}}(x)$ full row rank $\forall\, x$
- $A\bar{x} \leq b$ but $\exists\, h \in B$ s.t. $\bar{\mu}_h < 0 \implies \nu(P_{B'}) < \nu(P_B)$ with $B' = B \setminus \{h\}$
  Proof: w.l.o.g. $B = \mathcal{A}(x)$; if $B \subset \mathcal{A}(x)$ then $\nu(P_{B'})$ even smaller
  $A_B = A_{\mathcal{A}(x)}$ full rank $\equiv$ $A_h \notin \operatorname{range}(A_{B'}) \equiv$
  $A_h = v + d$ with $d \neq 0$, $v \perp d$, $v \in \operatorname{range}(A_{B'}) \implies A_{B'} d = 0$
  $\langle A_h, -d \rangle = -\langle v, d \rangle - \| d \|^2 < 0 \implies A_{\mathcal{A}(x)}(-d) \leq 0 \equiv -d \in F_X(x)$
  $\langle \nabla f(\bar{x}), -d \rangle = \bar{\mu}_h \langle A_h, d \rangle + \bar{\mu}_{B'} A_{B'}^T d < 0$: $d$ feasible and of descent
- Finitely terminates: once found the right $B$ the problem is over $+$
  cannot have the same $B$ twice since $f(x)$ strictly decreases [8, p. 477]
- "Just" have to search among $2^m$ possible $B$

- Important: always exploit all the structure of your problem
- QP with box constraints: (P) $\min\{ \frac{1}{2} x^T Q x + q x \ : \ \underline{x} \leq x \leq \overline{x} \}$
- Active constraint $\equiv$ inactive variable (fixed), "$B \subseteq N = \{1, \ldots, n\}$"
- $B = (L, U)$ , $L \cap U = \emptyset$ , $L \cup U \subset N$ , $F = N \setminus (L \cup U) \implies$
  $A_B x = b_B \equiv x = [x_L, x_F, x_U] = [\underline{x}_L, x_F, \overline{x}_U]$: only $x_F$ "free"
- W.l.o.g. $\underline{x} = 0$ $(x \leftarrow x + \underline{x}) \implies x = [0, x_F, \overline{x}_U]$
- $(P_B)$ $\min\{ \frac{1}{2} x_F^T Q_{FF} x_F + ( q_F + \overline{x}_U^T Q_{UF} ) x_F \}$ $[+\frac{1}{2} \overline{x}_U^T Q_{UU} \overline{x}_U + q_U \overline{x}_U]$
  unconstrained and in a (possibly, much) smaller space
- (LinI)-type condition obviously holds
- Initial feasible $x$ straightforward

**Exercise:** $(P_B)$ as written above does not have constraints: discuss how to compute $\bar{\mu}$ and/or how to replace it in the algorithm

- If (LinI)-type not satisfied, $B$ need be more carefully managed to handle possible degenerate steps ($\bar{\alpha} = 0$)
- For instance, only one constraint at a time is added to $B$ [8, Ex. 16.18], not very efficient in practice, different rules used (nontrivial)
- Exploit information from previous iteration to speed up KKT system solution: update factorizations [8, p. 478], use $x$ to warm-start iterative approaches ...
- Many different variants (direct / iterative, $H$ / $M$)
- Can be extended to $f(x)$ generic, but ($P_B$) general unconstrained problem
- No more exact solution, but (hopefully, fast) iterative approaches: e.g., (quasi-)Newton on the reduced problem [4, §10.2–.4]
- Which $\varepsilon$ / how many iterations? How about one of the gradient method?

## Outline

- Nonlinear problem with linear constraints: $(P)\ \ \min\{ f(x) : Ax \leq b \}$
- Another (primal) feasible approach: keep $Ax \leq b$, easy to initialize (**check**)
- If $-\nabla f(x) \in F_X(x)$ trivial: just LS / FS along $d = -\nabla f(x)$
  $(\Longleftarrow\ x \in \text{int}(X)\ \equiv\ \mathcal{A}(x) = \emptyset\ \Longrightarrow\ F_X(x) = \mathbb{R}^n)$
- If not: find $d \in F_X(x)$ "closer" to $-\nabla f(x)$, then LS/FS along $d$
- Projection of $d \in \mathbb{R}^n$ on $S \subset \mathbb{R}^n$: $p_S(d) = \min\{ \| d - x \| : x \in S \}$
- $p_{F_X(x)}(-\nabla f(x)) = \min\{ h(d) = \| d + \nabla f(x) \|^2 / 2 : A_{\mathcal{A}(x)} d \leq 0 \}$
  a convex QP with simple function on a polyhedral cone
- Not trivial but clearly doable, e.g. ASQPM (possibly streamlined, will see)
- $d^* = 0\ \Longrightarrow\ \lambda^* A(x) + \nabla f(x) = 0\ ,\ \lambda^* \geq 0$ (**check**) $\equiv$ $x$ optimal for $(P)$
  $\Longrightarrow$ stopping condition $\| d^* \| \leq \varepsilon$ (norm of projected gradient)
- $p_{F_X(x)}(\cdot)$ potentially costly, but can be very cheap with appropriate $X$

- Always exploit all the structure of your problem: box constraints $\underline{x} \leq x \leq \overline{x}$
- $X = X_1 \times X_2 \times \ldots \times X_n$ and $\| d - v \|^2 = \sum_{i=1}^{n} ( d_i - v_i )^2$ decomposable $\Longrightarrow$ $p_X( x )$ separable: $n$ independent problems $\Longrightarrow$ much easier & parallelizable

**procedure** $x = BCPGM\ ( f , \underline{x} , \overline{x} , x , \varepsilon )$
  **for**( ; ; )
    $d = -\nabla f( x )$; $\bar{\alpha} = \infty$;
    **foreach**( $i = 1 \ldots n$ s.t. $d_i \neq 0$ ) **do**
      **if**( $d_i < 0$ ) **then if**( $x_i = \underline{x}_i$ ) **then** $d_i = 0$ **else** $\bar{\alpha} \leftarrow \min\{ \bar{\alpha} , ( \underline{x}_i - x_i ) / d_i \}$
                **else if**( $x_i = \overline{x}_i$ ) **then** $d_i = 0$ **else** $\bar{\alpha} \leftarrow \min\{ \bar{\alpha} , ( \overline{x}_i - x_i ) / d_i \}$
    **if**( $\langle \nabla f( x ) , d \rangle \leq \varepsilon$ ) **then return**;
    $\alpha \leftarrow$ choose_step( $f , x , d , \bar{\alpha} , \varepsilon$ ); $x \leftarrow x + \alpha d$;

**Exercise:** justify why $d$ computed in the pseudo-code is the projected anti-gradient

**Exercise:** discuss what choose_step() should look like

- Other easy projections, e.g., balls or simplex constraints $\sum_{i=1}^{n} x_i = 1 \ , \ x \geq 0$

**Exercise:** propose a fast dual method to project for a simplex constraint

## Projected gradient method, Goldstein's version

- What if projection is easy on the whole of $X$, not only $F_X(x)$?
- Goldstein's projected gradient method: move first, project second

  $$y^i \leftarrow x^i - \alpha^i \nabla f(x^i);\ x^{i+1} \leftarrow p_X(y^i);$$

- Not a descent method, more like Heavy Ball
- Only converges with appropriate stepsize, typical $\alpha = 1/L$
- Convergence results $\approx$ unconstrained gradient [9, §5] (for good and bad):
  $O((L/\tau)\log(1/\varepsilon))$ for $f$ $\tau$-convex, $O(LD/\varepsilon)$ otherwise
- Projection cost can be very small (e.g., box constraints)

**Exercise:** Develop $p_X$ for box constraints . . . where have I seen it?

- Other easy projections, e.g., balls or simplex constraints [5] (sounds familiar?)

**Exercise:** develop $p_X$ for the ball in the 2-norm

- Practical convergence can be quite different

- General $Ax \leq b$: $p_{F_X(x)}$ can be too costly (not to mention $p_X$)
- Make it easier by projecting on $\partial F_X(x) = \{ d \in \mathbb{R}^n : A_{\mathcal{A}(x)} d = 0 \}$
  QP with easy objective and equality constraints $\Longrightarrow$ very easy
- In fact, $\bar{A} = A_{\mathcal{A}(x)}$ full row rank $\Longrightarrow$ closed formula (**check**)

$$\mu = -[\bar{A}\bar{A}^T]^{-1}\bar{A}\nabla f(x) \quad , \quad d = (I - \bar{A}^T[\bar{A}\bar{A}^T]^{-1}\bar{A})(-\nabla f(x))$$

- $d = 0$ may happen: good if $\mu \geq 0$ (**check**), un-good otherwise
- $d = 0$ surely happens if $\bar{A} \in \mathbb{R}^{n \times n}$ ($x$ a vertex, $\mathcal{A}(x)$ a base) (**check**)
- $A_{\mathcal{A}(x)}$ not full rank in general: must work with $A_B$ full rank, $B \subset \mathcal{A}(x)$
  $\Longrightarrow$ rather more complicated logic
- $f$ linear + streamlining $\rightsquigarrow$ primal simplex method [8, Chap. 13][7, Chap. 3]
- Can be extended to $G(x) \leq 0$ nonlinear (nontrivial) [3, p. 597][7, p. 371]
- When $B$ gives the optimal face $\approx$ unconstrained steepest descent $\Longrightarrow$
  convergence results analogous (with twists) [7, §12.5] (for good and bad)

**procedure** $x = PGM\,(\,f\,,\,A\,,\,b\,,\,x\,,\,\varepsilon\,)$ // invariant: $Ax \leq b$
  **for**( ; ; )
    $B \leftarrow$ maximal $\subseteq \mathcal{A}(\,x\,)$ s.t. $rank(\,A_B\,) = \#B$;
    **for**( ; ; )
      $d \leftarrow (\,I - A_B^T[\,A_B A_B^T\,]^{-1} A_B\,)(\,-\nabla f(\,x\,)\,)$;
      **if**( $\langle\,\nabla f(\,x\,)\,,\,d\,\rangle \leq \varepsilon$ ) **then**
        $\mu_B \leftarrow -[\,A_B A_B^T\,]^{-1} A_B \nabla f(\,x\,)$;
        **if**( $\mu_B \geq 0$ ) **then return**;
        $h \leftarrow \min\{\,i \in B\ :\ \mu_i < 0\,\}$; $B \leftarrow B \setminus \{\,h\,\}$; **continue**;
      $\bar{\alpha} \leftarrow \min\{\,\alpha_i = (\,b_i - A_i x\,)/A_i d\ :\ A_i d > 0\ ,\ i \notin B\,\}$;
      **if**( $\bar{\alpha} > 0$ ) **then break**;
      $k \leftarrow \min\{\,i \notin B\ :\ A_i d > 0\ :\ \alpha_i = 0\,\}$; $B \leftarrow B \cup \{\,k\,\}$;
    $\alpha \leftarrow \mathrm{LS}(\ f\,,\,x\,,\,d\,,\,\bar{\alpha}\,,\,\varepsilon\ )$; $x \leftarrow x + \alpha d$;

- pesky part: handling of linear independence
- Maximal $B$ easy to get via a greedy algorithm [15]

**Exercise:** streamline the algorithm's computations when $B$ is a base

- $d \neq 0$ is a descent direction, i.e., $\langle \nabla f(x), d \rangle < 0$ (**check**)

**Exercise:** is $d$ a feasible direction? If not, what happens? Discuss

- $B' = B \setminus \{h\}$, $\exists \xi$ s.t. $A_{B'}\xi = 0 \ \wedge \ A_h\xi < 0$ (**check**)

- $d = 0 \ \wedge \ \exists h \in B$ s.t. $\mu_h < 0 \implies \langle \nabla f(x), \xi \rangle < 0$ for any $\xi$ above (**check**)
  $\implies \exists x' \in \{x \in \mathbb{R}^n : A_{B'}x = b_{B'}, A_h x \leq b_h\}$ s.t. $f(x') < f(x)$
  $\equiv$ removing $h$ from $B$ the objective can (perhaps) strictly decrease

- Inner loop handles degenerate steps: $\bar{\alpha} = 0 \implies x$ does not change ($B$ does)

- Inner loop explores $\neq B \subset \mathcal{A}(x)$ s.t. $A_B$ full rank (if $A_{\mathcal{A}(x)}$ is not),
  finitely terminates by Bland's anti-cycle rule (min entering / leaving $i$) [6, §4.3]

- $B$ "changes little" at every iteration: update factorization of $A_B A_B^T$ ...

**Exercise:** streamline the algorithm for the case when $f(\cdot)$ is linear

## Outline

- Nonlinear problem with linear constraints: $(P)\ \ \min\{ f(x) : Ax \leq b \}$
- Projecting $\equiv\ \min\{ \| x - \bar{x} \|^2 : Ax \leq b \}$ may be too costly, but minimizing a linear function may be possible (special structure, e.g., network constraints)
- Solve nonlinear $(P)$ by solving a sequence Master Problems $\equiv$ LPs

**procedure** $x = FW(f, A, b, x, \varepsilon)$ // invariant: $Ax \leq b$
  **for**( ; ; )
    $(MP_x)$ $\bar{x} \leftarrow \text{argmin}\{ \langle \nabla f(x), z \rangle : Az \leq b \}$; $d \leftarrow \bar{x} - x$;
    **if**( $\langle \nabla f(x), d \rangle \geq -\varepsilon$ ) **then return**;
    $\alpha \leftarrow \text{LS}(\ f, x, d, 1, \varepsilon\ )$; $x \leftarrow x + \alpha d$;

- $\langle \nabla f(x), d \rangle = 0 \implies x$ local optimum

**Exercise:** prove it by exhibiting $\lambda^* \geq 0$ that satisfies (KKT) with $x$

- Otherwise $\langle \nabla f(x), d \rangle < 0 \ \equiv\ d$ a descent direction (**check**)
- $f$ convex $\implies v^* = f(x) + \langle \nabla f(x), d \rangle \leq \nu(P)$ (**check**) $\implies$
  $f(x) - \nu(P) \leq f(x) - v^* =$ readily available estimate of gap $\implies$
  $\langle \nabla f(x), d \rangle = 0 \implies x$ global optimum

- Initial $x$ not really needed: one LP (say, $\langle 0, z \rangle$ objective) to find it, if any
- Convergence easy enough, $\approx$ run-of-the-mill descent algorithm [2, Th. 3.8]
- One LP per iteration costly, exploit structure to make it efficient
- Convergence rather slow: trusting $L_x(\cdot)$ very far from $x$ where $\nabla f$ is computed
- Solution seen already: stabilize master problem
- But separable penalty $\gamma \| z - x \|_2$ in the objective $\rightsquigarrow$ QP $\approx$ projection
- Trust region stabilization: constraint $\| z - x \|_\infty \leq \tau \equiv$
  box constraints $x_i - \tau \leq z_i \leq x_i + \tau$, almost never make an LP harder
- Have to manage $\tau$ somehow (e.g., fixed), but often worth it
- Other ways to improve convergence speed, e.g., away step and/or
  use FW to identify optimal Active Set then exploit it [10]

## (Stabilised) Frank-Wolfe for $f \notin C^1 \equiv$ (Bundle) Cutting Plane

- What if $f \notin C^1$? Cannot use any $g \in \partial f(x)$, "crap" first-order information
- $f$ convex $\implies$ first-order information not so crap: globally valid
- What if I collect it along the way and use it all?
- $\{x^i\} \implies \mathscr{B} = \{(x^i, f^i = f(x^i), g^i \in \partial f(x^i))\}$ bundle
- $f_{\mathscr{B}}(x) = \max\{f^i + \langle g^i, x - x^i \rangle : (x^i, f^i, g^i) \in \mathscr{B}\}$
  cutting-plane model of $f$ – $(1+\varepsilon)$-order model
- $\bar{x} \leftarrow \operatorname{argmin}\{f_{\mathscr{B}}(x) : Ax \leq b\}$: constrained cutting-plane algorithm
  completely $\equiv$ unconstrained version, even better if $Ax \leq b$ compact
- $x \leftarrow \operatorname{argmin}\{f_{\mathscr{B}}(z) + \gamma \| z - \bar{x} \|^2 : Az \leq b\}$ constrained Bundle method
- More difficult to exploit structure, but not impossible [12]
- Many complicated details (dual ...), not for the faint of heart, not for today

sounds strangely familiar ...

- Want a better direction? Use a better model!
- Second-order model $\equiv$ constrained Newton's method

  $d_* \leftarrow \text{argmin} \left\{ \frac{1}{2} d^T \nabla^2 f(x) d + \langle \nabla f(x), d \rangle : A(x+d) \leq b \right\}$;

  $\alpha \leftarrow \text{LS}(f, x, d_*, 1, \varepsilon)$; $x \leftarrow x + \alpha d_*$;
- $\nabla^2 f(x) \succeq 0$ otherwise $\mathcal{NP}$-hard: Hessian modification or Trust Region constraint $\| d \| \leq \tau$ (may still be $\mathcal{NP}$-hard)
- Use $0 \preceq H \approx \nabla^2 f(x)$ with quasi-Newton formulæ
- Fast convergence if done properly [8, Chap. 18]
- Solve a constrained QP at each iteration, possibly with quadratic constraints
- Many complicated details, not for the faint of heart, not for today

## Outline

## Dual approach

- QP with linear constraints: $(P)\ \ \min\{\ \frac{1}{2}x^T Qx + qx\ :\ Ax \leq b\ \}$
- So far, kept $Ax \leq b$ and gotten $\lambda \geq 0$ in the end (primal approach)
- Can we do the reverse (dual approach)? Of course we can
- $\forall$ fixed $\lambda \geq 0$: $\psi(\lambda) = \min\left\{\ \frac{1}{2}x^T Qx + qx + \lambda(b - Ax)\ \right\} \leq \nu(P)$
  concave dual function, $x(\lambda)$ optimal solution $\implies b - Ax(\lambda) \in \partial\psi(\lambda)$
- Lagrangian dual $\ (D)\ \ \max\{\ \psi(\lambda)\ :\ \lambda \geq 0\ \}$ convex possibly nonsmooth
- Strong assumption $Q \succ 0 \implies$ unique $x(\lambda) = Q^{-1}(\lambda A - q)$
  $\implies \psi \in C^1$ (but, in general, $\psi \notin C^2$) and $\nabla\psi(\lambda) = b - Ax(\lambda)$
- $(D) \equiv (P)$: $\nu(D) = \nu(P)$ and $\{\ x(\lambda^i)\ \} \to x_*$ as $\{\ \lambda^i\ \} \to \lambda_*$

**Exercise:** discuss relaxing the strong assumption $Q \succ 0$

- Solve $(D)$ by any method for $C^1$-but-not $C^2$ functions (if you are lucky)
- $(D)$ constrained but constraints are very easy (projection trivial)
- Feasible primal solution $x_*$ only asymptotically, but
  valid lower bound $v^i = \psi(\lambda^i) \leq \nu(P)$ at every iteration
- If $X$ "simple", Lagrangian heuristic: $x^i = p_X(x(\lambda^i)) \implies$
  valid upper bound $f^i = f(x^i) \geq \nu(P)$ at every iteration $\implies$
  $f(x^i) - \nu(P) \leq f^i - v^i =$ readily available estimate of gap
- Can behave very differently from primal methods ($L/\tau$ much less of an issue)
- Extends to $f(x)$ (strictly) convex but must solve general nonlinear problem
- $f(x)$ not convex serious issue, $\psi$ has to be computed exactly
- Yet, methods $\exists$ that work with inexact computation of $\psi(\cdot)$ [1]

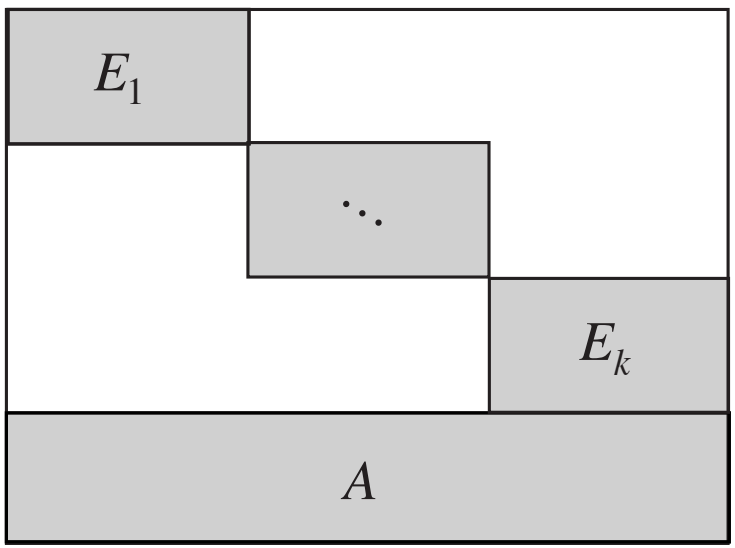
- Partial Lagrangian relaxation: $(P)\quad \min\{\, f(x) : Ax \le b\,,\ Ex \le d \,\}$

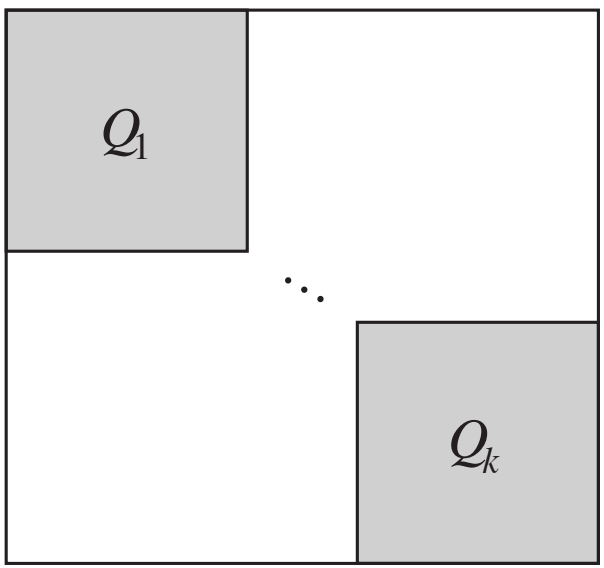
  $(R_\lambda)\quad \psi(\lambda) = \min\{\, f(x) + \lambda(b - Ax) : Ex \le d \,\}$

  complicating constraints $Ax \le b$ relaxed, easy constraints $Ex \le d$ kept

- $(R_\lambda)$ constrained but can exploit structure (and $\psi(\lambda) = -\infty$ less likely)

- Typical structure: $Ax \le b$ linking constraints $+$ $f$ separable



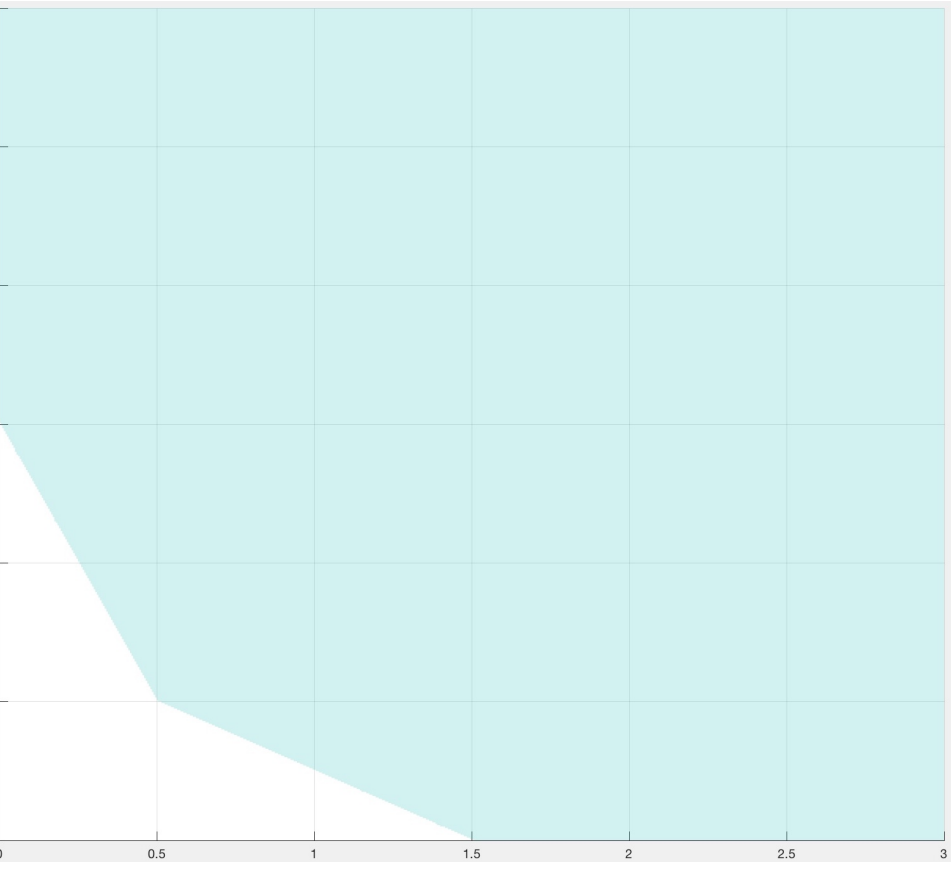

$\Longrightarrow$ $\psi(\lambda) = \sum_k \psi^k(\lambda)$ separable, algorithms can exploit it (parallelize)

- Many complicated details, not for the faint of heart [13], not for today

## Outline

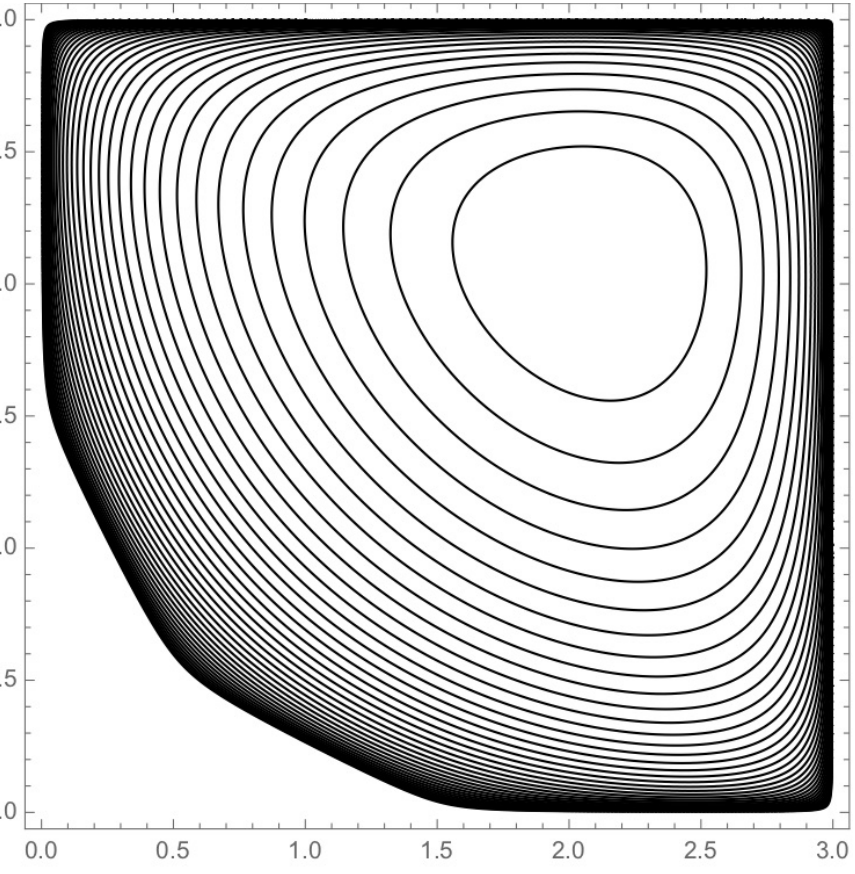
- Pros of dual methods: $(D) \approx$ unconstrained (would be with $Ax = b$)
- Cons of dual methods:
  - $\psi \notin C^2$, not even $\in C^1$ if $f$ not strictly convex
  - $x(\lambda)$ never feasible until the very end (unless Lagrangian heuristic)
- Would like: i) $(D)$ unconstrained; ii) $\psi \in C^2$; iii) $x(\lambda)$ feasible
- i) and iii) obvious: $f + \imath_X$ ... except $\imath_X \notin C^0$
- Would need something like $\imath_X$ but $\in C^2$
- Can get $C^2$ if you accept to solve almost $(P)$, but not quite

- $\gamma > 0$ parameter, $(P_\gamma)$ $\min \left\{ f_\gamma(x) = f(x) - \gamma \sum_{i=1}^{m} \ln(b_i - A_i x) \right\}$
  logarithmic barrier $f_\gamma \in C^2$ (if $f \in C^2$) and strictly convex (if $f$ convex)



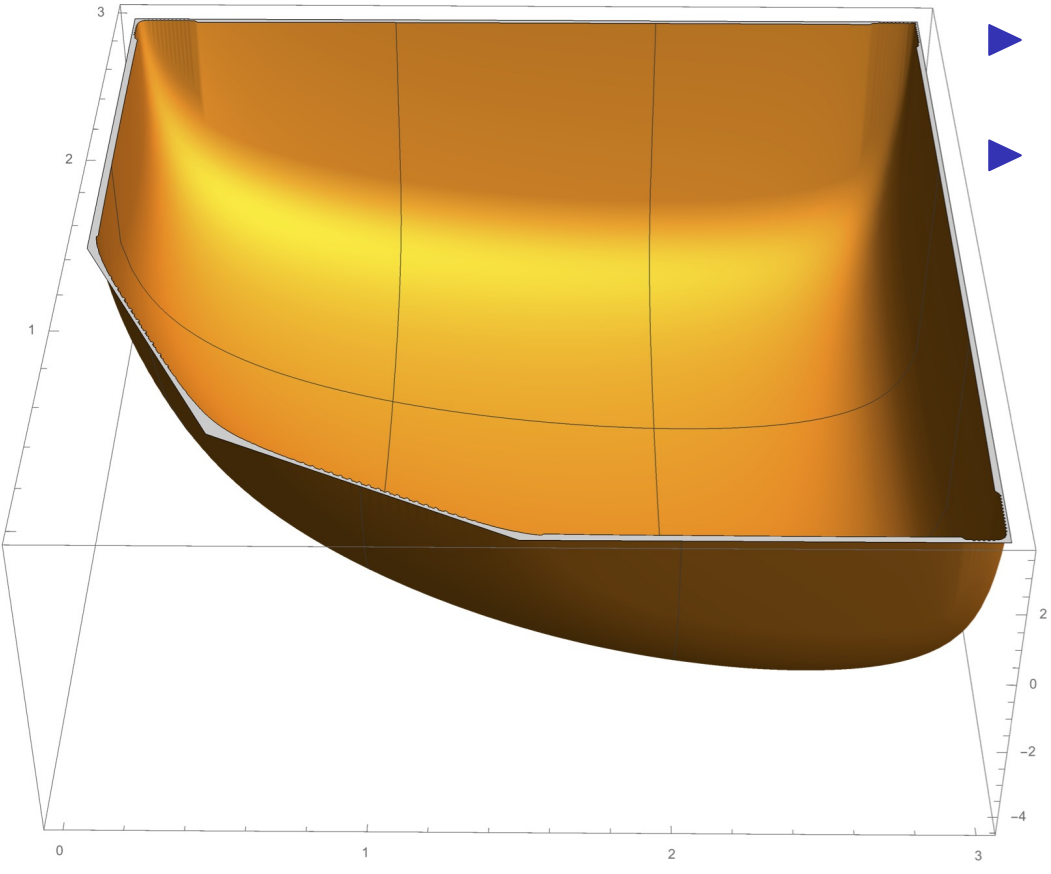

- $X = \{ x \in \mathbb{R}^n : Ax \leq b \}$

- $\gamma > 0$ parameter, $(P_\gamma)$ $\min \left\{ f_\gamma(x) = f(x) - \gamma \sum_{i=1}^{m} \ln(b_i - A_i x) \right\}$
  logarithmic barrier $f_\gamma \in C^2$ (if $f \in C^2$) and strictly convex (if $f$ convex)



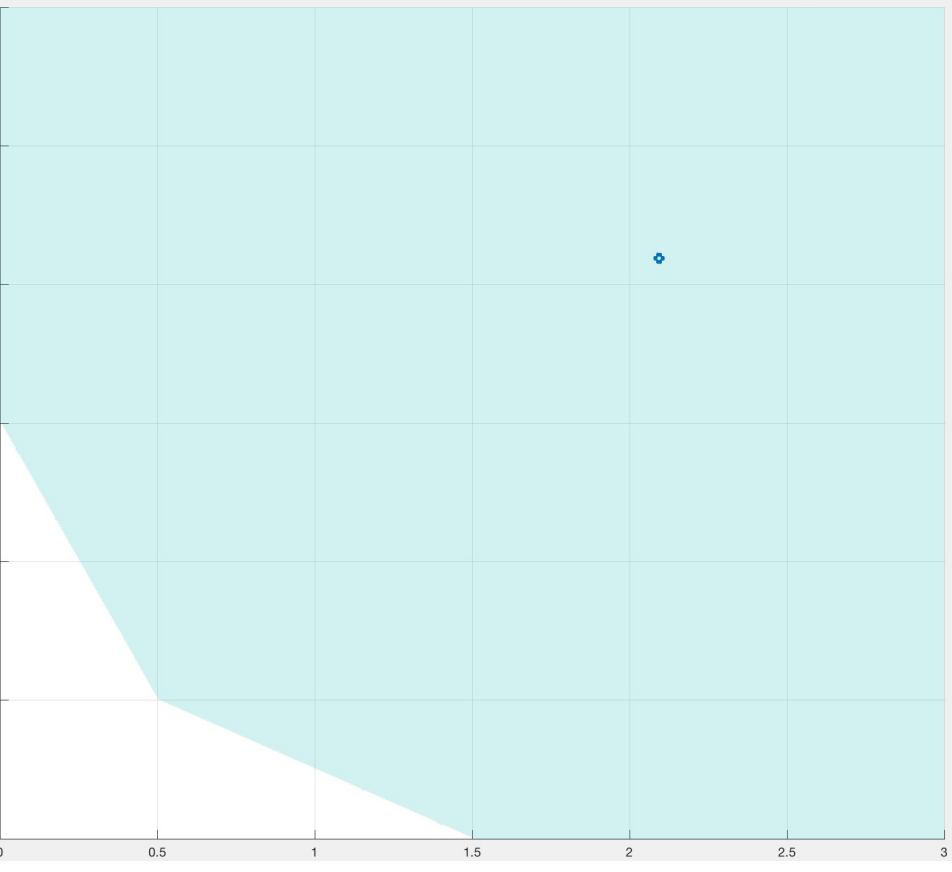

- $f_\gamma(x) = \infty$ for $x \notin X$ (like $\imath_X$)

- $\gamma > 0$ parameter, $(P_\gamma)$ $\min \left\{ f_\gamma(x) = f(x) - \gamma \sum_{i=1}^{m} \ln(b_i - A_i x) \right\}$
  logarithmic barrier $f_\gamma \in C^2$ (if $f \in C^2$) and strictly convex (if $f$ convex)



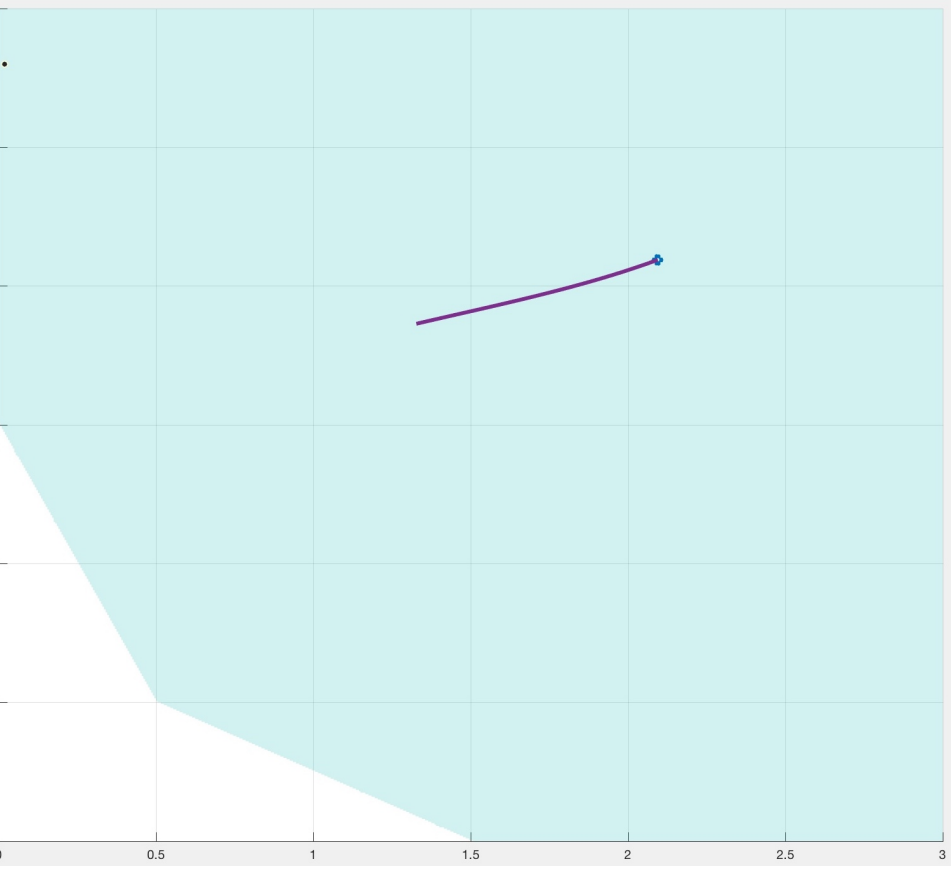

- $f_\gamma(x) = \infty$ for $x \notin X$ (like $\imath_X$)
- $f_\gamma(x) = \infty$ for $x \in \partial X$ (unlike $\imath_X$)

- $\gamma > 0$ parameter, $(P_\gamma)$ $\min \left\{ f_\gamma(x) = f(x) - \gamma \sum_{i=1}^{m} \ln(b_i - A_i x) \right\}$
  logarithmic barrier $f_\gamma \in C^2$ (if $f \in C^2$) and strictly convex (if $f$ convex)



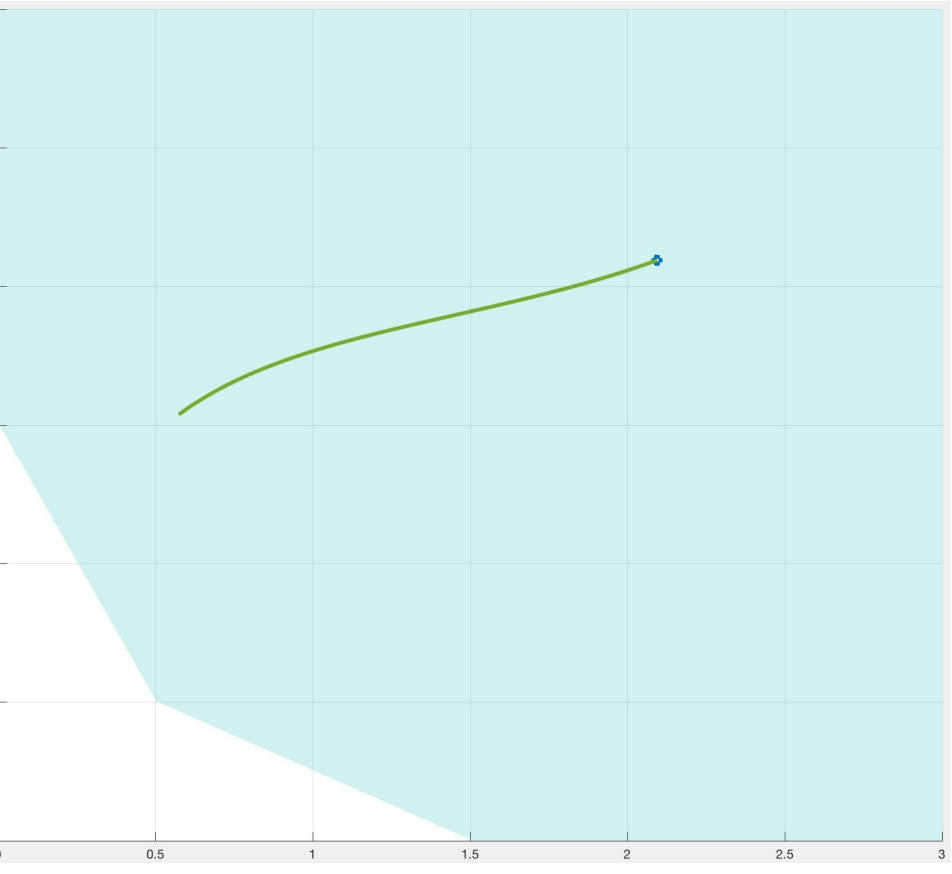

- $f_\gamma(x) = \infty$ for $x \notin X$ (like $\imath_X$)
- $f_\gamma(x) = \infty$ for $x \in \partial X$ (unlike $\imath_X$)
- $\forall\, \gamma > 0\ \exists!\, x_\gamma$ optimal of $(P_\gamma)$
- $x_\infty = \lim_{\gamma \to \infty} x_\gamma$ analytic center of $X$
  (maximize product of slacks)

- $\gamma > 0$ parameter, $(P_\gamma)$ $\min \left\{ f_\gamma(x) = f(x) - \gamma \sum_{i=1}^{m} \ln(b_i - A_i x) \right\}$
  logarithmic barrier $f_\gamma \in C^2$ (if $f \in C^2$) and strictly convex (if $f$ convex)



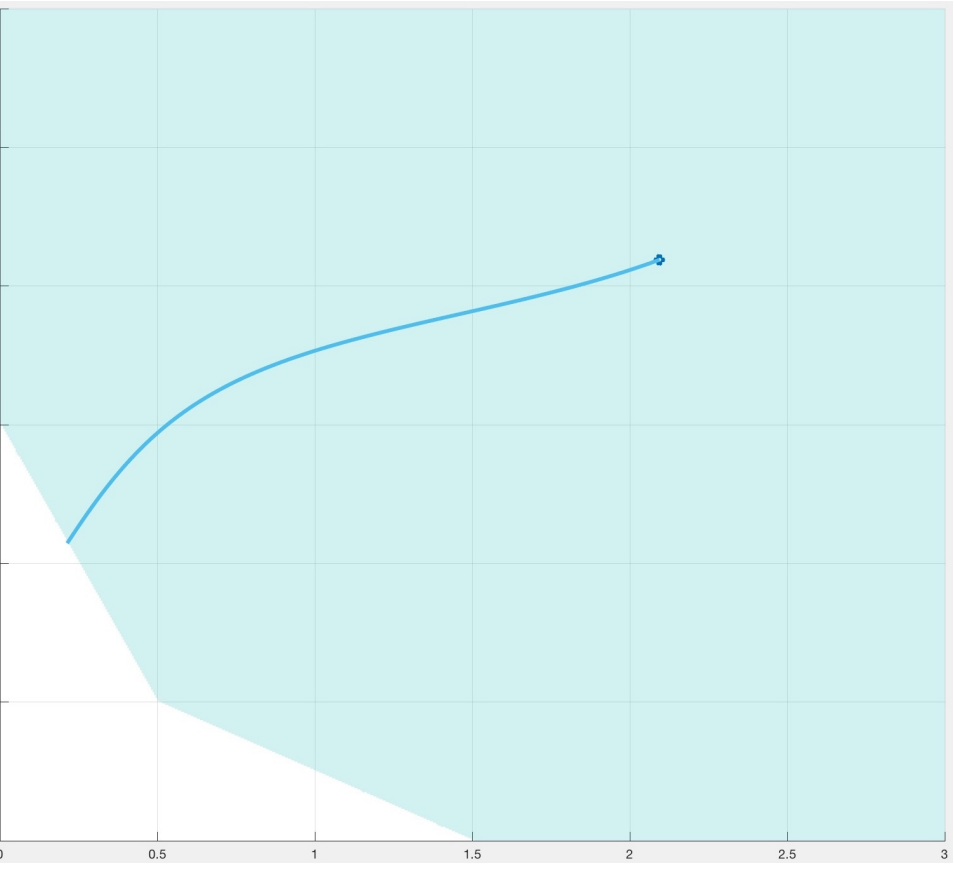

- $f_\gamma(x) = \infty$ for $x \notin X$ (like $\imath_X$)
- $f_\gamma(x) = \infty$ for $x \in \partial X$ (unlike $\imath_X$)
- $\forall\, \gamma > 0\ \exists!\, x_\gamma$ optimal of $(P_\gamma)$
- $x_\infty = \lim_{\gamma \to \infty} x_\gamma$ analytic center of $X$ (maximize product of slacks)
- As $\gamma \to 0$,

- $\gamma > 0$ parameter, $(P_\gamma)$ $\min \left\{ f_\gamma(x) = f(x) - \gamma \sum_{i=1}^{m} \ln(b_i - A_i x) \right\}$
  logarithmic barrier $f_\gamma \in C^2$ (if $f \in C^2$) and strictly convex (if $f$ convex)




- $f_\gamma(x) = \infty$ for $x \notin X$ (like $\imath_X$)
- $f_\gamma(x) = \infty$ for $x \in \partial X$ (unlike $\imath_X$)
- $\forall\, \gamma > 0\ \exists!\, x_\gamma$ optimal of $(P_\gamma)$
- $x_\infty = \lim_{\gamma \to \infty} x_\gamma$ analytic center of $X$
  (maximize product of slacks)
- As $\gamma \to 0$, $x_\gamma \to$

- $\gamma > 0$ parameter, $(P_\gamma)$ $\min \{ f_\gamma(x) = f(x) - \gamma \sum_{i=1}^{m} \ln(b_i - A_i x) \}$
  logarithmic barrier $f_\gamma \in C^2$ (if $f \in C^2$) and strictly convex (if $f$ convex)




- $f_\gamma(x) = \infty$ for $x \notin X$ (like $\imath_X$)
- $f_\gamma(x) = \infty$ for $x \in \partial X$ (unlike $\imath_X$)
- $\forall\, \gamma > 0\ \exists!\, x_\gamma$ optimal of $(P_\gamma)$
- $x_\infty = \lim_{\gamma \to \infty} x_\gamma$ analytic center of $X$
  (maximize product of slacks)
- As $\gamma \to 0$, $x_\gamma \to x_* =$
  analytic center of optimal face
- $\mathcal{C} = \{ x_\gamma : \gamma \in (0, \infty) \}$
  central path (smooth curve)

- Idea: start ($\approx$) at center $x_\infty$, ($\approx$) follow $\mathcal{C}$ to reach (very close to) $x_*$
- Always strictly fasible, never touch $\partial X$

- $-\log(\cdot)$ self-concordant [4, §9.6] $\implies$ $f_\gamma$ is for many $f$ (linear, quadratic, ...)
- Newton's method converges very quickly to $x_\gamma$ if started witin appropriate neighbourhood $\mathcal{N}$ of $\mathcal{C}$
- $x^i$ "close" to $x(\gamma^i)$: a few Newton's steps [4, §11.5.2] give $x^{i+1}$ "much closer" to $x(\gamma^i)$ $\implies$ "close" to $x(\gamma^{i+1})$ with $\gamma^{i+1} = \tau\gamma^i$
  $\implies$ linear convergence (and $\tau$ is "good" $\ll 1$)
- Overall $O(m\log(1/\varepsilon))$ iterations, can be made $O(\sqrt{m}\log(1/\varepsilon))$ and more like $O(\log(m)\log(1/\varepsilon))$ in practice [4, §11.5.3]
- Dimension independent on $n$, not on $m$ (but $\approx$ in practice),
- Each Newton's steps at least $O(n^3)$ $\equiv$ costly
- Best implementations for LP, QP, SOCP and SDP in fact primal-dual

- Focus on quadratic case: $(P)\ \ \min\{\ \frac{1}{2}x^T Qx + qx\ :\ Ax \leq b\ \}$
- Could compute Newton's step as usual

**Exercise:** compute $\nabla f_\mu(\,x\,)$, $\nabla^2 f_\mu(\,x\,)$, Newton's step

- Cleaner derivation out of KKT of $(P)$ "written with slacks"

$$Ax + s = b \quad , \quad s \geq 0 \qquad \text{(KKT-F)}$$
$$Qx + \lambda A = -q \quad , \quad \lambda \geq 0 \qquad \text{(KKT-G)}$$
$$\lambda_i s_i = 0 \qquad i = 1, ..., m \qquad \text{(KKT-CS)}$$

- One is solving the dual at the same time as the primal:

$$(D)\ \ \max\left\{\ -\lambda b - \tfrac{1}{2}x^T Qx\ :\ Qx + \lambda A = -q\ ,\ \lambda \geq 0\ \right\}$$

- "Slackened KKT" characterize $x(\,\gamma\,)$ and complementarity gap:

$$\lambda_i s_i = \gamma \qquad i = 1, ..., m \qquad \text{(KKT-CS-}\mu\text{)}$$

$$\Longrightarrow\ \textstyle\sum_{i=1}^m \lambda_i s_i = \gamma m = (\ \frac{1}{2}x^T Qx + qx\ ) - (\ -\lambda b - \frac{1}{2}x^T Qx\ ) \quad \textbf{(check)}$$

- Useful notation: $\Lambda$ , $S$ diagonal matrices with $\lambda_i$ , $s_i$ on the diagonal

  $\Lambda S u = \gamma u$ (KKT-CS-$\mu$)

- $x \to x + \Delta x$ , $s \to s + \Delta s$ , $\lambda \to \lambda + \Delta\lambda$ (current iterate + displacement)

  $\rightsquigarrow\ (\Lambda + \Delta\Lambda)(S + \Delta S)u = (S\Delta\Lambda + \Lambda\Delta S + \Lambda S + \Delta\Lambda\Delta S)u = \gamma u$

- Nonlinear system of equations (ignoring sign constraints),
  (KKT-CS-$\mu$) only nonlinear (bilinear) term

- Linearize $\equiv$ Newton's method $\equiv$ just ignore it

$$\begin{bmatrix} Q & A^T & 0 \\ A & 0 & I \\ 0 & S & \Lambda \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta\lambda \\ \Delta s \end{bmatrix} = \begin{bmatrix} -(Qx+q) - \lambda A \\ b - Ax - s \\ \gamma u - \Lambda S u - \Delta\Lambda\Delta S u \end{bmatrix} \approx \begin{bmatrix} r^P \\ r^D \\ \gamma u - \Lambda S u \end{bmatrix} \quad (*)$$

  $r^P = b - Ax - s = 0$ / $r^D = -(Qx+q) - \lambda A = 0$ if primal / dual feasible
  (if not unfeasibility $\searrow$ linearly $\implies$ feasible quickly)

- Solving $(*)$ by far the most costly step: exploit structure

- Exploit large-scale structure: substitute from last constraint (**check**) $\Longrightarrow$ modified Normal Equations (note: $\Lambda^{-1}S \succ 0$ diagonal)

$$\begin{bmatrix} Q & A^T \\ A & -\Lambda^{-1}S \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta\lambda \end{bmatrix} = \begin{bmatrix} r^P \\ r^D + s - \gamma\Lambda^{-1}u \end{bmatrix}$$

  structure-exploiting Krylov-like methods ...

- Exploit large-scale structure: substitute from second constraint (**check**) $\Longrightarrow$

  $[\,Q + A^T\Lambda S^{-1}A\,]\Delta x = r^P + A^T(\,\lambda - S^{-1}(\,\Lambda r^D - \gamma u\,)\,)$ Reduced KKT

  $M = Q + A^T\Lambda S^{-1}A \succ 0$ if $A$ has full column rank (it should)

  Cholesky factorization of $M$ (can be dense, permute rows of $A$ ...)

- Predictor-corrector variant: solve, add fixed term $\Delta\Lambda\Delta S u$ in r.h.s. of $(*)$, solve again re-using factorization, possibly iterate [8, Alg. 16.4][14, p. 15]

- Exploit all structure: $Dx = d$, box constraints $0 \le x \le u$, blocks, ...

**Exercise:** develop formulæ for $(P)$ $\min\{\,\frac{1}{2}x^TQx + qx \;:\; Ax = b\,,\; 0 \le x \le u\,\}$

**procedure** $x = IPMQP\,(\,Q\,,\,q\,,\,A\,,\,b\,,\,\varepsilon^P\,,\,\varepsilon^D\,,\,\varepsilon\,,\,\rho\,)$
  choose $(\,x\,,\,\lambda > 0\,,\,s > 0\,)$; $\gamma \leftarrow \langle\,\lambda\,,\,s\,\rangle$;
  **while**( $\|\,r^P\,\| > \varepsilon^P \;\vee\; \|\,r^P\,\| > \varepsilon^D \;\vee\; \gamma m > \varepsilon$ )
    solve $(*)$; $\alpha \leftarrow 0.995 \max\{\,\beta \,:\, \lambda + \beta\Delta\lambda \geq 0\,,\; s + \beta\Delta s \geq 0\,\}$;
    $x \leftarrow x + \alpha\Delta x$; $s \leftarrow s + \alpha\Delta s$; $\lambda \leftarrow \lambda + \alpha\Delta\lambda$; $\gamma \leftarrow \rho\gamma$;

- Until $\|\,r^P\,\| > \varepsilon^P \;\vee\; \|\,r^P\,\| > \varepsilon^D$ may choose $\alpha^P \neq \alpha^D$ [8, p. 483]
- New iterate primal and dual feasible if old was, otherwise less unfeasible
- Primal-dual algorithm: upper and lower bound on $\nu(\,P\,)$, converge as $\gamma \searrow 0$
- $\gamma = \rho(\lambda s)\,/\,m$ for $\rho < 1$ fixed (reasonable value $\rho = 1\,/\,m$),
  more sophisticated formulæ using $\alpha^P$ and/or $\alpha^D$ and for predictor-corrector
- Very good convergence in practice, but large time/memory cost per iteration
- "Straightforward" to extend to SOCP, SDP [4, §11.6]
- May have numerical problems (dividing by very small numbers)
  especially on empty / unbounded problems

## Outline

## Wrap up

- Constraints make things a lot more complex $\Longrightarrow$ intersting
- Many different cases, "structure constraints" $\times$ "structure objective"
  $\Longrightarrow$ very many different ways to exploit them
- The linear algebra is often crucial, makes or breaks a method
- We barely scratched the surface, there is lots more:
  - other barrier / penalty methods
  - other primal methods
  - algorithms for highly nonlinear constraints
  - and more, and more, ...
- Not to mention getting global optima in the nonconvex case
- learning usually does not need all this, but other applications do
- learning requires large size and speed: something's gotta give
- Still plenty of ways to do nice things

## (Wrap up)$^2$ – technical

- Models are important for algorithms, too (besides vice-versa)
- Models must be simple, but first- and second-order ones are!
- Want a better direction? Use a better model!
  If the world does not give you one, invent one yourself!
- Thank goodness you can go (much) faster than gradient,
  but there is only so much you can do with first-order methods
- Always keep it convex if possible, better if $C^1$, better still if $C^2$
- Duality an extremely useful tool, especially (but not only) in convex case
- Mind trade-offs: "fat" models $\rightsquigarrow$ fast convergence but high iteration cost
- If you don't know it estimate it, but be ready to revise your estimate
- Best choices in theory not best in practice (worst-case $\neq$ average case)
- A lot of details need be considered, numerical aspects crucial

- Dabble with math-based algorithms? Have to know (some) maths
- Learn simple things first: must know a Line Search to optimize in $\mathbb{R}^n$
- Algorithms can only get so far with nasty problems

  hence choose your problems (foes) wisely; learning most often does
- Always exploit all the structure of your problem
- There is no one-size-fits-all solution
- Linear algebra is crucial for doing optimization, vice-versa also quite true
- Your mileage may vary, so try, try, try!

- Dabble with math-based algorithms? Have to know (some) maths
- Learn simple things first: must know a Line Search to optimize in $\mathbb{R}^n$
- Algorithms can only get so far with nasty problems

  hence choose your problems (foes) wisely; learning most often does
- Always exploit all the structure of your problem
- There is no one-size-fits-all solution
- Linear algebra is crucial for doing optimization, vice-versa also quite true
- Your mileage may vary, so try, try, try!

Lots of Fun!

[1] W. van Ackooij, A. Frangioni "Incremental Bundle Methods Using Upper Models" *SIAM Journal on Optimization* 28, 379–410, 2018
http://pages.di.unipi.it/frangio/abstracts.html#SIOPT16

[2] S. Bubeck *Convex Optimization: Algorithms and Complexity*, arXiv:1405.4980v2, https://arxiv.org/abs/1405.4980, 2015

[3] M.S. Bazaraa, H.D. Sherali, C.M. Shetty *Nonlinear Programming: Theory and Algorithms*, John Wiley & Sons, 2006.

[4] S. Boyd, L. Vandenberghe *Convex Optimization*, https://web.stanford.edu/~boyd/cvxbook Cambridge University Press, 2008

[5] A. Frangioni, E. Gorgone "A Library for Continuous Convex Separable Quadratic Knapsack Problems" *EJOR* 229, 37–40, 2013
http://pages.di.unipi.it/frangio/abstracts.html#EJOR13

[6] J. Lee *A First Course in Linear Optimization v4.06*, 2022
https://github.com/jon77lee/JLee_LinearOptimizationBook/blob/master/JLee.4.06.zip

[7] D.G. Luenberger, Y. Ye *Linear and Nonlinear Programming*, Springer International Series in Operations Research & Management Science, 2008

[8] J. Nocedal, S.J. Wright, *Numerical Optimization – second edition*, Springer Series in Operations Research and Financial Engineering, 2006

[9] Y. Sun, *Notes on First-Order Methods for Minimizing Smooth Functions* https://web.stanford.edu/class/msande318/notes/notes-first-order-smooth.pdf, 2015

[10] I.M. Bomze, F. Rinaldi, D. Zeffiro "Active set complexity of the Away-step Frank-Wolfe Algorithm" arXiv:1912.11492v1 https://arxiv.org/abs/1912.11492, 2019

[11] E. de Klerk "The complexity of optimizing over a simplex, hypercube or sphere: a short survey" *Central European Journal of Operations Research* 16, 111–125, 2008 https://link.springer.com/content/pdf/10.1007/s10100-007-0052-9.pdf

[12] K.C. Kiwiel "An alternating linearization bundle method for convex optimization and nonlinear multicommodity flow problems" *Mathematical Programming* 130, 59–84, 2011

[13] A. Frangioni "The Long Road to Practical Decomposition Methods" http://www.di.unipi.it/~frangio/schools/Napoli-2021-I.pdf http://www.di.unipi.it/~frangio/schools/Napoli-2021-II.pdf *AIRO PhD School*, 2021

[14] J. Gondzio *Interior Point Methods 25 Years Later*, 2011 https://www.maths.ed.ac.uk/~gondzio/reports/ipmXXV.pdf

[15] Wikipedia – Greedy Algorithm https://en.wikipedia.org/wiki/Greedy_algorithm

[16] Wikipedia – Idempotent Matrix https://en.wikipedia.org/wiki/Idempotent_matrix

## Outline

- The clever idea is to construct a problem of the same class that is surely non-empty and whose optimal solution provides a feasible one of $(P)$ or prove no-one exists, such as $(F)$ $\min\{\, \frac{1}{2}\| v \|_2^2 \,:\, Ax \leq b + v \,,\; v \geq 0 \,\}$. $(F)$ is surely nonempty: take any $x$ (e.g., $x = 0$), then $v = \max\{\, Ax - b \,,\, 0 \,\}$ (e.g, $v = \max\{\, -b \,,\, 0 \,\}$) is such that $[\, x \,,\, v \,]$ is feasible to $(F)$. Also, $(F)$ cannot be unbounded below (see next exercise) as the objective is bounded below by 0. Thus, one can use the Active-Set approach starting from $[\, x \,,\, v \,]$ to get an optimal solution $[\, x^* \,,\, v^* \,]$ to $(F)$. Now, if $v^* = 0$ then $x^*$ is a feasible solution to $(P)$ that we can re-start the Active-Set approach from. If, instead, $v^* \neq 0$, this proves that $(P)$ has no feasible solution. Indeed, if any $x$ feasible for $(P)$ existed then it would correspond to a $[\, x \,,\, 0 \,]$ feasible for $(F)$ that would have a better objective than $v^*$, which is impossible since $v^*$ is provably a global optimum ($(F)$ is convex, in fact even if $(P)$ is not). This so-called "phase 0" of the approach can be conveniently integrated in the Active-Set method so that the restarting on $(P)$ once a feasible $x$ if found (if ever) can occur "naturally" exploiting all the currently available information; see, e.g., [8, p. 473] for details [**back**]

- For $x(\alpha) = x + \alpha d$, we want to find the maximum value of $\alpha$ s.t. $Ax(\alpha) \leq b$, which we know is $\geq 0$ since $x = x(0)$ is feasible. We look at every constraint individually, and write $A_i(x + \alpha d) \leq b_i \equiv \alpha(A_i d) \leq b_i - A_i x$. By feasibility, $b_i - A_i x \geq 0$: hence, if $A_i d \leq 0$ the relationship is true for all $\alpha \geq 0$. If, instead, $A_i d > 0$, then the relationship is only true if $\alpha \leq \alpha_i = (b_i - A_i x)/A_i d$. Since $x(\alpha)$ must satisfy all the constraints, $\alpha \leq \alpha_i$ must all for all $i$ s.t. $A_i d > 0$, and therefore $\alpha$ has to be the min of all these $\alpha_i$. It is easy to see that the min has to be $< \infty$, in fact $< 1$: this is because $x(1) = x + d = \bar{x}$ is unfeasible by construction when the control reaches the $\bar{\alpha}$ computation step [**back**]

- The first issue is that if $Q \not\succ 0$, even if possibly $Q \succeq 0$, then $(P_B)$ may not have a finite optimal solution because it may be unbounded below. This happens if $\exists\, d \in \ker(Q)$ (a linear combination of eigenvectors corresponding to 0 eigenvalues) s.t. $\langle q, d \rangle \neq 0$ and $A_B d = 0$, as implies $A_B(x + \alpha d) = = A_B x + \alpha A_B d = b_B \;\forall\, \alpha$, and $\varphi(\alpha) = f(x + \alpha d) = f(x) + \alpha\langle q, d \rangle$, which means that one can find feasible solutions to $(P_B)$ with arbitrarily large negative value. However, any such $d$—that one must be able to properly identify in order to prove unboundedness of $(P_B)$—can then be used instead of $\bar{x} - x$ as the direction of descent along which $\bar{\alpha}$ is found.

This reveals the second issue: in such a case, $(P)$ itself may be unbounded below. In fact, $\varphi(\alpha)$ is decreasing with $\alpha$ and unbounded below. If $\bar{\alpha} < \infty$, then the algorithm can proceed as usual. But $\bar{\alpha} = \infty$ may happen if $Ad \leq 0$, which implies that $x(\alpha) = x + \alpha d$ is feasible for all $\alpha \geq 0$. This means that the algorithm can be stopped as $(P)$ has been "solved", but specific checks (and a specific return code) are required [**back**]

- Given $\bar{\mu}_B \geq 0$, it is easy to see that $\lambda = [\,\bar{\mu}_B\,,\,0\,] \geq 0$ satisfies the KKT for $(P)$. In fact, $\nabla f(\bar{x}) + \bar{\mu}_B A_B = 0$ since $\bar{x}$ is optimal for $(P_B)$ and $\bar{\mu}_B$ are the corresponding optimal (unconstrained) Lagrangian multipliers, but this is equivalent to $\nabla f(\bar{x}) + \lambda A = 0$, i.e., (KKT-G). (KKT-F) is just $A\bar{x} \leq b$, that is satisfied since it is checked for $i \notin B$ in the algorithm and it surely satisfied at equality for $i \in B$ since $\bar{x}$ is feasible for $(P_B)$. Finally, (KKT-CS) holds: for $i \in B$ one has $A_i\bar{x} = b_i \implies \lambda_i(b_i - A_i\bar{x}) = \bar{\mu}_i(b_i - A_i\bar{x}) = 0$, while for $i \notin B$ one has $\lambda_i = 0$ and therefore $\lambda_i(b_i - A_i\bar{x}) = 0$ [**back**]

- Let $x_F^*$ be the solution of $(P_B)$, and $x^* = [\,0\,,\, x_F^*\,,\, \overline{x}_U\,]$ the corresponding complete solution. We only need $\bar{\mu}$, the Lagrangian multipliers of the active constraints, when $\underline{x}_F \leq x_F^* \leq \overline{x}_F$, i.e., $x^*$ is feasible. $\mu$ are the multipliers of the constraint $x_i \geq 0$ for $i \in L$ and those of the constraints $x_i \leq \overline{x}_i$ for $i \in U$. The values of each of these multipliers is immediately derived by the corresponding entry $g_i = [\,\nabla f(\,x^*\,)\,]_i$ of the gradient in $x^*$ and (KKT-G). In fact, the entry of (KKT-G) corresponding to $i \in L$ reads $g_i - \bar{\mu}_i = 0 \;\equiv\; \bar{\mu}_i = g_i$ (recall that the constraint is $-x_i \leq 0$), while for $i \in U$ reads $g_i + \bar{\mu}_i = 0 \;\equiv\; \bar{\mu}_i = -g_i$. Thus, the condition "$\bar{\mu} \geq 0$" reads "$g_i \geq 0$ for $i \in L$ and $g_i \leq 0$ for $i \in U$" This can be made sense of in the following way. If $g_i < 0$ for $i \in L$, $x_i^* - \alpha g_i = \alpha(-g_i) > 0$ for $\alpha > 0$; that is, a (small) step along the anti-gradient keeps the iterate $i$ inside the feasible region and decreases the function value, which means that $x^*$ cannot be optimal. Symmetrically, If $g_i > 0$ for $i \in U$, $x_i^* - \alpha g_i = \overline{x}_i - \alpha g_i < \overline{x}_i$, i.e., again, a (small) step $\alpha > 0$ along the anti-gradient keeps the iterate $i$ inside the feasible region and decreases the function value, which again means that $x^*$ cannot be optimal. The optimality condition is that none of these things happen [**back**]

- As we have seen already, feasibility of a set of linear inequalities can be cast as a, say, QP with linear constraints that can be solved with, say, an Active-Set method. In fact, one can alternatively use a Linear Program, i.e., $(F)\ \ \min\{\, uv \,:\, Ax \leq b + v\ ,\ v \geq 0\,\}$ and then use its finite optimal solution $[\, x^*\, ,\, v^*\,]$ (which must exists) exactly in the same way as the one from the QP from a few exercises back. LPs are somewhat cheaper than QPs to solve [**back**]

- $\nabla h(\, d\,) = d + \nabla f(\, x\,)$, hence (KKT-G) for $p_{F_X(\, x\,)}(\, -\nabla f(\, x\,)\,)$ reads $d^* + \nabla f(\, x\,) + \nu^* A_{\mathcal{A}(\, x\,)} = 0$. This $\nu^* \geq 0$ is not a dual solution for $(P)$ since it is of the wrong size: $\nu^* \in \mathbb{R}^k$, where $k = |\, \mathcal{A}(\, x\,)\,|$ is the number of active constraints in $x$, which in general is $< m$ (number of original constraints in $(P)$ = number of rows in $A$). However, this is easy to solve with the trick we have seen already when discussing (KKT-CS): just set $\lambda_i^* = \nu_i^*$ for $i \in \mathcal{A}(\, x\,)$ and $\lambda_i^* = 0$ for $i \notin \mathcal{A}(\, x\,)$. This $\lambda^* \in \mathbb{R}^m$ is such that $\lambda^* A = \nu^* A_{\mathcal{A}(\, x\,)}$ (and, of course, $\lambda^* \geq 0$), hence it still satisfies $d^* + \nabla f(\, x\,) + \lambda^* A_{\mathcal{A}(\, x\,)} = 0$, providing (KKT-S) for $(P)$ if $d^* = 0$. It is also obvious that $\lambda^* Ax = 0$ (it has been constructed precisely so that this holds) and $x$ is always kept feasible for $(P)$,

hence (KKT-F) and (KKT-CS) are satisfied as well and $x$ is optimal for $(P)$ [**back**]

- $X$ is decomposable, i.e., $X = X_1 \times X_2 \times \ldots \times X_n$ with $X_i = [\,\underline{x}_i\,,\,\overline{x}_i\,]$. Hence, $F_X(\,x\,)$ is decomposable: $F_X(\,x\,) = \bigotimes_{i=1}^{n} F_{X_i}(\,x_i\,)$. Now, the individual feasible direction cones are trivial. If $\underline{x}_i < x_i < \overline{x}_i$, then $F_{X_i}(\,x_i\,) = \mathbb{R}$ and the $i$-th entry of the projected anti-gradient is equal to the original entry: $d_i$ is not changed. If, instead, $x_i = \underline{x}_i$, then $F_{X_i}(\,x_i\,) = \mathbb{R}_+$: it is only feasible to increase $x_i$ but not to decrease it. Thus, if $d_i \geq 0$ it need not be changed, otherwise the projection problem $\min\{\,(\,z - d_i\,)^2 \;:\; z \geq 0\,\}$ clearly has optimal solution $z = 0$, hence $d_i = 0$. The converse happens if $x_i = \overline{x}_i$: increasing $x_i$ is forbidden, hence $d_i$ must be set to 0 if the $i$-th original entry of the anti-gradient is $> 0$ [**back**]

- choose_step() is just any regular Line Search over $d$, with maximum stepsize $\bar{\alpha}$: by convexity, $x + \alpha d$ is feasible for all $\alpha \in [\,0\,,\,\bar{\alpha}\,]$ (but it is no longer so for $\alpha > \bar{\alpha}$). The only delicate aspect is the stopping condition of the LS as a function of the "global" accuracy parameter $\varepsilon$, but this has amply been discussed already: one can stop when $|\,\varphi'(\,\alpha\,)\,| \leq \varepsilon \|\,\nabla f(\,x\,)\,\|$ (although it is an interesting exercise, left to the reader, whether or not one could use, say, $\|\,d\,\| \leq \|\,\nabla f(\,x\,)\,\|$ instead), or the Armijo-Wolfe stopping conditions, as this clearly is an example of twisted gradient method and therefore in principle subject to the Zoutendijk's Theorem. The convergence arguments are somewhat more involved than this discussion seems to imply since the maximum stepsize $\bar{\alpha}$ may make it impossible to satisfy Wolfe's condition. In fact, one may not even be able to find an $\alpha$ s.t. $\varphi'(\,\alpha\,) > 0$, as it may happen that $\varphi'(\,\alpha\,) < 0$ for all $\alpha \in [\,0\,,\,\bar{\alpha}\,]$; which is not a big deal since it means that $\bar{\alpha}$ is the minimum of $\varphi$ in the interval and the LS stops immediately. One should therefore expect the algorithm to be convergence, although the details of the proof are not immediate [**back**]

- It is easy to see that the projection problem is simple. Each variable $z_i$ is constrained to be non-negative ($z_i \geq 0$) if $x_i = \underline{x}_i$, non-positive ($z_i \leq 0$) if $x_i = \overline{x}_i$, and is unconstrained in sign otherwise. Also, a feasible direction $z$ must satisfy $\sum_{i=1}^{n} z_i = 0$. The Lagrangian relaxation w.r.t. that constraint has objective $\sum_{i=1}^{n} [\, g_i(\, z_i \,) = \, z_i - d_i \,)^2 \,/\, 2 - \mu z_i \,]$, and therefore is separable over the $z_i$ variables. Since $g_i'(\, z_i \,) = z_i - d_i - \mu$, the unconstrained minimum is $z_i^* = d_i + \mu$, a linear function of $\mu$: hence, the constrained optimum $z_i^*(\, \mu \,)$ is obtained by taking the min or max of $z_i^*$ with 0, depending on the sign constraint (if any), i.e., it is either a linear or a piecewise-linear function of $\mu$ with exactly two segments and the breakpoint in $z_i^* = d_i + \mu = 0 \;\equiv\; \mu = -d_i$. The optimal solution $\mu^*$ of the Lagrangian dual is such that $\sum_{i=1}^{n} z_i^*(\, \mu^* \,) = 0$ (this is the condition "(sub)gradient $= 0$" for the dual function); note that $\mu^*$ must exist since the projection problem is neither empty nor unbounded below and therefore it has an optimal solution, hence so has its dual since appropriate constraints qualifications trivially hold. It is now easy to explicitly write the function $\varphi'(\, \mu \,) = \sum_{i=1}^{n} z_i^*(\, \mu \,)$: it is piecewise-linear, continuous (since the individual $z_i^*(\, \mu \,)$ are) and it has at most $n$ breakpoints corresponding to the values $\mu = -d_i$ for the $z_i$ that have sign constraints. Once the breakpoints are properly ordered in $O(\, n \log(\, n \,) \,)$, a linear visit among them allows to find in

$O(n)$ the value $\mu^*$ s.t. $\varphi'(\mu^*) = 0$ and the corresponding optimal primal solution $z^* = z^*(\mu^*)$ (with minor adjustments required for the $z_i$ corresponding to the "critical" breakpoint). This is a special case of the classical dual approach to nonlinearly constrained convex quadratic knapsack problems, see, e.g., [5] and the references therein [**back**]

- Projection of $x$ over box constraints $[\underline{x}, \overline{x}]$ is trivial since it can be done independently for each variable: $x_i^* = \min\{\overline{x}_i, \max\{x_i, \underline{x}_i\}\}$. This is the formula for solving $x_i^* = \operatorname{argmin}\{h(z) = (z - x_i)^2/2 : \underline{x}_i \leq z \leq \overline{x}_i\}$, which it just the explicit formulation of the projection problem (the univariate minimization of a quadratic function), since $h'(z) = z - x_i$, thus the unconstrained minimum is $h'(z) = 0 \equiv z = x_i$ [**back**]

- The projection problem of $x$ over $\mathcal{B}_2(0, r)$, which is $\min\{ f(z) = \|z - x\|_2^2/2 : g(x) = \|x\|_2^2/2 - r^2/2 \leq 0 \}$, is very easy: if $\|x\|_2 \leq r$ then $z^* = x$, otherwise $z^* = x(r/\|x\|)$. Indeed, $\nabla f(z) = z - x$ and $\nabla g(z) = z$, thus (KKT-G) reads $z^* - x + \lambda^* z^* = 0 \equiv z^* = x/(1+\lambda^*)$: as announced, $z^*$ must be a positive scalar multiple of $x$. Now, $\|x\| > r$ but $\|z^*\| \leq r$ by (KKT-F), thus $1 + \lambda^* > 1 \equiv \lambda^* > 0$: then, (KKT-CS) gives $\|z^*\| = r \equiv \|x\|/(1+\lambda^*) = r \equiv 1 + \lambda^* = \|x\|/r \implies z^* = x(r/\|x\|)$ as desired [**back**]

- The Reduced KKT formulæ for the equality constrained quadratic problem $\min\{ x^T Q x/2 + qx : Ax = b \}$ read $[AQ^{-1}A^T]\mu = -b - AQ^{-1}q$ and $x = -Q^{-1}(A^T\mu + q)$. Here, $x = d$, $Q = I$, $q = \nabla f(x)$ (since the objective is $\|d + \nabla f(x)\|^2/2 = d^T d/2 + \nabla f(x)d +$ constant), $A = \bar{A}$, and $b = 0$: plugging them in gives $[\bar{A}\bar{A}^T]\mu = -\bar{A}\nabla f(x)$ and $d = -(\bar{A}^T[\bar{A}\bar{A}^T]^{-1}(-\bar{A}\nabla f(x)) + \nabla f(x)) = P(-\nabla f(x))$, with $P = I - \bar{A}^T[\bar{A}\bar{A}^T]^{-1}\bar{A}$ being the projection matrix [**back**]

- Obviously, because then $x$ is optimal. In fact, the "poorman's KKT conditions on the projected problem read $\nabla f(x) + \mu \bar{A} = 0$: since $\mu \geq 0$, with the already-seen trick of defining $\lambda_i^* = \mu_i$ for $i \in \mathcal{A}(x)$ and $\lambda_i^* = 0$ for $i \notin \mathcal{A}(x)$ one obtains a $\lambda^* \geq 0$ s.t. $\nabla f(x) + \lambda^* A = 0$ and $\lambda^*(b - Ax) = 0$, i.e., that satisfies all of the KKT conditions (feasibility is given for granted) [**back**]

- $\bar{A}$ being squared and full row rank means that it is nonsingular: hence $[\bar{A}\bar{A}^T]^{-1} = \bar{A}^{-T}\bar{A}^{-1}$, and therefore $P = I - \bar{A}^T[\bar{A}\bar{A}^T]^{-1}\bar{A} =$
$= I - \bar{A}^T\bar{A}^{-T}\bar{A}^{-1}\bar{A} = I - I = 0 \implies d = P(-\nabla f(x)) = 0$ [**back**]

- $B$ is a base $\equiv$ $A_B$ nonsingular $\implies$ $d = 0$ (as we have seen already) $\implies \langle \nabla f(x), d \rangle \leq \varepsilon$ surely holds. Also, $\mu_B = -[A_B A_B^T]^{-1} A_B \nabla f(x) =$
$= -A_B^{-T} A_B^{-1} A_B \nabla f(x) = -A_B^{-T} \nabla f(x)$, i.e., $\mu_B$ is the (unique, and existing) solution of the linear system $\mu_B^T A_B = -\nabla f(x)^T$ [**back**]

- The projection matrix $P = I - A_B^T[\,A_B A_B^T\,]^{-1}A_B$ is symmetric and idempotent, i.e., $PP = P$, which implies $P \succeq 0$ since its eigenvalues can only be 1 and 0 [16]. Symmetry is trivial, verifying the other property is just algebra:
$$\begin{aligned} PP &= (\,I - A_B^T[\,A_B A_B^T\,]^{-1}A_B\,)(\,I - A_B^T[\,A_B A_B^T\,]^{-1}A_B\,) = \\ &= I - 2A_B^T[\,A_B A_B^T\,]^{-1}A_B + (\,A_B^T[\,A_B A_B^T\,]^{-1}A_B\,)(\,A_B^T[\,A_B A_B^T\,]^{-1}A_B\,) = \\ &= I - 2A_B^T[\,A_B A_B^T\,]^{-1}A_B + A_B^T(\,[\,A_B A_B^T\,]^{-1}A_B A_B^T\,)[\,A_B A_B^T\,]^{-1}A_B = \\ &= I - 2A_B^T[\,A_B A_B^T\,]^{-1}A_B + A_B^T[\,A_B A_B^T\,]^{-1}A_B = I - A_B^T[\,A_B A_B^T\,]^{-1}A_B = P. \end{aligned}$$
Hence, $\langle\,\nabla f(\,x\,)\,,\,d\,\rangle = \langle\,\nabla f(\,x\,)\,,\,P(\,-\nabla f(\,x\,)\,)\,\rangle = -\nabla f(\,x\,)^T P \nabla f(\,x\,) < 0$ (in fact it could also be $= 0$, but in this case the algorithm stops) [**back**]

- The relevant property of $d$ is that $A_B d = 0$: hence, it is only guaranteed to be a feasible direction if $B = \mathcal{A}(\,x\,)$. In fact, $A_B(\,x + \alpha d\,) = A_B x + \alpha A_B d = = b_B + 0 = b_B$ (since $B \subseteq \mathcal{A}(\,x\,)$), i.e., all constraints in $B$ cannot be violated whatever step $\alpha$ is taken along $d$ (they will all remain active). Thus, if all the constraints $i \notin B$ are inactive, i.e., $A_i x < b_i$, then $x + \alpha d$ is feasible for a small enough nonzero step; in other words, $\bar{\alpha} > 0$. However, if there exists some $h \in \mathcal{A}(\,x\,)$ s.t. $h \notin B$, then $d$ is no longer guaranteed to be feasible. This happens if $A_h d > 0$: in fact, in this case each step $\alpha > 0$ would lead to

$A_h(\, x + \alpha d\,) = A_h x + \alpha A_h d = b_h + \alpha A_h d > b_h$, i.e., violating the $h$-th constraint. In this case $\bar{\alpha} = 0$ and a degenerate step is made where $x$ remains the same but $B$ does not, as (one of) the active constraint(s) $h \notin B$ is added to it. It is easy to see that $A_h$ is not linearly dependent from $A_B$: if this were true there would exist $\gamma$ s.t. $\gamma^T A_B = A_h$, but this would imply that $0 = \gamma^T A_B d = A_h d$ (since $A_B d = 0$) while we know that $A_h d > 0$. Thus, the new $B$ at the begining of the next iteration will be different (strictly larger) and a different $d$ will be generated. It is not trivial that this process terminates with either a feasible direction or proving that $x$ is optimal because $d = 0$ may eventually happen, leading to some indices to be removed from $B$ which could lead to cycling. This is known to be very unlikely in practice and it is avoided by Bland's anti-cycle rule [6, §4.3] [**back**]

- Because $A_B \in \mathbb{R}^{k \times n}$ is full row rank (which implies $k \leq n$), with a trick we have seen already we can write it (after reshuffling of columns if necessary) as $[\, A'_B \,,\, A''_B \,]$. Now, let $p \in \{\, 1 \,, \ldots \,, k \,\}$ be the position in $B$ of the row $h$, i.e., $[\, A_B \,]_p = A_h$, and consider the direction $\xi = [\, -(\, A'_B \,)^{-1} u_p \,,\, 0 \,]$ whose first $k$ components are the opposite of the $p$-th column of $(\, A'_B \,)^{-1}$ while the remaining $n - k$ ones are 0. Hence, $A_B \xi = [\, A'_B \,,\, A''_B \,][\, -(\, A'_B \,)^{-1} u_p \,,\, 0 \,] = = A'_B (\, A'_B \,)^{-1} (\, -u_p \,) + A''_B 0 = -u_p$, i.e., $A_i \xi = 0$ for $i \in B' = B \setminus \{\, h \,\}$ and $A_h \xi = -1 < 0$ **[back]**

- The "poorman's KKT" of the projection problem read $d + \nabla f(\, x \,) + \mu_B A_B$, and since $d = 0$ this gives $\nabla f(\, x \,) = -\mu_B A_B$. Hence, $\langle\, \nabla f(\, x \,) \,,\, \xi \,\rangle = \langle\, -\mu A_B \,,\, \xi \,\rangle = \langle\, -\mu_B \,,\, A_B \xi \,\rangle = -\mu_h \langle\, A_h \,,\, \xi \,\rangle < 0$ **[back]**

- The fundamental idea is that in the LP case one can ensure that $B$ is always a base, which streamlines many operations starting from computing $\mu_B$ and (not) $d$ $(= 0)$. Also, for the direction $\xi$ in the previous exercise one has $A'_B = A_B$ and $A''_B$ is void, hence $\xi = -A_B^{-1} u_p$ can be obtained at little cost since $A_B^{-1}$—or, equivalently, a factorization of $A_B$—need be computed to find $\mu_B$ in the first place. This $\xi$ is a descent direction: in fact, $\langle \nabla f(x), \xi \rangle = \langle -\mu_B^T A_B, \xi \rangle = = \langle -\mu_B^T, A_B \xi \rangle = \langle -\mu_B^T, -u_p \rangle = \mu_h < 0$. While $\xi$ is not the projection of the anti-gradient, hence the directional derivative is in principle less negative, it has a specific benefit: if the algorithm uses $d = \xi$ and performs a full step $\bar{\alpha}$, it easy to prove that $B'' = B \setminus \{h\} \cup \{k\}$ is another base, i.e., that $A_{B''}$ is nonsingular [6, Lemma 4.5], which allows to keep applying the streamlining at the next iteration. And since $f(\cdot)$, clearly $\bar{\alpha}$ is the optimal step, which also implies that the Line Search is also useless and can be avoided entirely. Note that the special cases $\bar{\alpha} = \infty$ (which, as we have already seen, implies that $(P)$ is unbounded below) ane $\bar{\alpha} = 0$ (a degenerate step, discussed already) can happen but are not a problem. Hence, in the LP case the projected gradient algorithm can be streamlined as to always have $B$ as a basis: doing this results in the well-known (primal) simplex algorithm for Linear Programs [6, §4.2]. There actually is a caveat: one needs not only a feasible starting point, but a

feasible starting basis (in other words, the starting point needs be a vertex of the polyhedron). We have already seen how computing the starting point, or proving there is none, can be recasted as an LP: it is possible to work out the details [6, §4.4] to prove that for such an LP we can easily construct a feasible (for the auxiliary problem) starting basis, and that upon termination it provides a feasible (for the original problem) basis or the proof that no feasible point exists. Doing this in details requires to weed out weird cases such as that $A$ has not full column rank, i.e., no basis exists; this is also doable, but the details are not important here [**back**]

- The Lagrangian dual $\max\{\min\{\langle \nabla f(x), z\rangle + \lambda(Az - b)\} : \lambda \geq 0\}$ of $(MP_x)$ has a bounded below inner min $\iff \nabla f(x) + \lambda A = 0$, in which case its optimal value is 0: hence, $\max\{\lambda(-b) : \lambda A = -\nabla f(x), \lambda \geq 0\}$ is the linear dual of $(MP_x)$, whose optimal solution $\lambda^* \geq 0$ thus satisfies (KKT-G) $\nabla f(x) + \lambda^* A = 0$. (KKT-F) is kept satisfied by $x$ all along. Since $\bar{x}$ is optimal for $(MP_x)$ it satisfies its (KKT-CS) $\lambda^*(b - A\bar{x}) = 0$. But $\langle \nabla f(x), d\rangle = 0 \equiv \langle \nabla f(x), \bar{x}\rangle = \langle \nabla f(x), x\rangle$: hence, $x$ is also optimal for $(MP_x)$, thus it satisfies its (KKT-CS) $\lambda^*(b - Ax) = 0$, which are (KKT-CS) for $(P)$ [**back**]

- $\langle \nabla f(x), d \rangle = \langle \nabla f(x), \bar{x} - x \rangle \leq 0$ since $\bar{x}$ is the optimal solution of the master problem, hence $\langle \nabla f(x), \bar{x} \rangle \leq \langle \nabla f(x), z \rangle \; \forall z$ s.t. $Az \leq b$ and therefore also for $z = x$: since it is not $= 0$, it can only be $< 0$ [**back**]

- This hinges on the (sub)gradient inequality $f(z) \geq f(x) + \langle \nabla f(x), z - x \rangle$ which is valid $\forall z \in \mathbb{R}^n$, i.e., to the fact that the first-order model of $f(\cdot)$ is a lower approximation of $f(\cdot)$ everywhere since $f(\cdot)$ is convex. Thus, minimizing independently over the feasible region on the two sides of the inequality gives $\nu(P) = \min\{ f(z) : Az \leq b \} \geq f(x) + \min\{ \langle \nabla f(x), z - x \rangle : Az \leq b \} = f(x) + \langle \nabla f(x), \bar{x} - x \rangle = f(x) + \langle \nabla f(x), d \rangle = v^*$ [**back**]

- The first consequence of $Q \not\succ 0$ is that the Lagrangian relaxation does not in general have a unique optimal solution, hence $\psi \notin C^1$. Yet, $\psi$ remains concave and one can use convex nonsmooth algorithms to solve $(D)$. An issue is that as $\{ \lambda^i \} \to \lambda_*$ it is no longer true that $\{ x(\lambda^i) \} \to x_*$ automatically holds: however, this can be solved by, e.g., Proximal Bundle methods whose dual solution of the Master Problem can be used to construct aggregated solutions $\{ \tilde{x}^i \} \to x_*$ as hinted at in the relevant deck of slides. Another issue, however, is that for some $\lambda$ the Lagrangian may not have an optimal solution at all since it is unbounded below. This may happen even if $Q \succeq 0$, as there can be a direction $d$ of 0 curvature ($d^T Q d = 0$) where the linear part of the objective is also not null, i.e., $qd + \lambda(b - Ad) \neq 0$. One should then add linear constraints on $\lambda(b - Ad) = qd$ to $(D)$ to "neutralise" all these directions; these are in principle infinitely many but clearly only a finite set of constraints is needed (say, those corresponding to all eigenvectors of $Q$ corresponding to 0 eigenvalues). Algorithmically speaking, one could also avoid do insert all these constraints from the beginning and only do this "on demand" when a $\lambda^i$ is generated such that $\psi(\lambda^i) = -\infty$, as detecting unboundedness of the Lagrangian relaxation amounts precisely at finding one of the offending $d$; yet, by adding (many) "complex" linear constraints to the much simpler $\lambda \geq 0$ in

($D$) may make it significantly more costly to solve (but see, for instance, ▸ Frank-Wolfe for $f \notin C^1$ for ways to do that with a Bundle method).
$Q$ having negative curvature directions is more complicated, as then $\psi(\lambda) = -\infty \; \forall \lambda$ and ($D$) $\equiv$ ($P$) no longer holds (not surprising since the latter is no longer convex while the former always is). One possible approach here is to add to ($P$) other constraints that are not relaxed and that make the dual function bounded; e.g., identify some $\mathcal{B}_p(0, r)$ that contains all feasible solutions (if any exists, as it may not) and leave the corresponding constraint in the Lagrangian relaxation. This is still tricky in that, for instance, balls in the 1- or $\infty$-norm would still lead to $\mathcal{NP}$-hard relaxations, but a ball in the 2-norm or some other "simple" sets could instead be used [11] [**back**]

- $f_\gamma$ is the sum of two terms: one is a simple quadratic function whose gradient and Hessian are trivial, so we concentrate on the second, which in turn is the sum of $m$ terms $h_i(x) = -\ln(b_i - A_i x)$. Hence we study gradient and Hessian of each $h_i(\cdot)$ and then sum (and multiply everything by $\gamma$). It is also convenient to introduce the vector $s \in \mathbb{R}^m$ of the slacks of the constraints, i.e., $s_i = b_i - A_i x$: for all $h_i(\cdot)$ to be well-defined, necessarily $s > 0$. Now, the multivariate chain rule gives $\nabla h_i(x) = -A_i^T(1/s_i)$; note that $A_i$ is the $i$-th row of $A$ and therefore is a row—horizontal—$n$-vector while by default the gradient must be a column—vertical—$n$-vector. Introducing $S \in \mathbb{R}^{m \times m} = \mathrm{diag}(s)$, one can finally write $\nabla f_\gamma(x) = Qx + q + \gamma A^T S^{-1} u$; note that $S^{-1}u$ is the $m$-vector with entries $1/s_i$. Since the first derivative w.r.t. $x_j$ of $h_i(\cdot)$ is $-A_{ij}(1/s_i)$, the second-order partial derivative w.r.t. both $x_j$ and $x_k$ is $A_{ij}A_{ik}(1/s_i^2)$ (recall that $[z^{-1}]' = -1/z^2$): hence, $\nabla^2 h_i(x) = A_i^T A_i (1/s_i^2)$. Thus, $\nabla^2 f_\gamma(x) = Q + \gamma A^T S^{-2} A$, and Newton's direction then reads $d = -[Q + \gamma A^T S^{-2} A]^{-1}(Qx + q + \gamma A^T S^{-1} u,)$. Qualitatively speaking, the term "$S^{-2}$" is somewhat worrying: for any constraint $i$ that is active in the optimal solution, i.e., $i \in \mathcal{A}(x^*)$, one would want to drive $s_i$ to be "very small"; say, 1e-8 to 1e-12. But then, $s_i^2$ would be "very very small", and therefore $1/s_i^2$ would be "very very large", i.e., many orders of magnitude

larger than the $s_i$ of non-active constraints. This may make the $A^T S^{-2} A$ term extremely ill-conditioned, although this is somehow balanced by the fact that $\gamma \to 0$; but overall, it should be apparent that the approach can be prone to numerical difficulties. This is one of the reasons why primal-dual approaches are preferred: not only they tend to be more robust (we will see some "$^{-1}$", but no "$^{-2}$" there), they can also be easily complemented with crossover techniques that, using both primal and dual information, try to guess $\mathcal{A}(x^*)$ and "jump on the right active set" when the algorithm has reached close enough to $x^*$, hybridising the barrier method with efficient simplex-based ones exactly at its final stage where the numerical difficulties are more significant [**back**]

- It should be recalled that the $x$ variables in $(D)$, as we have discussed when deriving it, are formally distinct from those in $(P)$; however, it is true that they are meant to ultimately become the same in primal-dual optimal solutions, and anyway there is only one copy of $x$ in the KKT conditions. One then has $x^T Qx/2 + qx - (-\lambda b - x^T Qx/2) = x^T Qx + qx + \lambda b = x^T(Qx + q) + \lambda b =$ [ using $Qx + q = -\lambda A$ ] $= -\lambda Ax + \lambda b = \lambda(b - Ax) =$ [ using $s = b - Ax$ ] $= \lambda s$ [**back**]

- We are using the last constraint in $(*)$ to rewrite $\Delta s = \Lambda^{-1}(\,\gamma u - S\Delta\lambda\,) - s$ (we can do it because $\lambda > 0$ and therefore $\Lambda^{-1}$ is well-defined, and we are using $\Lambda^{-1}\Lambda S u = S u = s$), and then we substitute this in the second constraint in $(*)$ to get $A\Delta x + \Delta s = r^D \implies A\Delta x - [\,\Lambda^{-1}S\,]\Delta\lambda = r^D + s - \gamma\Lambda^{-1}u$ [**back**]

- On top of the previous development, we are now using the second constraint in $(*)$ to further rewrite $A\Delta x - \Lambda^{-1}S\Delta\lambda = r^D + s - \gamma\Lambda^{-1}u \implies \Delta\lambda = = S^{-1}(\,\gamma u - \Lambda r^D\,) + \Lambda S^{-1}A\Delta x - \lambda$ (again, $s > 0 \implies S^{-1}$ is well-defined, plus we have used $[\,\Lambda^{-1}S\,]^{-1}s = \Lambda S^{-1}S u = \Lambda u = \lambda$ and $[\,\Lambda^{-1}S\,]^{-1}\gamma\Lambda^{-1}u = = S^{-1}\Lambda\gamma\Lambda^{-1}u = \gamma S^{-1}u$); plug this in the first constraint and rearrange [**back**]

- We first have to compute $(D)$. This can be done in different ways, one of which is to rewrite $(P)$ in an equivalent form that has the shape of a QP for which we have already derived the dual. We rather take the direct route: the Lagrangian relaxation is $(P_{\mu,\lambda})$ $\min_x\{\, x^T Qx\,/\,2 + qx + \mu(\, b - Ax\,) - \lambda_- x + \lambda_+(\, x - u\,)\,\}$, that only has solution if $Qx + q - \mu A - \lambda_- + \lambda_+ = 0$. Thus, $(D)$ $\min\{\, x^T Qx\,/\,2 + \mu b - \lambda_+ u \;:\; Qx - \mu A - \lambda_- + \lambda_+ = -q\,,\; \lambda_- \geq 0\,,\; \lambda_+ \geq 0\,\}$. The slackened KKT conditions for $(P)$ and $(D)$ therefore read $Ax = b$, $0 \leq x \leq u$, $Qx - \mu A - \lambda_- + \lambda_+ = -q$, $\lambda_- x = \gamma u$, $\lambda_+(\, u - x\,) = \gamma u$. Due to the simplicity of the upper bound constraints $x \leq u$ we can avoid to introduce formal slack variables, as they would just be $s = u - x$, $s \geq 0$. We now introduce the "current iterate plus displacement" notation, i.e., $x \to x + \Delta x$ , $\mu \to \mu + \Delta\mu$ , $\lambda_- \to \lambda_- + \Delta\lambda_-$ , $\lambda_+ \to \lambda_+ + \Delta\lambda_+$, as well as the primal and dual residuals of the equality constraints, respectively $r^P = b - Ax$ and $r^D = \mu A + \lambda_- - \lambda_+ - Qx - q$. The two slackened (KKT-CS) then are $(\, x + \Delta x\,)(\, \lambda_- + \Delta\lambda_-\,) = \gamma u$ and $(\, u - x - \Delta x\,)(\, \lambda_+ + \Delta\lambda_+\,) = \gamma u$, which we rewrite $X\Delta\lambda_- + \Lambda_-\Delta x = \gamma u - X\Lambda_- u - \Delta X\Delta\Lambda_- u$ and $(\, U - X\,)\Delta\lambda_+ - \Lambda_+\Delta x = \gamma u - (U - X)\Lambda_+ u + \Delta X\Delta\Lambda_+ u$, with the usual

notation whereby an upper case $S$ indicates the diagonal matrix having as diagonal entries the lower case vector $s$. Thus, the slackened KKT system reads

$$\begin{bmatrix} Q & -A^T & I & -I \\ A & 0 & 0 & 0 \\ \Lambda_- & 0 & X & 0 \\ \Lambda_+ & 0 & 0 & U-X \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta\mu \\ \Delta\lambda_+ \\ \Delta\lambda_- \end{bmatrix} = \begin{bmatrix} r^D \\ r^P \\ \gamma u - X\Lambda_- u - \Delta X \Delta\Lambda_- u \\ \gamma u - (U-X)\Lambda_+ u + \Delta X \Delta\Lambda_+ u \end{bmatrix}$$

We now proceed as usual by removing the bilinear terms in $\Delta x$ and $\Delta\lambda_-$ / $\Delta\lambda_+$ from the right-hand-side of the last two constraints (with the provision that they could be iteratively re-inserted with the fixed value of the previous iteration in a predictor-corrector approach). This allows to start by solving the last two constraints over $\Delta\lambda_-$ and $\Delta\lambda_+$, yielding $\Delta\lambda_- = \gamma X^{-1} u - \lambda_-$ and $\Delta\lambda_+ = \gamma (U-X)^{-1} u - \lambda_+$ (again, $x > 0$ and $x < u$ make $X^{-1}$ and $(U-X)^{-1}$ well-defined, and we have used similar tricks as before). Substituting this in the first constraint gives a Normal Equation version in $[\Delta x\,,\, \Delta\mu]$ only, and then substituting away $\Delta\mu$ gives a Reduced KKT version. The lengthy and tedious final formulæ are left as final exercise [**back**]