

## LP Methods.S4

### Interior Point Methods

---

All forms of the simplex method reach the optimum by traversing a series of basic solutions. Since each basic solution represents an extreme point of the feasible region, the track followed by the algorithm moves around the boundary of the feasible region. In the worst case, it may be necessary to examine most if not all of the extreme points. This can be cripplingly inefficient given that the number of extreme points grows exponentially with  $n$  for  $m$  fixed.

The running time of an algorithm as a function of the problem size is known as its *computational complexity*. In practice, the simplex method works surprisingly well, exhibiting linear complexity; i.e., proportional to  $n + m$ . Researchers, however, have long tried to develop solution algorithms whose worst-case running times are polynomial functions of the problem size. The first success was attributed to the Russian mathematician, Khachian, who proposed the *ellipsoid method* which has a running time proportional to  $n^6$  (see Bertsimas and Tsitliklis (1997) for a full discussion of the approach). Though theoretically efficient, code developers were never able to realize an implementation that matched the performance of concurrent simplex codes.

Just about the time when interest in the ellipsoid method was waning, a new technique to solve linear programs was proposed by Karmarkar (1984). His idea was to approach the optimal solution from the strict interior of the feasible region. This led to a series of *interior point methods* (IPMs) that combined the advantages of the simplex algorithm with the geometry of the ellipsoid algorithm. IPMs are of interest from a theoretical point of view because they have polynomial complexity, and from a practical point of view because they have produced solutions to many industrial problems that were hitherto intractable.

There are at least three major types of IPMs: (1) the potential reduction algorithm which most closely embodies the constructs of Karmarkar, (2) the affine scaling algorithm which is perhaps the simplest to implement, and (3) path following algorithms which arguably combine excellent behavior in theory and practice. In this section, we highlight a member of the third category known as the primal-dual path following algorithm which has become the method of choice in large scale implementations.

The primal-dual path following algorithm is an example of an IPM that operates simultaneously on the primal and dual linear programming problems. Consider the following example where the slack variables in the dual are denoted by the vector  $\mathbf{z}$  to correspond to the notation in the literature.

<i>Primal model</i>	<i>Dual model</i>
Maximize $z_P = 2x_1 + 3x_2$	Minimize $z_D = 8x_1 + 6x_2$
subject to $2x_1 + x_2 + x_3 = 8$	subject to $2x_1 + x_2 - z_1 = 2$
$x_1 + 2x_2 + x_4 = 6$	$x_1 + 2x_2 - z_2 = 3$
$x_j \geq 0, j = 1, \dots, 4$	$x_1 - z_3 = 0$
	$x_2 - z_4 = 0$
	$z_j \geq 0, j = 1, \dots, 4$

Fig. 3 and 4 shown the progress of such an algorithm starting at point #1 which is seen to be in the interior of the feasible region. In general, the algorithm iteratively modifies the primal and dual solutions until the optimality conditions of primal feasibility, dual feasibility and complementary slackness are satisfied to a sufficient degree. This event is signaled when the duality gap, the difference between the primal and dual objective functions, is sufficiently small. For purposes of discussion, “interior of the feasible region” is taken to mean that the values of the primal and dual variables are always strictly greater than zero.

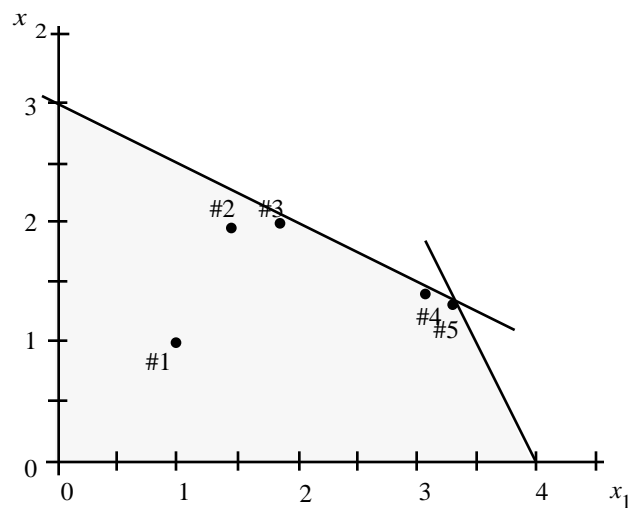


Figure 3. Path of the primal solution for the interior point algorithm

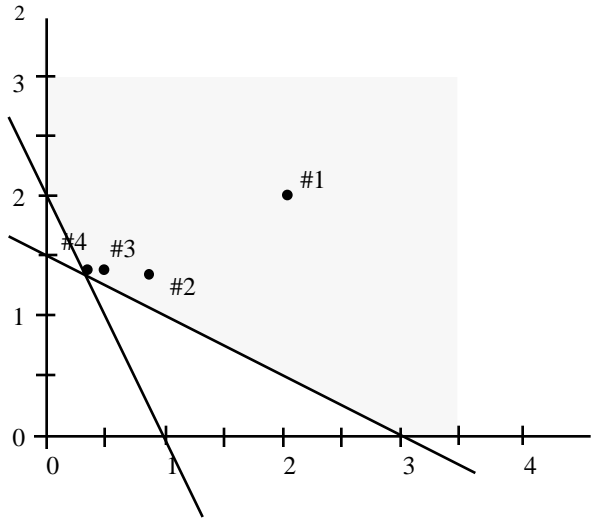


Figure 4. Path of the dual solution for the interior point algorithm

### The Primal and Dual Problems

In developing the algorithm, we will work with the primal and dual problems as defined in Table 9. The primal problem is assumed to consist of  $m$  nonredundant equations in  $n$  variables, and is given in equality form. This means that the  $n$  dual variables  $\pi$  are unrestricted in sign. The  $m$ -dimensional vector of nonnegative slack variables,  $\mathbf{z}$ , transforms the dual inequalities to equations.

Table 9. Primal and dual problems for IPM

(P) Maximize $z_P = \mathbf{c}\mathbf{x}$ subject to $\mathbf{A}\mathbf{x} = \mathbf{b}$ $\mathbf{x} \geq \mathbf{0}$	(D) Minimize $z_D = \pi\mathbf{b}$ subject to $\pi\mathbf{A} - \mathbf{z} = \mathbf{c}$
---	--

### Basic Ideas

The use of path following algorithms to solve linear programs is based on three ideas:

- (i) the application of the Lagrange multiplier method of classical calculus to transform an equality constrained optimization problem into an unconstrained one;

- (ii) the transformation of an inequality constrained optimization problem into a sequence of unconstrained problems by incorporating the constraints in a logarithmic barrier function that imposes a growing penalty as the boundary ( $x_j = 0, z_j = 0$  for all  $j$ ) is approached;
- (iii) the solution of a set of nonlinear equations using Newton's method, thereby arriving at a solution to the unconstrained optimization problem.

When solving the sequence of unconstrained problems, as the strength of the barrier function is decreased, the optimum follows a well-defined path (hence the term "path following") that ends at the optimal solution to the original problem. In subsequent sections, we outline the three components of the algorithm.

### The Lagrangian Approach

A well known procedure for determining the minimum or maximum of a function subject to equality constraints is the Lagrange multiplier method. The details are given in Chapter 11, Nonlinear Programming Methods. For now, consider the general problem

Maximize  $f(\mathbf{x})$

subject to  $g_i(\mathbf{x}) = 0, i = 1, \dots, m$

where  $f(\mathbf{x})$  and  $g_i(\mathbf{x})$  are scalar functions of the  $n$ -dimensional vector  $\mathbf{x}$ .

The Lagrangian for this problem is

$$L(\mathbf{x}, \boldsymbol{\pi}) = f(\mathbf{x}) - \sum_{i=1}^m \pi_i g_i(\mathbf{x})$$

where the variables  $\boldsymbol{\pi} = (\pi_1, \pi_2, \dots, \pi_m)$  are the Lagrange multipliers.

Necessary conditions for a stationary point (maximum or minimum) of the constrained optimization of  $f(\mathbf{x})$  are that the partial derivatives of the Lagrangian with respect to the components of  $\mathbf{x}$  and  $\boldsymbol{\pi}$  be zero; i.e.,

$$\frac{\partial L}{\partial x_j} = 0, j = 1, \dots, n \quad \text{and} \quad \frac{\partial L}{\partial \pi_i} = 0, i = 1, \dots, m$$

For linear constraints ( $\mathbf{a}_i \mathbf{x} - b_i = 0$ ), the conditions are sufficient for a maximum if the function  $f(\mathbf{x})$  is concave and sufficient for a minimum if  $f(\mathbf{x})$  is convex.

### The Barrier Approach

For the primal-dual pair of LPs in Table 9, the only essential inequalities are the nonnegativity conditions. The idea of the barrier approach, as developed by Fiacco and McCormick (1968), is to start from a point in the strict interior of the inequalities ( $x_j > 0, z_j > 0$  for all  $j$ ) and construct a barrier that prevents any variable from reaching a boundary (e.g.,  $x_j = 0$ ). Adding “ $\log(x_j)$ ” to the objective function of the primal, for example, will cause the objective to decrease without bound as  $x_j$  approaches 0. The difficulty with this idea is that if the constrained optimum is on the boundary (that is, one or more  $x_j^* = 0$ , which is always the case in linear programming), then the barrier will prevent us from reaching it. To get around this difficulty, we use a barrier parameter  $\mu$  that balances the contribution of the true objective function with that of the barrier term. The modified problems are given in Table 10.

Table 10. Primal and dual barrier problems

<p>(P) Maximize <math>B_P(\mu) = \mathbf{c}\mathbf{x} + \mu \sum_{j=1}^n \log(x_j)</math></p> <p>subject to <math>\mathbf{A}\mathbf{x} = \mathbf{b}</math></p>	<p>(D) Minimize <math>B_D(\mu) = \boldsymbol{\pi}\mathbf{b} - \mu \sum_{j=1}^n \log(z_j)</math></p> <p>subject to <math>\boldsymbol{\pi}\mathbf{A} - \mathbf{z} = \mathbf{c}</math></p>
--	---

The parameter  $\mu$  is required to be positive and controls the magnitude of the barrier term. Because function  $\log(x)$  takes on very large negative values as  $x$  approaches zero from above, as long as  $\mathbf{x}$  remains positive the optimal solution to the barrier problems will be interior to the nonnegative orthants ( $x_j > 0$  and  $z_j > 0$  for all  $j$ ). The barrier term is added to the objective function for a maximization problem and subtracted for a minimization problem. The new formulations have nonlinear objective functions with linear equality constraints, and can be solved with the Lagrangian technique for  $\mu > 0$  fixed. The solution to these problems will approach the solution to the original problem as  $\mu$  approaches zero.

Table 11 shows the development of the necessary optimal conditions for the barrier problems. These conditions are also sufficient because the primal Lagrangian is concave and the dual Lagrangian is convex. Note that the dual variables  $\boldsymbol{\pi}$  are the Lagrange multipliers of the primal, and the primal variables  $\mathbf{x}$  are the Lagrange multipliers of the dual.

Table 11. Necessary conditions for the barrier problems

<p><i>Lagrangian</i></p> $L_P = \mathbf{c}\mathbf{x} + \mu \sum_{j=1}^n \log(x_j) - \pi(\mathbf{A}\mathbf{x} - \mathbf{b})$	<p><i>Lagrangian</i></p> $L_D = \pi\mathbf{b} - \mu \sum_{j=1}^n \log(z_j) - (\pi\mathbf{A} - \mathbf{z} - \mathbf{c})\mathbf{x}$
$\frac{L_P}{x_j} = 0$ $c_j - \sum_{i=1}^m a_{ij} \pi_i + \frac{\mu}{x_j} = 0$ $-\pi_j + \frac{\mu}{x_j} = 0$ $z_j x_j = \mu, \quad j = 1, \dots, n$ <p>(<math>\mu</math>-complementary slackness)</p>	$\frac{L_D}{z_j} = 0$ $-\frac{\mu}{z_j} + \pi_j = 0$ $z_j x_j = \mu, \quad j = 1, \dots, n$ <p>(<math>\mu</math>-complementary slackness)</p>
$\frac{L_P}{i} = 0$ $\sum_{j=1}^n a_{ij} x_j = b_i, \quad i = 1, \dots, m$ <p>(primal feasibility)</p>	$\frac{L_D}{i} = 0$ $\sum_{j=1}^n a_{ij} x_j = b_i, \quad i = 1, \dots, m$ <p>(primal feasibility)</p>
	$\frac{L_D}{x_j} = 0$ $\sum_{i=1}^m a_{ij} \pi_i - z_j = c_j, \quad j = 1, \dots, n$ <p>(dual feasibility)</p>

Thus the optimal conditions are nothing more than primal feasibility, dual feasibility, and complementary slackness satisfied to within a tolerance of

$\mu$ . Theory tells us that when  $\mu$  goes to zero we get the solution to the original problem; however, we can't just set  $\mu$  to zero because that would destroy the convergence properties of the algorithm.

To facilitate the presentation, we define two  $n \times n$  diagonal matrices containing the components of  $\mathbf{x}$  and  $\mathbf{z}$ , respectively; i.e.,

$$\mathbf{X} = \text{diag}\{x_1, x_2, \dots, x_n\}$$

$$\mathbf{Z} = \text{diag}\{z_1, z_2, \dots, z_n\}$$

Also, let  $\mathbf{e} = (1, 1, \dots, 1)^T$  be a column vector of size  $n$ . Using this notation, the necessary and sufficient conditions derived in Table 11 for the simultaneous solution of both the primal and dual barrier problems can be written as

$$\text{Primal feasibility: } \mathbf{A}\mathbf{x} - \mathbf{b} = \mathbf{0} \quad (m \text{ linear equations})$$

$$\text{Dual feasibility: } \mathbf{A}^T \boldsymbol{\pi}^T - \mathbf{z} - \mathbf{c}^T = \mathbf{0} \quad (n \text{ linear equations})$$

$$\mu\text{-Complementary slackness: } \mathbf{X}\mathbf{Z}\mathbf{e} - \mu\mathbf{e} = \mathbf{0} \quad (n \text{ nonlinear equations})$$

We must now solve this set of nonlinear equations for the variables  $(\mathbf{x}, \boldsymbol{\pi}, \mathbf{z})$ .

### Stationary Solutions Using Newton's Method

Newton's method is an iterative procedure for numerically solving a set of nonlinear equations. To motivate the discussion, consider the single variable problem of finding  $y$  to satisfy the nonlinear equation

$$f(y) = 0$$

where  $f$  is once continuously differentiable. Let  $y^*$  be the unknown solution. At some point  $y^k$ , one can calculate the functional value,  $f(y^k)$ , and the first derivative,  $f'(y^k)$ . Using the derivative as a first order approximation for how the function changes with  $y$ , we can predict the amount of change  $= y^{k+1} - y^k$  required to bring the function to zero.

This idea is illustrated in Fig. 5.

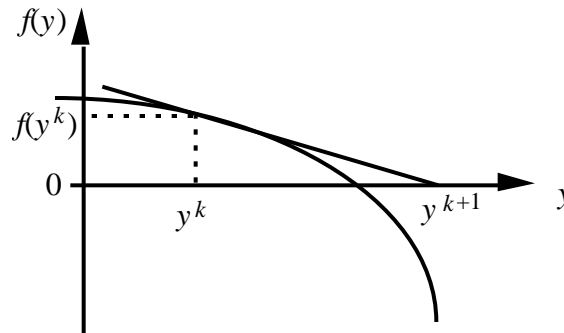


Figure 5. Newton's method for function of a single variable

Taking the first order Taylor series expansion of  $f(y)$  around  $y^k$  gives

$$f(y^{k+1}) \approx f(y^k) + f'(y^k)(y^{k+1} - y^k)$$

Setting the approximation of  $f(y^{k+1})$  to zero and solving for  $y^{k+1}$  gives

$$y^{k+1} = y^k - \frac{f(y^k)}{f'(y^k)}$$

The point  $y^{k+1} = y^k - \frac{f(y^k)}{f'(y^k)}$  is an approximate solution to the equation. It can be shown that if one starts at a point  $y^0$  sufficiently close to  $y^*$ , the value of  $y^k$  will approach  $y^*$  as  $k \rightarrow \infty$ .

The method extends to multidimensional functions. Consider the general problem of finding the  $r$ -dimensional vector  $\mathbf{y}$  that solves the set of  $r$  equations

$$f_i(\mathbf{y}) = 0, \quad i = 1, \dots, r$$

$$\text{or } \mathbf{f}(\mathbf{y}) = \mathbf{0}$$

Let the unknown solution to the equations be  $\mathbf{y}^*$ . The  $n \times n$  Jacobian matrix describes the first order variations of these functions with the components of  $\mathbf{y}$ . The Jacobian at  $\mathbf{y}^k$  is



$$\mathbf{J}(\mathbf{y}^k) = \begin{array}{cccc} \frac{f_1}{y_1} & \frac{f_1}{y_2} & \cdots & \frac{f_1}{y_n} \\ \frac{f_2}{y_1} & \frac{f_2}{y_2} & \cdots & \frac{f_2}{y_n} \\ \vdots & \vdots & & \vdots \\ \frac{f_n}{y_1} & \frac{f_n}{y_2} & \cdots & \frac{f_n}{y_n} \end{array}$$

All the partial derivatives are evaluated at  $\mathbf{y}^k$ . Now, taking the first order Taylor series expansion around the point  $\mathbf{y}^k$ , and setting it to zero gives

$$\mathbf{f}(\mathbf{y}^k) + \mathbf{J}(\mathbf{y}^k)\mathbf{d} = \mathbf{0}$$

where  $\mathbf{d} = \mathbf{y}^{k+1} - \mathbf{y}^k$  is an  $n$ -dimensional vector whose components represent the change of position for the  $k+1^{\text{st}}$  iteration. Solving for  $\mathbf{d}$  leads to

$$\mathbf{d} = -\mathbf{J}(\mathbf{y}^k)^{-1}\mathbf{f}(\mathbf{y}^k)$$

The point  $\mathbf{y}^{k+1} = \mathbf{y}^k + \mathbf{d}$  is an approximation for the solution to the set of equations. Once again, if one starts at an initial point  $\mathbf{y}^0$  sufficiently close to  $\mathbf{y}^*$ , the value of  $\mathbf{y}^k$  will approach  $\mathbf{y}^*$  for large values of  $k$ .

### Newton's Method for Solving Barrier Problems

We are now ready to use Newton's method to solve the optimality conditions for the barrier problems given in Table 11 for a fixed value of  $\mu$ . For  $\mathbf{y} = (\mathbf{x}, \boldsymbol{\pi}, \mathbf{z})$  and  $r = 2n+m$ , the corresponding equations and Jacobian are

$$\mathbf{Ax} - \mathbf{b} = \mathbf{0}$$

$$\mathbf{A}^T \boldsymbol{\pi}^T - \mathbf{z} - \mathbf{c}^T = \mathbf{0}$$

$$\mathbf{XZ}\mathbf{e} - \mu\mathbf{e} = \mathbf{0}$$

$$\mathbf{A} \quad \mathbf{0} \quad \mathbf{0}$$

$$\mathbf{J}(\mathbf{y}) = \begin{array}{ccc} \mathbf{0} & \mathbf{A}^T & -\mathbf{I} \\ \mathbf{Z} & \mathbf{0} & \mathbf{X} \end{array}$$

$$\mathbf{Z} \quad \mathbf{0} \quad \mathbf{X}$$

Assume that we have a starting point  $(\mathbf{x}^0, \boldsymbol{\pi}^0, \mathbf{z}^0)$  satisfying  $\mathbf{x}^0 > \mathbf{0}$  and  $\mathbf{z}^0 > \mathbf{0}$ , and denote by

$$\delta_P = \mathbf{b} - \mathbf{Ax}^0$$

$$\delta_D = \mathbf{c}^T - \mathbf{A}^T(\boldsymbol{\pi}^0)^T + \mathbf{z}^0$$

the primal and dual residual vectors at this starting point. The optimality conditions can be written as

$$\mathbf{J}(\mathbf{y})\mathbf{d} = -\mathbf{f}(\mathbf{y})$$

$$\begin{array}{ccc} \mathbf{A} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}^T & -\mathbf{I} \\ \mathbf{Z} & \mathbf{0} & \mathbf{X} \end{array} \begin{array}{c} \mathbf{d}_x \\ \mathbf{d} \\ \mathbf{d}_z \end{array} = \begin{array}{c} \delta_p \\ \delta_D \\ \mu\mathbf{e} - \mathbf{XZ}\mathbf{e} \end{array} \quad (7)$$

where the  $(2n+m)$ -dimensional vector  $\mathbf{d} = (\mathbf{d}_x, \mathbf{d}, \mathbf{d}_z)^T$  in (7) is called the Newton direction. We must now solve for the individual components of  $\mathbf{d}$ .

In explicit form, the above system is

$$\begin{aligned} \mathbf{A}\mathbf{d}_x &= \delta_p \\ \mathbf{A}^T\mathbf{d} - \mathbf{d}_z &= \delta_D \\ \mathbf{Z}\mathbf{d}_x + \mathbf{X}\mathbf{d}_z &= \mu\mathbf{e} - \mathbf{XZ}\mathbf{e} \end{aligned}$$

The first step is to find  $\mathbf{d}$ . With some algebraic manipulation, left as an exercise, we get

$$(\mathbf{AZ}^{-1}\mathbf{XA}^T)\mathbf{d} = -\mathbf{b} + \mu\mathbf{AZ}^{-1}\mathbf{e} + \mathbf{AZ}^{-1}\mathbf{X}\delta_D$$

$$\text{or } \mathbf{d} = (\mathbf{AZ}^{-1}\mathbf{XA}^T)^{-1}(-\mathbf{b} + \mu\mathbf{AZ}^{-1}\mathbf{e} + \mathbf{AZ}^{-1}\mathbf{X}\delta_D) \quad (8)$$

Note that  $\mathbf{Z}^{-1} = \text{diag}\{1/z_1, 1/z_2, \dots, 1/z_n\}$  and is trivial to compute. Further multiplications and substitutions give

$$\mathbf{d}_z = -\delta_D + \mathbf{A}^T\mathbf{d} \quad (9)$$

$$\text{and } \mathbf{d}_x = \mathbf{Z}^{-1}(\mu\mathbf{e} - \mathbf{XZ}\mathbf{e} - \mathbf{X}\mathbf{d}_z) \quad (10)$$

From these results, we can see in part why it is necessary to remain in the interior of the feasible region. In particular, if either  $\mathbf{Z}^{-1}$  or  $\mathbf{X}^{-1}$  does not exist the procedure breaks down.

Once the Newton direction has been computed,  $\mathbf{d}_x$  is used as a search direction in the  $x$ -space and  $(\mathbf{d}, \mathbf{d}_z)$  as a search direction in the  $(\pi, z)$ -space. That is, we move from the current point  $(\mathbf{x}^0, \pi^0, \mathbf{z}^0)$  to a new point  $(\mathbf{x}^1, \pi^1, \mathbf{z}^1)$  by taking a step in the direction  $(\mathbf{d}_x, \mathbf{d}, \mathbf{d}_z)$ . The step

sizes,  $\alpha_P$  and  $\alpha_D$ , are chosen in the two spaces to preserve  $\mathbf{x} > \mathbf{0}$  and  $\boldsymbol{\pi} > \mathbf{0}$ . This requires a ratio test similar to that performed in the simplex algorithm. The simplest approach is to use

$$\alpha_P = \min_j \frac{-x_j}{(\mathbf{d}_x)_j} : (\mathbf{d}_x)_j < 0$$

$$\alpha_D = \min_j \frac{-z_j}{(\mathbf{d}_z)_j} : (\mathbf{d}_z)_j < 0$$

where  $\alpha$  is the step size factor that keeps us from actually touching the boundary. Typically,  $\alpha = 0.995$ . The notation  $(\mathbf{d}_x)_j$  refers to the  $j$ th component of the vector  $\mathbf{d}_x$ . The new point is

$$\mathbf{x}^1 = \mathbf{x}^0 + \alpha_P \mathbf{d}_x$$

$$\boldsymbol{\pi}^1 = \boldsymbol{\pi}^0 + \alpha_D \mathbf{d}$$

$$\mathbf{z}^1 = \mathbf{z}^0 + \alpha_D \mathbf{d}_z$$

which completes one iteration. Ordinarily, we would now resolve Eqs. (8) - (10) at  $(\mathbf{x}^1, \boldsymbol{\pi}^1, \mathbf{z}^1)$  to find a new Newton direction and hence a new point. Rather than iterating in this manner until the system converged for the current value of  $\mu$ , it is much more efficient to reduce  $\mu$  at every iteration. The primal-dual method itself suggests how to update  $\mu$ .

It is straightforward to show that the Newton step reduces the duality gap — the difference between the dual and primal objective values at the current point. Assume that  $\mathbf{x}^0$  is primal feasible and  $(\boldsymbol{\pi}^0, \mathbf{z}^0)$  is dual feasible (Lustig et. al (1989) discuss the more general case). Let “gap(0)” denote the current duality gap.

$$\begin{aligned} \text{gap}(0) &= \boldsymbol{\pi}^0 \mathbf{b} - \mathbf{c} \mathbf{x}^0 \\ &= \boldsymbol{\pi}^0 (\mathbf{A} \mathbf{x}^0) - (\boldsymbol{\pi}^0 \mathbf{A} - \mathbf{z}^0)^T \mathbf{x}^0 \quad (\text{primal and dual} \\ &\quad \text{feasibility}) \\ &= (\mathbf{z}^0)^T \mathbf{x}^0 \end{aligned}$$

If we let  $\alpha = \min\{\alpha_P, \alpha_D\}$  then

$$\text{gap}(\alpha) = (\mathbf{z}^0 + \alpha \mathbf{d}_z)^T (\mathbf{x}^0 + \alpha \mathbf{d}_x)$$

and with a little algebra, it can be shown that  $\text{gap}(\mu) < \text{gap}(0)$  as long as

$$\mu < \frac{\text{gap}(0)}{n} \quad (11)$$

In our computations we have used

$$\mu^k = \frac{\text{gap}(\alpha^k)}{n^2} = \frac{(\mathbf{z}^k)^T \mathbf{x}^k}{n^2}$$

which indicates that the value of  $\mu^k$  is proportional to the duality gap.

Termination occurs when the gap is sufficiently small, say, less than  $10^{-8}$ .

### Iterative Primal-Dual Interior Point Algorithm

We now summarize the basic steps of the algorithm. The following inputs are assumed:

- (i) the data of the problem  $(\mathbf{A}, \mathbf{b}, \mathbf{c})$ , where the  $m \times n$  matrix  $\mathbf{A}$  has full row rank;
- (ii) initial primal and dual feasible solutions  $\mathbf{x}^0 > \mathbf{0}$ ,  $\mathbf{z}^0 > \mathbf{0}$ ,  $\pi^0$ ;
- (iii) the optimality tolerance  $\epsilon > 0$  and the step size parameter  $\alpha \in (0, 1)$ .

*Step 1 (Initialization)* Start with some feasible point  $\mathbf{x}^0 > \mathbf{0}$ ,  $\mathbf{z}^0 > \mathbf{0}$ ,  $\pi^0$  and set the iteration counter  $k = 0$ .

*Step 2 (Optimality test)* If  $(\mathbf{z}^k)^T \mathbf{x}^k < \epsilon$  stop; otherwise, go to Step 3.

*Step 3 (Compute Newton directions)* let

$$\mathbf{X}^k = \text{diag } x_1^k, x_2^k, \dots, x_n^k$$

$$\mathbf{Z}^k = \text{diag } z_1^k, z_2^k, \dots, z_n^k$$

$$\mu^k = \frac{(\mathbf{z}^k)^T \mathbf{x}^k}{n}$$

Solve the following linear system equivalent to (7) to get  $\mathbf{d}_x^k$ ,  $\mathbf{d}_z^k$ , and  $\mathbf{d}_\pi^k$ .

$$\mathbf{A} \mathbf{d}_x = \mathbf{0}$$

$$\mathbf{A}^T \mathbf{d} - \mathbf{d}_z = \mathbf{0}$$

$$\mathbf{Z} \mathbf{d}_x + \mathbf{X} \mathbf{d}_z = \mu \mathbf{e} - \mathbf{X} \mathbf{Z} \mathbf{e}$$

Note that  $\delta_p = \mathbf{0}$  and  $\delta_D = \mathbf{0}$  due to the feasibility of the initial point.

Step 4 (Find step lengths) Let

$$p = \min_j \frac{-x_j^k}{(\mathbf{d}_x^k)_j} : (\mathbf{d}_x^k)_j < 0$$

$$D = \min_j \frac{-z_j^k}{(\mathbf{d}_z^k)_j} : (\mathbf{d}_z^k)_j < 0$$

Step 5 (Update solution) Take a step in the Newton direction to get

$$\mathbf{x}^{k+1} = \mathbf{x}^k + p \mathbf{d}_x^k$$

$$\pi^{k+1} = \pi^k + D \mathbf{d}^k$$

$$\mathbf{z}^{k+1} = \mathbf{z}^k + D \mathbf{d}_z^k$$

Put  $k = k + 1$  and go to Step 2.

## Implementation Issues

From a theoretical point of view, it has been shown that for a slightly different choice of  $\mu^k$  and the step length, the algorithm takes about

$$\sqrt{n} \log\left(\frac{\epsilon}{\epsilon_0}\right)$$

iterations to reduce the duality gap from  $\epsilon_0$  to  $\epsilon$  in the worst case. The observed average behavior is closer to  $O(\log(n) \log(\epsilon_0/\epsilon))$ . In fact, extensive testing indicates that IPMs are surprisingly insensitive to problem size. Convergence usually takes between 20 and 80 iterations. Most of the work is at Step 2 where the system of linear equations has to be solved. The most

computationally intensive component of this step involves forming the matrix  $(\mathbf{AZ}^{-1}\mathbf{XA}^T)$  and then inverting it to get  $\mathbf{d}$  as indicated in Eq. (8). This consumes about 90% of the overall effort. It should be mentioned that one iteration of an IPM is much more time consuming than one iteration of a simplex-type algorithm as measured by a pivot.

Virtually all interior point methods, whether path following, affine scaling, or potential reduction, require the solution of a linear system similar to that in Step 2. Much research has gone into figuring out how to do this efficiently. In today's codes, the matrix  $\mathbf{AZ}^{-1}\mathbf{XA}^T$  is never inverted explicitly; Cholesky factorization and back substitution are used to solve a series of related triangular linear systems. Specifically, this involves finding the lower triangular  $n \times n$  matrix  $\mathbf{L}$  such that  $\mathbf{AZ}^{-1}\mathbf{XA}^T = \mathbf{LL}^T$  and then solving

$$\mathbf{LL}^T\mathbf{d} = \mathbf{r}$$

where  $\mathbf{r}$  is the right-hand side of Eq. (8). This can be done by first solving  $\mathbf{Lg} = \mathbf{r}$  for  $\mathbf{g}$  and then solving  $\mathbf{L}^T\mathbf{d} = \mathbf{g}$  for  $\mathbf{d}$ , both by simple substitution. This approach can be implemented in a highly efficient manner when  $\mathbf{AZ}^{-1}\mathbf{XA}^T$  is a large sparse matrix. Although the matrix changes from iteration to iteration, the pattern of zeros remains the same implying that the calculations can be streamed by eliminating almost all multiplications by zero.

In the presentation of the path following algorithm, it was assumed that an initial feasible point was available. Surprisingly, this need not be the case for the primal-dual approach. The algorithm can be started at any interior point with the values of  $\delta_p$  and  $\delta_D$  in Step 2 set appropriately. Other IPMs, including the primal (only) path following method, require the equivalent of a phase 1 step to find a feasible point. Rather than explicitly solving a phase 1 problem, though, they generally take a big- $M$  approach that involves the introduction of one artificial variable,  $x_{n+1}$ , in the primal problem.

To conclude this section, we offer a few comments on the relative performance of simplex-based methods versus interior point methods. Although the determination of which is better depends on the problem and the particular instance, some qualitative insights are available. In particular, simplex methods tend to bog down when solving large, massively degenerate problems. Such problems typically arise in transportation and scheduling applications that have network models at their core. The accompanying formulations are often very sparse thus making them prime candidates for IPMs.

When the formulations are dense, or in some cases when only a few columns of the  $\mathbf{A}$  matrix are dense, IPMs are not likely to perform well. Such problems give rise to dense Cholesky factors,  $\mathbf{L}$ , which nullify

the efficiencies associated with solving triangular systems. Dense problems, however, are not in general degenerate so simplex-type methods are the better choice.

**Example**

As an example, consider the problem that started this section. In general terms we repeat the primal and dual problems from Table 9.

(P) Maximize $z_P = \mathbf{c}\mathbf{x}$ subject to $\mathbf{A}\mathbf{x} = \mathbf{b}$ $\mathbf{x} \geq \mathbf{0}$	(D) Minimize $z_D = \boldsymbol{\pi}\mathbf{b}$ subject to $\boldsymbol{\pi}\mathbf{A} - \mathbf{z} = \mathbf{c}$
---	--

For the specific case of the example we have  $n = 4$  and  $m = 2$ .

$$\mathbf{c} = [2, 3, 0, 0], \mathbf{c} = [2, 3, 0, 0] \mathbf{A} = \begin{matrix} 2 & 1 & 1 & 0 \\ 1 & 2 & 0 & 1 \end{matrix}, \mathbf{b} = \begin{matrix} 8 \\ 6 \end{matrix}$$

$$\mathbf{x} = \begin{matrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{matrix}, \mathbf{z} = \begin{matrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{matrix}, \boldsymbol{\pi} = [\pi_1, \pi_2]$$

For the initial solution we select the interior solutions shown in Table 12<sup>1</sup>. The primal feasibility vector is  $\delta_P$  and the dual feasibility vector is  $\delta_D$ . Both are zero since the  $\mathbf{x}$  and  $\mathbf{z}$  are interior to their respective problems. The vector labeled “Comp. Slack.” is the  $\mu$ -Complementary slackness vector computed as

$$\mathbf{XZ}\mathbf{e} - \mu\mathbf{e}.$$

Since this vector is not zero, the solution is not optimum.

Table 12. Initial solution for the example

<sup>1</sup> The tables in the section were constructed by the Interior Point demonstration in the Teach LP add-in.

Current Solution		Pi		
	X	Z	1	2
1	1	4	2	2
2	1	3		
3	5	2		
4	3	2		
		Iter.:	1	
		Gap	23	
		Mu	1.4375	
Feasibility Vectors				
	Primal	Dual	Comp. Slack.	
1	0	0	1	-2.563
2	0	0	2	-1.563
		3	3	-8.563
		4	4	-4.563

We solve the set of equations in step 3 of the algorithm to find the direction vectors shown in Table 13. The first three columns refer to the directions:  $\mathbf{d}_\pi$ ,  $\mathbf{d}_z$ , and  $\mathbf{d}_x$ . The last two columns are the ratio vectors that determine the maximum flow change in the  $\mathbf{z}$  and  $\mathbf{x}$  vectors to remain interior to the region. We use  $\alpha = 0.8$ , so the example does not move too close to the boundaries of the region. This factor multiplies the minimum of each ratio to find the step size for the dual and primal variables. The cell labeled Z Step is the value of  $\alpha \cdot \min(\text{Ratio}(Z))$  and the cell labeled X Step is  $\alpha \cdot \min(\text{Ratio}(X))$ .

Table 13 The direction and ratio vectors for the initial solution

Direction Vectors			Ratio Vectors			
	d(Pi)	d(Z)	d(X)	Ratio(Z)	Ratio(X)	
1	-1.351	-3.477	0.2286	1.1505	- - -	
2	-0.774	-2.9	0.4457	1.0346	- - -	
		3	-0.903	1.48	5.5378	
		4	-1.12	2.5835	2.6785	
			Min	1.0346	Min	2.6785
			Z Step	0.8277	X Step	2.1428

Table 14 shows next solution obtained by multiplying  $\mathbf{d}_\pi$  and  $\mathbf{d}_z$  by  $\alpha$ , and  $\mathbf{d}_x$  by  $\beta$  and adding the change to the vectors  $\pi$ ,  $\mathbf{z}$  and  $\mathbf{x}$ . The primal and dual solutions remain interior and the gap is reduced.

Table 14. The solution at the 2<sup>nd</sup> iteration



Current Solution		Pi	
	X	Z	1 2
1	1.4898	1.1223	0.8815 1.3592
2	1.9551	0.6	
3	3.0653	0.8815	
4	0.6	1.3592	
		Iter.:	2
		Gap	6.3626
		Mu	0.3977
Feasibility Vectors			
	Primal	Dual	Comp. Slack.
1	-2E-15	4E-16	-1.274
2	-2E-15	3E-16	-0.775
3		0	-2.304
4		0	-0.418

Table 15 shows the nine iterations necessary to obtain a gap of 0.001. Observe that  $z_p$  increases from its initial value while  $z_D$  decreases from its initial value. The converge to the same value at the optimal. The optimal solution is at intersection of the two constraints, where the slack variables  $x_3$  and  $x_4$  are both zero. The interior method never really reaches the intersection, but it comes as close as desired.

Table 15. The sequence of solutions

Iteration	Primal Solution					Dual Solution		
	$z_p$	$x_1$	$x_2$	$x_3$	$x_4$	$z_D$	$\pi_1$	$\pi_2$
1	5	1	1	5	3	28	2	2
2	8.845	1.4898	1.9551	3.0653	0.6	15.208	0.8815	1.3592
3	9.764	1.8872	1.9964	2.2292	0.12	11.761	0.3964	1.4317
4	10.419	3.0608	1.4325	0.4458	0.0741	11.025	0.3111	1.4228
5	10.598	3.2836	1.3437	0.0892	0.029	10.744	0.3276	1.3538
6	10.651	3.3201	1.337	0.0227	0.0058	10.685	0.3328	1.3371
7	10.664	3.3306	1.3341	0.0046	0.0012	10.67	0.3332	1.3341
8	10.666	3.3328	1.3335	0.0009	0.0002	10.667	0.3333	1.3335
9	10.667	3.3332	1.3334	0.0002	5E-05	10.667	0.3333	1.3334