Appendix A
Mathematical Review

The purpose of this appendix is to set down for reference and review some basic definitions, notation, and relations that are used frequently in the text.

A.1 Sets

If x is a member of the set S, we write $x \in S$. We write $y \notin S$ if y is not a member of S.

A set S may be specified by listing its elements between braces; such as, for example, $S = \{1, 2, 3, 4\}$. Alternatively, a set can be specified in the form $S = \{x : \boldsymbol{P}(x)\}$ as the set of elements satisfying property P; such as $S = \{x : 1 \leqslant x \leqslant 4,\ x \text{ integer}\}$

The *union* of two sets S and T is denoted $S \cup T$ and is the set consisting of the elements that belong to either S or T. The *intersection* of two sets S and T is denoted $S \cap T$ and is the set consisting of the elements that belong to both S and T. If S is a *subset* of T, that is, if every member of S is also a member of T, we write $S \subset T$ or $T \supset S$.

The empty set is denoted ϕ or $\emptyset$. There are two ways that operations such as minimization over a set are represented. Specifically we write either

$$\min_{x \in S} f(x) \text{ or } \min\{f(x) : x \in S\}$$

to denote the minimum value of f over the set S. The set of x's in S that achieve the minimum is denoted argmin $\{f(x) : x \in S\}$.

© Springer International Publishing Switzerland 2016

D.G. Luenberger, Y. Ye, *Linear and Nonlinear Programming*, International Series in Operations Research & Management Science 228,
DOI 10.1007/978-3-319-18842-3

Sets of Real Numbers

If a and b are real numbers, $[a,\ b]$ denotes the set of real numbers x satisfying $a \leqslant x \leqslant b$. A rounded, instead of square, bracket denotes strict inequality in the definition. Thus $(a,\ b]$ denotes all x satisfying $a < x \leqslant b$.

If S is a set of real numbers bounded above, then there is a smallest real number y such that $x \leqslant y$ for all $x \in S$. The number y is called the *least upper bound* or *supremum* of S and is denoted

$$\sup_{x \in S}(x) \text{ or } \sup\{x : x \in S\}.$$

Similarly, the *greatest lower bound* or *infimum* of a set S is denoted

$$\inf_{x \in S}(x) \text{ or } \inf\{x : x \in S\}.$$

A.2 Matrix Notation

A *matrix* is a rectangular array of numbers, called *elements*. The matrix itself is denoted by a boldface letter. When specific numbers are not used, the elements are denoted by italicized lower-case letters, having a double subscript. Thus we write

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & & & \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$

for a matrix $\mathbf{A}$ having m rows and n columns. Such a matrix is referred to as an $m \times n$ matrix. If we wish to specify a matrix by defining a general element, we use the notation $\mathbf{A} = [a_{ij}]$.

An $m \times n$ matrix all of whose elements are zero is called a *zero matrix* and denoted $\mathbf{0}$. A *square* matrix (a matrix with $m = n$) whose elements are $a_{ij} = 0$ for $i \neq j$, and $a_{ii} = 1$ for $i = 1, 2,\ \ldots,\ n$ is said to be an *identity matrix* and denoted $\mathbf{I}$.

The *sum* of two $m \times n$ matrices $\mathbf{A}$ and $\mathbf{B}$ is written $\mathbf{A} + \mathbf{B}$ and is the matrix whose elements are the sum of the corresponding elements in $\mathbf{A}$ and $\mathbf{B}$. The *product* of a matrix $\mathbf{A}$ and a scalar λ, written $\lambda\mathbf{A}$ or $\mathbf{A}\lambda$, is obtained by multiplying each element of $\mathbf{A}$ by λ. The *product* $\mathbf{AB}$ of an $m \times n$ matrix $\mathbf{A}$ and an $n \times p$ matrix $\mathbf{B}$ is the $m \times p$ matrix $\mathbf{C}$ with elements $c_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj}$.

The *transpose* of an $m \times n$ matrix $\mathbf{A}$ is the $n \times m$ matrix $\mathbf{A}^T$ with elements $a_{ij}^T = a_{ji}$. A (square) matrix $\mathbf{A}$ is *symmetric* if $\mathbf{A}^T = \mathbf{A}$. A square matrix $\mathbf{A}$ is *nonsingular* if there is a matrix $\mathbf{A}^{-1}$, called the *inverse* of $\mathbf{A}$, such that $\mathbf{A}^{-1}\mathbf{A} = \mathbf{I} = \mathbf{A}\mathbf{A}^{-1}$. The *determinant* of a square matrix $\mathbf{A}$ is denoted by det $(\mathbf{A})$. The determinant is nonzero if and only if the matrix is nonsingular. Two square $n \times n$ matrices $\mathbf{A}$ and $\mathbf{B}$ are *similar* if there is a nonsingular matrix $\mathbf{S}$ such that $\mathbf{B} = \mathbf{S}^{-1}\mathbf{AS}$.

Matrices having a single row are referred to as *row vectors*; matrices having a single column are referred to as *column vectors. Vectors* of either type are usually denoted by lower-case boldface letters. To economize page space, row vectors are written $\mathbf{a} = [a_1, a_2, \ldots, a_n]$ and column vectors are written $\mathbf{a} = (a_1, a_2, \ldots, a_n)$. Since column vectors are used frequently, this notation avoids the necessity to display numerous columns. To further distinguish rows from columns, we write $\mathbf{a} \in E^n$ if $\mathbf{a}$ is a column vector with n components, and we write $\mathbf{b} \in E_n$ if $\mathbf{b}$ is a row vector with n components.

It is often convenient to partition a matrix into submatrices. This is indicated by drawing partitioning lines through the matrix, as for example,

$$\mathbf{A} = \left[\begin{array}{cc|cc} a_{11} & a_{12} & a_{13} & a_{14} \\ \hline a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \end{array}\right] = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix}.$$

The resulting submatrices are usually denoted $\mathbf{A}_{ij}$, as illustrated.

A matrix can be partitioned into either column or row vectors, in which case a special notation is convenient. Denoting the columns of an $m \times n$ matrix $\mathbf{A}$ by $\mathbf{a}_j$, $j = 1, 2, \ldots, n$, we write $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n]$. Similarly, denoting the rows of $\mathbf{A}$ by $\mathbf{a}^i$, $i = 1, 2, \ldots, m$, we write $\mathbf{A} = (\mathbf{a}^1, \mathbf{a}^2, \ldots, \mathbf{a}^m)$. Following the same pattern, we often write $\mathbf{A} = [\mathbf{B}, \mathbf{C}]$ for the partitioned matrix $\mathbf{A} = [\mathbf{B}|\mathbf{C}]$.

A.3 Spaces

We consider the n-component vectors $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ as elements of a vector space. The space itself, n-dimensional Euclidean space, is denoted E^n. Vectors in the space can be added or multiplied by a scalar, by performing the corresponding operations on the components. We write $\mathbf{x} \geqslant \mathbf{0}$ if each component of $\mathbf{x}$ is nonnegative.

The *line segment* connecting two vectors $\mathbf{x}$ and $\mathbf{y}$ is denoted $[\mathbf{x}, \mathbf{y}]$ and consists of all vectors of the form $\alpha\mathbf{x} + (1 - \alpha)\mathbf{y}$ with $0 \leqslant \alpha \leqslant 1$.

The *scalar product* of two vectors $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ and $\mathbf{y} = (y_1, y_2, \ldots, y_n)$ is defined as $\mathbf{x}^T\mathbf{y} = \mathbf{y}^T\mathbf{x} = \sum_{i=1}^n x_i y_i$. The vectors $\mathbf{x}$ and $\mathbf{y}$ are said to be *orthogonal* if $\mathbf{x}^T\mathbf{y} = 0$. The *magnitude* or *norm* of a vector $\mathbf{x}$ is $|\mathbf{x}| = (\mathbf{x}^T\mathbf{x})^{1/2}$. For any two vectors $\mathbf{x}$ and $\mathbf{y}$ in E^n, the *Cauchy-Schwarz Inequality* holds: $|\mathbf{x}^T\mathbf{y}| \leqslant |\mathbf{x}| \cdot |\mathbf{y}|$.

A set of vectors $\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_k$ is said to be *linearly dependent* if there are scalars $\lambda_1, \lambda_2, \ldots, \lambda_k$, not all zero, such that $\sum_{i=1}^k \lambda_i \mathbf{a}_i = \mathbf{0}$. If no such set of scalars exists, the vectors are said to be *linearly independent*. A *linear combination* of the vectors $\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_k$ is a vector of the form $\sum_{i=1}^k \lambda_j \mathbf{a}_i$. The set of vectors that are linear combinations of $\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_k$ is the set *spanned* by the vectors. A linearly independent set of vectors that span E^n is said to be a *basis* for E^n. Every basis for E^n contains exactly n vectors.

The *rank* of a matrix $\mathbf{A}$ is equal to the maximum number of linearly independent columns in $\mathbf{A}$. This number is also equal to the maximum number of linearly independent rows in $\mathbf{A}$. The $m \times n$ matrix $\mathbf{A}$ is said to be of *full rank* if the rank of $\mathbf{A}$ is equal to the minimum of m and n.

A *subspace* M of E^n is a subset that is closed under the operations of vector addition and scalar multiplication; that is, if $\mathbf{a}$ and $\mathbf{b}$ are vectors in M, then $\lambda\mathbf{a} + \mu\mathbf{b}$ is also in M for every pair of scalars λ, μ. The dimension of a subspace M is equal to the maximum number of linearly independent vectors in M. If M is a subspace of E^n, the *orthogonal complement* of M, denoted $M^\perp$, consists of all vectors that are orthogonal to every vector in M. The orthogonal complement of M is easily seen to be a subspace, and together M and $M^\perp$ span E^n in the sense that every vector $\mathbf{x} \in E^n$ can be written uniquely in the form $\mathbf{x} = \mathbf{a} + \mathbf{b}$ with $\mathbf{a} \in M$, $\mathbf{b} \in M^\perp$. In this case $\mathbf{a}$ and $\mathbf{b}$ are said to be the *orthogonal projections* of $\mathbf{x}$ onto the subspaces M and $M^\perp$, respectively.

A correspondence $\mathbf{A}$ that associates with each point in a space X a point in a space Y is said to be a *mapping from* X to Y. For convenience this situation is symbolized by $\mathbf{A} : X \to Y$. The mapping $\mathbf{A}$ may be either linear or nonlinear. The norm of linear mapping $\mathbf{A}$ is defined as $|\mathbf{A}| = \max\limits_{|\mathbf{x}|\leq 1} |\mathbf{Ax}|$. It follows that for any $\mathbf{x}$, $|\mathbf{Ax}| \leq |\mathbf{A}| \cdot |\mathbf{x}|$.

A.4 Eigenvalues and Quadratic Forms

Corresponding to an $n \times n$ square matrix $\mathbf{A}$, a scalar λ and a nonzero vector $\mathbf{x}$ satisfying the equation $\mathbf{Ax} = \lambda\mathbf{x}$ are said to be, respectively, an eigenvalue and eigenvector of $\mathbf{A}$. In order that λ be an eigenvalue it is clear that it is necessary and sufficient for $\mathbf{A} - \lambda\mathbf{I}$ to be singular, and hence $\det(\mathbf{A} - \lambda\mathbf{I}) = 0$. This last result, when expanded, yields an nth-order polynomial equation which can be solved for n (possibly nondistinct) complex roots λ which are the eigenvalues of $\mathbf{A}$.

Now, for the remainder of this section, assume that $\mathbf{A}$ is symmetric. Then the following properties hold:

(i) The eigenvalues of $\mathbf{A}$ are real.
(ii) Eigenvectors associated with distinct eigenvalues are orthogonal.
(iii) There is an orthogonal basis for E^n, each element of which is an eigenvector of $\mathbf{A}$.

If the basis $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_n$ in (iii) is normalized so that each element has magnitude unity, then defining the matrix $\mathbf{Q} = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_n]$ we note that $\mathbf{Q}^T\mathbf{Q} = \mathbf{I}$ and hence $\mathbf{Q}^T = \mathbf{Q}^{-1}$. A matrix with this property is said to be an *orthogonal* matrix. Also, we observe, in this case, that

$$\mathbf{Q}^{-1}\mathbf{AQ} = \mathbf{Q}^T\mathbf{AQ} = \mathbf{Q}^T[\mathbf{Au}_1, \mathbf{Au}_2, \ldots, \mathbf{Au}_n]$$
$$= \mathbf{Q}^T[\lambda_1\mathbf{u}_1, \lambda_2\mathbf{u}_2, \ldots, \lambda_n\mathbf{u}_n].$$

Thus

$$\mathbf{Q}^{-1}\mathbf{A}\mathbf{Q} = \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_n \end{bmatrix},$$

and therefore $\mathbf{A}$ is similar to a diagonal matrix.

A symmetric matrix $\mathbf{A}$ is said to be *positive definite* if the *quadratic form* $\mathbf{x}^T\mathbf{A}\mathbf{x}$ is positive for all nonzero vectors $\mathbf{x}$. Similarly, we define $\mathbf{A}$ to be *positive semidefinite, negative definite*, or *negative semidefinite* if $\mathbf{x}^T\mathbf{A}\mathbf{x} \geqslant 0, < 0$, or $\leqslant 0$ for all $\mathbf{x}$. The matrix $\mathbf{A}$ is *indefinite* if $\mathbf{x}^T\mathbf{A}\mathbf{x}$ is positive for some $\mathbf{x}$ and negative for others.

It is easy to obtain a connection between definiteness and the eigenvalues of $\mathbf{A}$. For any $\mathbf{x}$ let $\mathbf{y} = \mathbf{Q}^{-1}\mathbf{x}$ where $\mathbf{Q}$ is defined as above. Then $\mathbf{x}^T\mathbf{A}\mathbf{x} = \mathbf{y}^T\mathbf{Q}^T\mathbf{A}\mathbf{Q}\mathbf{y} = \sum_{i=1}^n \lambda_i y_i^2$. Since the y_i's are arbitrary (since $\mathbf{x}$ is), it is clear that $\mathbf{A}$ is positive definite (or positive semidefinite) if and only if all eigenvalues of $\mathbf{A}$ are positive (or nonnegative).

Through diagonalization we can also easily show that a positive semidefinite matrix $\mathbf{A}$ has a positive semidefinite (symmetric) square root $\mathbf{A}^{1/2}$ satisfying $\mathbf{A}^{1/2} \cdot \mathbf{A}^{1/2} = \mathbf{A}$. For this we use $\mathbf{Q}$ as above and define

$$\mathbf{A}^{1/2} = \mathbf{Q} \begin{bmatrix} \lambda_1^{1/2} & & & \\ & \lambda_2^{1/2} & & \\ & & \ddots & \\ & & & \lambda_n^{1/2} \end{bmatrix} \mathbf{Q}^T,$$

which is easily verified to have the desired properties.

A.5 Topological Concepts

A sequence of vectors $\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_k, \ldots$, denoted by $\{\mathbf{x}_{k=0}\}_k^\infty$, or if the index set is understood, by simply $\{\mathbf{x}_k\}$, is said to *converge* to the limit $\mathbf{x}$ if $|\mathbf{x}_k - \mathbf{x}| \to 0$ as $k \to \infty$ (that is, if given $\varepsilon > 0$, there is a N such that $k \geqslant N$ implies $|\mathbf{x}_k - \mathbf{x}| < \varepsilon$). If $\{\mathbf{x}_k\}$ converges to $\mathbf{x}$, we write $\mathbf{x}_k \to \mathbf{x}$ or $\lim \mathbf{x}_k = \mathbf{x}$.

A point $\mathbf{x}$ is a *limit point* of the sequence $\{\mathbf{x}_k\}$ if there is a subsequence of $\{\mathbf{x}_k\}$ convergent to $\mathbf{x}$. Thus $\mathbf{x}$ is a limit point of $\{\mathbf{x}_k\}$ if there is a subset $\mathcal{K}$ of the positive integers such that $\{\mathbf{x}_k\}_{k\in\mathcal{K}}$ converges to $\mathbf{x}$.

A *ball (sphere) around* $\mathbf{x}$ is a set of the form $\{\mathbf{y} : |\mathbf{y} - \mathbf{x}| < (=)\ \varepsilon\}$ for some $\varepsilon > 0$. Such a ball is also referred to as the *neighborhood* of $\mathbf{x}$ of radius ε.

A subset S of E^n is *open* if around every point in S there is a sphere that is contained in S. Equivalently, S is open if given $\mathbf{x} \in S$ there is an $\varepsilon > 0$ such that $|\mathbf{y} - \mathbf{x}| < \varepsilon$ implies $\mathbf{y} \in S$. Thus the sphere $\{\mathbf{x} : |\mathbf{x}| < 1\}$ is open. In general, open sets can be characterized as sets having no sharp boundaries. The *interior* of any set S in E^n is the set of points $\mathbf{x} \in S$ which are the center of some sphere contained in S.

It is denoted $\overset{\circ}{S}$. The interior of a set is always open; indeed it is the largest open set contained in S. The interior of the set $\{\mathbf{x} : |\mathbf{x}| \leqslant 1\}$ is the sphere $\{\mathbf{x} : |\mathbf{x}| < 1\}$.

A set P is *closed* if every point that is arbitrarily close to the set P is a member of P. Equivalently, P is closed if $\mathbf{x}_k \to \mathbf{x}$ with $\mathbf{x}_k \in P$ implies $\mathbf{x} \in P$. Thus the set $\{\mathbf{x} : |\mathbf{x}| \leqslant 1\}$ is closed. The *closure* of any set P in E^n is the smallest closed set containing P. It is denoted $\overline{S}$. The *boundary* of a set is that part of the closure that is not in the interior.

A set is *compact* if it is both closed and bounded (that is, if it is closed and is contained within some sphere of finite radius). An important result, due to Weierstrass, is that if S is a compact set and $\{\mathbf{x}_k\}$ is a sequence each member of which belongs to S, then $\{\mathbf{x}_k\}$ has a limit point in S (that is, there is subsequence converging to a point in S).

Corresponding to a bounded sequence $\{r_k\}_{k=0}^{\infty}$ of real numbers, if we let $s_k = \sup\{r_i : i \geqslant k\}$ then $\{s_k\}$ converges to some real number s_o. This number is called the *limit superior* of $\{r_k\}$ and is denoted $\overline{\lim\limits_{k\to\infty}}(r_k)$.

A.6 Functions

A real-valued function f defined on a subset of E^n is said to be *continuous* at $\mathbf{x}$ if $\mathbf{x}_k \to \mathbf{x}$ implies $f(\mathbf{x}_k) \to f(\mathbf{x})$. Equivalently, f is continuous at $\mathbf{x}$ if given $\varepsilon > 0$ there is a $\delta > 0$ such that $|\mathbf{y}-\mathbf{x}| < \delta$ implies $|f(\mathbf{y})-f(\mathbf{x})| < \varepsilon$. An important result connected with continuous functions is a *theorem of Weierstrass*: A continuous function f defined on a compact set S has a minimum point in S; that is, there is an $\mathbf{x}^* \in S$ such that for all $\mathbf{x} \in S$, $f(\mathbf{x}) \geqslant f(\mathbf{x}^*)$.

A set of real-valued functions $f_1, f_2, \ldots, f_m$ on E^n can be regarded as a single vector function $\mathbf{f} = (f_1, f_2, \ldots, f_m)$. This function assigns a vector $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_m(\mathbf{x}))$ in E^m to every vector $\mathbf{x} \in E^n$. Such a vector-valued function is said to be *continuous* if each of its component functions is continuous.

If each component of $\mathbf{f} = (f_1, f_2, \ldots, f_m)$ is continuous on some open set of E^n, then we write $\mathbf{f} \in C$. If in addition, each component function has first partial derivatives which are continuous on this set, we write $\mathbf{f} \in C^1$. In general, if the component functions have continuous partial derivatives of order p, we write $\mathbf{f} \in C^p$.

If $f \in C^1$ is a real-valued function on E^n, $f(\mathbf{x}) = f(x_1, x_2, \ldots, x_n)$, we define the *gradient* of f to be the vector

$$\nabla f(\mathbf{x}) = \left[\frac{\partial f(\mathbf{x})}{\partial x_1}, \frac{\partial f(\mathbf{x})}{\partial x_2}, \cdots, \frac{\partial f(\mathbf{x})}{\partial x_n}\right].$$

We sometimes use the alternative notation $f_{\mathbf{x}}(\mathbf{x})$ for $\nabla f(\mathbf{x})$. In matrix calculations the gradient is considered to be a row vector.

If $f \in C^2$ then we define the *Hessian* of f at $\mathbf{x}$ to be the $n \times n$ matrix denoted $\boldsymbol{\nabla}^2 f(\mathbf{x})$ or $\mathbf{F}(\mathbf{x})$ as

$$\mathbf{F}(\mathbf{x}) = \left[\frac{\partial^2 f(\mathbf{x})}{\partial x_i \partial x_j}\right].$$

Since

$$\frac{\partial^2 f}{\partial x_i \partial x_j} = \frac{\partial^2 f}{\partial x_j \partial x_i},$$

it is easily seen that the Hessian is symmetric.

For a vector-valued function $\mathbf{f} = (f_1, f_2, \ldots, f_m)$ the situation is similar. If $\mathbf{f} \in C^1$, the first derivative is defined as the $m \times n$ matrix

$$\boldsymbol{\nabla}\mathbf{f}(\mathbf{x}) = \left[\frac{\partial f_i(\mathbf{x})}{\partial x_j}\right].$$

If $\mathbf{f} \in C^2$ it is possible to define the m Hessians $\mathbf{F}_1(\mathbf{x}), \mathbf{F}_2(\mathbf{x}), \ldots, \mathbf{F}_m(\mathbf{x})$ corresponding to the m component functions. The second derivative itself, for a vector function, is a third-order tensor but we do not require its use explicitly. Given any $\boldsymbol{\lambda}^T = [\lambda_1, \lambda_2, \ldots, \lambda_m] \in E_m$, we note, however, that the real-valued function $\boldsymbol{\lambda}^T\mathbf{f}$ has gradient equal to $\boldsymbol{\lambda}^T\boldsymbol{\nabla}\mathbf{f}(\mathbf{x})$ and Hessian, denoted $\boldsymbol{\lambda}^T\mathbf{F}(\mathbf{x})$, equal to

$$\boldsymbol{\lambda}^T\mathbf{F}(\mathbf{x}) = \sum_{i=1}^{m} \lambda_i \mathbf{F}_i(\mathbf{x}).$$

Also see Sect. 7.4 for a discussion of convex functions.

Taylor's Theorem

A group of results that are used frequently in analysis are referred to under the general heading of Taylor's Theorem or Mean Value Theorems. If $f \in C^1$ in a region containing the line segment $[\mathbf{x}_1, \mathbf{x}_2]$, then there is a θ, $0 \leqslant \theta \leqslant 1$ such that

$$f(\mathbf{x}_2) = f(\mathbf{x}_1) + \boldsymbol{\nabla} f(\theta\mathbf{x}_1 + (1-\theta)\mathbf{x}_2)(\mathbf{x}_2 - \mathbf{x}_1).$$

Furthermore, if $f \in C^2$ then there is a θ, $0 \leqslant \theta \leqslant 1$ such that

$$\begin{aligned} f(\mathbf{x}_2) &= f(\mathbf{x}_1) + \boldsymbol{\nabla} f(\mathbf{x}_1)(\mathbf{x}_2 - \mathbf{x}_1) \\ &\quad + \frac{1}{2}(\mathbf{x}_2 - \mathbf{x}_1)^T\mathbf{F}(\theta\mathbf{x}_1 + (1-\theta)\mathbf{x}_2)(\mathbf{x}_2 - \mathbf{x}_1), \end{aligned}$$

where $\mathbf{F}$ denotes the Hessian of f.

Implicit Function Theorem

Suppose we have a set of m equations in n variables

$$h_i(\mathbf{x}) = 0, \ i = 1, 2, \ \ldots, \ m.$$

The implicit function theorem addresses the question as to whether if $n - m$ of the variables are fixed, the equations can be solved for the remaining m variables. Thus selecting m variables, say $x_1, \ x_2, \ \ldots, \ x_m$, we wish to determine if these may be expressed in terms of the remaining variables in the form

$$x_i = \phi_i(x_{m+1}, \ x_{m+2}, \ \ldots, \ x_n), \qquad i = 1, 2, \ \ldots, \ m.$$

The functions ϕ_i, if they exist, are called *implicit* functions.

Theorem. *Let* $\mathbf{x}^0 = (x_1^0, \ x_2^0, \ \ldots, \ x_n^0)$ *be a point in* E^n *satisfying the properties:*

i) The functions $h_i \in C^p, \ i = 1, 2, \ \ldots, \ m$ *in some neighborhood of* $\mathbf{x}^0$*, for some* $p \geqslant 1$.
ii) $h_i(\mathbf{x}^0) = 0, \ i = 1, 2, \ \ldots, \ m.$
iii) The $m \times m$ *Jacobian matrix*

$$\mathbf{J} = \begin{bmatrix} \frac{\partial h_1(\mathbf{x}^0)}{\partial x_1} & \cdots & \frac{\partial h_1(\mathbf{x}^0)}{\partial x_m} \\ \vdots & & \vdots \\ \frac{\partial h_m(\mathbf{x}^0)}{\partial x_1} & \cdots & \frac{\partial h_m(\mathbf{x}^0)}{\partial x_m} \end{bmatrix}.$$

is nonsingular.

Then there is a neighborhood of $\hat{\mathbf{x}}^0 = (x_{m+1}^0, \ x_{m+2}^0, \ \ldots, \ x_n^0) \in E^{n-m}$ *such that for* $\hat{\mathbf{x}} = (x_{m+1}, \ x_{m+2}, \ \ldots, \ x_n)$ *in this neighborhood there are functions* $\phi_i(\hat{\mathbf{x}}), \ i = 1, 2, \ \ldots, \ m$ *such that*

i) $\phi_i \in C^p$.
ii) $x_i^0 = \phi_i(\hat{\mathbf{x}}^0), \qquad i = 1, 2, \ \ldots, \ m.$
iii) $h_i(\phi_1(\hat{\mathbf{x}}), \ \phi_2(\hat{\mathbf{x}}), \ \ldots, \ \phi_m(\hat{\mathbf{x}}), \hat{\mathbf{x}}) = 0, \ i = 1, 2, \ \ldots, \ m.$

Example 1. Consider the equation $x_1^2 + x_2 = 0$. A solution is $x_1 = 0, \ x_2 = 0$. However, in a neighborhood of this solution there is no function ϕ such that $x_1 = \phi(x_2)$. At this solution condition (iii) of the implicit function theorem is violated. At any other solution, however, such a ϕ exists.

Example 2. Let $\mathbf{A}$ be an $m \times n$ matrix ($m < n$) and consider the system of linear equations $\mathbf{Ax} = \mathbf{b}$. If $\mathbf{A}$ is partitioned as $\mathbf{A} = [\mathbf{B}, \ \mathbf{C}]$ where $\mathbf{B}$ is $m \times m$ then condition (iii) is satisfied if and only if $\mathbf{B}$ is nonsingular. This condition corresponds, of course, exactly with what the theory of linear equations tells us. In view of this example, the implicit function can be regarded as a nonlinear generalization of the linear theory.

o, *O* Notation

If g is a real-valued function of a real variable, the notation $g(x) = O(x)$ means that $g(x)$ goes to zero at least as fast as x does. More precisely, it means that there is a $K \geqslant 0$ such that

$$\left|\frac{g(x)}{x}\right| \leqslant K \text{ as } x \to 0.$$

The notation $g(x) = o(x)$ means that $g(x)$ goes to zero faster than x does; or equivalently, that K above is zero.

Appendix B
Convex Sets

B.1 Basic Definitions

Concepts related to convex sets so dominate the theory of optimization that it is essential for a student of optimization to have knowledge of their most fundamental properties. In this appendix is compiled a brief summary of the most important of these properties.

Definition. A set C in E^n is said to be *convex* if for every $\mathbf{x}_1,\ \mathbf{x}_2 \in C$ and every real number $\alpha,\ 0 < \alpha < 1$, the point $\alpha\mathbf{x}_1 + (1 - \alpha)\mathbf{x}_2 \in C$.

This definition can be interpreted geometrically as stating that a set is convex if, given two points in the set, every point on the line segment joining these two points is also a member of the set. This is illustrated in Fig. B.1.

The following proposition shows that certain familiar set operations preserve convexity.

Proposition 1. *Convex sets in E^n satisfy the following relations:*

i) If C is a convex set and β is a real number, the set

$$\beta C = \{\mathbf{x} : \mathbf{x} = \beta\mathbf{c},\ \mathbf{c} \in C\}$$

is convex.

ii) If C and D are convex sets, then the set

$$C + D = \{\mathbf{x} : \mathbf{x} = \mathbf{c} + \mathbf{d},\ \mathbf{c} \in C,\ \mathbf{d} \in D\}$$

is convex.

iii) The intersection of any collection of convex sets is convex.

The proofs of these three properties follow directly from the definition of a convex set and are left to the reader. The properties themselves are illustrated in Fig. B.2.

© Springer International Publishing Switzerland 2016

D.G. Luenberger, Y. Ye, *Linear and Nonlinear Programming*, International Series in Operations Research & Management Science 228,
DOI 10.1007/978-3-319-18842-3

Another important concept is that of forming the smallest convex set containing a given set.

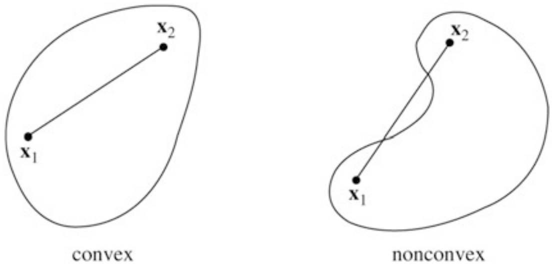

Fig. B.1 Convexity

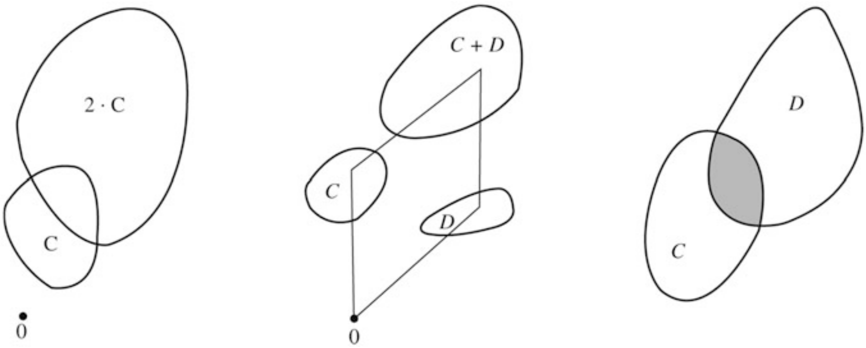

Fig. B.2 Properties of convex sets

Definition. Let S be a subset of E^n. The *convex hull* of S, denoted co(S), is the set which is the intersection of all convex sets containing S. The *closed convex hull* of S is defined as the closure of co(S).

Finally, we conclude this section by defining a *cone* and a *convex cone*. A convex cone is a special kind of convex set that arises quite frequently.

Definition. A set C is a *cone* if $\mathbf{x} \in C$ implies $\alpha\mathbf{x} \in C$ for all $\alpha > 0$. A cone that is also convex is a *convex cone*.

Some cones are shown in Fig. B.3. Their basic property is that if a point $\mathbf{x}$ belongs to a cone, then the entire half line from the origin through the point (but not the origin itself) also must belong to the cone.

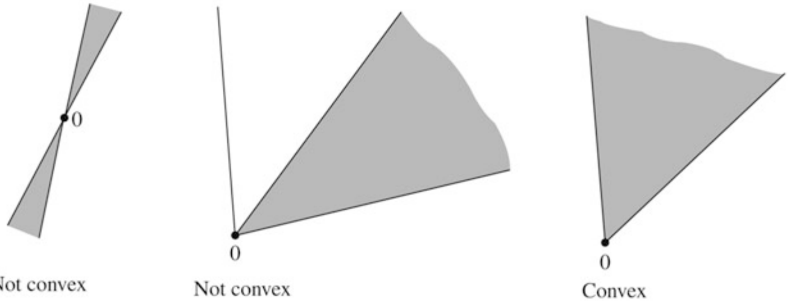

Fig. B.3 Cones

B.2 Hyperplanes and Polytopes

The most important type of convex set (aside from single points) is the hyperplane. Hyperplanes dominate the entire theory of optimization, appearing under the guise of Lagrange multipliers, duality theory, or gradient calculations.

The most natural definition of a hyperplane is the logical generalization of the geometric properties of a plane in three dimensions. We start by giving this geometric definition. For computations and for a concrete description of hyperplanes, however, there is an equivalent algebraic definition that is more useful. A major portion of this section is devoted to establishing this equivalence.

Definition. A set V in E^n is said to be a *linear variety*, if, given any $\mathbf{x}_1, \mathbf{x}_2 \in V$, we have $\lambda\mathbf{x}_1 + (1-\lambda)\mathbf{x}_2 \in V$ for all real numbers λ.

Note that the only difference between the definition of a linear variety and a convex set is that in a linear variety the entire line passing through any two points, rather than simply the line segment between them, must lie in the set. Thus in three dimensions the nonempty linear varieties are points, lines, two-dimensional planes, and the whole space. In general, it is clear that we may speak of the dimension of a linear variety. Thus, for example, a point is a linear variety of dimension zero and a line is a linear variety of dimension one. In the general case, the dimension of a linear variety in E^n can be found by translating it (moving it) so that it contains the origin and then determining the dimension of the resulting set, which is then a subspace of E^n.

Definition. A *hyperplane* in E^n is an $(n-1)$-dimensional linear variety.

We see that hyperplanes generalize the concept of a two-dimensional plane in three-dimensional space. They can be regarded as the largest linear varieties in a space, other than the entire space itself.

We now relate this abstract geometric definition to an algebraic one.

Proposition 2. *Let* $\mathbf{a}$ *be a nonzero n-dimensional column vector, and let c be a real number. The set*

$$H = \{\mathbf{x} \in E^n : \mathbf{a}^T\mathbf{x} = c\}$$

is a hyperplane in E^n.

Proof. It follows directly from the linearity of the equation $\mathbf{a}^T\mathbf{x} = c$ that H is a linear variety. Let $\mathbf{x}_1$ be any vector in H. Translating by $-\mathbf{x}_1$ we obtain the set $M = H - \mathbf{x}_1$ which is a linear subspace of E^n. This subspace consists of all vectors $\mathbf{x}$ satisfying $\mathbf{a}^T\mathbf{x} = 0$; in other words, all vectors orthogonal to $\mathbf{a}$. This is clearly an $(n-1)$-dimensional subspace. ∎

Proposition 3. *Let H be a hyperplane in* E^n. *Then there is a nonzero n- dimensional vector and a constant c such that*

$$H = \{\mathbf{x} \in E^n : \mathbf{a}^T\mathbf{x} = c\}.$$

Proof. Let $\mathbf{x}_1 \in H$ and translate by $-\mathbf{x}_1$ obtaining the set $M = H - \mathbf{x}_1$. Since H is a hyperplane, M is an $(n-1)$-dimensional subspace. Let $\mathbf{a}$ be any nonzero vector that is orthogonal to this subspace, that is, $\mathbf{a}$ belongs to the one-dimensional subspace $M^\perp$. Clearly $M = \{\mathbf{x} : \mathbf{a}^T\mathbf{x} = \mathbf{0}\}$. Letting $c = \mathbf{a}^T\mathbf{x}_1$ we see that if $\mathbf{x}_2 \in H$ we have $\mathbf{x}_2 - \mathbf{x}_1 \in M$ and thus $\mathbf{a}^T\mathbf{x}_2 - \mathbf{a}^T\mathbf{x}_1 = 0$ which implies $\mathbf{a}^T\mathbf{x}_2 = c$. Thus $H \subset \{\mathbf{x} : \mathbf{a}^T\mathbf{x} = c\}$. Since H is, by definition, of dimension $n-1$ and $\{\mathbf{x} : \mathbf{a}^T\mathbf{x} = c\}$ is of dimension $n-1$ by Proposition 2, these two sets must be equal. ∎

Combining Propositions 2 and 3, we see that a hyperplane is the set of solutions to a single linear equation. This is illustrated in Fig. B.4. We now use hyperplanes to build up other important classes of convex sets.

Definition. Let $\mathbf{a}$ be a nonzero vector in E^n and let c be a real number. Corresponding to the hyperplane $H = \{\mathbf{x} : \mathbf{a}^T\mathbf{x} = c\}$ are the *positive and negative closed half spaces*

$$H_+ = \{\mathbf{x} : \mathbf{a}^T\mathbf{x} \geqslant c\}$$
$$H_- = \{\mathbf{x} : \mathbf{a}^T\mathbf{x} \leqslant c\}$$

and the *positive* and *negative open half spaces*

$$\mathring{H}_+ = \{\mathbf{x} : \mathbf{a}^T\mathbf{x} > c\}$$
$$\mathring{H}_- = \{\mathbf{x} : \mathbf{a}^T\mathbf{x} < c\}.$$

It is easy to see that half spaces are convex sets and that the union of H_+ and H_- is the whole space.

Definition. A set which can be expressed as the intersection of a finite number of closed half spaces is said to be a *convex polytope*.

We see that convex polytopes are the sets obtained as the family of solutions to a set of linear inequalities of the form

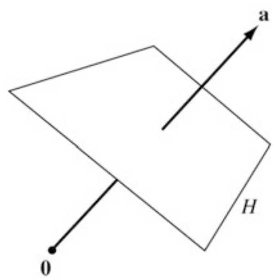

Fig. B.4 Hyperplane

Fig. B.5 Polytopes

$$
\begin{aligned}
\mathbf{a}_1^T\mathbf{x} &\leqslant b_1\\
\mathbf{a}_2^T\mathbf{x} &\leqslant b_2\\
\vdots \quad &\quad \vdots\\
\mathbf{a}_m^T\mathbf{x} &\leqslant b_m,
\end{aligned}
$$

since each individual inequality defines a half space and the solution family is the intersection of these half spaces. (If some $\mathbf{a}_i = \mathbf{0}$, the resulting set can still, as the reader may verify, be expressed as the intersection of a finite number of half spaces.)

Several polytopes are illustrated in Fig. B.5. We note that a polytope may be empty, bounded, or unbounded. The case of a nonempty bounded polytope is of special interest and we distinguish this case by the following.

Definition. A nonempty bounded polytope is called a *polyhedron*.

B.3 Separating and Supporting Hyperplanes

The two theorems in this section are perhaps the most important results related to convexity. Geometrically, the first states that given a point outside a convex set, a hyperplane can be passed through the point that does not touch the convex set. The second, which is a limiting case of the first, states that given a boundary point of a convex set, there is a hyperplane that contains the boundary point and contains the convex set on one side of it.

Theorem 1. *Let C be a convex set and let $\mathbf{y}$ be a point exterior to the closure of C. Then there is a vector $\mathbf{a}$ such that $\mathbf{a}^T\mathbf{y} < \inf_{\mathbf{x}\in C} \mathbf{a}^T\mathbf{x}$.*

Proof. Let

$$\delta = \inf_{\mathbf{x}\in C} |\mathbf{x}-\mathbf{y}| > 0.$$

There is an $\mathbf{x}_0$ on the boundary of C such that $|\mathbf{x}_0-\mathbf{y}| = \delta$. This follows because the continuous function $f(\mathbf{x}) = |\mathbf{x}-\mathbf{y}|$ achieves its minimum over any closed and bounded set and it is clearly only necessary to consider $\mathbf{x}$ in the intersection of the closure of C and the sphere of radius 2δ centered at $\mathbf{y}$.

We shall show that setting $\mathbf{a} = \mathbf{x}_0 - \mathbf{y}$ satisfies the conditions of the theorem. Let $\mathbf{x} \in C$. For any α, $0 \leqslant \alpha \leqslant 1$, the point $\mathbf{x}_0 + \alpha(\mathbf{x}-\mathbf{x}_0) \in \overline{C}$ and thus

$$|\mathbf{x}_0 + \alpha(\mathbf{x}-\mathbf{x}_0) - \mathbf{y}|^2 \geqslant |\mathbf{x}_0 - \mathbf{y}|^2.$$

Expanding,

$$2\alpha(\mathbf{x}_0-\mathbf{y})^T(\mathbf{x}-\mathbf{x}_0) + \alpha^2|\mathbf{x}-\mathbf{x}_0|^2 \geqslant 0.$$

Thus, considering this as $\alpha \to 0+$, we obtain

$$(\mathbf{x}_0-\mathbf{y})^T(\mathbf{x}-\mathbf{x}_0) \geqslant 0$$

or,

$$\begin{aligned}(\mathbf{x}_0-\mathbf{y})^T\mathbf{x} \geqslant (\mathbf{x}_0-\mathbf{y})^T\mathbf{x}_0 &= (\mathbf{x}_0-\mathbf{y})^T\mathbf{y} + (\mathbf{x}_0-\mathbf{y})^T(\mathbf{x}_0-\mathbf{y}) \\ &= (\mathbf{x}_0-\mathbf{y})^T\mathbf{y} + \delta^2.\end{aligned}$$

Setting $\mathbf{a} = \mathbf{x}_0 - \mathbf{y}$ proves the theorem. ∎

The geometrical interpretation of Theorem 1 is that, given a convex set C and a point $\mathbf{y}$ exterior to the closure of C, there is a hyperplane containing $\mathbf{y}$ that contains C in one of its open half spaces. We can easily extend this theorem to include the case where $\mathbf{y}$ is a boundary point of C.

Theorem 2. *Let C be a convex set and let $\mathbf{y}$ be a boundary point of C. Then there is a hyperplane containing $\mathbf{y}$ and containing C in one of its closed half spaces.*

Proof. Let $\{\mathbf{y}_k\}$ be a sequence of vectors, exterior to the closure of C, converging to $\mathbf{y}$. Let $\{\mathbf{a}_k\}$ be the sequence of corresponding vectors constructed according to Theorem 1, normalized so that $|\mathbf{a}_k| = 1$, such that

$$\mathbf{a}_k^T\mathbf{y}_k < \inf_{\mathbf{x}\in C} \mathbf{a}_k^T\mathbf{x}.$$

Since $\{\mathbf{a}_k\}$ is a bounded sequence, it has a convergent subsequence $\{\mathbf{a}_k\}$, $k \in \mathcal{K}$ with limit $\mathbf{a}$. For this vector we have for any $\mathbf{x} \in C$.

$$\mathbf{a}^T\mathbf{y} = \lim_{k\in\mathcal{K}} \mathbf{a}_k^T\mathbf{y}_k \leqslant \lim_{k\in\mathcal{K}} \mathbf{a}_k^T\mathbf{x} = \mathbf{a}\mathbf{x}. \; ∎$$

Definition. A hyperplane containing a convex set C in one of its closed half spaces and containing a boundary point of C is said to be a *supporting hyperplane* of C.

In terms of this definition, Theorem 2 says that, given a convex set C and a boundary point y of C, there is a hyperplane supporting C at $\mathbf{y}$.

It is useful in the study of convex sets to consider the *relative interior* of a convex set C defined as the largest subset of C that contains no boundary points of C.

Another variation of the theorems of this section is the one that follows, which is commonly known as the Separating Hyperplane Theorem.

Theorem 3. *Let B and C be convex sets with no common relative interior points. (That is the only common points are boundary points.) Then there is a hyperplane separating B and D. In particular, there is a nonzero vector $\mathbf{a}$ such that* $\sup_{\mathbf{b}\in B} \mathbf{a}^T\mathbf{b} \leq \inf_{\mathbf{c}\in C} \mathbf{a}^T\mathbf{c}$.

Proof. Consider the set $G = C - B$. It is easily shown that G is convex and that $\mathbf{0}$ is not a relative interior point of G. Hence, Theorem 1 or Theorem 2 applies and gives the appropriate hyperplane. ∎

B.4 Extreme Points

Definition. A point $\mathbf{x}$ in a convex set C is said to be an *extreme point* of C if there are no two distinct points $\mathbf{x}_1$ and $\mathbf{x}_2$ in C such that $\mathbf{x} = \alpha\mathbf{x}_1 + (1-\alpha)\mathbf{x}_2$ for some α, $0 < \alpha < 1$.

For example, in E^2 the extreme points of a square are its four corners; the extreme points of a circular disk are all points on the boundary. Note that a linear variety consisting of more than one point has no extreme points.

Lemma 1. *Let C be a convex set, H a supporting hyperplane of C, and T the intersection of H and C. Every extreme point of T is an extreme point of C.*

Proof. Suppose $\mathbf{x}_0 \in T$ is not an extreme point of C. Then $\mathbf{x}_0 = \alpha\mathbf{x}_1 + (1-\alpha)\mathbf{x}_2$ for some $\mathbf{x}_1,\ \mathbf{x}_2 \in C,\ \mathbf{x}_1 \neq \mathbf{x}_2, 0 < \alpha < 1$. Let H be described as $H = \{\mathbf{x} : \mathbf{a}^T\mathbf{x} = c\}$ with C contained in its closed positive half space. Then

$$\mathbf{a}^T\mathbf{x}_1 \geqslant c, \quad \mathbf{a}^T\mathbf{x}_2 \geqslant c.$$

But, since $\mathbf{x}_0 \in H$,

$$c = \mathbf{a}^T\mathbf{x}_0 = \alpha\mathbf{a}^T\mathbf{x}_1 + (1-\alpha)\mathbf{a}^T\mathbf{x}_2,$$

and thus $\mathbf{x}_1$ and $\mathbf{x}_2 \in H$. Hence $\mathbf{x}_1,\ \mathbf{x}_2 \in T$ and $\mathbf{x}_0$ is not an extreme point of T. ∎

Theorem 4. *A closed bounded convex set in E^n is equal to the closed convex hull of its extreme points.*

Proof. The proof is by induction on the dimension of the space E^n. The statement is easily seen to be true for $n = 1$. Suppose that it is true for $n-1$. Let C be a closed bounded convex set in E^n, and let K be the closed convex hull of the extreme points of C. We wish to show that $K = C$.

Assume there is $\mathbf{y} \in C$ $\mathbf{y} \notin K$. Then by Theorem 1, Sect. B.3, there is a hyperplane separating $\mathbf{y}$ and K; that is, there is $\mathbf{a} \neq \mathbf{0}$, such that $\mathbf{a}^T\mathbf{y} < \inf_{\mathbf{x}\in K} \mathbf{a}^T\mathbf{x}$. Let $c_0 = \inf\limits_{\mathbf{x}\in C}(\mathbf{a}^T\mathbf{x})$. The number c_0 is finite and there is an $\mathbf{x}_0 \in C$ for which $\mathbf{a}^T\mathbf{x}_0 = c_0$, because by Weierstrass' Theorem, the continuous function $\mathbf{a}^T\mathbf{x}$ achieves its minimum over any closed bounded set. Thus the hyperplane $H = \{\mathbf{x} : \mathbf{a}^T\mathbf{x} = c_0\}$ is a supporting hyperplane to C. It is disjoint from K since $c_0 < \inf\limits_{\mathbf{x}\in K}(\mathbf{a}^T\mathbf{x})$.

Let $T = H \cap C$. Then T is a bounded closed convex subset of H which can be regarded as a space of dimension $n - 1$. T is nonempty, since it contains $\mathbf{x}_0$. Thus, by the induction hypothesis, T contains extreme points; and by Lemma 1 these are also extreme points of C. Thus we have found extreme points of C not in K, which is a contradiction. ∎

Let us investigate the implications of this theorem for convex polyhedra. We recall that a convex polyhedron is a bounded polytope. Being the intersection of closed half spaces, a convex polyhedron is also closed. Thus any convex polyhedron is the closed convex hull of its extreme points. It can be shown (see Sect. 2.5) that any polytope has at most a finite number of extreme points and hence a convex polyhedron is equal to the convex hull of a finite number of points. The converse can also be established, yielding the following two equivalent characterizations.

Theorem 5. *A convex polyhedron can be described either as a bounded intersection of a finite number of closed half spaces, or as the convex hull of a finite number of points.*

Appendix C
Gaussian Elimination

This appendix describes the method for solving systems of linear equations that has proved to be, not only the most popular, but also the fastest and least susceptible to round-off error accumulation—the method of Gaussian elimination. Attention is directed toward explaining this classical elimination technique itself and its relation to the theory of **LU** decomposition of a non-singular square matrix.

We first note how easily triangular systems of equations can be solved. Thus the system

$$\begin{aligned} a_{11}x_1 \qquad\qquad\qquad\qquad &= b_1 \\ a_{21}x_1 + a_{22}x_2 \qquad\qquad &= b_2 \\ \vdots \qquad\qquad\qquad\qquad\qquad &\quad \vdots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n &= b_n \end{aligned}$$

can be solved recursively as follows:

$$\begin{aligned} x_1 &= b_1/a_{11} \\ x_2 &= (b_2 - a_{21}x_1)/a_{22} \\ &\vdots \\ x_n &= \ (b_n - a_{n1}x_1 - a_{n2}x_2 \ldots - a_{nn-1}x_{n-1})/a_{nn}, \end{aligned}$$

provided that each of the diagonal terms a_{ii}, $i = 1, 2, \ldots, n$ is nonzero (as they must be if the system is nonsingular). This observation motivates us to attempt to reduce an arbitrary system of equations to a triangular one.

Definition. A square matrix $\mathbf{C} = [c_{ij}]$ is said to be *lower triangular* if $c_{ij} = 0$ for $i < j$. Similarly, $\mathbf{C}$ is said to be *upper triangular* if $c_{ij} = 0$ for $i > j$.

In matrix notation, the idea of Gaussian elimination is to somehow find a decomposition of a given $n \times n$ matrix $\mathbf{A}$ in the form $\mathbf{A} = \mathbf{LU}$ where $\mathbf{L}$ is a lower triangular and $\mathbf{U}$ an upper triangular matrix. The system

© Springer International Publishing Switzerland 2016
D.G. Luenberger, Y. Ye, *Linear and Nonlinear Programming*, International Series in Operations Research & Management Science 228,
DOI 10.1007/978-3-319-18842-3

$$\mathbf{Ax} = \mathbf{b} \tag{C.1}$$

can then be solved by solving the two triangular systems

$$\mathbf{Ly} = \mathbf{b}, \quad \mathbf{Ux} = \mathbf{y}. \tag{C.2}$$

The calculation of **L** and **U** together with solution of the first of these systems is usually referred to as *forward elimination*, while solution of the second triangular system is called *back substitution*.

Every nonsingular square matrix **A** has an **LU** decomposition, provided that interchanges of rows of **A** are introduced if necessary. This interchange of rows corresponds to a simple reordering of the system of equations, and hence amounts to no loss of generality in the method. For simplicity of notation, however, we assume that no such interchanges are required.

We turn now to the problem of explicitly determining **L** and **U**, by elimination, for a nonsingular matrix **A**. Given the system, we attempt to transform it so that zeros appear below the main diagonal. Assuming that $a_{11} \neq 0$ we subtract multiples of the first equation from each of the others in order to get zeros in the first column below a_{11}. If we define $m_{k1} = a_{k1}/a_{11}$ and let

$$\mathbf{M}_1 = \begin{bmatrix} 1 & & & \\ -m_{21} & 1 & & \\ -m_{31} & & 1 & \\ \bullet & & & \\ \bullet & & & \\ \bullet & & & \\ -m_{n1} & & & 1 \end{bmatrix},$$

the resulting new system of equations can be expressed as

$$\mathbf{A}^{(2)}\mathbf{x} = \mathbf{b}^{(2)}$$

with

$$\mathbf{A}^{(2)} = \mathbf{M}_1\mathbf{A}, \;\; \mathbf{b}^{(2)} = \mathbf{M}_1\mathbf{b}.$$

The matrix $\mathbf{A}^{(2)} = [a_{ij}^{(2)}]$ has $a_{k1}^{(2)} = 0, \; k > 1$.

Next, assuming $a_{22}^{(2)} \neq 0$, multiples of the second equation of the new system are subtracted from equations 3 through n to yield zeros below $a_{22}^{(2)}$ in the second column. This is equivalent in premultiplying $\mathbf{A}^{(2)}$ and $\mathbf{b}^{(2)}$ by

$$\mathbf{M}_2 = \begin{bmatrix} 1 & 0 & & \\ 0 & 1 & & \\ \bullet & -m_{32} & 1 & \\ \bullet & -m_{42} & & \\ \bullet & \bullet & & \\ & \bullet & & \\ & \bullet & & \\ & -m_{n2} & & 1 \end{bmatrix},$$

where $m_{k2} = a_{k2}^{(2)}/a_{22}^{(2)}$. This yields $\mathbf{A}^{(3)} = \mathbf{M}_2\mathbf{A}^{(2)}$ and $\mathbf{b}^{(3)} = \mathbf{M}_2\mathbf{A}^{(2)}$.

Proceeding in this way we obtain $\mathbf{A}^{(n)} = \mathbf{M}_{n-1}\mathbf{M}_{n-2}\ldots\mathbf{M}_1\mathbf{A}$, an upper triangular matrix which we denote by $\mathbf{U}$. The matrix $\mathbf{M} = \mathbf{M}_{n-1}\mathbf{M}_{n-2}\ldots\mathbf{M}_1$ is a lower triangular matrix, and since $\mathbf{MA} = \mathbf{U}$ we have $\mathbf{A} = \mathbf{M}^{-1}\mathbf{U}$. The matrix $\mathbf{L} = \mathbf{M}^{-1}$ is also lower triangular and becomes the $\mathbf{L}$ of the desired $\mathbf{LU}$ decomposition for $\mathbf{A}$.

The representation for $\mathbf{L}$ can be made more explicit by noting that $\mathbf{M}_k^{-1}$ is the same as $\mathbf{M}_k$ except that the off-diagonal terms have the opposite sign. Furthermore, we have $\mathbf{L} = \mathbf{M}^{-1} = \mathbf{M}_1^{-1}\mathbf{M}_2^{-1}\ldots\mathbf{M}_{n-1}^{-1}$ which is easily verified to be

$$\mathbf{L} = \begin{bmatrix} 1 & 0 & & & & & \\ m_{21} & 1 & & & & & \\ m_{31} & m_{32} & 1 & & & & \\ \bullet & \bullet & & \bullet & & & \\ \bullet & \bullet & & & \bullet & & \\ \bullet & \bullet & & & & \bullet & \\ m_{n1} & m_{n2} & \bullet\bullet\bullet & & & & 1 \end{bmatrix} \bullet$$

Hence $\mathbf{L}$ can be evaluated directly in terms of the calculations required by the elimination process. Of course, an explicit representation for $\mathbf{M} = \mathbf{L}^{-1}$ would actually be more useful but a simple representation for $\mathbf{M}$ does not exist. Thus we content ourselves with the explicit representation for $\mathbf{L}$ and use it in (C.2).

If the original system (C.1) is to be solved for a single $\mathbf{b}$ vector, the vector $\mathbf{y}$ satisfying $\mathbf{Ly} = \mathbf{b}$ is usually calculated simultaneously with $\mathbf{L}$ in the form $\mathbf{y} = \mathbf{b}^{(n)} = \mathbf{Mb}$. The final solution $\mathbf{x}$ is then found by a single back substitution, from $\mathbf{Ux} = \mathbf{y}$. Once the $\mathbf{LU}$ decomposition of $\mathbf{A}$ has been obtained, however, the solution corresponding to any right-hand side can be found by solving the two systems (C.2).

In practice, the diagonal element $a_{kk}^{(k)}$ of $\mathbf{A}^{(k)}$ may become zero or very close to zero. In this case it is important that the kth row be interchanged with a row that is below it. Indeed, for considerations of numerical accuracy, it is desirable to continuously introduce row interchanges of this type in such a way to insure $|m_{ij}| \leqslant 1$ for all i, j. If this is done, the Gaussian elimination procedure has exceptionally good stability properties.

Appendix D
Basic Network Concepts

This appendix describes some of the basic graph and network terminology and concepts necessary for the development of this alternative approach.

A *graph* consists of a finite collection of elements called *nodes* together with a subset of unordered pairs of the nodes called *arcs*. The nodes of a graph are usually numbered, say, $1, 2, 3, \ldots, n$. An arc between nodes i and j is then represented by the unordered pair $(i, \ j)$. A graph is typically represented as shown in Fig. D.1. The nodes are designated by circles, with the number inside each circle denoting the index of that node. The arcs are represented by the lines between the nodes.

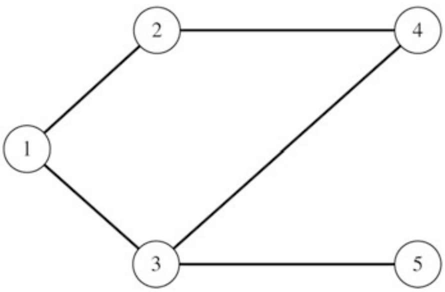

Fig. D.1 A graph

There are a number of other elementary definitions associated with graphs that are useful in describing their structure. A *chain* between nodes i and j is a sequence of arcs connecting them. The sequence must have the form $(i, \ k_1), \ (k_1, \ k_2), \ (k_2, \ k_3), \ \ldots, \ (k_m, \ j)$. In Fig. D.1, (1, 2), (2, 4), (4, 3) is a chain between nodes 1 and 3. If a direction of movement along a chain is specified—say from node i to node j—it is then called a *path* from i to j. A *cycle* is a chain leading from node i back to node i. The chain (1, 2), (2, 4), (4, 3), (3, 1) is a cycle for the graph in Fig. D.1.

© Springer International Publishing Switzerland 2016
D.G. Luenberger, Y. Ye, *Linear and Nonlinear Programming*, International Series in Operations Research & Management Science 228,
DOI 10.1007/978-3-319-18842-3

A graph is *connected* if there is a chain between any two nodes. Thus, the graph of Fig. D.1 is connected. A graph is a *tree* if it is connected and has no cycles. Removal of any one of the arcs (1, 2), (1, 3), (2, 4), (3, 4) would transform the graph of Fig. D.1 into a tree. Sometimes we consider a tree within a graph G, which is just a tree made up of a subset of arcs from G. Such a tree is a *spanning* tree if it touches all nodes of G. It is easy to see that a graph is connected if and only if it contains a spanning tree.

In *directed graphs* a sense of orientation is given to each arc. In this case an arc is considered to be an *ordered pair* of nodes (i, j), and we say that the arc is from node i to node j. This is indicated on the graph by having an arrow on the arc pointing from i to j as shown in Fig. D.2. When working with directed graphs, some node pairs may have an arc in both directions between them. Rather than explicitly indicating both arcs in such a case, it is customary to indicate a single undirected arc. The notions of paths and cycles can be directly applied to directed graphs. In addition we say that node j is *reachable* from i if there is a path from node i to j.

In addition to the visual representation of a directed graph characterized by Fig. D.2, another common method of representation is in terms of a graph's node-arc incidence matrix. This is constructed by listing the nodes vertically and the arcs horizontally. Then in the column under arc (i, j), a $+1$ is placed in the position corresponding to node i and a -1 is placed in the position corresponding to node j. The incidence matrix for the graph of Fig. D.2 is shown in Table D.1.

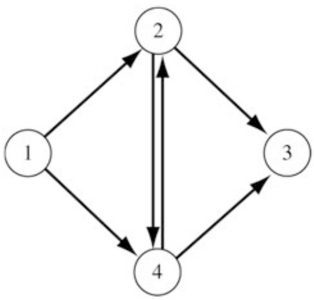

Fig. D.2 A directed graph

	(1,2)	(1,4)	(2,3)	(2,4)	(4,2)
1	1	1			
2	–1		1	1	–1
3			–1		
4		–1		–1	1

Table D.1 Incidence matrix for example

Clearly, all information about the structure of the graph is contained in the node-arc incidence matrix. This representation is often very useful for computational purposes, since it is easily stored in a computer.

D.1 Flows in Networks

A graph is an effective way to represent the communication structure between nodes. When there is the possibility of *flow* along the arcs, we refer to the directed graph as a *network*. In applications the network might represent a transportation system or a communication network, or it may simply be a representation used for mathematical purposes (such as in the assignment problem).

A flow in a given directed arc $(i, \ j)$ is a number $x_{ij} \geqslant 0$. Flows in the arcs of the network must jointly satisfy a conservation criterion at each node. Specifically, unless the node is a *source* or *sink* as discussed below, flow cannot be created or lost at a node; the total flow into a node must equal the total flow out of the node. Thus at each such node i

$$\sum_{j=1}^{n} x_{ij} - \sum_{k=1}^{n} x_{ki} = 0.$$

The first sum is the total flow *from* i, and the second sum is the total flow *to* i. (Of course x_{ij} does not exist if there is no arc from i to j.) It should be clear that for nonzero flows to exist in a network without sources or sinks, the network must contain a cycle.

In many applications, some nodes are in fact designated as *sources* or *sinks* (or, alternatively, supply nodes or demand nodes). The net flow *out* of a source may be positive, and the level of this net flow may either be fixed or variable, depending on the application. Similarly, the net flow *into* a sink may be positive.

D.2 Tree Procedure

Recall that node j is *reachable* from node i in a directed graph if there is a path from node i to node j. For simple graphs, determination of reachability can be accomplished by inspection, but for large graphs it generally cannot. The problem can be solved systematically by a process of repeatedly labeling and scanning various nodes in the graph. This procedure is the backbone of a number of methods for solving more complex graph and network problems, as illustrated later. It can also be used to establish quickly some important theoretical results.

Assume that we wish to determine whether a path from node 1 to node m exists. At each step of the algorithm, each node is either unlabeled, labeled but unscanned, or labeled and scanned. The procedure consists of these steps:

Step 1. Label node 1 with any mark. All other nodes are unlabeled.

Step 2. For any labeled but unscanned node i, scan the node by finding all unlabeled nodes reachable from i by a single arc. Label these nodes with an i.

Step 3. If node m is labeled, stop; a *breakthrough* has been achieved—a path exists. If no unlabeled nodes can be labeled, stop; no connecting path exists. Otherwise, go to Step 2.

The process is illustrated in Fig. D.3, where a path between nodes 1 and 10 is sought. The nodes have been labeled and scanned in the order 1, 2, 3, 5, 6, 8, 4, 7, 9, 10. The labels are indicated close to the nodes. The arcs that were used in the scanning processes are indicated by heavy lines. Note that the collection of nodes and arcs selected by the process, regarded as an undirected graph, form a tree—a graph without cycles. This, of course, accounts for the name of the process, the tree procedure. If one is interested only in determining whether a connecting path exists and does not need to find the path itself, then the labels need only be simple check marks rather than node indices. However, if node indices are used as labels, then after successful completion of the algorithm, the actual connecting path can be found by tracing backward from node m by following the labels. In the example, one begins at 10 and moves to node 7 as indicated; then to 6, 3, and 1. The path follows the reverse of this sequence.

It is easy to prove that the algorithm does indeed resolve the issue of the existence of a connecting path. At each stage of the process, either a new node is labeled, it is impossible to continue, or node m is labeled and the process is successfully terminated. Clearly, the process can continue for at most $n - 1$ stages, where n is the number of nodes in the graph. Suppose at some stage it is impossible to continue. Let S be the set of labeled nodes at that stage and let $\bar{S}$ be the set of unlabeled nodes. Clearly, node 1 is contained in S, and node m is contained in $\bar{S}$. If there were a path connecting node 1 with node m, then there must be an arc in that path from a node k in S to a node in $\bar{S}$. However, this would imply that node k was not scanned, which is a contradiction. Conversely, if the algorithm does continue until reaching node m, then it is clear that a connecting path can be constructed backward as outlined above.

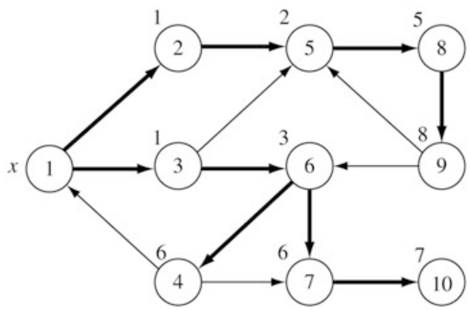

Fig. D.3 The scanning procedure

D.3 Capacitated Networks

In some network applications it is useful to assume that there are upper bounds on the allowable flow in various arcs. This motivates the concept of a capacitated network. A *capacitated network* is a network in which some arcs are assigned non-negative capacities, which define the maximum allowable flow in those arcs. The capacity of an arc (i, j) is denoted k_{ij}, and this capacity is indicated on the graph by placing the number k_{ij} adjacent to the arc. Figure 2.1 shows an example of a network with the capacities indicated. Thus the capacity from node 1 to node 2 is 12, while that from node 2 to node 1 is 6.

Bibliography

[A1] J. Abadie, J. Carpentier, Generalization of the Wolfe reduced gradient method to the case of nonlinear constraints, in *Optimization*, ed. by R. Fletcher (Academic, London, 1969), pp. 37–47

[A2] H. Akaike, On a successive transformation of probability distribution and its application to the analysis of the optimum gradient method. Ann. Inst. Stat. Math. **11**, 1–17 (1959)

[3] A.Y. Alfakih, A. Khandani, H. Wolkowicz, Solving euclidean distance matrix completion problems via semidefinite programming. Comput. Opt. Appl. **12**, 13–30 (1999)

[A3] F. Alizadeh, Combinatorial optimization with interior point methods and semi-definite matrices, Ph.D. thesis, University of Minnesota, Minneapolis, 1991

[A4] F. Alizadeh, Optimization over the positive semi-definite cone: interior-point methods and combinatorial applications, in *Advances in Optimization and Parallel Computing*, ed. by P.M. Pardalos (North Holland, Amsterdam, 1992), pp. 1–25

[6] E.D. Andersen, MOSEK: high performance software for large-scale LP, QP, SOCP, SDP and MIP, http://www.mosek.com/ (1997)

[A5] E.D. Andersen, Y. Ye, On a homogeneous algorithm for the monotone complementarity problem. Math. Prog. **84**, 375–400 (1999)

[A6] K.M. Anstreicher, D. den Hertog, C. Roos, T. Terlaky, A long step barrier method for convex quadratic programming. Algorithmica **10**, 365–382 (1993)

[A7] H.A. Antosiewicz, W.C. Rheinboldt, Numerical analysis and functional analysis, in *Survey of Numerical Analysis*, ed. by J. Todd, Chap. 14 (McGraw-Hill, New York, 1962)

[A8] L. Armijo, Minimization of functions having lipschitz continuous first-partial derivatives. Pac. J. Math. **16**(1), 1–3 (1966)

© Springer International Publishing Switzerland 2016

D.G. Luenberger, Y. Ye, *Linear and Nonlinear Programming*, International Series in Operations Research & Management Science 228,
DOI 10.1007/978-3-319-18842-3

[A9] K.J. Arrow, L. Hurwicz, Gradient method for concave programming, I.: local results, in *Studies in Linear and Nonlinear Programming*, ed. by K.J. Arrow, L. Hurwicz, H. Uzawa (Stanford University Press, Stanford, CA, 1958)

[12] E. Balas, S. Ceria, G. Cornuejols, A lift-and-project cutting plane algorithm for mixed 0-1 programs. Math. Program. **58**, 295–324 (1993)

[B1] R.H. Bartels, A numerical investigation of the simplex method. Technical Report No. CS 104, Computer Science Department, Stanford University, Stanford, CA (31 July 1968)

[B2] R.H. Bartels, G.H Golub, The simplex method of linear programming using LU decomposition. Commun. ACM **12**(5), 266–268 (1969)

[15] A. Barvinok, A remark on the rank of positive semidefinite matrices subject to affine constraints. Discrete Comput. Geom. **25**, 23–31 (2001)

[16] A. Barvinok, *A Course in Convexity*. Graduate Studies in Mathematics, vol. 54 (American Mathematical Society, Providence, RI, 2002)

[17] J. Barzilai, J.M. Borwein, Two-point step size gradient methods. IMA J. Numer. Anal. **8**, 141–148 (2008)

[B3] D.A., Bayer, J.C. Lagarias, The nonlinear geometry of linear programming, part i: affine and projective scaling trajectories. Trans. Am. Math. Soc **314**(2), 499–526 (1989)

[B4] D.A. Bayer, J.C. Lagarias, The nonlinear geometry of linear programming, part ii: legendre transform coordinates. Trans. Am. Math. Soc. **314**(2), 527–581 (1989)

[B5] M.S. Bazaraa, J.J. Jarvis, *Linear Programming and Network Flows* (Wiley, New York, 1977)

[B6] M.S. Bazaraa, J.J. Jarvis, H.F. Sherali, Karmarkar's projective algorithm (Chap. 8.4), pp. 380–394; Analysis of Karmarkar's algorithm (Chap. 8.5), pp. 394–418, in *Linear Programming and Network Flows*, 2nd edn. (Wiley New York, 1990)

[B7] E.M.L. Beale, in *Numerical Methods, Nonlinear Programming*, ed. by J. Abadie (North-Holland, Amsterdam, 1967)

[23] A. Beck, M. Teboulle, A fast iterative shrinkage-thresholding algorithm for linear inverse problems. SIAM J. Imaging Sci. **2**, 183–202 (2009)

[B8] F.S. Beckman, The solution of linear equations by the conjugate gradient method, in *Mathematical Methods for Digital Computers*, ed. by A. Ralston, H.S. Wilf, vol. 1 (Wiley, New York, 1960)

[25] S.J. Benson, Y. Ye, X. Zhang, Solving large-scale sparse semidefinite programs for combinatorial optimization. SIAM J. Optim. **10**, 443–461 (2000)

[26] A. Ben-Tal, A. Nemirovski, Structural design via semidefinite programming, in *Handbook on Semidefinite Programming* (Kluwer, Boston, 2000), pp. 443–467

[B9] D.P. Bertsekas, Partial conjugate gradient methods for a class of optimal control problems. IEEE Trans. Autom. Control **19**, 209–217 (1973)

[B10] D.P. Bertsekas, Multiplier methods: a survey. Automatica **12**(2), 133–145 (1976)

[B11] D.P. Bertsekas, *Constrained Optimization and Lagrange Multiplier Methods* (Academic, New York, 1982)

[B12] D.P. Bertsekas, *Nonlinear Programming* (Athena Scientific, Belmont, 1995)

[31] D. Bertsimas, Y. Ye, Semidefinite relaxations, multivariate normal distributions, and order statistics, in *Handbook of Combinatorial Optimization* (Springer, New York, 1999), pp. 1473–1491

[B13] D.M. Bertsimas, J.N. Tsitsiklis, *Linear Optimization* (Athena Scientific, Belmont, 1997)

[B14] M.C. Biggs, Constrained minimization using recursive quadratic programming: some alternative sub-problem formulations, in *Towards Global Optimization*, ed. by L.C.W. Dixon, G.P. Szego (North-Holland, Amsterdam, 1975)

[B15] M.C. Biggs, On the convergence of some constrained minimization algorithms based on recursive quadratic programming. J. Inst. Math. Appl. **21**, 67–81 (1978)

[B16] G. Birkhoff, Three observations on linear algebra. Rev. Univ. Nac. Tucumán, Ser. A. **5**, 147–151 (1946)

[B17] P. Biswas, Y. Ye, Semidefinite programming for ad hoc wireless sensor network localization, in *Proceedings of the 3rd IPSN*, 2004, pp. 46–54

[B18] R.E. Bixby, Progress in linear programming. ORSA J. Comput. **6**(1), 15–22 (1994)

[B19] R.G. Bland, New finite pivoting rules for the simplex method. Math. Oper. Res. **2**(2), 103–107 (1977)

[B20] R.G. Bland, D. Goldfarb, M.J. Todd, The ellipsoidal method: a survey. Oper. Res. **29**, 1039–1091 (1981)

[B21] L. Blum, F. Cucker, M. Shub, S. Smale, *Complexity and Real Computation* (Springer, New York, 1996)

[41] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, Distributed optimization and statistical learning via the alternating direction method of multipliers. Found. Trends Mach. Learn. **3**, 1–122 (2010)

[B22] S. Boyd, L.E. Ghaoui, E. Feron, V. Balakrishnan, *Linear Matrix Inequalities in System and Control Science* (SIAM, Philadelphia, 1994)

[B23] S. Boyd, L. Vandenberghe, *Convex Optimization* (Cambridge University Press, Cambridge, 2004)

[B24] C.G. Broyden, Quasi-Newton methods and their application to function minimization. Math. Comput. **21**, 368–381 (1967)

[B25] C.G. Broyden, The convergence of a class of double rank minimization algorithms: parts I and II. J. Inst. Math. Appl. **6**, 76–90, 222–231 (1970)

[B26] T. Butler, A.V. Martin, On a method of courant for minimizing functionals. J. Math. Phys. **41**, 291–299 (1962)

[C1] C.W. Carroll, The created response surface technique for optimizing nonlinear restrained systems. Oper. Res. **9**(12), 169–184 (1961)

[C2] A. Charnes, Optimality and degeneracy in linear programming. Econometrica **20**, 160–170 (1952)

[C3] A. Charnes, C.E. Lemke, The bounded variables problem. ONR Research Memorandum 10, Graduate School of Industrial Administration, Carnegie Institute of Technology, Pittsburgh (1954)

[50] C.H. Chen, B.S. He, Y.Y. Ye, X.M. Yuan, The direct extension of ADMM for multi-block convex minimization problems is not necessarily convergent. Math. Program. (2014). doi:10.1007/s10107-014-0826-5

[C4] A. Cohen, Rate of convergence for root finding and optimization algorithms. Ph.D. dissertation, University of California, Berkeley, 1970

[C5] S.A. Cook, The complexity of theorem-proving procedures, in *Proceedings of 3rd ACM Symposium on the Theory of Computing*, 1971, pp. 151–158

[C6] R.W. Cottle, *Linear Programming*. Lecture Notes for MS& E 310 (Stanford University, Stanford, 2002)

[C7] R. Cottle, J.S. Pang, R.E. Stone, Interior-Point Methods (Chap. 5.9), in *The Linear Complementarity Problem* (Academic, Boston, 1992), pp. 461–475

[C8] R. Courant, Calculus of variations and supplementary notes and exercises (mimeographed notes), supplementary notes by M. Kruskal and H. Rubin, revised and amended by J. Moser, New York University (1962)

[C9] J.B. Crockett, H. Chernoff, Gradient methods of maximization. Pac. J. Math. **5**, 33–50 (1955)

[C10] H. Curry, The method of steepest descent for nonlinear minimization problems. Q. Appl. Math. **2**, 258–261 (1944)

[58] Y.-H. Dai, R. Fletcher, Projected Barzilai-Borwein methods for large-scale box-constrained quadratic programming. Numer. Math. **100**, 21–47 (2005)

[D1] J.W. Daniel, The conjugate gradient method for linear and nonlinear operator equations. SIAM J. Numer. Anal. **4**(1), 10–26 (1967)

[D2] G.B. Dantzig, Maximization of a linear function of variables subject to linear inequalities (Chap. XXI), in *Activity Analysis of Production and Allocation*, ed. by T.C. Koopmans. Cowles Commission Monograph, vol. 13 (Wiley, New York, 1951)

[D3] G.B., Dantzig, Application of the simplex method to a transportation problem, in *Activity Analysis of Production and Allocation*, ed. by T.C. Koopmans (Wiley, New York, 1951), pp. 359–373

[D4] G.B. Dantzig, Computational algorithm of the revised simplex method. RAND Report RM-1266, The RAND Corporation, Santa Monica, CA (1953)

[D5] G.B. Dantzig, Variables with upper bounds in linear programming. RAND Report RM-1271, The RAND Corporation, Santa Monica, CA (1954)

[D6] G.B. Dantzig, *Linear Programming and Extensions* (Princeton University Press, Princeton, 1963)

[D7] G.B. Dantzig, L.R. Ford Jr., D.R. Fulkerson, A primal-dual algorithm, in *Linear Inequalities and Related Systems*. Annals of Mathematics Study, vol. 38 (Princeton University Press, Princeton, NJ, 1956), pp. 171–181

[D8] G.B. Dantzig, A. Orden, P. Wolfe, Generalized simplex method for minimizing a linear form under linear inequality restraints. RAND Report RM-1264, The RAND Corporation, Santa Monica, CA (1954)

[D9] G.B. Dantzig, M.N. Thapa, *Linear Programming 1: Introduction* (Springer, New York, 1997)

[D10] G.B. Dantzig, M.N. Thapa, *Linear Programming 2: Theory and Extensions* (Springer, New York, 2003)

[D11] G.B. Dantzig, P. Wolfe, Decomposition principle for linear programs. Oper. Res. **8**, 101–111 (1960)

[D12] W.C. Davidon, Variable metric method for minimization. Research and Development Report ANL-5990 (Ref.) U.S. Atomic Energy Commission, Argonne National Laboratories (1959)

[D13] W.C. Davidon, Variance algorithm for minimization. Comput. J. **10**, 406–410 (1968)

[72] E. de Klerk, C. Roos, T. Terlaky, Initialization in semidefinite programming via a self–dual skew–symmetric embedding. Oper. Res. Lett. **20**, 213–221 (1997)

[D14] R.S. Dembo, S.C. Eisenstat, T. Steinhaug, Inexact Newton methods. SIAM J. Numer. Anal. **19**(2), 400–408 (1982)

[D15] J.E. Dennis Jr., J.J. Moré, Quasi-Newton methods, motivation and theory. SIAM Rev. **19**, 46–89 (1977)

[D16] J.E. Dennis Jr., R.E. Schnabel, Least change secant updates for quasi-Newton methods. SIAM Rev. **21**, 443–469 (1979)

[D17] L.C.W. Dixon, Quasi-Newton algorithms generate identical points. Math. Prog. **2**, 383–387 (1972)

[E1] B.C. Eaves, W.I. Zangwill, Generalized cutting plane algorithms. Working Paper No. 274, Center for Research in Management Science, University of California, Berkeley (July 1969)

[78] J. Eckstein, D.P. Bertsekas, On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators. Math. Program. **55**, 293–318 (1992)

[E2] H. Everett III, Generalized lagrange multiplier method for solving problems of optimum allocation of resources. Oper. Res. **11**, 399–417 (1963)

[F1] D.K. Faddeev, V.N. Faddeeva, *Computational Methods of Linear Algebra* (W. H. Freeman, San Francisco, CA, 1963)

[F2] S.C. Fang, S. Puthenpura, *Linear Optimization and Extensions* (Prentice-Hall, Englewood Cliffs, NJ, 1994)

[F3] W. Fenchel, *Convex Cones, Sets, and Functions.* Lecture Notes (Department of Mathematics, Princeton University, Princeton, NJ, 1953)

[F4] A.V. Fiacco, G.P. McCormick, *Nonlinear Programming: Sequential Unconstrained Minimization Techniques* (Wiley, New York, 1968)

[F5] A.V. Fiacco, G.P. McCormick, *Nonlinear Programming: Sequential Unconstrained Minimization Techniques* (Wiley, New York, 1968). Reprint: Volume 4 of *SIAM Classics in Applied Mathematics* (SIAM Publications, Philadelphia, PA, 1990)

[F6] R. Fletcher, A new approach to variable metric algorithms. Comput. J. **13**(13), 317–322 (1970)

[F7] R. Fletcher, An exact penalty function for nonlinear programming with inequalities. Math. Program. **5**, 129–150 (1973)

[F8] R. Fletcher, Conjugate gradient methods for indefinite systems. Numerical Analysis Report, 11. Department of Mathematics, University of Dundee, Scotland (September 1975)

[F9] R. Fletcher, *Practical Methods of Optimization 1: Unconstrained Optimization* (Wiley, Chichester, 1980)

[F10] R. Fletcher, *Practical Methods of Optimization 2: Constrained Optimization* (Wiley, Chichester, 1981)

[F11] R. Fletcher, M.J.D. Powell, A rapidly convergent descent method for minimization. Comput. J. **6**, 163–168 (1963)

[F12] R. Fletcher, C.M. Reeves, Function minimization by conjugate gradients. Comput. J. **7**, 149–154 (1964)

[F13] L.K. Ford Jr., D.K. Fulkerson, *Flows in Networks* (Princeton University Press, Princeton, NJ, 1962)

[F14] G.E. Forsythe, On the asymptotic directions of the s-dimensional optimum gradient method. Numer. Math. **11**, 57–76 (1968)

[F15] G.E. Forsythe, C.B. Moler, *Computer Solution of Linear Algebraic Systems* (Prentice-Hall, Englewood Cliffs, NJ, 1967)

[F16] G.E. Forsythe, W.R. Wasow, *Finite-Difference Methods for Partial Differential Equations* (Wiley, New York, 1960)

[96] M. Fortin, R. Glowinski, On decomposition-coordination methods using an augmented Lagrangian, in *Augmented Lagrangian Methods: Applications to the Solution of Boundary Problems*, ed. by M. Fortin, R. Glowinski (North-Holland, Amsterdam, 1983)

[F17] K. Fox, *An Introduction to Numerical Linear Algebra* (Clarendon Press, Oxford, 1964)

[98] M. Frank, P. Wolfe, An algorithm for quadratic programming. Naval Res. Logist. Q. **3**, 95–110 (1956)

[F18] R.M. Freund, Polynomial-time algorithms for linear programming based only on primal scaling and projected gradients of a potential function. Math. Program. **51**, 203–222 (1991)

[F19] K.R. Frisch, The logarithmic potential method for convex programming. Unpublished Manuscript, Institute of Economics, University of Oslo, Oslo (1955)

[G1] D. Gabay, Reduced quasi-Newton methods with feasibility improvement for nonlinear constrained optimization, in *Mathematical Programming Studies*, vol. 16 (North-Holland, Amsterdam, 1982), pp. 18–44

[102] D. Gabay, B. Mercier, A dual algorithm for the solution of nonlinear variational problems via finite element approximations. Comput. Math. Appl. **2**, 17–40 (1976)

[G2] D. Gale, *The Theory of Linear Economic Models* (McGraw-Hill, New York, 1960)

[G3] U.M. Garcia-Palomares, O.L. Mangasarian, Superlinearly convergent quasi-Newton algorithms for nonlinearly constrained optimization problems. Math. Program. **11**, 1–13 (1976)

[G4] S.I. Gass, *Linear Programming*, 3rd edn. (McGraw-Hill, New York, 1969)

[G5] P.E. Gill, W. Murray, M.A. Saunders, J.A. Tomlin, M.H. Wright, On projected Newton barrier methods for linear programming and an equivalence to Karmarkar's projective method. Math. Program. **36**, 183–209 (1986)

[G6] P.E., Gill, W. Murray, Quasi-Newton methods for unconstrained optimization. J. Inst. Math. Appl. **9**, 91–108 (1972)

[G7] P.E. Gill, W. Murray, M.H. Wright, *Practical Optimization* (Academic, London, 1981)

[109] R. Glowinski, A. Marrocco, Approximation par éléments finis d'ordre un et résolution par pénalisation-dualité d'une classe de problèmes non linéaires. R.A.I.R.O. R2 **2**, 41–76 (1975)

[G8] M.X. Goemans, D.P. Williamson, Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. J. Assoc. Comput. Mach. **42**, 1115–1145 (1995)

[G9] D. Goldfarb, A family of variable metric methods derived by variational means. Math. Comput. **24**, 23–26 (1970)

[112] D. Goldfarb, G. Iyengar, Robust portfolio selection problems. Math. Oper. Res. **28**, 1–38 (2002)

[G10] D. Goldfarb, M.J. Todd, Linear programming, in *Optimization*, ed. by G.L. Nemhauser, A.H.G. Rinnooy Kan, M.J. Todd. Handbooks in Operations Research and Management Science, vol. 1 (North Holland, Amsterdam, 1989), pp. 141–170

[G11] D. Goldfarb, D. Xiao, A primal projective interior point method for linear programming. Math. Program. **51**, 17–43 (1991)

[G12] A.A. Goldstein, On steepest descent. SIAM J. Control **3**, 147–151 (1965)

[G13] C.C. Gonzaga, An algorithm for solving linear programming problems in $O(n^3L)$ operations, in *Progress in Mathematical Programming: Interior Point and Related Methods*, ed. by N. Megiddo (Springer, New York, 1989), pp. 1–28

[G14] C.C. Gonzaga, M.J. Todd, An $O(\sqrt{n}L)$-iteration large-step primal-dual affine algorithm for linear programming. SIAM J. Optim. **2**, 349–359 (1992)

[G15] J. Greenstadt, Variations on variable metric methods. Math. Comput. **24**, 1–22 (1970)

[H1] G. Hadley, *Linear Programming* (Addison-Wesley, Reading, MA, 1962)

[H2] G. Hadley, *Nonlinear and Dynamic Programming* (Addison-Wesley, Reading, MA, 1964)

[H3] S.P. Han, A globally convergent method for nonlinear programming. J. Optim. Theory Appl. **22**(3), 297–309 (1977)

[H4] H. Hancock, *Theory of Maxima and Minima* (Ginn, Boston, 1917)

[H5] J. Hartmanis, R.E. Stearns, On the computational complexity of algorithms. Trans. Am. Math. Soc. **117**, 285–306 (1965)

[124] B.S. He, X.M. Yuan, On the $O(1/n)$ convergence rate of the Douglas-Rachford alternating direction method. SIAM J. Numer. Anal. **50**, 700–709 (2012)

[H6] D. den Hertog, Interior point approach to linear, quadratic and convex programming, algorithms and complexity, Ph.D. thesis, Faculty of Mathematics and Informatics, TU Delft, BL Delft, 1992

[H7] M.R. Hestenes, The conjugate gradient method for solving linear systems, in *Proceeding of Symposium in Applied Mathematics*, vol. VI, Numerical Analysis (McGraw-Hill, New York 1956), pp. 83–102

[H8] M.R. Hestenes, Multiplier and gradient methods. J. Opt. Theory Appl. **4**(5), 303–320 (1969)

[H9] M.R. Hestenes, *Conjugate-Direction Methods in Optimization* (Springer, Berlin, 1980)

[H10] M.R. Hestenes, E.L. Stiefel, Methods of conjugate gradients for solving linear systems. J. Res. Natl. Bur. Stand. Sect. B **49**, 409–436 (1952)

[H11] F.L. Hitchcock, The distribution of a product from several sources to numerous localities. J. Math. Phys. **20**, 224–230 (1941)

[H12] P. Huard, Resolution of mathematical programming with nonlinear constraints by the method of centers, in *Nonlinear Programming*, ed. by J. Abadie (North Holland, Amsterdam, 1967), pp. 207–219

[H13] H.Y. Huang, Unified approach to quadratically convergent algorithms for function minimization. J. Optim. Theory Appl. **5**, 405–423 (1970)

[H14] L. Hurwicz, Programming in linear spaces, in *Studies in Linear and Nonlinear Programming*, ed. by K.J. Arrow, L. Hurwicz, H. Uzawa (Stanford University Press, Stanford, CA, 1958)

[I1] E. Isaacson, H.B. Keller, *Analysis of Numerical Methods* (Wiley, New York, 1966)

[J1] W. Jacobs, The caterer problem. Naval Res. Logist. Q. **1**, 154–165 (1954)

[J2] F. Jarre, Interior-point methods for convex programming. Appl. Math. Optim. **26**, 287–311 (1992)

[137] W.B. Johnson, J. Lindenstrauss, Extensions of lipshitz mapping into Hilbert space. Comtemp. Math. **26**, 189–206 (1984)

[K1] S. Karlin, *Mathematical Methods and Theory in Games, Programming, and Economics*, vol. I (Addison-Wesley, Reading, MA, 1959)

[K2] N.K. Karmarkar, A new polynomial-time algorithm for linear programming. Combinatorica **4**, 373–395 (1984)

[K3] J.E. Kelley, The cutting-plane method for solving convex programs. J. Soc. Ind. Appl. Math. **VIII**(4), 703–712 (1960)

[K4] L.G. Khachiyan, A polynomial algorithm for linear programming. Doklady Akad. Nauk USSR **244**, 1093–1096 (1979). Translated in Soviet Math. Doklady **20**, 191–194 (1979)

[K5] V. Klee, G.J. Minty, How good is the simplex method, in *Inequalities III*, ed. by O. Shisha (Academic, New York, 1972)

[K6] M. Kojima, S. Mizuno, A. Yoshise, A polynomial-time algorithm for a class of linear complementarity problems. Math. Program. **44**, 1–26 (1989)

[K7] M. Kojima, S. Mizuno, A. Yoshise, An $O(\sqrt{n}L)$ iteration potential reduction algorithm for linear complementarity problems. Math. Program. **50**, 331–342 (1991)

[K8] T.C. Koopmans, Optimum utilization of the transportation system, in *Proceedings of the International Statistical Conference*, Washington, DC, 1947

[K9] J. Kowalik, M.R. Osborne, *Methods for Unconstrained Optimization Problems* (Elsevier, New York, 1968)

[K10] H.W. Kuhn, The Hungarian method for the assignment problem. Naval Res. Logist. Q. **2**, 83–97 (1955)

[K11] H.W. Kuhn, A.W. Tucker, Nonlinear programming, in *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*, ed. by J. Neyman (University of California Press, Berkeley/Los Angeles, CA, 1961), pp. 481–492

[L1] C. Lanczos, *Applied Analysis* (Prentice-Hall, Englewood Cliffs, NJ, 1956)

[150] J.B. Lasserre, Global optimization with polynomials and the problem of moments related. SIAM J. Optim. **11**, 796–817 (2001)

[151] M. Laurent, Matrix completion problems. Encycl. Optim. **3**, 221–229 (2001)

[L2] E. Lawler, *Combinatorial Optimization: Networks and Matroids* (Holt, Rinehart, and Winston, New York, 1976)

[L3] C. Lemarechal, R. Mifflin, Nonsmooth optimization, in *IIASA Proceedings III* (Pergamon Press, Oxford, 1978)

[L4] C.E. Lemke, The dual method of solving the linear programming problem. Naval Res. Logist. Q. **1**(1), 36–47 (1954)

[L5] E.S. Levitin, B.T. Polyak, Constrained minimization methods. Zh. vychisl. Math. Math. Fiz **6**(5), 787–823 (1966)

[156] M.S. Lobo, L. Vandenberghe, S. Boyd, Applications of second-order cone programming. Linear Algebra Appl. **284**, 193–228 (1998)

[L6] C. Loewner, Über monotone Matrixfunktionen. Math. Zeir. **38**, 177–216 (1934). Also see C. Loewner, Advanced matrix theory, mimeo notes, Stanford University, 1957

[L7] F.A. Lootsma, Boundary properties of penalty functions for constrained minimization, Doctoral dissertation, Technical University, Eindhoven, May 1970

[159] L. Lovász, A. Shrijver, Cones of matrices and setfunctions, and $0-1$ optimization. SIAM J. Optim. **1**, 166–190 (1990)

[160] Z. Lu, L. Xiao, On the complexity analysis of randomized block-coordinate descent methods. Math. Program. (2013). doi: 10.1007/s10107-014-0800-2

[L8] D.G. Luenberger, *Optimization by Vector Space Methods* (Wiley, New York, 1969)

[L9] D.G. Luenberger, Hyperbolic pairs in the method of conjugate gradients. SIAM J. Appl. Math. **17**(6), 1263–1267 (1969)

[L10] D.G. Luenberger, A combined penalty function and gradient projection method for nonlinear programming, Internal Memo, Department of Engineering-Economic Systems, Stanford University (June 1970)

[L11] D. G. Luenberger, The conjugate residual method for constrained minimization problems. SIAM J. Numer. Anal. **7**(3), 390–398 (1970)
[L12] D.G. Luenberger, Control problems with kinks. IEEE Trans. Autom. Control **AC-15**(5), 570–575 (1970)
[L13] D.G. Luenberger, Convergence rate of a penalty-function scheme. J. Optim. Theory Appl. **7**(1), 39–51 (1971)
[L14] D.G. Luenberger, The gradient projection method along geodesics. Manag. Sci. **18**(11), 620–631 (1972)
[L15] D.G. Luenberger, *Introduction to Linear and Nonlinear Programming*, 1st edn. (Addison-Wesley, Reading, MA, 1973)
[L16] D.G. Luenberger, *Linear and Nonlinear Programming*, 2nd edn. (Addison-Wesley, Reading, MA, 1984)
[L17] D.G. Luenberger, An approach to nonlinear programming. J. Optim. Theory Appl. **11**(3), 219–227 (1973)
[171] Z.Q. Luo, W. Ma, A.M. So, Y. Ye, S. Zhang, Semidefinite relaxation of quadratic optimization problems. IEEE Signal Process. Mag. **27**, 20–34 (2010)
[L18] Z.Q. Luo, J. Sturm, S. Zhang, Conic convex programming and self-dual embedding. Optim. Methods Softw. **14**, 169–218 (2000)
[L19] I.J. Lustig, R.E. Marsten, D.F. Shanno, On implementing mehrotra's predictor-corrector interior point method for linear programming. SIAM J. Optim. **2**, 435–449 (1992)
[M1] N. Maratos, Exact penalty function algorithms for finite dimensional and control optimization problems, Ph.D. thesis, Imperial College Sci. Tech., University of London, 1978
[M2] G.P. McCormick, Optimality criteria in nonlinear programming, in *Nonlinear Programming, SIAM-AMS Proceedings*, vol. IX, 1976, pp. 27–38
[M3] L. McLinden, The analogue of *Moreau*'*s* proximation theorem, with applications to the nonlinear complementarity problem. Pac. J. Math. **88**, 101–161 (1980)
[M4] N. Megiddo, Pathways to the optimal set in linear programming, in *Progress in Mathematical Programming: Interior Point and Related Methods*, ed. by N. Megiddo (Springer, New York, 1989), pp. 131–158
[M5] S. Mehrotra, On the implementation of a primal-dual interior point method. SIAM J. Optim. **2**(4), 575–601 (1992)
[M6] S. Mizuno, M.J. Todd, Y. Ye, On adaptive step primal-dual interior point algorithms for linear programming. Math. Oper. Res. **18**, 964–981 (1993)
[180] R.D.C. Monteiro, B.F. Svaiter, Iteration-complexity of block-decomposition algorithms and the alternating direction method of multipliers. SIAM J. Optim. **23**, 475–507 (2013)
[M7] R.D.C. Monteiro, I. Adler, Interior path following primal-dual algorithms: part i: linear programming. Math. Program. **44**, 27–41 (1989)
[M8] D.D. Morrison, Optimization by least squares. SIAM J. Numer. Anal. **5**, 83–88 (1968)

[M9] B.A. Murtagh, *Advanced Linear Programming* (McGraw-Hill, New York, 1981)

[M10] B.A. Murtagh, R.W.H. Sargent, A constrained minimization method with quadratic convergence (Chap. 14), in *Optimization*, ed. by R. Fletcher (Academic, London, 1969)

[M11] K.G. Murty, *Linear and Combinatorial Programming* (Wiley, New York, 1976)

[M12] K.G. Murty, The Karmarkar's algorithm for linear programming (Chap. 11.4.1), in *Linear Complementarity, Linear and Nonlinear Programming*. Sigma Series in Applied Mathematics, vol. 3 (Heldermann Verlag, Berlin, 1988), pp. 469–494

[N1] S.G., Nash, A. Sofer *Linear and Nonlinear Programming* (McGraw-Hill Companies, New York, 1996)

[188] Y. Nesterov, Efficiency of coordinate descent methods on huge-scale optimization problems. SIAM J. Optim. **22**, 341–362 (2012)

[189] Y.E. Nesterov, Semidefinite relaxation and nonconvex quadratic optimization. Optim. Methods Softw. **9**, 141–160 (1998)

[190] Y. Nesterov, A method of solving a convex programming problem with convergence rate $O((1/k^2))$. Soviet Math. Doklady **27**(2), 372–376 (1983)

[191] Y. Nesterov, M.J. Todd, Y. Ye, Infeasible-start primal-dual methods and infeasibility detectors for nonlinear programming problems. Math. Program. **84**, 227–267 (1999)

[N2] Y. Nesterov, A. Nemirovskii, *Interior Point Polynomial Methods in Convex Programming: Theory and Algorithms* (SIAM Publications, Philadelphia, 1994)

[N3] Y. Nesterov, M.J. Todd, Self-scaled barriers and interior-point methods for convex programming. Math. Oper. Res. **22**(1) 1–42 (1997)

[N4] Y. Nesterov, *Introductory Lectures on Convex Optimization: A Basic Course* (Kluwer, Boston, 2004)

[O1] W. Orchard-Hays, Background development and extensions of the revised simplex method. RAND Report RM-1433, The RAND Corporation, Santa Monica, CA (1954)

[O2] A. Orden, Application of the simplex method to a variety of matrix problems, in *Directorate of Management Analysis: "Symposium on Linear Inequalities and Programming"*, ed. by A. Orden, L. Goldstein (DCS/Comptroller, Headquarters, U.S. Air Force, Washington, DC, 1952), pp. 28–50

[O3] A. Orden, The transshipment problem. Manag. Sci. **2**(3), 276–285 (1956)

[O4] S.S. Oren, Self-scaling variable metric (ssvm) algorithms ii: implementation and experiments. Manag. Sci. **20**, 863–874 (1974)

[O5] S.S. Oren, D.G. Luenberger, Self-scaling variable metric (ssvm) algorithms i: criteria and sufficient conditions for scaling a class of algorithms. Manag. Sci. **20**, 845–862 (1974)

[O6] S.S. Oren, E. Spedicato, Optimal conditioning of self-scaling variable metric algorithms. Math. Program. **10**, 70–90 (1976)

[O7] J.M. Ortega, W.C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables* (Academic, New York, 1970)

[P1] C.C. Paige, M.A. Saunders, Solution of sparse indefinite systems of linear equations. SIAM J. Numer. Anal. **12**(4), 617–629 (1975)

[P2] C. Papadimitriou, K. Steiglitz, *Combinatorial Optimization Algorithms and Complexity* (Prentice-Hall, Englewood Cliffs, NJ, 1982)

[204] P. Parrilo, Semidefinite programming relaxations for semialgebraic problems. Math. Program. **96**, 293–320 (2003)

[205] G. Pataki, On the rank of extreme matrices in semidefinite programs and the multiplicity of optimal eigenvalues. Math. Oper. Res. **23**, 339–358 (1998)

[P3] A. Perry, A modified conjugate gradient algorithm, Discussion Paper No. 229, Center for Mathematical Studies in Economics and Management Science, North-Western University, Evanston, IL (1976)

[P4] E. Polak, *Computational Methods in Optimization: A Unified Approach* (Academic, New York, 1971)

[P5] E. Polak, G. Ribiere, Note sur la Convergence de Methods de Directions Conjugres. Revue Francaise Informat. Recherche Operationnelle **16**, 35–43 (1969)

[P6] M.J.D. Powell, An efficient method for finding the minimum of a function of several variables without calculating derivatives. Comput. J. **7**, 155–162 (1964)

[P7] M.J.D. Powell, A method for nonlinear constraints in minimization problems, in *Optimization*, ed. by R. Fletcher Powell (Academic, London, 1969), pp. 283–298

[P8] M.J.D. Powell, On the convergence of the variable metric algorithm. Mathematics Branch, Atomic Energy Research Establishment, Harwell, Berkshire, England, (October 1969)

[P9] M.J.D. Powell, Algorithms for nonlinear constraints that use lagrangian functions. Math. Program. **14**, 224–248 (1978)

[P10] B.N. Pshenichny, Y.M. Danilin, *Numerical Methods in Extremal Problems* (translated from Russian by V. Zhitomirsky) (MIR Publishers, Moscow, 1978)

[214] M. Ramana, An exact duality theory for semidefinite programming and its complexity implications. Math. Program. **77**, 129–162 (1997)

[215] M. Ramana, L. Tuncel H. Wolkowicz, Strong duality for semidefinite programming. SIAM J. Optim. **7**, 641–662 (1997)

[R1] J. Renegar, A polynomial-time algorithm, based on Newton's method, for linear programming. Math. Program. **40**, 59–93 (1988)

[R2] J. Renegar, *A Mathematical View of Interior-Point Methods in Convex Optimization* (Society for Industrial and Applied Mathematics, Philadelphia, 2001)

[R3] R.T. Rockafellar, The multiplier method of hestenes and powell applied to convex programming. J. Optim. Theory Appl. **12**, 555–562 (1973)

[219] R.T. Rockafellar, *Convex Analysis* (Princeton University Press, Princeton, NJ, 1970)

[R4] C. Roos, T. Terlaky, J.-Ph. Vial, *Theory and Algorithms for Linear Optimization: An Interior Point Approach* (Wiley, Chichester, 1997)

[R5] J. Rosen, The gradient projection method for nonlinear programming, I. Linear contraints. J. Soc. Ind. Appl. Math. **8**, 181–217 (1960)

[R6] J. Rosen, The gradient projection method for nonlinear programming, II. Non-linear constraints. J. Soc. Ind. Appl. Math. **9**, 514–532 (1961)

[S1] R. Saigal, *Linear Programming: Modern Integrated Analysis* (Kluwer Academic Publisher, Boston, 1995)

[S2] B. Shah, R. Buehler, O. Kempthorne, Some algorithms for minimizing a function of several variables. J. Soc. Ind. Appl. Math. **12**, 74–92 (1964)

[S3] D.F. Shanno, Conditioning of quasi-Newton methods for function minimization. Math. Comput. **24**, 647–656 (1970)

[S4] D.F. Shanno, Conjugate gradient methods with inexact line searches. Math. Oper. Res. **3**(3) 244–2560 (1978)

[S5] A. Shefi, Reduction of linear inequality constraints and determination of all feasible extreme points, Ph.D. dissertation, Department of Engineering-Economic Systems, Stanford University, Stanford, CA, October 1969

[228] W.F. Sheppard, On the calculation of the double integral expressing normal correlation. Trans. Camb. Philos. Soc. **19**, 23–66 (1900)

[S6] M. Simonnard, *Linear Programming*, translated by William S. Jewell (Prentice-Hall, Englewood Cliffs, NJ, 1966)

[S7] M. Slater, Lagrange multipliers revisited: a contribution to non-linear programming. Cowles Commission Discussion Paper, Math 403 (November 1950)

[231] A.M. So, Y. Ye, Theory of semidefinite programming for sensor network localization. Math. Program. **109**, 367–384 (2007)

[232] A.M. So, Y. Ye, J. Zhang, A unified theorem on SDP rank reduction. Math. Oper. Res. **33**, 910–920 (2008)

[S8] G. Sonnevend, An ‘analytic center’ for polyhedrons and new classes of global algorithms for linear (smooth, convex) programming, in *System Modelling and Optimization: Proceedings of the 12th IFIP-Conference held in Budapest, Hungary, September 1985*, ed. by A. Prekopa, J. Szelezsan, B. Strazicky. Lecture Notes in Control and Information Sciences, vol. 84 (Springer, Berlin, 1986), pp. 866–876

[S9] G.W. Stewart, A modification of Davidon’s minimization method to accept difference approximations of derivatives. J. ACM **14**, 72–83 (1967)

[S10] E.L. Stiefel, Kernel polynomials in linear algebra and their numerical applications. Nat. Bur. Stand. Appl. Math. Ser. **49**, 1–22 (1958)

[S11] J.F. Sturm, Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. Optim. Methods Softw. **11&12**, 625–633 (1999)

[S12] J. Sun, L. Qi, An interior point algorithm of $O(\sqrt{n}|\ln(\epsilon)|)$ iterations for C^1-convex programming. Math. Program. **57**, 239–257 (1992)

[T1] A. Tamir, Line search techniques based on interpolating polynomials using function values only. Manag. Sci. **22**(5), 576–586 (1976)

[T2] K. Tanabe, Complementarity-enforced centered Newton method for mathematical programming, in *New Methods for Linear Programming*, ed. by K. Tone (The Institute of Statistical Mathematics, Tokyo, 1987), pp. 118–144

[T3] R.A. Tapia, Quasi-Newton methods for equality constrained optimization: equivalents of existing methods and new implementation, in *Symposium on Nonlinear Programming III*, ed. by O. Mangasarian, R. Meyer, S. Robinson (Academic, New York, 1978), pp. 125–164

[T4] M.J. Todd, A low complexity interior point algorithm for linear programming. SIAM J. Optim. **2**, 198–209 (1992)

[T5] M.J. Todd, Y. Ye, A centered projective algorithm for linear programming. Math. Oper. Res. **15**, 508–529 (1990)

[T6] K. Tone, Revisions of constraint approximations in the successive qp method for nonlinear programming problems. Math. Program. **26**(2), 144–152 (1983)

[T7] D.M. Topkis, A note on cutting-plane methods without nested constraint sets. ORC 69-36, Operations Research Center, College of Engineering, Berkeley, CA (December 1969)

[T8] D.M. Topkis, A.F. Veinott Jr., On the convergence of some feasible direction algorithms for nonlinear programming. J. SIAM Control **5**(2), 268–279 (1967)

[T9] J.F. Traub, *Iterative Methods for the Solution of Equations* (Prentice-Hall, Englewood Cliffs, NJ, 1964)

[T10] L. Tunçel, Constant potential primal-dual algorithms: a framework. Math. Prog. **66**, 145–159 (1994)

[T11] R. Tutuncu, An infeasible-interior-point potential-reduction algorithm for linear programming, Ph.D. thesis, School of Operations Research and Industrial Engineering, Cornell University, Ithaca, NY, 1995

[T12] P. Tseng, Complexity analysis of a linear complementarity algorithm based on a Lyapunov function. Math. Program. **53**, 297–306 (1992)

[V1] P.M. Vaidya, An algorithm for linear programming which requires $O((m+n)n^2 + (m+n)^{1.5}nL)$ arithmetic operations. Math. Prog. **47**, 175–201 (1990). Condensed version in: Proceedings of the 19th Annual ACM Symposium on Theory of Computing, 1987, pp. 29–38

[V2] L. Vandenberghe, S. Boyd, Semidefinite programming. SIAM Rev. **38**(1) 49–95 (1996)

[V3] R.J. Vanderbei, *Linear Programming: Foundations and Extensions* (Kluwer Academic Publishers, Boston, 1997)

[V4] S.A. Vavasis, *Nonlinear Optimization: Complexity Issues* (Oxford Science, New York, NY, 1991)

[V5] A.F. Veinott Jr., The supporting hyperplane method for unimodal programming. Oper. Res. XV **1**, 147–152 (1967)

[V6] Y.V. Vorobyev, *Methods of Moments in Applied Mathematics* (Gordon and Breach, New York, 1965)

[W1] D.J. Wilde, C.S. Beightler, *Foundations of Optimization* (Prentice-Hall, Englewood Cliffs, NJ, 1967)

[W2] R.B. Wilson, A simplicial algorithm for concave programming, Ph.D. dissertation, Harvard University Graduate School of Business Administration, 1963
[W3] P. Wolfe, A duality theorem for nonlinear programming. Q. Appl. Math. **19**, 239–244 (1961)
[W4] P. Wolfe, On the convergence of gradient methods under constraints. IBM Research Report RZ 204, Zurich (1966)
[W5] P. Wolfe, Methods of nonlinear programming (Chap. 6), in *Nonlinear Programming*, ed. by J. Abadie. Interscience (Wiley, New York, 1967), pp. 97–131
[W6] P. Wolfe, Convergence conditions for ascent methods. SIAM Rev. **11**, 226–235 (1969)
[W7] P. Wolfe, Convergence theory in nonlinear programming (Chap. 1), in *Integer and Nonlinear Programming*, ed. by J. Abadie (North-Holland Publishing Company, Amsterdam, 1970)
[W8] S.J. Wright, *Primal-Dual Interior-Point Methods* (SIAM, Philadelphia, 1996)
[264] G. Xue, Y. Ye, Efficient algorithms for minimizing a sum of Euclidean norms with applications. SIAM J. Optim. **7**, 1017–1036 (1997)
[265] Y. Ye, Approximating quadratic programming with bound and quadratic constraints. Math. Program. **84**, 219–226 (1999)
[Y1] Y. Ye, An $O(n^3L)$ potential reduction algorithm for linear programming. Math. Program. **50**, 239–258 (1991)
[Y2] Y. Ye, M.J. Todd, S. Mizuno, An $O(\sqrt{n}L)$-iteration homogeneous and self-dual linear programming algorithm. Math. Oper. Res. **19**, 53–67 (1994)
[Y3] Y. Ye, *Interior Point Algorithms* (Wiley, New York, 1997)
[269] Y. Ye, A new complexity result on solving the markov decision problem. Math. Oper. Res. **30**, 733–749 (2005)
[Z1] W.I. Zangwill, Nonlinear programming via penalty functions. Manag. Sci. **13**(5), 344–358 (1967)
[Z2] W.I. Zangwill, *Nonlinear Programming: A Unified Approach* (Prentice-Hall, Englewood Cliffs, NJ, 1969)
[Z3] Y. Zhang, D. Zhang, On polynomiality of the mehrotra-type predictor-corrector interior-point algorithms. Math. Program. **68**, 303–317 (1995)
[Z4] G. Zoutendijk, *Methods of Feasible Directions* (Elsevier, Amsterdam, 1960)

Index

© Springer International Publishing Switzerland 2016

D.G. Luenberger, Y. Ye, *Linear and Nonlinear Programming*, International Series in Operations Research & Management Science 228,
DOI 10.1007/978-3-319-18842-3