

1 Interior Methods for Linear Programs

First we consider “vanilla” linear programs of the form

LP	minimize	$c^T x$			
	x				
	subject to	$Ax = b,$		$: y$	
		$x \geq 0,$		$: z$	

where $A \in \mathbb{R}^{m \times n}$ and (y, z) are the dual variables for the general constraints and bounds. We assume $m \leq n$ and $\text{rank}(A) = m$. The dual linear program is

LD	minimize	$-b^T y$			
	x				
	subject to	$A^T y + z = c,$		$: w$	
		$z \geq 0,$		$: x$	

where $w = -x$ (so will not appear further). We assume that there exists a point (x, y, z) that is *primal-dual feasible*:

$$Ax = b, \quad x \geq 0, \quad A^T y + z = c, \quad z \geq 0.$$

We further assume that the *interior-point condition* is satisfied: that there exists a primal-dual feasible point (x, y, z) that is *strictly interior*:

$$Ax = b, \quad x > 0, \quad A^T y + z = c, \quad z > 0.$$

Some important results follow from the feasibility assumption. First note that if (x, y, z) is primal-dual feasible, then

$$c^T x = x^T A^T y + x^T z = b^T y + x^T z \geq b^T y,$$

so that $x^T z$ is an important quantity. Indeed it is zero at an optimal solution.

Strong Duality There exists a primal-dual feasible point (x^*, y^*, z^*) such that $c^T x^* = b^T y^*$, from which it follows that $x^{*T} z^* = 0$.

Strict Complementarity (Goldman and Tucker [5]) There exists a primal-dual feasible point (x^*, y^*, z^*) such that $x^{*T} z^* = 0$ and $x^* + z^* > 0$.

Interior methods (often called *interior-point methods* or IPMs) differ from primal or dual simplex methods in their handling of the bounds on x and z and their treatment of the complementarity condition $x^T z = 0$.

First note that the optimality conditions for LP and LD may be stated as

$$Ax = b, \tag{1}$$

$$A^T y + z = c, \tag{2}$$

$$Xz = 0, \tag{3}$$

$$x, z \geq 0, \tag{4}$$

where $X = \text{diag}(x_j)$ and constraints (3)–(4) are another way of imposing the complementarity condition. (They are a more direct statement of the requirement that at least one of each pair (x_j, z_j) be zero, $j = 1:n$.) We could replace (3) by the single equation $x^T z = 0$, or we could replace (3)–(4) by the MATLAB-type vector expression $\min(x, z) = 0$. However, interior methods advanced dramatically following Megiddo's 1986 proposal to work with a perturbed form of $Xz = 0$ [9].

Simplex methods satisfy the complementarity condition at all times. Primal Simplex satisfies (1)–(3) and $x \geq 0$ while iterating until $z \geq 0$. Dual Simplex satisfies (1)–(3) and $z \geq 0$ while iterating until $x \geq 0$.

In contrast, *primal-dual interior methods* satisfy $x > 0$ and $z > 0$ throughout while iterating to satisfy (1)–(3). A key concept is to parameterize the complementarity equation and work with the equations

$$Ax = b, \tag{5}$$

$$A^T y + z = c, \tag{6}$$

$$Xz = \mu e, \tag{7}$$

where e is a vector of 1s and $\mu > 0$. The conditions $x > 0, z > 0$ are understood to hold throughout. The interior-point condition ensures that a solution exists for at least some $\mu > 0$. In fact, (5)–(7) are the unique solution of the convex problem

LP(μ)	$\begin{aligned} & \underset{x}{\text{minimize}} && c^T x - \mu \sum_j \ln(x_j) \\ & \text{subject to} && Ax = b, \quad x > 0, \end{aligned}$
-------------	---

which is well defined for all $\mu > 0$. The sequence of solutions $\{(x(\mu), y(\mu), z(\mu))\}$ for $\mu > 0$ is called the *central path* for LP and LD.

Primal-dual IPMs apply *Newton's method for nonlinear equations* to system (5)–(7) with μ decreasing towards zero in discrete stages. A vital concept is the *proximity* of the current estimate (x, y, z) to the central path. The current value of μ is retained for each Newton step until some measure of proximity is suitably small. Then μ is reduced (e.g., to $(1 - \theta)\mu$, $\theta \in (0, 1)$) and Newton's method continues.

Much research has occurred on interior methods since the mid 1980s (a revival after much earlier work on penalty and barrier methods). The monograph by Peng, Roos and Terlaky [15] summarizes much of the theory behind modern IPMs for LP and other problems, and gives an intriguing new approach to measuring proximity. See also Nocedal and Wright [10].

1.1 The Newton System

Linearizing (5–7) at the current estimate (x, y, z) gives the following matrix equation for the Newton direction $(\Delta x, \Delta y, \Delta z)$:

$$\begin{pmatrix} A & & \\ & A^T & I \\ Z & & X \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta z \end{pmatrix} = \begin{pmatrix} r_1 \\ r_2 \\ r_3 \end{pmatrix} \equiv \begin{pmatrix} b - Ax \\ c - A^T y - z \\ \mu e - Xz \end{pmatrix}, \quad (8)$$

where $Z = \text{diag}(z_j)$. We can expect A to be a very large sparse matrix. In some applications, A may be an *operator* for which products Av and $A^T w$ can be computed for arbitrary v and w .

Note that X and Z are positive-definite diagonal matrices with no large elements but more and more *small* elements as $\mu \rightarrow 0$ (since $x_j z_j \rightarrow \mu$ as Newton's method converges for any given μ). Thus, X and Z both become increasingly ill-conditioned.

This need not imply that system (8) is ill-conditioned (although it may be). If the iterates stay reasonably near the central path, we know that either x_j or z_j will be larger than $\sqrt{\mu}$. Scaling the j th row of X and Z by $\max\{x_j, z_j\}$ should keep the condition of the 3×3 block matrix similar to the condition of A .

We can make (8) structurally symmetric by reordering as follows:

$$\begin{pmatrix} & I & A^T \\ Z & X & \\ A & & \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta z \\ \Delta y \end{pmatrix} = \begin{pmatrix} r_2 \\ r_3 \\ r_1 \end{pmatrix}. \quad (9)$$

This may be helpful for some sparse-matrix solvers of the future. Nevertheless, systems (8) and (9) are large. Much research has been devoted to finding efficient and reliable numerical methods for solving such systems by eliminating blocks of variables to reduce their size. The search continues.

First, it seems reasonable to regard I as a safe block pivot to eliminate Δz . Permuting I to the top left gives

$$\begin{pmatrix} I & & A^T \\ X & Z & \\ & & A \end{pmatrix} \begin{pmatrix} \Delta z \\ \Delta x \\ \Delta y \end{pmatrix} = \begin{pmatrix} r_2 \\ r_3 \\ r_1 \end{pmatrix}. \quad (10)$$

Subtracting X times the first equation from the second gives

$$\begin{pmatrix} Z & -XA^T \\ A & \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} = \begin{pmatrix} r_3 - Xr_2 \\ r_1 \end{pmatrix} \quad \text{and} \quad \Delta z = r_2 - A^T \Delta y. \quad (11)$$

Alternatively we can permute X to the top left:

$$\begin{pmatrix} X & Z & \\ I & & A^T \\ & & A \end{pmatrix} \begin{pmatrix} \Delta z \\ \Delta x \\ \Delta y \end{pmatrix} = \begin{pmatrix} r_3 \\ r_2 \\ r_1 \end{pmatrix}. \quad (12)$$

Using X as a (dangerous!) block pivot and defining $r_4 = r_2 - X^{-1}r_3$ gives

$$\begin{pmatrix} -X^{-1}Z & A^T \\ A & \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} = \begin{pmatrix} r_4 \\ r_1 \end{pmatrix} \quad \text{and} \quad X\Delta z = r_3 - Z\Delta x. \quad (13)$$

The hard part is solving the 2×2 block systems in (11) or (13). Those who like to live dangerously (or don't know any better!) use Z in (11) or $X^{-1}Z$ in (13) as a block pivot to eliminate Δx , giving

$$A(Z^{-1}X)A^T\Delta y = r_1 + AZ^{-1}(Xr_2 - r_3), \quad Z\Delta x = r_3 - Xr_2.$$

Defining $D^2 = XZ^{-1}$ gives

$$AD^2A^T\Delta y = r_1 + AD^2r_4, \quad \Delta x = D^2(A^T\Delta y - r_4). \quad (14)$$

Often r_1 reaches zero before the other residuals (if a full step Δx is taken). System (14) is then the “normal equations” for the least-squares problem

$$\min_{\Delta y} \|Dr_4 - DA^T\Delta y\|^2.$$

Although this casts immediate doubt, system (14) has better numerical properties than one might think, as long as the iterates stay near the central path (r_3 small); see Wright [20, 21]. Many implementations are based on (14).

The main advantage of the normal-equations approach is that standard sparse Cholesky factorizations may be applied to AD^2A^T . A single call to the Analyze procedure suffices because only D changes. Most of the work for solving an LP problem goes into factorizing only 20 or 30 matrices with constant sparsity pattern. Since parallel Cholesky factorizations exist (e.g., PARDISO [13], POOCLAPACK [16, 7], WSMP [23]), it becomes clear that interior methods are much easier to parallelize than simplex methods.

1.2 Augmented Systems

A mechanical difficulty with (14) is that if A contains one or more rather dense columns, the matrix AD^2A^T will itself be very dense (let alone its factorization). Various devices have been proposed to alleviate this difficulty, but each tends to cast further numerical doubt on the normal-equations approach.

Returning to the 2×2 system (11), we may symmetrize it in various ways. For example, a similarity transformation involving $\Delta x = X^{1/2}\Delta\bar{x}$ preserves the eigenvalues (but unfortunately not the singular values!):

$$\begin{pmatrix} -Z & X^{1/2}A^T \\ AX^{1/2} & \end{pmatrix} \begin{pmatrix} \Delta\bar{x} \\ \Delta y \end{pmatrix} = \begin{pmatrix} X^{1/2}r_2 - X^{-1/2}r_3 \\ r_1 \end{pmatrix}. \quad (15)$$

Alternatively, with $\Delta x = \beta\Delta\tilde{x}$, the equivalent system

$$\begin{pmatrix} -\beta I & DA^T \\ AD & \end{pmatrix} \begin{pmatrix} \Delta\tilde{x} \\ \Delta y \end{pmatrix} = \begin{pmatrix} r_4 \\ r_1/\beta \end{pmatrix} \quad (16)$$

has good numerical properties if β is judiciously small (and r_1 is not too large). Fourer and Mehrotra [3] obtained good performance applying their own *symmetric indefinite* LBL^T factorizer (involving block-diagonal B with blocks of order 1 or 2).

System (13) can also be solved using an LBL^T factorization. Wright [22] has analyzed this approach for both non-degenerate and degenerate LP problems and shown it to be more reliable than expected from the block-pivot on X .

1.3 Quasi-Definite Systems

Sparse *Cholesky-type* factorizations of augmented systems became viable with the concept of *symmetric quasi-definite matrices* (Vanderbei [18]) and the advent of LOQO [8, 19]. By judicious formulation of the LP problem itself, Vanderbei ensures that the augmented systems have the form

$$M = \begin{pmatrix} -E & A^T \\ A & F \end{pmatrix}, \quad (17)$$

where E and F are positive definite. Factorizations $PMP^T = LDL^T$ exist for arbitrary symmetric permutations P , with D diagonal but indefinite.

Such factorizations are easier to justify with the help of certain perturbations to the LP problem, as discussed next.

2 Regularized Linear Programs

To improve the reliability of Newton's method, and to generate quasi-definite formulations with guaranteed stability, we consider perturbations to problem LP. We define a *regularized linear program* to be

$\begin{array}{ll} \text{LP}(\gamma, \delta) & \text{minimize}_{x,r} \quad c^T x + \frac{1}{2} \ \gamma x\ ^2 + \frac{1}{2} \ r\ ^2 \\ & \text{subject to} \quad Ax + \delta r = b, \quad x \geq 0, \end{array}$
--

where γ and δ are typically 10^{-3} or 10^{-4} on machines with today's normal 15–16 digit floating-point arithmetic. (We assume that the data (A, b, c) have been scaled to be of order 1.) With positive perturbations, $\text{LP}(\gamma, \delta)$ is really a strictly convex quadratic program with a unique, bounded, optimal solution (x, r, y, z) . Small values of γ and δ help keep this unique solution near a solution of the unperturbed LP. For least-squares applications we set $\delta = 1$. (Such problems tend to be easier to solve.)

2.1 The Barrier Approach

As before, we replace the non-negativity constraints by the log barrier function to obtain a sequence of convex subproblems with decreasing values of μ :

$\begin{array}{ll} \text{LP}(\gamma, \delta, \mu) & \text{minimize}_{x,r} \quad c^T x + \frac{1}{2} \ \gamma x\ ^2 + \frac{1}{2} \ r\ ^2 - \mu \sum_j \ln x_j \\ & \text{subject to} \quad Ax + \delta r = b, \end{array}$
--

where $\mu > 0$ and $x > 0$ are understood. The first-order optimality conditions state that the gradient of the subproblem objective should be a linear combination of the gradients of the primal constraint. Thus,

$$\begin{aligned} Ax + \delta r &= b \\ A^T y &= c + \gamma^2 x - \mu X^{-1} e \\ \delta y &= r, \end{aligned}$$

where $X = \text{diag}(x)$, y is a vector of dual variables, and e is a vector of 1s. Defining $z = \mu X^{-1}e$ and immediately converting to the equivalent condition $Xz = \mu e$, we obtain a system of nonlinear equations that has a unique solution for each μ :

$$\begin{aligned} Ax + \delta^2 y &= b \\ A^T y + z &= c + \gamma^2 x \\ Xz &= \mu e, \end{aligned} \quad (18)$$

where we have eliminated $r = \delta y$. These are the parameterized nonlinear equations for LP(γ, δ) corresponding to (5)–(7) for the vanilla LP. Since the perturbations appear as γ^2 and δ^2 , they tend to be negligible on well behaved problems (with $\|x\|$ and $\|y\|$ of order 1). Otherwise they help prevent those norms from becoming large.

We now apply Newton's method for nonlinear equations, with a steplength restriction to ensure that the estimates of x and z remain strictly positive.

2.2 The Newton System

Linearizing (18) at the current estimate (x, y, z) gives the system

$$\begin{pmatrix} A & \delta^2 I & \\ -\gamma^2 I & A^T & I \\ Z & & X \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta z \end{pmatrix} = \begin{pmatrix} r_1 \\ r_2 \\ r_3 \end{pmatrix} \equiv \begin{pmatrix} b - Ax - \delta^2 y \\ c + \gamma^2 x - A^T y - z \\ \mu e - Xz \end{pmatrix}, \quad (19)$$

where $Z = \text{diag}(z_j)$. The analogue of (13) is

$$\begin{pmatrix} -(X^{-1}Z + \gamma^2 I) & A^T \\ A & \delta^2 I \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} = \begin{pmatrix} r_4 \\ r_1 \end{pmatrix} \quad (20)$$

with $r_4 = r_2 - X^{-1}r_3$ again. Defining $D^2 = (X^{-1}Z + \gamma^2 I)^{-1}$, we have

$$(AD^2A^T + \delta^2 I)\Delta y = AD^2r_4 + r_1, \quad (21)$$

with $\Delta x = D^2(A^T\Delta y - r_4)$ as before. Regularization reduces the condition of both D^2 and $(AD^2A^T + \delta^2 I)$, thereby helping the normal-equations approach.

2.3 Quasi-Definite Systems

With γ and δ positive, we recognize system (20) to be symmetric quasi-definite (SQD). Thus, an indefinite Cholesky-type factorization exists for any symmetric permutation. The question is, under what conditions are the SQD factors *stable*?

First note that if M in (17) is SQD, then the matrix $\bar{M} = M\bar{I}$ is positive definite:

$$\begin{aligned} \bar{I} &= \begin{pmatrix} -I & \\ & I \end{pmatrix}, & z &= \begin{pmatrix} x \\ y \end{pmatrix}, \\ \bar{M} = M\bar{I} &= \begin{pmatrix} E & A^T \\ -A & F \end{pmatrix}, & z^T \bar{M} z &= x^T E x + y^T F y \geq 0. \end{aligned}$$

More generally, we find that for any permutation P ,

$$PMP^T = LDL^T \text{ if and only if } P\bar{M}P^T = L\tilde{D}U,$$

where $\tilde{I} \equiv P\bar{I}P^T$, $\tilde{D} \equiv D\tilde{I}$, and $U = \tilde{I}L^T\tilde{I}$. (Both L and U have unit diagonals, and D and \tilde{D} are indefinite but nonsingular diagonal matrices.) Thus, Golub and Van Loan's analysis of LDU factors of unsymmetric positive-definite systems [6] provides a similar analysis of LDL^T factors of SQD matrices. This observation was exploited by Gill et al.[4] to show that the factorization $PMP^T = LDL^T$ is stable for every permutation P if

- (i) $\|A\|$ is not too large compared to $\|E\|$ and $\|F\|$;
- (ii) $\text{diag}(E, F)$ is not too ill-conditioned.

Regarding system (20) as $Mv = r$, we have

$$M = \begin{pmatrix} -E & A^T \\ A & \delta^2 I \end{pmatrix}, \quad E = D^{-2} = X^{-1}Z + \gamma^2 I$$

and we find that the *effective condition number* of M is

$$\text{Econd}(M) \approx \frac{1}{\min\{\gamma^2, \delta^2\}} \text{cond}(M).$$

Hence the LDL^T approach should be stable until (x, y, z) approaches a solution (with $\text{cond}(M)$ becoming increasingly large).

A more uniform bound was obtained later [17] by writing (20) as the system

$$\begin{pmatrix} -\delta I & DA^T \\ AD & \delta I \end{pmatrix} \begin{pmatrix} \Delta \hat{x} \\ \Delta y \end{pmatrix} = \begin{pmatrix} Dr_4 \\ r_1/\delta \end{pmatrix} \equiv \hat{M}\hat{v} = \hat{r}, \quad (22)$$

where $\Delta x = \delta D \Delta \hat{x}$. This is still an SQD system and the same theory shows that

$$\text{cond}(\hat{M}) \approx \frac{\|AD\|}{\delta} \approx \frac{1}{\gamma\delta}, \quad \text{Econd}(\hat{M}) \approx \frac{1}{\gamma^2\delta^2}$$

(assuming $\|A\| \approx 1$). Hence, indefinite Cholesky factorization should be stable for all primal-dual iterations as long as $\gamma^2\delta^2 \gg \epsilon$ (where ϵ is the floating-point precision—typically 2.2×10^{-16} on today's machines). Thus we need $\gamma\delta \gg 10^{-8}$. For least-squares problems with $\delta = 1$, this is readily arranged. For regularized LPs, $\gamma = \delta = 10^{-3}$ is safe.

Such factorizations were implemented successfully within IBM's mathematical programming system OSL. The sparse Cholesky solver in WSMP [23] is applied to either M or $AD^2A^T + \delta^2 I$ (whichever is more sparse). A major benefit is that any dense columns in A are handled sensibly without special effort.

2.4 Least-Squares Formulation

With $\delta > 0$, we have an alternative to SQD systems and normal equations. We may write (21) as a least-squares problem even when $r_1 \neq 0$:

$$\min_{\Delta y} \left\| \begin{pmatrix} Dr_4 \\ r_1/\delta \end{pmatrix} - \begin{pmatrix} DA^T \\ \delta I \end{pmatrix} \Delta y \right\|^2. \quad (23)$$

This may be solved *inexactly* by a conjugate-gradient-type iterative solver such as LSQR [11, 12]. It is especially useful when A is an *operator*, but is also applicable when A is explicit. Clearly δ should not be too small (and $\delta = 1$ is ideal).

This approach is used in SOL's MATLAB solver PDCO [14].

3 Separable Convex Optimization

PDCO (Primal-Dual Method for Convex Optimization) is a MATLAB solver for optimization problems of the form

$$\begin{array}{ll} \text{NP} & \begin{array}{l} \text{minimize}_x \quad \phi(x) \\ \text{subject to} \quad Ax = b, \quad \ell \leq x \leq u, \end{array} \end{array}$$

where $\phi(x)$ is a convex function with known gradient $g(x)$ and Hessian $H(x)$, and $A \in \mathbb{R}^{m \times n}$. The format of NP is suitable for any linear constraints. For example, a double-sided constraint $\alpha \leq a^T \tilde{x} \leq \beta$ ($\alpha < \beta$) should be entered as $a^T \tilde{x} - \xi = 0$, $\alpha \leq \xi \leq \beta$, where \tilde{x} and ξ are relevant parts of x .

To allow for constrained least-squares problems, and to ensure unique primal and dual solutions (and improve solver stability), we regularize the problem as

$$\begin{array}{ll} \text{NP2} & \begin{array}{l} \text{minimize}_{x,r} \quad \phi(x) + \frac{1}{2} \|D_1 x\|^2 + \frac{1}{2} \|r\|^2 \\ \text{subject to} \quad Ax + D_2 r = b, \quad \ell \leq x \leq u, \end{array} \end{array}$$

where D_1, D_2 are positive-definite diagonal matrices specified by the user. The diagonals of D_1 are typically small (10^{-3} or 10^{-4}). Similarly for D_2 if the constraints in NP should be satisfied reasonably accurately. For least-squares applications, some of the diagonals of D_2 will be 1. Note that some elements of ℓ and u may be $-\infty$ and $+\infty$ respectively, but we expect no large numbers in A, b, D_1, D_2 . If $\|D_2\|$ is small, we would expect A to be under-determined ($m < n$). If $D_2 = I$, A may have any shape.

3.1 The Barrier Approach

First we introduce slack variables x_1, x_2 to convert the bounds to non-negativity constraints:

$$\begin{array}{ll} \text{NP3} & \begin{array}{l} \text{minimize}_{x,r,x_1,x_2} \quad \phi(x) + \frac{1}{2} \|D_1 x\|^2 + \frac{1}{2} \|r\|^2 \\ \quad \quad \quad \quad Ax + D_2 r = b \\ \text{subject to} \quad \quad \quad x - x_1 = \ell \\ \quad \quad \quad \quad \quad \quad x + x_2 = u \\ \quad \quad \quad \quad \quad \quad x_1, x_2 \geq 0. \end{array} \end{array}$$

Then we replace the non-negativity constraints by the log barrier function, obtaining a sequence of convex subproblems with decreasing values of μ ($\mu > 0$):

$$\begin{array}{ll} \text{NP}(\mu) & \begin{array}{ll} \text{minimize}_{x,r,x_1,x_2} \quad \phi(x) + \frac{1}{2} \|D_1 x\|^2 + \frac{1}{2} \|r\|^2 - \mu \sum_j \ln([x_1]_j [x_2]_j) & \\ \quad \quad \quad \quad Ax + D_2 r = b & : y \\ \text{subject to} \quad \quad \quad x - x_1 = \ell & : z_1 \\ \quad \quad \quad \quad \quad \quad -x - x_2 = -u, & : z_2 \end{array} \end{array}$$

where y, z_1, z_2 denote dual variables for the associated constraints. With $\mu > 0$, most variables are strictly positive: $x_1, x_2, z_1, z_2 > 0$.

The KKT conditions for the barrier subproblem involve the three *primal* equations of $\text{NP}(\mu)$, along with four *dual* equations stating that the gradient of the subproblem objective should be a linear combination of the gradients of the primal constraints:

$$\begin{aligned} Ax + D_2 r &= b \\ x - x_1 &= \ell \\ -x - x_2 &= -u \\ A^T y + z_1 - z_2 &= g(x) + D_1^2 x && : x \\ D_2 y &= r && : r \\ X_1 z_1 &= \mu e && : x_1 \\ X_2 z_2 &= \mu e, && : x_2 \end{aligned}$$

where $X_1 = \text{diag}(x_1)$, $X_2 = \text{diag}(x_2)$, and similarly for Z_1, Z_2 later. The last two equations are commonly called the perturbed complementarity conditions. Initially they are in a different form. The dual equation for x_1 is really

$$-z_1 = \nabla(-\mu \ln(x_1)) = -\mu X_1^{-1} e,$$

where e is a vectors of 1's. Thus, $x_1 > 0$ implies $z_1 > 0$, and multiplying by $-X_1$ gives the equivalent equation $X_1 z_1 = \mu e$ as stated.

3.2 Newton's Method

We now eliminate $r = D_2 y$ and apply Newton's method:

$$\begin{aligned} A(x + \Delta x) + D_2^2(y + \Delta y) &= b \\ (x + \Delta x) - (x_1 + \Delta x_1) &= \ell \\ -(x + \Delta x) - (x_2 + \Delta x_2) &= -u \\ A^T(y + \Delta y) + (z_1 + \Delta z_1) - (z_2 + \Delta z_2) &= g + H\Delta x + D_1^2(x + \Delta x) \\ X_1 z_1 + X_1 \Delta z_1 + Z_1 \Delta x_1 &= \mu e \\ X_2 z_2 + X_2 \Delta z_2 + Z_2 \Delta x_2 &= \mu e, \end{aligned}$$

where g and H are the current objective gradient and Hessian. To solve this Newton system, we work with three sets of residuals:

$$\begin{pmatrix} \Delta x - \Delta x_1 \\ -\Delta x - \Delta x_2 \end{pmatrix} = \begin{pmatrix} r_\ell \\ r_u \end{pmatrix} \equiv \begin{pmatrix} \ell - x + x_1 \\ -u + x + x_2 \end{pmatrix}, \quad (24)$$

$$\begin{pmatrix} X_1 \Delta z_1 + Z_1 \Delta x_1 \\ X_2 \Delta z_2 + Z_2 \Delta x_2 \end{pmatrix} = \begin{pmatrix} c_\ell \\ c_u \end{pmatrix} \equiv \begin{pmatrix} \mu e - X_1 z_1 \\ \mu e - X_2 z_2 \end{pmatrix}, \quad (25)$$

$$\begin{pmatrix} A\Delta x + D_2^2 \Delta y \\ -H_1 \Delta x + A^T \Delta y + \Delta z_1 - \Delta z_2 \end{pmatrix} = \begin{pmatrix} r_1 \\ r_2 \end{pmatrix} \equiv \begin{pmatrix} b - Ax - D_2^2 y \\ g + D_1^2 x - A^T y - z_1 + z_2 \end{pmatrix}, \quad (26)$$

where $H_1 = H + D_1^2$. We use (24) and (25) to replace two sets of vectors in (26). With

$$\begin{pmatrix} \Delta x_1 \\ \Delta x_2 \end{pmatrix} = \begin{pmatrix} -r_\ell + \Delta x \\ -r_u - \Delta x \end{pmatrix}, \quad \begin{pmatrix} \Delta z_1 \\ \Delta z_2 \end{pmatrix} = \begin{pmatrix} X_1^{-1}(c_\ell - Z_1 \Delta x_1) \\ X_2^{-1}(c_u - Z_2 \Delta x_2) \end{pmatrix}, \quad (27)$$

$$\begin{aligned} H_2 &\equiv H + D_1^2 + X_1^{-1}Z_1 + X_2^{-1}Z_2 \\ w &\equiv r_2 - X_1^{-1}(c_\ell + Z_1 r_\ell) + X_2^{-1}(c_u + Z_2 r_u) \end{aligned} \quad (28)$$

we find that

$$\begin{pmatrix} -H_2 & A^T \\ A & D_2^2 \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} = \begin{pmatrix} w \\ r_1 \end{pmatrix}. \quad (29)$$

3.3 Solving for $(\Delta x, \Delta y)$

If $\phi(x)$ is a general convex function with known Hessian H , system (29) may be treated by direct or iterative solvers. Since it is an SQD system, sparse LDL^T factors should be sufficiently stable under the same conditions as for regularized LP ($\|A\| \approx 1$ and $\gamma\delta \gg 10^{-8}$).

Alternatively a sparse Cholesky factorization $H_2 = LL^T$ may be practical, where $H_2 = H +$ diagonal terms, and L is a nonsingular permuted triangle. This is trivial if $\phi(x)$ is a separable function, since H and H_2 in (28) are then diagonal. System (29) may then be solved by eliminating either Δx or Δy :

$$(A^T D_2^{-2} A + H_2) \Delta x = A^T D_2^{-2} r_1 - w, \quad D_2^2 \Delta y = r_1 - A \Delta x, \quad (30)$$

or

$$(A H_2^{-1} A^T + D_2^2) \Delta y = A H_2^{-1} w + r_1, \quad H_2 \Delta x = A^T \Delta y - w. \quad (31)$$

Sparse Cholesky factorization may again be applicable, but if an iterative solver must be used it is preferable to regard them as least-squares problems suitable for LSQR or CGLS [11]:

$$\min_{\Delta x} \left\| \begin{pmatrix} D_2^{-1} A \\ L^T \end{pmatrix} \Delta x - \begin{pmatrix} D_2^{-1} r_1 \\ -L^{-T} w \end{pmatrix} \right\|^2, \quad D_2 \Delta y = D_2^{-1}(r_1 - A \Delta x), \quad (32)$$

or

$$\min_{\Delta y} \left\| \begin{pmatrix} L^{-1} A^T \\ D_2 \end{pmatrix} \Delta y - \begin{pmatrix} L^{-1} w \\ D_2^{-1} r_1 \end{pmatrix} \right\|^2, \quad L^T \Delta x = L^{-1}(A^T \Delta y - w). \quad (33)$$

The right-most vectors in (32)–(33) are part of the residual vectors for the least-squares problems (and may be by-products from the least-squares solver).

PDSCO, the predecessor of PDCO, is used within the Basis Pursuit signal decomposition software called Atomizer [1]. PDCO has been applied to some large network problems with the entropy function $\sum x_j \ln x_j$ as objective. A problem with 50,000 constraints and 660,000 variables solves in about 3 minutes on a 2GHz PC, requiring less than 100 total LSQR iterations. (The entropy function keeps all variables safely away from zero. It seems to be the most “friendly” convex objective ever.)

See [2] for a fine overview of interior methods for *general* nonlinear optimization.

References

- [1] S. S. Chen, D. L. Donoho, and M. A. Saunders. Atomic decomposition by basis pursuit. *SIAM Review*, 43(1):129–159, 2001.
- [2] A. Forsgren, P. E. Gill, and M. H. Wright. Interior methods for nonlinear optimization. *SIAM Review*, 44(4):525–597, 2002.
- [3] R. Fourer and S. Mehrotra. Performance of an augmented system approach for solving least-squares problems in an interior-point method for linear programming. *Math. Program.*, 19:26–31, August 1991.
- [4] P. E. Gill, M. A. Saunders, and J. R. Shinnerl. On the stability of the Cholesky factorization for symmetric quasi-definite systems. *SIAM J. Matrix Anal. Appl.*, 17:35–46, 1996.
- [5] A. J. Goldman and A. W. Tucker. Theory of linear programming. In H. W. Kuhn and A. W. Tucker, editors, *Linear Inequalities and Related Systems, Annals of Mathematical Studies 38*, pages 63–97. Princeton University Press, Princeton, NJ, 1956.
- [6] G. H. Golub and C. F. Van Loan. Unsymmetric positive definite linear systems. *Linear Algebra and its Applications*, 28:85–98, 1979.
- [7] B. C. Gunter, W. C. Reiley, and R. A. van de Geijn. Implementation of out-of-core Cholesky and QR factorizations with POOCLAPACK.
- [8] LOQO optimization software. <http://orfe.princeton.edu/~loqo>.
- [9] N. Megiddo. Pathways to the optimal set in linear programming. In N. Megiddo, editor, *Progress in Mathematical Programming : Interior Point and Related Methods*, pages 131–158. Springer Verlag, New York, NY, 1989. Identical version in : *Proceedings of the 6th Mathematical Programming Symposium of Japan, Nagoya, Japan*, pages 1–35, 1986.
- [10] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer Series in Operations Research. Springer Verlag, New York, second edition, 2006.
- [11] C. C. Paige and M. A. Saunders. LSQR: An algorithm for sparse linear equations and sparse least squares. *ACM Trans. Math. Software*, 8(1):43–71, 1982.
- [12] C. C. Paige and M. A. Saunders. Algorithm 583; LSQR: Sparse linear equations and least-squares problems. *ACM Trans. Math. Software*, 8(2):195–209, 1982.
- [13] PARDISO parallel sparse solver. <http://www.pardiso-project.org>.
- [14] PDCO convex optimization software (MATLAB). <http://www.stanford.edu/group/SOL/software.html>.
- [15] J. Peng, C. Roos, and T. Terlaky. *Self-Regularity: A New Paradigm for Primal-Dual Interior Point Algorithms*. Princeton University Press, Princeton, NJ, 2002.
- [16] W. C. Reiley and R. A. van de Geijn. POOCLAPACK: Parallel out-of-core linear algebra package. Technical Report CS-TR-99-33, citeseer.ist.psu.edu/article/reiley99pooclapack.html, 1999.
- [17] M. A. Saunders. Cholesky-based methods for sparse least squares: The benefits of regularization. In L. Adams and J. L. Nazareth, editors, *Linear and Nonlinear Conjugate Gradient-Related Methods*, pages 92–100. SIAM, Philadelphia, 1996. Proceedings of AMS-IMS-SIAM Joint Summer Research Conference, University of Washington, Seattle, WA (July 9–13, 1995).
- [18] R. J. Vanderbei. Symmetric quasi-definite matrices. *SIAM J. Optim.*, 5:100–113, 1995.
- [19] R. J. Vanderbei. *Linear Programming: Foundations and Extensions*. Kluwer, Boston, London and Dordrecht, second edition, 2001.
- [20] S. J. Wright. Stability of linear equations solvers in interior-point methods. *SIAM J. Matrix Anal. Appl.*, 16:1287–1307, 1995.
- [21] S. J. Wright. *Primal-Dual Interior-Point Methods*. SIAM, Philadelphia, 1997.
- [22] S. J. Wright. Stability of augmented system factorizations in interior-point methods. *SIAM J. Matrix Anal. Appl.*, 18:191–222, 1997.
- [23] WSMP: Watson Sparse Matrix Package. <http://www-users.cs.umn.edu/~agupta/wsmpl.html>.