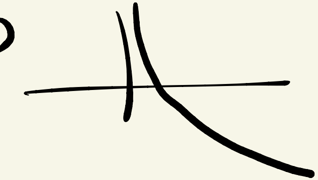
Interior point Algorithm

- perturbed optimality conditions
- Newton's method
- primal-dual method for LP's.

Barrier function

$-\log(x)$

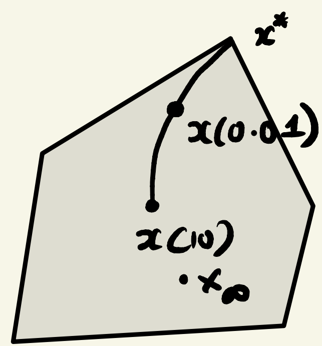


(P_t) $\quad \underset{x}{\text{minimize}} \quad B_t(x) \quad \text{s.t.} \quad Ax=b.$

- minimizer of the barrier problem depends on t:

  x_t solves P_t.

- minimizer of P_t is unique for each t because of convexity of B_t.



$$x_\infty = \underset{Ax=b}{\arg\min} -\sum_i \log(x_i)$$

Primal barrier method

solve a sequence of linearly constrained nonlinear functions:

choose $x_0 > 0$, $t_0 > 0 \ (\simeq 1)$, $\tau < 1$

repeat

x_{k+1} argmin $B_t(x)$ subject to $Ax = b$

$t_{k+1} \leftarrow \tau t_k$

until t_k is "small"

under mild conditions $x_k \to x^*$.

Perturbed optimality conditions

primal program:

$$\min_x \left(c^T x - t \underbrace{\sum_j \overbrace{\log(x_i)}^{\left(-\frac{t}{x_1}, \dots, -\frac{t}{x_n}\right)}}\right) =: f(x)$$

subject to $Ax = b$ $(x > 0)$

dual program:

$$\max_{y,z} \; b^T y + t \sum_{j=1}^{m} \log(z_j)$$

subject $A^T y + z = c$, $z > 0$

$$\mathcal{L}(x,y) = c^T x - t \sum_i \log(x_i) + y^T(b - Ax)$$

Recall: $\min_x f(x)$ s.t. $Ax = b$ optimality was: $\nabla f(x^*) = A^T y$

$Ax = b$

$$\begin{bmatrix} \nabla_y f \\ \nabla_z f \end{bmatrix} = \begin{bmatrix} A \\ I \end{bmatrix} w$$

$Aw = -b$

$-tZ^{-1}e = w$

$A^T y + z = c$

$$\underline{\begin{bmatrix} A^T & I \end{bmatrix}} \begin{bmatrix} y \\ z \end{bmatrix} = c$$

optimality conditions:

$$c - tX^{-1}e = A^T y, \quad X = \begin{bmatrix} x_1 & & 0 \\ & \ddots & \\ 0 & & x_n \end{bmatrix}$$

$Ax = b$

We can tie these optimality conditions together by identifying

$x \equiv -w$ $\quad -tZ^{-1}e = -x \iff t\frac{1}{z_i} = x_i \iff x_i z_i = t$

optimality conditions:

$$c - tX^{-1}e = A^Ty, \quad X = \begin{bmatrix} x_1 & & 0 \\ & \ddots & \\ 0 & & x_n \end{bmatrix}$$

$$Ax = b$$

$$Aw = -b$$

$$-tZ^{-1}e = w$$

$$A^Ty + z = c$$

We can tie those optimality conditions together by identifying $x \equiv -w$

$$-tZ^{-1}e = -x \iff t\frac{1}{z_i} = x_i \iff x_i z_i = t$$

we can write optimality condition simultaneously as:

$$\left.\begin{array}{lll} Ax = b, & x \geq 0, & m \text{ equations} \\ A^Ty + z = c, & z \geq 0, & n \text{ equation} \\ x_i z_i = t, & i = 1, \dots, n & \end{array}\right\} \Rightarrow \# \, 2n + m \text{ equation.}$$

Newton's Method

define $F_t(x,y,z) := \begin{bmatrix} Ax-b \\ A^Ty+z-c \\ Xz-te \end{bmatrix}$

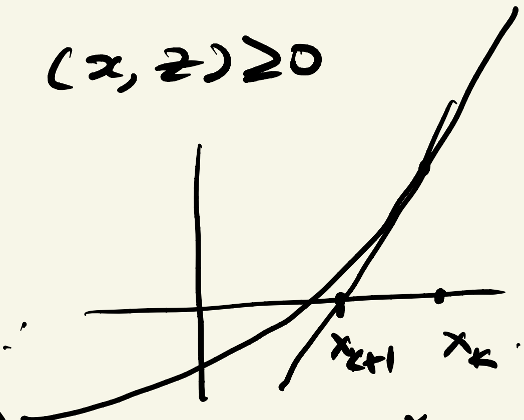
$x_i z_i = t, \; i=1,\dots,n$

$\iff Xz = te$

An approximate solution (x,y,z) with $(x,z) \geq 0$ satisfies

$$F_t(x,y,z) = 0$$

Apply Newton's method for root finding:



$$(x_{k+1}, y_{k+1}, z_{k+1}) = (x_k, y_k, z_k) + \alpha(p_x, p_y, p_z)$$

$$F_t^{k+1} = F_t^k + J^k \underbrace{(x - x^k)}_{p}$$

where p satisfies:

$$J_k p = -F_k \iff \begin{bmatrix} A & 0 & 0 \\ 0 & A^T & I \\ Z_k & 0 & X_k \end{bmatrix} \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} = \begin{bmatrix} b - Ax_k \\ c - z_k - A^T y_k \\ te - X_k z_k \end{bmatrix}$$

Primal-dual method for LP

choo $x_0 > 0,\ y_0,\ z_0 > 0,\ \tau < 1$, $\quad x_i z_i = t$

$\Rightarrow x_i^T z_i = nt$

$\gamma_0 \leftarrow x_0^T z$

$t_0 = \tau \gamma_0 / n$

while $t_k > \varepsilon$

solve $J_k p = -F_k$ for $p = (p_x, p_y, p_z)$ $\leftarrow$ Newton step.

$$\beta_x^k = \min\left\{1,\ 0.995 \min_{\{j \mid p_j^x < 0\}} \frac{-x^k}{p_j^x}\right\}$$

$$\beta_z^k = \min\left\{1,\ 0.995 \min_{\{j \mid p_j^z < 0\}} \frac{-z^k}{p_j^z}\right\}$$

$x_{k+1} = x_k + \beta_x^k p_x$, $\quad y_{k+1} = y_k + \beta_z^k p_y$, $\quad z_{k+1} = z_k + \beta_z^k p_z$

$t_k \leftarrow \tau \gamma_k / n$

Linear Algebra

The main work is in computing the step direction (p_x, p_y, p_z):

$$\begin{bmatrix} A & 0 & 0 \\ 0 & A^T & I \\ Z_k & 0 & X_k \end{bmatrix} \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} = \begin{bmatrix} b - A x_k \\ c - A^T y - z_k \\ te - X_k z_k \end{bmatrix} =: \begin{bmatrix} r_d \\ r_p \\ r_t \end{bmatrix}$$

Solve this 2x2 block system by removing p_z:

$$\begin{bmatrix} -X_k^{-1} Z_k & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} p_x \\ p_y \end{bmatrix} = \begin{bmatrix} r_d \\ r_d - X_k^{-1} r_t \end{bmatrix}$$

$$\boxed{Z_k p_x + X_k p_z = r_t \Rightarrow p_z = X_k^{-1} r_t - X_k^{-1} Z_k p_k}$$