Stanford University, ICME

CME 338 Large-Scale Numerical Optimization

Instructor: Michael Saunders Spring 2019

Notes 9: **PDCO: Primal-Dual Interior Methods**

1 Interior methods for linear optimization

First we consider "vanilla" LO problems of the form

LO	$\underset{x}{\text{minimize}}$	$c^T x$	
	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 problem is

LD	$\underset{y,\, z}{\text{minimize}}$	$-b^T y$	
	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* to the bounds:

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

Some important results follow from the feasibility assumption. First note that if x is primal feasible and (y, z) is 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 If (x^*, y^*, z^*) is a primal-dual feasible point, $c^T x^* = b^T y^*$ and $x^{*T} z^* = 0$.

Strict Complementarity (Goldman and Tucker [10]) 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 LO 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 a nonlinear way of imposing the complementarity condition. (They imply 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, IPMs advanced dramatically following Megiddo's 1986 proposal to work with a perturbed form of $Xz = 0$ [18].

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 system of nonlinear equations

$$\begin{aligned} Ax &= b, \\ A^T y + z &= c, \\ Xz &= \mu e, \end{aligned} \tag{5}$$

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) gives the unique solution of the convex problem

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

which is well defined for all $\mu > 0$. The objective of CO(μ) is called the log-barrier function, and IPMs are often called *barrier methods*. The infinite sequence of solutions $\{(x(\mu), y(\mu), z(\mu))\}$ for $\mu > 0$ is the *central path* for LO and LD.

Primal-dual IPMs apply *Newton's method for nonlinear equations* to system (5) with μ decreasing toward 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; for example, $\max(Xz)/\min(Xz) \leq 1000$. Then μ is reduced in some way, e.g., to $(1 - \alpha)\mu$, $\alpha \in (0, 0.995]$, 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 [26] summarizes much of the theory behind IPMs for LO and other problems, and gives a novel approach to measuring proximity. See also Wright [34], Nocedal and Wright [21], Hinder [14]. A finite termination strategy proposed by Ye [37] makes use of the intriguing Goldman-Tucker theorem to guess the sets $\mathcal{B}$ and $\mathcal{N}$ for which $x_{\mathcal{B}} > 0$ and $z_{\mathcal{N}} > 0$, based on the condition $x_j \geq z_j \Rightarrow j \in \mathcal{B}$; see Mehrotra and Ye [19], Wright [34, pp. 146–149].

1.1 The Newton system

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

$$\begin{pmatrix} A & & \\ & A^T & I \\ Z & & X \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta z \end{pmatrix} = \begin{pmatrix} r_1 \\ r_2 \\ r_3 \end{pmatrix} \equiv \begin{pmatrix} b - Ax \\ c - A^T y - z \\ \mu e - Xz \end{pmatrix}, \tag{6}$$

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, w.

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

This need not imply that system (6) is ill-conditioned (although it may be). If the iterates stay reasonably near the central path, we know from strict complementarity that either $x_j = O(1)$, $z_j = O(\mu)$ or vice versa. Scaling the jth row of X and Z by $\max\{x_j, z_j\}$ should hence keep the condition of the 3×3 block matrix similar to the condition of A.

We can make (6) structurally symmetric by interchanging the first two rows:

$$\begin{pmatrix} & A^T & I \\ A & & \\ Z & & X \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta z \end{pmatrix} = \begin{pmatrix} r_2 \\ r_1 \\ r_3 \end{pmatrix}. \tag{K_3}$$

This is equivalent to the symmetric system

$$\begin{pmatrix} & A^T & Z^{1/2} \\ A & & \\ Z^{1/2} & & X \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \\ Z^{-1/2}\Delta z \end{pmatrix} = \begin{pmatrix} r_2 \\ r_1 \\ Z^{-1/2}r_3 \end{pmatrix}, \tag{$K_{3.5}$}$$

which may be helpful for some sparse-matrix solvers of the future. Indeed, Greif, Moulding, and Orban [12] analyze the eigenvalues of a similar system for convex quadratic optimization and advocate working directly with that system rather than eliminating variables.

Nevertheless, systems (6)–($K_{3.5}$) 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. First, it seems reasonable to regard I as a safe block pivot to eliminate Δz from (6). Permuting I to the top left gives

$$\begin{pmatrix} I & & A^T \\ X & Z & \\ & A & \end{pmatrix} \begin{pmatrix} \Delta z \\ \Delta x \\ \Delta y \end{pmatrix} = \begin{pmatrix} r_2 \\ r_3 \\ r_1 \end{pmatrix}. \tag{7}$$

Subtracting X times the first equation from the second gives the unsymmetric system

$$\begin{pmatrix} Z & -XA^T \\ A & \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} = \begin{pmatrix} r_3 - Xr_2 \\ r_1 \end{pmatrix} \quad \text{and} \quad \Delta z = r_2 - A^T\Delta y. \tag{K_{2u}}$$

Alternatively we can permute X to the top left:

$$\begin{pmatrix} X & Z & \\ I & & A^T \\ & A & \end{pmatrix} \begin{pmatrix} \Delta z \\ \Delta x \\ \Delta y \end{pmatrix} = \begin{pmatrix} r_3 \\ r_2 \\ r_1 \end{pmatrix}. \tag{8}$$

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

$$\begin{pmatrix} -X^{-1}Z & A^T \\ A & \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} = \begin{pmatrix} r_4 \\ r_1 \end{pmatrix} \quad \text{and} \quad X\Delta z = r_3 - Z\Delta x. \tag{K_2}$$

The hard part is solving the 2×2 block systems in (K_{2u}) or (K_2). Those who like to live dangerously (or don't know any better!) use Z in (K_{2u}) or $X^{-1}Z$ in (K_2) as a block pivot to eliminate Δx, giving

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

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

$$AD^2A^T\Delta y = r_1 + AD^2r_4, \qquad \Delta x = D^2\big(A^T\Delta y - r_4\big). \tag{K_1}$$

Often r_1 reaches zero before the other residuals (if a full step Δx is taken). System (K_1) 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 (K_1) has better numerical properties than one might think, as long as the iterates stay near the central path (r_3 small); see Wright [33, 34]. Many implementations are based on (K_1).

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 (ordering) procedure suffices because only D changes. Most of the work for solving an LO problem goes into factorizing only 20 to 50 matrices with constant sparsity pattern. Since parallel Cholesky factorizations exist (MUMPS [20], PARDISO [24], POOCLAPACK [28, 13], WSMP [36]), it becomes clear that interior methods are much easier to parallelize than simplex methods.

1.2 Augmented systems

A mechanical difficulty with (K_1) 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 (K_{2u}), 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!):

$$\begin{pmatrix} -Z & X^{1/2}A^T \\ AX^{1/2} & \end{pmatrix} \begin{pmatrix} \Delta\bar{x} \\ \Delta y \end{pmatrix} = \begin{pmatrix} X^{1/2}r_2 - X^{-1/2}r_3 \\ r_1 \end{pmatrix}. \qquad (K_{2.5})$$

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

$$\begin{pmatrix} -\beta I & DA^T \\ AD & \end{pmatrix} \begin{pmatrix} \Delta\widetilde{x} \\ \Delta y \end{pmatrix} = \begin{pmatrix} r_4 \\ r_1/\beta \end{pmatrix} \qquad (9)$$

has good numerical properties if β is judiciously small (and r_1 is not too large). Fourer and Mehrotra [7] obtained good performance applying their own *symmetric indefinite* LBL$^{\mathrm{T}}$ factorizer (involving block-diagonal B with blocks of order 1 or 2).

System (K_2) can also be solved using an LBL$^{\mathrm{T}}$ factorization. Wright [35] has analyzed this approach for both non-degenerate and degenerate LO 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 [31]) and the advent of LOQO [17, 32]. By judicious formulation of the LO problem itself, Vanderbei ensures that the augmented systems have the form

$$M = \begin{pmatrix} -E & A^T \\ A & F \end{pmatrix}, \qquad (10)$$

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 LO problem, as discussed next.

2 Regularized linear optimization

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

LO(γ, δ)	$\underset{x,\,r}{\text{minimize}}$	$c^Tx + \frac{1}{2}\|\gamma x\|^2 + \frac{1}{2}\|r\|^2$
	subject to	$Ax + \delta r = b, \quad x \geq 0,$

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, LO(γ, δ) is really a strictly convex quadratic problem 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 LO. 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:

CO(γ, δ, μ)	$\underset{x,\,r}{\text{minimize}}$	$c^Tx + \frac{1}{2}\|\gamma x\|^2 + \frac{1}{2}\|r\|^2 - \mu \sum_j \ln x_j$
	subject to	$Ax + \delta r = b,$

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,

$$Ax + \delta r = b, \qquad A^Ty = c + \gamma^2 x - \mu X^{-1}e, \qquad \delta y = r,$$

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:

$$\begin{aligned} Ax + \delta^2 y &= b \\ A^Ty + z &= c + \gamma^2 x \\ Xz &= \mu e, \end{aligned} \tag{11}$$

where we have eliminated $r = \delta y$. These are the parameterized nonlinear equations for LO(γ, δ) corresponding to (5) for the vanilla LO problem. 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 (11) at the current estimate (x, y, z) gives the system analogous to (6):

$$\begin{pmatrix} A & \delta^2 I & \\ -\gamma^2 I & A^T & I \\ Z & & X \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta z \end{pmatrix} = \begin{pmatrix} r_1 \\ r_2 \\ r_3 \end{pmatrix} \equiv \begin{pmatrix} b - Ax - \delta^2 y \\ c + \gamma^2 x - A^Ty - z \\ \mu e - Xz \end{pmatrix}, \tag{12}$$

where $Z = \text{diag}(z_j)$. The analogue of (K_{2u}) is

$$\begin{pmatrix} Z + \gamma^2 X & -XA^T \\ A & \delta^2 I \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} = \begin{pmatrix} r_3 - Xr_2 \\ r_1 \end{pmatrix} \quad \text{and} \quad \Delta z = r_2 + \gamma^2 \Delta x - A^T \Delta y. \tag{K_{2ur}}$$

The analogue of (K_2) is

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

with $r_4 = r_2 - X^{-1}r_3$ again and $\Delta z = r_2 + \gamma^2 \Delta x - A^T \Delta y$. A friendlier version analogous to $(K_{2.5})$ is

$$\begin{pmatrix} -(Z + \gamma^2 X) & X^{1/2}A^T \\ AX^{1/2} & \delta^2 I \end{pmatrix} \begin{pmatrix} \Delta \bar{x} \\ \Delta y \end{pmatrix} = \begin{pmatrix} X^{1/2} r_4 \\ r_1 \end{pmatrix} \tag{$K_{2.5r}$}$$

with $\Delta x = X^{1/2} \Delta \bar{x}$. Defining $D^2 = (X^{-1}Z + \gamma^2 I)^{-1}$, we have

$$\big(AD^2A^T + \delta^2 I\big)\Delta y = AD^2 r_4 + r_1 \tag{K_{1r}}$$

analogous to (K_1) with $\Delta x = D^2\big(A^T \Delta y - r_4\big)$ 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 (K_{2r}) 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 (10) is SQD, then the matrix $\bar{M} = M\bar{I}$ is positive definite:

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

More generally, we find that for any permutation P,

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

where $\widetilde{I} \equiv P\bar{I}P^T$, $\widetilde{D} \equiv D\widetilde{I}$, and $U = \widetilde{I}L^T\widetilde{I}$. (Both L and U have unit diagonals, and D and $\widetilde{D}$ are indefinite but nonsingular diagonal matrices.) Thus, Golub and Van Loan's analysis of LDU factorization of unsymmetric positive-definite systems without permutations [11] provides a similar analysis of $\mathrm{LDL^T}$ factors of SQD matrices. This observation was exploited by Gill et al. [9] to show that $PMP^T = LDL^T$ is stable for every permutation P if

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

Regarding system (K_{2r}) as $Mv = r$, we have

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

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

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

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

A more uniform bound was obtained later by Saunders [29] by writing (K_{2r}) as the system

$$\begin{pmatrix} -\delta I & DA^T \\ AD & \delta I \end{pmatrix} \begin{pmatrix} \Delta\widehat{x} \\ \Delta y \end{pmatrix} = \begin{pmatrix} Dr_4 \\ r_1/\delta \end{pmatrix} \quad \equiv \quad \hat{M}\hat{v} = \hat{r}, \tag{13}$$

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

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

(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 LO problems, $\gamma = \delta = 10^{-3}$ is safe.

Such factorizations were implemented successfully within IBM's OSL (Optimization Subroutine Library). The sparse Cholesky solver in WSMP [36] was applied to either M or $AD^2A^T + \delta^2 I$ (whichever is more sparse). A major benefit was that any dense columns in A were 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 (K_{1r}) as a least-squares problem even when $r_1 \neq 0$:

$$\min_{\Delta y} \left\| \begin{pmatrix} Dr_4 \\ r_1/\delta \end{pmatrix} - \begin{pmatrix} DA^T \\ \delta I \end{pmatrix} \Delta y \right\|^2. \tag{14}$$

This may be solved *inexactly* by a conjugate-gradient-type iterative solver such as LSQR [22, 23] or LSMR [5]. 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 was used by PDSCO in the Basis Pursuit DeNoising (BPDN) signal decomposition software Atomizer [4]. PDSCO (S = separable) has been superseded by PDCO.

2.5 Exact regularization for linear and quadratic optimization

To ensure numerical stability in solving the above regularized systems, the parameters γ and δ cannot be too small. Values as large as 10^{-3} or 10^{-4} in problem LO(γ, δ) sometimes perturb the underlying LO problem more than a user may wish for. To retain the linear-algebra benefits of regularization while eliminating the perturbation to the solution, Friedlander and Orban [8] solve a sequence of regularized problems of the form

QO(ρ, δ)	$\underset{x,\,r}{\text{minimize}}$	$c^Tx + \frac{1}{2}x^TQx + \frac{1}{2}\rho\|x - x_k\|^2 + \frac{1}{2}\delta\|r + y_k\|^2$
	subject to	$Ax + \delta r = b, \quad x \geq 0,$

where $\rho, \delta > 0$ and (x_k, y_k) are current estimates of the optimal variables (x, y). The positive semidefinite matrix Q generalizes the class of problems.

3 Convex optimization with linear constraints

PDCO (Primal-Dual Method for Convex Optimization) [25] is a MATLAB solver for optimization problems that are nominally of the form

CO	$\underset{x}{\text{minimize}}$	$\phi(x)$
	subject to	$Ax = b, \quad \ell \leq x \leq u,$

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 CO is suitable for any linear constraints. For example, a double-sided constraint $\alpha \leq a^T\widetilde{x} \leq \beta$ $(\alpha < \beta)$ should be entered as $a^T\widetilde{x} - \xi = 0$, $\alpha \leq \xi \leq \beta$, where $\widetilde{x}$ and ξ are relevant parts of x.

To allow for constrained least-squares problems, and to ensure unique primal and dual solutions (and improved stability), PDCO really solves the regularized problem

CO2	$\underset{x,\,r}{\text{minimize}}$	$\phi(x) + \frac{1}{2}\|D_1x\|^2 + \frac{1}{2}\|r\|^2$
	subject to	$Ax + D_2r = b, \quad \ell \leq x \leq u,$

where D_1, D_2 are specified positive-definite diagonal matrices. The diagonals of D_1 are typically small (10^{-3} or 10^{-4}). Similarly for D_2 if the constraints in CO should be satisfied reasonably accurately. For least-squares applications, some 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:

CO3	$\underset{x,\,r,\,x_1,\,x_2}{\text{minimize}}$	$\phi(x) + \frac{1}{2}\|D_1 x\|^2 + \frac{1}{2}\|r\|^2$
	subject to	$Ax + D_2 r = b$
		$x - x_1 = \ell$
		$x + x_2 = u$
		$x_1, x_2 \geq 0.$

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

CO(μ)	$\underset{x,\,r,\,x_1,\,x_2}{\text{minimize}}$	$\phi(x) + \frac{1}{2}\|D_1 x\|^2 + \frac{1}{2}\|r\|^2 - \mu \sum_j \ln([x_1]_j [x_2]_j)$	
		$Ax + D_2 r = b$	$: y$
	subject to	$x - x_1 = \ell$	$: z_1$
		$-x - x_2 = -u,$	$: z_2$

where y, z_1, z_2 denote dual variables for the associated constraints. With $\mu > 0$, most variables are strictly positive: $x_1, x_2, z_1, z_2 > 0$. (Exceptions: If $\ell_j = -\infty$ or $u_j = \infty$, the corresponding equation is omitted and the jth element of x_1 or x_2 doesn't exist.)

The KKT conditions for the barrier subproblem involve the three *primal* equations of CO(μ), 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:

$$\begin{pmatrix} \Delta x - \Delta x_1 \\ -\Delta x - \Delta x_2 \end{pmatrix} = \begin{pmatrix} r_\ell \\ r_u \end{pmatrix} \equiv \begin{pmatrix} \ell - x + x_1 \\ -u + x + x_2 \end{pmatrix}, \tag{15}$$

$$\begin{pmatrix} X_1\Delta z_1 + Z_1\Delta x_1 \\ X_2\Delta z_2 + Z_2\Delta x_2 \end{pmatrix} = \begin{pmatrix} c_\ell \\ c_u \end{pmatrix} \equiv \begin{pmatrix} \mu e - X_1 z_1 \\ \mu e - X_2 z_2 \end{pmatrix}, \tag{16}$$

$$\begin{pmatrix} A\Delta x + D_2^2\Delta y \\ -H_1\Delta x + A^T\Delta y + \Delta z_1 - \Delta z_2 \end{pmatrix} = \begin{pmatrix} r_1 \\ r_2 \end{pmatrix} \equiv \begin{pmatrix} b - Ax - D_2^2 y \\ g + D_1^2 x - A^T y - z_1 + z_2 \end{pmatrix}, \tag{17}$$

where $H_1 = H + D_1^2$. We use (15) and (16) to replace two sets of vectors in (17). With

$$\begin{pmatrix} \Delta x_1 \\ \Delta x_2 \end{pmatrix} = \begin{pmatrix} -r_\ell + \Delta x \\ -r_u - \Delta x \end{pmatrix}, \qquad \begin{pmatrix} \Delta z_1 \\ \Delta z_2 \end{pmatrix} = \begin{pmatrix} X_1^{-1}(c_\ell - Z_1\Delta x_1) \\ X_2^{-1}(c_u - Z_2\Delta x_2) \end{pmatrix}, \tag{18}$$

$$\begin{aligned} H_2 &\equiv H + D_1^2 + X_1^{-1}Z_1 + X_2^{-1}Z_2 \\ w &\equiv r_2 - X_1^{-1}(c_\ell + Z_1 r_\ell) + X_2^{-1}(c_u + Z_2 r_u) \end{aligned} \tag{19}$$

we find that

$$\begin{pmatrix} -H_2 & A^T \\ A & D_2^2 \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} = \begin{pmatrix} w \\ r_1 \end{pmatrix}. \tag{20}$$

3.3 Keeping x safe

If the objective is the entropy function $\phi(x) = \sum x_j \ln(x_j)$, for example, it is essential to keep $x > 0$. However, problems CO3 and CO(μ) treat x as having no bounds (except when $x - x_1 = \ell$ and $x + x_2 = u$ are satisfied in the limit). Thus (since April 2010), PDCO safeguards x at every iteration by setting

$$\begin{aligned} x_j &= \ \ [x_1]_j \ \text{ if } \ell_j = 0 \quad (\text{and } \ell_j < u_j), \\ x_j &= -[x_2]_j \ \text{ if } u_j = 0 \quad (\text{and } \ell_j < u_j). \end{aligned}$$

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

If $\phi(x)$ is a general convex function with known Hessian H, (20) may be treated by direct or iterative solvers. Since it is an SQD system, sparse LDL$^{\mathrm{T}}$ factors should be sufficiently stable under the same conditions as for regularized LO: $\|A\| \approx 1$, $\|H\| \approx 1$, and $\gamma\delta \gg 10^{-8}$, where γ and δ are the minimum values of the diagonals of D_1 and D_2. If `K` represents the sparse matrix in (20), PDCO with `Method = 21` uses the following lines of code to achieve reasonable efficiency:

```
s          = symamd(K);    % sqd ordering (first iteration only)
rhs        = [w; r1];
thresh     = eps;          % eps ~= 2e-16 suppresses partial pivoting
[L,U,p]    = lu(K(s,s),thresh,'vector');    % expect p = I
sqdsoln    = U\(L\rhs(s));
sqdsoln(s) = sqdsoln;   dx = sqdsoln(1:n);   dy = sqdsoln(n+1:n+m);
```

The `symamd` ordering can be reused for all iterations, and with row interchanges effectively suppressed by `thresh = eps`, we expect the *original* sparse LU factorization in MATLAB to do no further permutations (`p = 1:n+m`).

With `Method = 22`, PDCO solves (20) directly using MA57:

```
thresh = 0;      % tells MA57 to keep its sparsity-preserving order
[L,D,P,S] = ldl(K,thresh);
if nnz(D)~=m+n
   error('[L,D,P,S] = ldl(K,0) gave non-diagonal D')
end
sqdsoln = S*(P*(L'\(D\(L\(P'*(S*rhs))))));   dx = ...;   dy = ...;
```

Alternatively a sparse Cholesky factorization $H_2 = LL^T$ may be practical, where $H_2 = H +$ diagonal terms (19), and L is a nonsingular permuted triangle. This is trivial if $\phi(x)$ is a separable function, since H and H_2 in (19) are then diagonal. System (20) may then be solved by eliminating either Δx or Δy:

$$(A^T D_2^{-2} A + H_2)\Delta x = A^T D_2^{-2} r_1 - w, \qquad D_2^2 \Delta y = r_1 - A\Delta x, \tag{21}$$

$$\text{or} \quad (AH_2^{-1}A^T + D_2^2)\Delta y = AH_2^{-1}w + r_1, \qquad H_2\Delta x = A^T\Delta y - w. \tag{22}$$

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 LSMR:

$$\min_{\Delta x} \left\| \begin{pmatrix} D_2^{-1}A \\ L^T \end{pmatrix} \Delta x - \begin{pmatrix} D_2^{-1} r_1 \\ -L^{-T} w \end{pmatrix} \right\|^2, \qquad D_2\Delta y = D_2^{-1}(r_1 - A\Delta x), \tag{23}$$

$$\text{or} \quad \min_{\Delta y} \left\| \begin{pmatrix} L^{-1}A^T \\ D_2 \end{pmatrix} \Delta y - \begin{pmatrix} L^{-1} w \\ D_2^{-1} r_1 \end{pmatrix} \right\|^2, \qquad L^T\Delta x = L^{-1}(A^T\Delta y - w). \tag{24}$$

The right-most vectors in (23)–(24) are part of the residual vectors for the least-squares problems (and may be by-products from the least-squares solver).

3.5 Success

PDCO has been applied to some large web-traffic network problems with the entropy function $\sum x_j \ln x_j$ as objective [30]. Search directions were obtained by applying LSQR to (24) with diagonal $H = X^{-1}$ and $L = H_2^{-1/2}$, and $D_1 = 0$, $D_2 = 10^{-3}I$. A problem with 50,000 constraints and 660,000 variables (an explicit sparse A) solves in about 3 minutes on a 2GHz PC, requiring less than 100 total LSQR iterations. At the time (2003), this was unexpectedly remarkable performance. Both the entropy function and the network matrix A are evidently amenable to interior methods.

More recently, PDCO has proved ideal for analyzing Low-Field NMR data [1, 2] to determine the contents of olive oil, biodiesel, organic waste, etc. For 1D data [1], PDCO was applied to the convex problem

$$\min_{f,\,r} \ \lambda_1\|f\|_1 + \tfrac{1}{2}\lambda_2\|f\|_2^2 + \tfrac{1}{2}\|r\|_2^2 \tag{25}$$

$$\text{s.t.} \ \ Kf + r = s, \quad f \geq 0, \tag{26}$$

where K is the discrete Laplace transform, f is the unknown spectrum vector, s is the relaxation signal with time constant T_2, and r is a residual vector that allows for noise in the signal. With K being a linear operator, LSQR was used to compute search directions for the dual variables y. Remarkably few iterations of LSQR were required (much fewer than the dimensions of K).

For 2D data [2] with time constants T_1 and T_2, the formulation is

$$\min_{f,\,r} \ \lambda_1\|f\|_1 + \tfrac{1}{2}\lambda_2\|f\|_2^2 + \tfrac{1}{2}\|r\|_2^2 \tag{27}$$

$$\text{s.t.} \ \ K_1 F K_2 + R = S, \quad F \geq 0, \tag{28}$$

where F, R, S are matrix forms of the vectors f, r, s. LSMR handles the linear operators K_1 and K_2 to obtain search directions for y. A 40×200 example gave a problem of size $m = 8000$, $n = 61952$ requiring 166 PDCO iterations, 16849 LSMR iterations (about 100 for each PDCO iteration), and about 1 minute of elapsed time in MATLAB. The sparsity of the solution f probably explains why LSMR converged so rapidly each time (to high precision). The choice of regularization parameters λ_1, λ_2 is discussed in [3].

4 Interior methods for general NLO

Reference [6] gives an overview of the theory of interior methods for *general* nonlinear optimization.

With exact second derivatives increasingly available (in particular within GAMS and AMPL), general-purpose software for large-scale nonconvex optimization with nonlinear constraints is becoming increasingly powerful and popular. Some commercial examples are IPOPT, KNITRO, LOQO, and PENOPT [15, 16, 17, 27]. To allow for general (non-diagonal) Hessians, they all use sparse direct methods on indefinite systems analogous to (20). To allow for nonconvex problems, the matrix H_2 must be modified in some way (often by the addition of multiples of I).

References

[1] P. Berman, O. Levi, Y. Parmet, M. Saunders, and Z. Wiesman. Laplace inversion of low-resolution NMR relaxometry data using sparse representation methods. *Concepts in Magnetic Resonance Part A*, 42A:3:72–88, 2013.

[2] S. Campisi-Pinto, O. Levi, D. Benson, M. Cohen, M. T. Resende, M. Saunders, C. Linder, and Z. Wiesman. Analysis of the regularization parameters of primal-dual interior method for convex objectives applied to 1H Low Field Nuclear Magnetic Resonance data processing. *Applied Magnetic Resonance*, 49:1129–1150, 2018.

[3] S. Campisi-Pinto, O. Levi, D. Benson, M. T. Resende, M. Saunders, C. Linder, and Z. Wiesman. Simulation-based sensitivity analysis of regularization parameters for robust and effective reconstruction of 2D spin-spin vs spin-lattice time domain spectra of complex materials from proton LF-NMR energy relaxation signals. *J. Applied NMR*, submitted, 2019.

[4] S. S. Chen, D. L. Donoho, and M. A. Saunders. Atomic decomposition by basis pursuit. *SIAM Review*, 43(1):129–159, 2001. SIGEST article.

[5] D. C.-L. Fong and M. Saunders. LSMR: An iterative algorithm for least-squares problems. *SIAM J. Sci. Comput.*, 33(5):2950–2971, 2011.

[6] A. Forsgren, P. E. Gill, and M. H. Wright. Interior methods for nonlinear optimization. *SIAM Review*, 44(4):525–597, 2002.

[7] 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.

[8] M. P. Friedlander and D. Orban. A primal–dual regularized interior-point method for convex quadratic programs. *Math. Prog. Comp.*, 4(1):71–107, 2012.

[9] P. E. Gill, M. A. Saunders, and J. R. Shinnerl. On the stability of Cholesky factorization for symmetric quasi-definite systems. *SIAM J. Matrix Anal. Appl.*, 17:35–46, 1996.

[10] 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.

[11] G. H. Golub and C. F. Van Loan. Unsymmetric positive definite linear systems. *Linear Algebra and its Applications*, 28:85–98, 1979.

[12] C. Greif, E. Moulding, and D. Orban. Bounds on eigenvalues of matrices arising from interior-point methods. *SIAM J. Optim.*, 24(1):49–83, 2014.

[13] B. C. Gunter, W. C. Reiley, and R. A. van de Geijn. Implementation of out-of-core Cholesky and QR factorizations with POOCLAPACK.

[14] O. Hinder. *Principled Algorithms for Finding Local Minima.* PhD thesis, Department of Management Science and Engineering, Stanford University, June 2019.

[15] IPOPT open source NLP solver. `https://projects.coin-or.org/Ipopt`.

[16] KNITRO optimization software. `https://www.artelys.com/tools/knitro`.

[17] LOQO optimization software. `http://orfe.princeton.edu/~loqo`.

[18] 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.

[19] S. Mehrotra and Y. Ye. Finding an interior point in the optimal face of linear programs. *Math. Program.*, 62:497–515, 1993.

[20] MUMPS: a multifrontal massively parallel sparse direct solver. `http://mumps.enseeiht.fr/`.

[21] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer Series in Operations Research. Springer Verlag, New York, second edition, 2006.

[22] C. C. Paige and M. A. Saunders. LSQR: An algorithm for sparse linear equations and sparse least squares. *ACM Trans. Math. Softw.*, 8(1):43–71, 1982a.

[23] C. C. Paige and M. A. Saunders. Algorithm 583; LSQR: Sparse linear equations and least-squares problems. *ACM Trans. Math. Softw.*, 8(2):195–209, 1982b.

[24] PARDISO parallel sparse solver. `http://www.pardiso-project.org`.

[25] PDCO: MATLAB convex optimization software. `http://stanford.edu/group/SOL/software/pdco/`.

[26] J. Peng, C. Roos, and T. Terlaky. *Self-Regularity: A New Paradigm for Primal-Dual Interior Point Algorithms*. Princeton University Press, Princeton, NJ, 2002.

[27] PENOPT optimization systems for nonlinear programming, bilinear matrix inequalities, and linear semidefinite programming. `http://www.penopt.com`.

[28] 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.

[29] 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).

[30] M. A. Saunders and J. A. Tomlin. Interior-point solution of large-scale entropy maximization problems. Presented at 18th International Symposium on Mathematical Programming, Copenhagen, Denmark, Aug 18–22, 2003. `http://stanford.edu/group/SOL/talks.html`.

[31] R. J. Vanderbei. Symmetric quasi-definite matrices. *SIAM J. Optim.*, 5:100–113, 1995.

[32] R. J. Vanderbei. *Linear Programming: Foundations and Extensions*. Kluwer, Boston, London and Dordrecht, second edition, 2001.

[33] S. J. Wright. Stability of linear equations solvers in interior-point methods. *SIAM J. Matrix Anal. Appl.*, 16:1287–1307, 1995.

[34] S. J. Wright. *Primal-Dual Interior-Point Methods*. SIAM, Philadelphia, 1997.

[35] S. J. Wright. Stability of augmented system factorizations in interior-point methods. *SIAM J. Matrix Anal. Appl.*, 18:191–222, 1997.

[36] WSMP: Watson Sparse Matrix Package. `https://researcher.watson.ibm.com/researcher/view_group.php?id=1426`.

[37] Y. Ye. On the finite convergence of interior-point algorithms for linear programming. *Math. Program.*, 57:325–336, 1992.