Simple Efficient Solutions for Semidefinite Programming

Henry Wolkowicz [*]

October 4, 2001

University of Waterloo
Department of Combinatorics & Optimization
Waterloo, Ontario N2L 3G1, Canada
Research Report CORR 2001-49 [1]

Key words: Semidefinite Programming, large sparse problems, inexact Gauss-Newton method, preconditioned conjugate gradients, optimal diagonal preconditioner, Max-Cut Problem.

Abstract

This paper provides a simple approach for solving a semidefinite program, SDP. As is common with many other approaches, we apply a primal-dual method that uses the perturbed optimality equations for SDP, $F_\mu(X, y, Z) = 0$, where X, Z are $n \times n$ symmetric matrices and $y \in \Re^m$. However, as in Kruk et al [19], we look at this as an overdetermined system of nonlinear (bilinear) equations on vectors X, y, Z which has a zero residual at optimality. We do not use any symmetrization on this system. That the vectors X, Z are symmetric matrices is ignored. What is different in this paper is a preprocessing step that results in one single, well-conditioned, overdetermined bilinear equation. We do not form a Schur complement form for the linearized system. In addition, asymptotic q-quadratic convergence is obtained with a *crossover* approach that switches to affine scaling without maintaining the positive (semi)definiteness of X and Z. This results in a marked reduction in the number and the complexity of the iterations.

We use the long accepted method for nonlinear least squares problems, the Gauss-Newton method. For large sparse data, we use an *inexact Gauss-Newton* approach with a preconditioned conjugate gradient method applied directly on the linearized equations in matrix space. This does not require any operator formations into a Schur complement system or matrix representations of linear operators.

[*] Research supported by The Natural Sciences and Engineering Research Council of Canada. E-mail hwolkowicz@uwaterloo.ca

[1] A modified version of this paper, *Semidefinite Programming and Some Closest Matrix Approximation Problems*, was presented at the 1st Annual McMaster Optimization Conference (MOPTA 01), August 2-4, 2001 Hamilton, Ontario.

[1] URL for paper and MATLAB programs:
http://orion.math.uwaterloo.ca/~hwolkowi/henry/reports/ABSTRACTS.html

To illustrate the approach, we apply it to the well known SDP relaxation of the Max-Cut problem. The cost of an iteration consists almost entirely in the solution of a (generally well-conditioned) least squares problem of size $n^2 \times \binom{n+1}{2}$. This system consists of one large (sparse) block with $\binom{n}{2}$ columns (one CG iteration cost is one sparse matrix multiplication) and one small (dense) block with n columns (one CG iteration cost is one matrix row scaling). Exact primal and dual feasibility are guaranteed throughout the iterations. We include the derivation of the *optimal diagonal preconditioner*. The numerical tests show that the algorithm exploits sparsity and obtains q-quadratic convergence.

Contents

1 Introduction

The many applications, elegant theory, and practical algorithms for Semidefinite Programming (SDP) has, arguably, made SDP the hottest area of optimization during the last five years. It's popularity, however, remains concentrated among specialists rather than mainstream nonlinear programming practitioners and users. Most of the current algorithmic approaches use symmetrizations and apply Newton's method. The next (complicated and costly) step is to construct the matrix for the *Schur complement system*. In general, this results in a dense ill-conditioned system. There is a lack of SDP solvers that can efficiently exploit sparsity and avoid the ill-conditioning of the Schur complement system. This raises doubts whether SDP will replace the simpler Linear Programming (LP) even in the cases where SDP provides stronger relaxations.

The main purpose of this paper is to illustrate a simple alternative algorithmic development for SDP completely based on standard principles from numerical analysis, i.e. on the (inexact) Gauss-Newton method with preconditioned conjugate gradients. The only additional tool we use is the definition of a linear operator and its adjoint. There is no need to construct matrix representations of operators. And, no ill-conditioned system is formed. We illustrate this approach on the well-known Max-Cut problem.

The primal SDP we consider is

$$\text{(PSDP)} \qquad \begin{array}{rrl} p^* := & \max & \operatorname{trace} CX \\ & \text{s.t.} & \mathcal{A}X = b \\ & & X \succeq 0. \end{array} \tag{1.1}$$

Its dual is

$$\text{(DSDP)} \qquad \begin{array}{rrl} d^* := & \min & b^T y \\ & \text{s.t.} & \mathcal{A}^* y - Z = C \\ & & Z \succeq 0, \end{array} \tag{1.2}$$

where $X, Z \in \mathcal{S}^n$, $\mathcal{S}^n$ denotes the space of $n \times n$ real symmetric matrices, and $\succeq$ denotes positive semidefiniteness. $\mathcal{A} : \mathcal{S}^n \to \Re^m$ is a linear operator and $\mathcal{A}^*$ is the adjoint operator.

SDP looks just like a linear program and many of the properties from linear programming follow through. (We discuss the similarity with linear programming and its influence on the algorithmic development for SDP in Section 1.1 below.) *Weak duality* $p^* \leq d^*$ holds. However, as in general convex programming, *strong duality* can fail; there can exist a nonzero duality gap at optimality $p^* < d^*$ and/or the dual may not be attained. (Strong duality can be ensured using a suitable constraint qualification.)

The primal-dual pair leads to the elegant primal-dual optimality conditions, i.e. primal feasibility, dual feasibility, and complementary slackness.

Theorem 1.1 *Suppose that suitable constraint qualifications hold for (PSDP), (DSDP). The primal-dual variables (X, y, Z), with $X, Z \succeq 0$, are optimal for the primal-dual pair of SDPs if and only if*

$$F(X, y, Z) := \begin{pmatrix} \mathcal{A}^* y - Z - C \\ \mathcal{A}X - b \\ ZX \end{pmatrix} = 0. \tag{1.3}$$

■

Note that $F : \mathcal{S}^n \times \Re^m \times \mathcal{S}^n \to \mathcal{S}^n \times \Re^m \times \mathcal{M}^n$, where $\mathcal{M}^n$ is the space of $n \times n$ matrices, since the product ZX is not necessarily symmetric. (Z, X are diagonal matrices in the LP case. This is one of the differences between SDP and LP that, perhaps, has had the most influence on algorithmic development for SDP.) Currently, the successful primal-dual algorithms are path following algorithms that use Newton's method to solve (symmetrizations of) the perturbed optimality conditions:

$$F_\mu(X,y,Z) := \begin{pmatrix} \mathcal{A}^*y - Z - C \\ \mathcal{A}X - b \\ ZX - \mu I \end{pmatrix} = 0. \tag{1.4}$$

1.1 Historical Remarks and Motivation

Though studied as early as the 1960s, the popularity of SDP increased tremendously in the early 1990s, see e.g. the historical summary in [27]. This was partly a result of the many powerful applications and elegant theory of this area; but the main influence followed from the knowledge that interior-point methods can solve SDPs; and, the field of optimization had just completed a tremendous upheaval following the great success and excitement of interior-point methods for Linear Programming, (LP), developed in the 1980s. Many of the researchers from the LP area moved into SDP.

As in LP, the early successful algorithms were developed using the log-barrier approach. The optimality conditions of the log-barrier problems led to the

$$X - \mu Z^{-1} = 0 \tag{1.5}$$

form of the perturbed complementary slackness equation. However, it was quickly discovered that the algorithms based on this form were inefficient, i.e. the result was slow convergence with numerical difficulties arising from ill-conditioning when the barrier parameter μ was close to 0. Premultiplication by Z (a form of preconditioning) led to the $ZX - \mu I = 0$ form in (1.4) given above, i.e. the form that paralleled the classical approach in Fiacco-McCormick [12], and the one that exhibited great success for LPs.

However, one could not apply Newton's method to the nonsquare system (1.4) and Newton's method was the driving force for the success in interior-point methods for LP. Therefore, the effort was then put into *symmetrization schemes* so that techniques from LP could be applied. Furthermore, after the symmetrization schemes are applied, the *Schur complement* approach reduces the size of the linear system to be solved. Often, forming this Schur complement system requires more work than solving the resulting reduced system of linear equations. Moreover, both the symmetrized system and this reduced system are, in general, ill-conditioned even if the original system was not. And, contrary to the case of linear programming, it is not easy to avoid this ill-conditioning.

These symmetrization schemes can be considered from the view of preconditioning of the optimality conditions (1.4), with the form (1.5), using some preconditioner. Suppose that we start with the optimality conditions that arise from the log-barrier problem.

$$\begin{pmatrix} \mathcal{A}^*y - Z - C \\ \mathcal{A}X - b \\ X - \mu Z^{-1} \end{pmatrix} = 0. \tag{1.6}$$

Premultiplication by Z (a form of preconditioning) leads to a less nonlinear system, avoids the ill-conditioning, and results in the more familiar $(ZX - \mu I)$ form.

$$\begin{pmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & Z\cdot \end{pmatrix} \begin{pmatrix} \mathcal{A}^*y - Z - C \\ \mathcal{A}X - b \\ X - \mu Z^{-1} \end{pmatrix} = \begin{pmatrix} \mathcal{A}^*y - Z - C \\ \mathcal{A}X - b \\ ZX - \mu I \end{pmatrix} = 0. \tag{1.7}$$

We now precondition a second time using the linear operator $\mathcal{P}$ with symmetrization operator $\mathcal{S}$.

$$\mathcal{P}F_\mu(x,y,Z) = \begin{pmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & \mathcal{S} \end{pmatrix} \begin{pmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & Z\cdot \end{pmatrix} \begin{pmatrix} \mathcal{A}^*y - Z - C \\ \mathcal{A}X - b \\ X - \mu Z^{-1} \end{pmatrix} = 0. \qquad (1.8)$$

However, the symmetrizations used in the literature, in general, reverse the previous process, i.e. after the symmetrization the ill-conditioning problem has returned. This framework encompasses many different symmetrized systems with various acronyms, similar to the area of quasi-Newton methods, e.g. AHO, HKM, NT, GT, MTW. The survey of search directions by Todd [25] presented several search directions and their theoretical properties, including: well-definedness, scale invariance, and primal-dual symmetry. For example, section 4 in that paper studies twenty different primal-dual search directions. (Though the Gauss-Newton direction studied here is not included.)

The point of view and motivation taken in this paper is that the framework of symmetrization in itself (contrary to the premultiplication by Z) can be *counterproductive*. This can be seen from basic texts in Numerical Analysis and the preconditioning point of view in (1.8). Let us ignore that some of the variables are in matrix space, and look at the overdetermined nonlinear system (1.4), $F_\mu(x,y,Z) = 0$, with zero residual. Then the approach in any standard text (e.g. [9]) is to apply the Gauss-Newton method. This method has many desirable features. Preconditioning (from the left, since it is scale invariant from the right) is recommended if it results in *improved conditioning of the problem*, or in a problem that is *less nonlinear*. This view can be used to motivate our GN approach, i.e. the symmetrization schemes used in the literature are not motivated by conditioning considerations. But, on the other hand, they attempt to recreate the LP type framework. In instances studied in [19], theoretical and empirical evidence shows that conditioning worsened for these symmetrization schemes, i.e. the opposite effect of preconditioning was observed. In particular, many of the symmetrized directions are ill-posed (singular Jacobian) at $\mu = 0$, resulting in numerical difficulties in obtaining high accuracy approximations of optimal solutions.

Further motivation for GN arises from the need to exploit sparsity in many applications of SDP. The symmetrizations and the corresponding Schur complement systems make it extremely difficult to exploit sparsity. Many attempts have been made. But in each case the matrix problem is changed to a vector problem and then the structure of the original problem is used in the vector problem. For the GN method, a standard approach to solving large sparse problems is to use a preconditioned conjugate gradient method. Again, as above, this does not need to take into account that some of the variables are in a matrix space. There is no need to change back and forth between matrices and vectors. One should consider the function as an operator between vector spaces, i.e. a black box for the Gauss-Newton (GN) method with preconditioned conjugate gradients (PCG). To fully exploit this approach, we eliminate the linear equations from the optimality conditions in a preprocessing phase.

For simplicity, we restrict ourselves in this paper to the semidefinite relaxation of the Max-Cut problem; and, we derive and use the *optimal diagonal preconditioner* in the PCG method. Our approach is currently being applied to several other problems with more sophisticated preconditioners, see e.g. [1, 24, 15].

1.1.1 Related Work

Several recent papers have concentrated on exploiting the special structure of the SDP relaxation for the Max-Cut problem (see (P) below). A discussion of several of the methods is given in Burer-Montreiro [5]. In particular, Benson et al [3] present an interior-point method based on potential-reduction and dual-scaling; while, Helmberg-Rendl [17] use a bundle-trust approach on a nondifferentiable function arising from the Lagrangian dual. Both of these methods exploit the small dimension n of the dual problem compared to the dimension $\binom{n+1}{2} = n(n+1)/2$ of the primal problem. Moreover, the dual feasibility equation is sparse if the matrix of the quadratic form Q is sparse. Therefore each iteration is inexpensive. However, these are not primal-dual methods and do not easily obtain high accuracy approximations of optimal solutions without many iterations.

The method in [5] uses the transformation $X = VV^T$ and discards the semidefinite constraint. The problem reduces to a first order gradient projection method. They argue for using a search direction based on first order information only (rather than second order information as used in interior-point methods) since this results in fast and inexpensive iterations. However, the cost is that they need many iterations, contrary to interior-point methods which need relatively few. Therefore, their approach is useful in obtaining optimal solutions to moderate accuracy. Their formulation has $\binom{n}{2} = n(n-1)/2$ variables (unknowns).

Similarly, Our algorithm has $\binom{n+1}{2}$ variables. However, it is based on the primal-dual framework. We make the opposite arguement, (common in nonlinear programming), i.e. *it is important to start with a good search direction.* We find a Gauss-Newton search direction using a PCG approach. We use an *inexact Newton* framework, e.g. [8], to lower the cost of each iteration. In fact, restricting CG to one iteration results in a first order method with the same cost per iteration as that in [5]. In our algorithm, we try to stay well-centered until we reach the region of quadratic convergence for affine scaling. Since we have a well-posed system, (full rank Jacobian at optimality), we can obtain q-quadratic convergence and highly accurate approximations of optimal solutions. The cost of each CG iteration is a sparse matrix multiplication (essentially $Z\Delta X$) and a matrix scaling (essentially $\mathrm{Diag}\,(\Delta y)X$).

1.2 Further Background and Notation

The Max-Cut problem, e.g. [11], consists in finding a partition of the set of vertices of a given undirected graph with weights on the edges so that the sum of the weights of the edges cut by the partition is maximized. This NP-hard discrete optimization problem can be formulated as the following (quadratic) program (e.g. Q is a multiple of the Laplacian matrix of the graph).

$$\text{(MC0)} \qquad \begin{array}{rll} \mu^* := & \max & v^TQv \\ & \text{s.t.} & v_i^2 = 1, \quad i = 1, \ldots, n. \end{array} \tag{1.9}$$

Using Lagrangian relaxation, (see e.g. [2]), one can derive the following semidefinite relaxation of (MC0).

$$
\text{(P)} \qquad \begin{array}{rll} \mu^* \leq \nu^* := & \max & \operatorname{trace} QX \\ & \text{s.t.} & \operatorname{diag}(X) = e \\ & & X \succeq 0, X \in \mathcal{S}^n \,, \end{array} \tag{1.10}
$$

where diag denotes the vector formed from the diagonal elements and e denotes the (column) vector of ones. For our purposes we assume, without loss of generality, that $\operatorname{diag}(Q) = 0$. This relaxation has been extensively studied. It has been found to be suprisingly strong both in theory and in practice. (See e.g. the survey paper [13].)

We work with the trace inner product in matrix space

$$
\langle M, N \rangle = \operatorname{trace} M^T N, \qquad M, N \in \mathcal{M}^n \,.
$$

This definition follows through to $\mathcal{S}^n$, the space of $n \times n$ symmetric matrices. The induced norm is the *Frobenius norm*, $\|M\|_F = \sqrt{\operatorname{trace} M^T M}$. Upper case letters $M, X, \ldots$ are used to denote matrices; while lower case letters are used for vectors.

1.2.1 Adjoint Operators; Generalized Inverses; Projections

We define several linear operators on vectors and matrices. We also need the adjoints when finding the dual SDP and also when applying the conjugate gradient method. Though not essential for our development, we include information on generalized inverses and projections.

$$
v = \operatorname{vec}(M) := \begin{pmatrix} M_{11} \\ M_{21} \\ \ldots \\ M_{12} \\ M_{22} \\ M_{32} \\ \ldots \end{pmatrix} \in \Re^{n^2}, \quad \text{columnwise if } M \in \mathcal{M}_{m,n}
$$

takes the general rectangular matrix M and forms a vector from its columns. The inverse mapping Mat $:= \operatorname{vec}^{-1}$.

$$
x = \operatorname{u2svec}(X) := \sqrt{2} \begin{pmatrix} X_{12} \\ X_{13} \\ X_{23} \\ X_{14} \\ \ldots \end{pmatrix} \in \Re^{\binom{n}{2}} \quad \text{(columnwise, strict upper triangular if } X \in \mathcal{S}^n \text{)}
$$

is $\sqrt{2}$ times the vector obtained columnwise from the strictly upper triangular part of the symmetric matrix X; $\begin{pmatrix} n \\ 2 \end{pmatrix} = \frac{n(n-1)}{2}$. (The multiplication by $\sqrt{2}$ makes the mapping an isometry. Define vector q similarly, $q = \operatorname{u2svec}(Q)$.) Let u2sMat $:= \operatorname{u2svec}^{\dagger}$ denote the *Moore-Penrose generalized inverse* mapping into $\mathcal{S}^n$. This is an inverse mapping if we restrict to the subspace of matrices with zero diagonal.

The adjoint operator $\mathrm{u2sMat}^* = \mathrm{u2svec}$, since

$$\begin{array}{rcl}\langle \mathrm{u2sMat}\,(v), S\rangle &=& \mathrm{trace}\, \mathrm{u2sMat}\,(v) S \\ &=& v^T \mathrm{u2svec}\,(S) = \langle v, \mathrm{u2svec}\,(S)\rangle\,.\end{array}$$

Let

$$\mathrm{offDiag}\,(S) := S - \mathrm{Diag}\,(\mathrm{diag}\,(S)),$$

where $\mathrm{diag}\,(S)$ denotes the diagonal of S and $\mathrm{diag}^*(v) = \mathrm{diag}^\dagger(v) = \mathrm{Diag}\,(v)$ is the adjoint operator, i.e. the diagonal matrix with diagonal elements from the vector v. Then $\mathrm{diag}\,\mathrm{Diag} = I$ on $\Re^n$,

$$\mathrm{Diag}\,\mathrm{diag}^* = \mathrm{Diag}\,\mathrm{diag}^\dagger$$

is the orthogonal projection on the subspace of diagonal matrices in $\mathcal{S}^n$, the range of Diag, and

$$\mathrm{u2sMat}\,\mathrm{u2sMat}^* = \mathrm{u2sMat}\,\mathrm{u2sMat}^\dagger = \mathrm{offDiag} = \mathrm{offDiag}^*,$$

is the orthogonal projection onto the subspace of matrices with 0 diagonal, the range of u2sMat.

For a linear constraint $\mathcal{A}\,(X) = b$, where the linear operator $\mathcal{A}$ is rank m, let the linear operator $\mathcal{N} : \Re^{\binom{n+1}{2}-m} \rightarrow \mathcal{S}^n$ have range equal to the null space of $\mathcal{A}$. (We assume, without loss of generality that $\mathcal{A}$ is onto.) Then

$$\begin{array}{rll}\mathcal{A}\,(X) = b & \text{iff} & X = \mathcal{A}^\dagger b + (I - \mathcal{A}\mathcal{A}^\dagger)W, \quad \text{for some } W \in \mathcal{S}^n \\ & \text{iff} & X = \mathcal{A}^\dagger b + \mathcal{N}\,(w), \quad \text{for some } w \in \Re^{\binom{n+1}{2}-m}.\end{array} \qquad (1.11)$$

When we apply (1.11) to (P), we obtain

$$\mathrm{diag}\,(X) = e \quad \text{iff} \quad X := I + \mathrm{u2sMat}\,(x), \quad \text{for some } x \in \Re^{\binom{n}{2}}. \qquad (1.12)$$

Below, we equate $X := \mathrm{u2sMat}\,(x) + I$, $Z := \mathrm{Diag}\,(y) - Q$. The following operators map $\mathcal{S}^n \rightarrow \Re^{n^2}$. These are the two operators used in the optimality conditions and in the Gauss-Newton method.

$$\boxed{\mathcal{X}(\cdot) := \mathrm{vec}\,(\mathrm{Diag}\,(\cdot)X); \quad \mathcal{Z}(\cdot) := \mathrm{vec}\,(Z\mathrm{u2sMat}\,(\cdot)). \qquad (1.13)}$$

We evaluate the adjoint operators. Let $A \circ B$ denote the Hadamard (elementwise) product of the matrices A, B. Let $w = \mathrm{vec}\,(W)$.

$$\begin{array}{rcl}\langle w, \mathcal{X}(v)\rangle &=& \mathrm{trace}\, W^T \mathrm{Diag}\,(v) X \\ &=& \mathrm{trace}\, \mathrm{Diag}\,(v) X W^T \\ &=& v^T \mathrm{diag}\,(X \mathrm{Mat}\,(w)^T) \\ &=& (e^T (X \circ \mathrm{Mat}\,(w)^T)) v \\ &=& ((X \circ \mathrm{Mat}\,(w)) e)^T v \\ &=& \langle \mathcal{X}^*(w), v\rangle\,.\end{array}$$

$$\begin{array}{rcl}\langle w, \mathcal{Z}(v)\rangle & = & \operatorname{trace} W^T Z \mathrm{u2sMat}\,(v) \\ & = & \operatorname{trace} \mathrm{u2sMat}\,(v) \frac{1}{2}\left\{ZW + W^T Z\right\} \\ & = & v^T \frac{1}{2} \mathrm{u2svec}\,\left\{Z\mathrm{Mat}\,(w) + \mathrm{Mat}\,(w)^T Z\right\} \\ & = & \langle \mathcal{Z}^*(w), v\rangle\,. \end{array}$$

Summary:

$$\boxed{\begin{array}{rcl}\mathcal{X}^*(w) & = & \operatorname{diag}\left(X\mathrm{Mat}\,(w)^T\right) = (X \circ \mathrm{Mat}\,(w))e; \\ \mathcal{Z}^*(w) & = & \frac{1}{2}\mathrm{u2svec}\,\left\{Z\mathrm{Mat}\,(w) + (Z\mathrm{Mat}\,(w))^T\right\}. \end{array} \qquad (1.14)}$$

2 Duality and Optimality Conditions

Recall the primal SDP.

$$(P) \qquad \begin{array}{rll} \nu^* := & \max & \operatorname{trace} QX \\ & \text{subject to} & \operatorname{diag}\,(X) = e \\ & & X \succeq 0. \end{array} \qquad (2.1)$$

To obtain optimality conditions we use a dual problem. Slater's constraint qualification (strict feasibility) holds for (P), which implies that we have strong duality with the Lagrangian dual (e.g. [23])

$$(D) \qquad \begin{array}{rll} \nu^* = & \min & e^T y \\ & \text{subject to} & \operatorname{Diag}\,(y) - Z = Q \\ & & Z \succeq 0. \end{array} \qquad (2.2)$$

Weak duality for feasible variables can be expressed as:

$$0 \leq e^T y - \operatorname{trace} QX = e^T y - \operatorname{trace}\,(\operatorname{Diag}\,(y) - Z)X = (e - \operatorname{diag}\,(X))^T y + \operatorname{trace} ZX = \operatorname{trace} ZX.$$

Therefore, a zero duality gap is equivalent to $\operatorname{trace} ZX = 0$. Moreover, since $X, Z \succeq 0$, this is equivalent to $ZX = 0$. Since Slater's condition is also satisfied for the dual program, we have primal attainment and get the following well-known characterization of optimality for (P).

Theorem 2.1 *The primal-dual variables X, y, Z with $X \succeq 0, Z \succeq 0$ are optimal for (P),(D) if and only if*

$$\begin{array}{cl} \operatorname{Diag}\,(y) - Z = Q & \textit{(dual feasibility)} \\ \operatorname{diag}\,(X) = e & \textit{(primal feasibility)} \\ ZX = 0 & \textit{(complementary slackness)} \end{array}$$

■

2.1 Preprocessing

Since the diagonal of X is fixed, we can use the constant $K = e^T \operatorname{diag}(Q)$ in the objective function. We could also set $\operatorname{diag}(Q) = 0$ so that $K = 0$. Therefore we assume without loss of generality that

$$\operatorname{Diag}(Q) = 0.$$

The simplicity of the primal feasibility equation yields an equivalent problem to (P) with the representation $x = \operatorname{u2svec}(X),\ X = \operatorname{u2sMat}(x) + I$. This representation is a key element in our approach, i.e. we found an initial positive solution using an operator whose range is the null space of $\mathcal{A}$. In the general SDP notation, we found $\mathcal{A}X = b$ is equivalent to $X = \mathcal{N}(x) + \hat{X}$, where $\mathcal{N}$ is a linear operator with range equal to the null space of $\mathcal{A}$ and $\hat{X}$ is a particular solution that satisfies $X = \mathcal{N}(x) + \hat{X} \succ 0$. This approach can be directly applied to general problems with constraints of the form $\mathcal{A}X \preceq b,\ X \succeq 0$. Obtaining an initial feasible starting point can be done using the self-dual embedding e.g. [6, 7].

We can now substitute for both Z, X and eliminate the first two (linear) blocks of equations. We obtain the following **single** block of equations for optimality. (By abuse of notation, we keep the symbol F for the nonlinear operator. The meaning is clear from the context.)

Theorem 2.2 *The primal-dual variables X, y, Z, with $X = \operatorname{u2sMat}(x + I) \succeq 0$, $Z = (\operatorname{Diag}(y) - Q) \succeq 0$, are optimal for (P),(D) if and only if they satisfy the single bilinear optimality equation*

$$F(x,y) := (\operatorname{Diag}(y) - Q)(\operatorname{u2sMat}(x) + I) = 0, \quad F(x,y) : \Re^{\binom{n}{2}} \times \Re^n \to \mathcal{M}^n . \tag{2.3}$$

■

This leads to the single perturbed optimality conditions that we use for our primal-dual method.

$$F_\mu(x,y) := (\operatorname{Diag}(y) - Q)(\operatorname{u2sMat}(x) + I) - \mu I = 0, \quad F_\mu(x,y) : \Re^{\binom{n}{2}} \times \Re^n \to \mathcal{M}^n . \tag{2.4}$$

When we implement GN we use $\operatorname{vec}(F_\mu(x,y)) = 0$.

This is a nonlinear (bilinear) overdetermined system. The linearization (or Gauss-Newton equation) for the search direction $\Delta x, \Delta y$ is (vec is understood)

$$\begin{array}{rcl} -F_\mu(x,y) & = & F'_\mu(x,y)\begin{pmatrix}\Delta x \\ \Delta y\end{pmatrix} \\ & = & \mathcal{Z}(\Delta x) + \mathcal{X}(\Delta y) \\ & = & (\operatorname{Diag}(y) - Q)\operatorname{u2sMat}(\Delta x) + \operatorname{Diag}(\Delta y)(\operatorname{u2sMat}(x) + I). \end{array} \tag{2.5}$$

This is a linear, full rank, overdetermined system and we find the least squares solution. We note that the first part $\mathcal{Z}(\Delta x)$ is the large part of the system since it has $n(n-1)/2$ variables. However, the operator $\mathcal{Z}$ is sparse if the data Q is sparse. The second part is the small part since it only has n variables, though the operator $\mathcal{X}$ is usually dense. This is the size of the Schur complement system that arises in the standard approaches for SDP. Sparse least squares techniques can be applied. In particular, the distinct division into two sets of columns can be exploited using projection and multifrontal methods, e.g. [4, 14, 20, 21].

The above system is full rank at each iteration, and, in addition, it has the strong property of being full rank at optimality, i.e. it does not necessarily get ill-conditioned as μ approaches 0. This is seen empirically by our numerical tests. We now prove this for a general primal-dual SDP pair.

Theorem 2.3 *Consider the primal-dual SDP pair (PSDP),(DSDP). Suppose that $\mathcal{A}$ is onto (full rank), $\mathcal{N}$ defines the null space as in (1.11), and X, y, Z are primal-dual optimal solutions for (P) and (D) that satisfy strict complementary slackness, i.e. $X + Z \succ 0$. Then the matrix of the system*

$$\begin{aligned} -F_\mu(x,y) &= F'_\mu(x,y)\begin{pmatrix}\Delta x\\ \Delta y\end{pmatrix} \\ &= (\mathcal{A}^*(y) - Q)\mathcal{N}(\Delta x) + \mathcal{A}^*(\Delta y)(\mathcal{N}(x) + \mathcal{A}^\dagger b) \end{aligned} \tag{2.6}$$

(F', Jacobian of $F(x,y)$) is full rank.

Proof. That the original overdetermined system $F'(X,y,Z)$ is full rank is proved in [19]. We observe that the system is obtained by block Gaussian elimination, which is equivalent to premultiplication by invertible operators. ■

3 Primal-Dual Interior-Point Algorithm

The primal-dual framework we use is given by the following.

Algorithm 3.1 *(p-d; GN with PCG)*

```
• Initialization:
    •• Input data: a real symmetric $n \times n$ matrix $Q$ (set $\operatorname{diag}(Q) = 0$)
    •• Positive tolerances: $\delta_1$ (relative duality gap), $\delta_2$ (accuracy for lss)
    •• Find initial feasible points:
            set:        $x^0 = 0; \quad y^0 = 2\operatorname{Diag}(q) - Q,$
       where $q_j = \|Q_{:,j}\|_2$;
       (guarantees: both $X := (\operatorname{u2sMat}(x) + I), Z := (\operatorname{diag}(y) - Q) \succ 0$.
    •• Set initial parameters:
            $\text{gap} = \operatorname{trace} ZX; \quad \mu = \text{gap}/n; \quad \text{optval} = \operatorname{trace} QX; \quad k = 0.$
• while $\{\frac{\text{gap}}{|\text{optval}|+1}\} > \delta_1$
```

•• **solve for the search direction** *(to tolerance* $\max\left\{\frac{\delta_2}{10}, \frac{\min\{\mu^{\frac{1}{3}},1\}}{10^6}\right\}$*, in a least squares sense with PCG, using k-th iterate* $(x,y) = (x^k, y^k)$*)*

$$F'_{\sigma\mu}(x,y)\begin{pmatrix}\Delta x\\ \Delta y\end{pmatrix} = -F_{\sigma\mu}(x,y),$$

where σ_k *centering,* $\mu_k = \operatorname{trace}(\operatorname{Diag}(y) - Q)(\operatorname{u2sMat}(x) + I)/n$

$$x^{k+1} = x + \alpha_k \Delta x, \quad y^{k+1} = y + \alpha_k \Delta y; \quad \alpha_k > 0,$$

so that both $(X := \operatorname{u2sMat}(x^{k+1}) + I), (Z := \operatorname{Diag}(y^{k+1}) - Q) \succ 0$*, before the crossover; so that sufficient decrease in* $\|ZX\|_F$ *is guaranteed, after the crossover.*

•• **update**

$$k \leftarrow k+1 \quad \textit{and then update}$$

$$\mu_k, \sigma_k \quad (\sigma_k = 0 \textit{ after the crossover}).$$

• **end (while)**.

• **Conclusion:** *optimal* X *is approx.* $\operatorname{u2sMat}(x) + I$

We use equation (2.4) and the linearization (2.5) to develop the primal-dual interior-point Algorithm 3.1, (This modifies the standard approach in e.g. [28].) We include a centering parameter σ_k (rather than the customary predictor-corrector approach). Our approach differs in that we have eliminated, in advance, the primal and dual linear feasibility equations. We work with an overdetermined nonlinear system rather than a square symmetrized system; thus we use a Gauss-Newton approach, [19]. We use a crossover step, i.e. we use affine scaling, $\sigma = 0$, and we do not backtrack to preserve positive definiteness of Z, X once we have found the region of quadratic convergence. The search direction is found using a preconditioned conjugate gradient method, lsqr [22]. The cost of each CG iteration is a (sparse) matrix multiplication and a diagonal matrix scaling, see e.g. (2.5).

There are many advantages for this algorithm:

•Primal and dual feasibility is exact during each iteration (assuming that the the basis for the null space was computed precisely);

•there is no (costly, dense) Schur complement system to form;

•There is no need to find Z^{-1} (which becomes ill-conditioned as μ goes to 0);

•the sparsity of the data Q is exploited completely;

•by the robustness of the algorithm, there is no need to enforce positivity of Z, X once μ gets small enough; q-quadratic convergence is obtained;

•the entire work of the algorithm lies in finding the search direction at each iteration by solving a (sparse) least squares problem using a CG type algorithm. Each iteration of the CG algorithm involves a (sparse) matrix-matrix multiplication and a diagonal row scaling of a matrix. The more efficiently we can solve these least squares problems, the faster our algorithm will be. Better preconditioners, better solvers, and better parallelization in the future will improve the algorithm;

•the techniques can be extended directly to general SDPs, depending on efficient (sparse) representations of the primal (and/or dual) feasibility equation.

3.1 Preconditioning

3.1.1 Optimal Diagonal Column Preconditioning

We begin with the simplest of the preconditioners. This has been identified as a successful preconditioner for least squares problems, see [16, Sect. 10.5], [26], and [10, Prop. 2.1(v)]. In the latter reference, it was shown to be the optimal diagonal preconditioner in the sense that, for $A, m \times n, m \geq n$ a full rank matrix, the solution of the optimization problem $\min \omega((AD)^T(AD))$, over all positive diagonal matrices D, with condition number $\omega(K) = \frac{\operatorname{trace}(K)/n}{\det(K)^{\frac{1}{n}}}$, is given by $D_{ii} = \frac{1}{\|A_{:i}\|}$. And, ω is a measure of the condition number, in the sense that it is bounded above and below by a constant times the standard condition number (ratio of largest to smallest singular values).

We first find the columns of the operator $F'(x,y) = \mathcal{Z}(x) + \mathcal{X}(y)$. The columns are ordered using $k = 1, 2, \dots$ where k represents $(i,j),\ 1 \leq i < j \leq n$ for the upper triangular part of X; and then represents $i = 1, \dots, n$ for the elements of y.

1.

$$\begin{array}{rcl} \mathcal{Z}(e_k) & = & \operatorname{vec}(\operatorname{Diag}(y) - Q)\operatorname{u2sMat}(e_k) \\ & = & \frac{1}{\sqrt{2}}\operatorname{vec}(\operatorname{Diag}(y) - Q)\left(e_i e_j^T + e_j e_i^T\right) \\ & = & \frac{1}{\sqrt{2}}\operatorname{vec}\left\{\left(y_i e_i e_j^T + y_j e_j e_i^T\right) - \left(Q_{:i} e_j^T + Q_{:j} e_i^T\right)\right\}. \end{array}$$

Therefore

$$\begin{array}{rcl} \|\mathcal{Z}(e_k)\|_F^2 & = & \frac{1}{2}\left\{\|y_i e_i - Q_{:i}\|^2 + \|y_j e_j - Q_{:j}\|^2\right\} \\ & = & \frac{1}{2}\left\{\|Q_{:i}\|^2 + \|Q_{:j}\|^2 + y_i^2 + y_j^2\right\}, \end{array} \tag{3.1}$$

since $\operatorname{diag}(Q) = 0$ by assumption. We see that this calculation is inexpensive since one need only find the norms and sums of the columns of the sparse matrix Q once, at the beginning of the algorithm.

2.

$$\mathcal{X}(e_i) = \operatorname{Diag}(e_i)(\operatorname{u2sMat}(x) + I).$$

Therefore

$$\|\mathcal{X}(e_i)\|_F^2 = \|X_{i,:}\|^2. \tag{3.2}$$

3.2 Crossover Criteria

Advantages of the GN approach include full rank of the Jacobian at optimality and a zero residual. Therefore, there is a local neighbourhood of quadratic convergence and once in this neighbourhood, we can safely apply affine scaling with step lengths of one without backtracking to maintain positive definiteness. Standard convergence results, e.g. [9, 18] show that the Gauss-Newton method applied to $F(x,y) = 0$ is locally q-quadrically convergent, since the Jacobian at the optimum is full column rank (one to one operator). We follow the notation in [9, Theorem 10.2.1] and discuss several constants used to determine the region of quadratic convergence. We let $\Delta s = \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix}$. Since we have a zero residual, the corresponding constant $\sigma = 0$. Since

$$\begin{array}{rcl} \|F'(\Delta s)\|_F & = & \|Z\mathrm{u2sMat}\,(\Delta x) + \mathrm{Diag}\,(\Delta y)X\|_F \\ & \leq & \|Z\mathrm{u2sMat}\,(\Delta x)\|_F + \|\mathrm{Diag}\,(\Delta y)X\|_F \\ & \leq & \|Z\|_F\|\mathrm{u2sMat}\,(\Delta x)\|_F + \|\mathrm{Diag}\,(\Delta y)\|_F\|X\|_F \\ & = & \|Z\|_F\|\Delta x\|_2 + \|\Delta y\|_2\|X\|_F \\ & \leq & \sqrt{\|Z\|_F^2 + \|X\|_F^2}\|\Delta s\|_2, \text{ (by Cauchy-Schwartz inequality)} \end{array}$$

the bound on the norm of the Jacobian is

$$\alpha = \sqrt{\|Z\|_F^2 + \|X\|_F^2}.$$

$$\begin{array}{rcl} \|F'(s-\bar{s})(\Delta s)\|_F & = & \|(Z-\bar{Z})\mathrm{u2sMat}\,(\Delta x) + \mathrm{Diag}\,(\Delta y)(X-\bar{X})\|_F \\ & \leq & \|(Z-\bar{Z})\|_F\|\mathrm{u2sMat}\,(\Delta x)\|_F + \|\mathrm{Diag}\,(\Delta y)\|_F\|(X-\bar{X})\|_F \\ & = & \|(y-\bar{y})\|_2\|\Delta x\|_2 + \|\Delta y\|_2\|(x-\bar{x})\|_2. \end{array}$$

Therefore the Lipschitz constant is

$$\gamma = 1. \tag{3.3}$$

Now suppose that the optimum s^* is unique and the smallest singular value satisfies $\sigma_{\min}(F'(s)) \geq \sqrt{K}$, for all s in an ϵ_1 neighbourhood of s^*, for some constant $K > 0$. Following [9, Page 223], we define

$$\epsilon := \min\left\{\epsilon_1, \frac{1}{K\alpha\gamma}\right\} = \min\left\{\epsilon_1, \frac{1}{K\sqrt{\|Z^*\|_F^2 + \|X^*\|_F^2}}\right\}.$$

Then q-quadratic convergence is guaranteed once the current iterate is in this ϵ neighbourhood of the optimum. One possible heuristic for this is to start the crossover if $\sigma_{\min}(F'(s)) \geq 2\|ZX\|_2$. In our tests we started the crossover when the relative duality gap $\frac{\mathrm{trace}\,ZX}{|\mathrm{trace}\,QX|+1} < .1$. This simpler heuristic never failed to converge, though q-quadratic convergence was not immediately detected.

4 Numerical Tests

We have performed preliminary testing to illustrate some of the features of the method. We see that the crossover to using a steplength of 1, not maintaining interiority, and using affine scaling, has a significant effect on both the number of iterations and the conditioning of the linear system. The number of iterations are approximately halved. The best results were obtained by staying well-centered before the crossover. This is in line with what we know from the theory. This appeared to help with the conditioning of the Jacobian and lowered the number of CG iterations. For simplicity, the crossover was done when the relative duality gap was less than .1 and we used a steplength of 1 after the crossover, rather than a line search to guarantee sufficient decrease in $\|ZX\|_F$. In all our tests on random problems, we never failed to converge to a high accuracy solution.

The tests were done using MATLAB 6.0.0.88 (R12) on a SUNW, Ultra $5-10$ with one GB of RAM using SunOS 5.8 (denoted by SUN), as well as on a Toshiba Pentium II, 300

MHZ, with 128 MB of RAM (denoted by PII). 99% of the cputime was spent on finding the search direction, i.e. in the PCG part of the code in finding the (approximate) least squares solution of the Gauss-Newton equation. Randomly generated problems of size up to $n = 550$ were solved in a reasonable time to high accuracy. To prevent zero columns in Q, the first upper (lower) diagonal of Q was set to a vector of ones.

The cost for the early iterations was very low, e.g. 21, 50 CG iterations, 24, 58 cpu seconds for the first two iterations for $n = 365$ on the SUN. This cost increased as the relative duality gap (and the requested tolerance for the least squares solution) decreased. Low accuracy solutions were obtained quickly, e.g. one decimal accuracy after 4 to 5 iterations. The cost per iteration increased steadily and then levelled off near the optimum.

In Table 4.1 we show the decrease in the number of iterations compared to the crossover tolerance, i.e. an earlier crossover leads to faster convergence. However, there can be loss of positive definiteness at the end, as observed by the small (less than the tolerance in absolute value) negative eigenvalues. The low number of iterations and high accuracy in the solution are notable. In Tables 4.2 to 4.6 we see the effect of preconditioning, dimension, and density on the cputime of the algorithm. The density is $1/n$ and $3/n$, i.e. the density is increasing since the number of variables is order n^2. We used two machines with different RAM to illustrate that sparsity is exploited. (The Pentium II had only 128 MB RAM.) We were able to solve very large problems as long as the data (matrix Q) was sparse. Tables 4.5 and 4.5 illustrate the importance of the optimal diagonal preconditioning.

A typical plot of the number of iterations versus the values of $-log_{10}(\|ZX\|_F)$ is given in Figure 4.1. Figure 4.2 illustrates the number of nonzeros in Q versus the increase in total cputime (when dimension runs from 15 to 105 with increments of 10).

4.1 Tables and Figures

Crossover toler. in relative gap	nmbr of major iterations	norm(ZX) at end	min. eig. pos. violation in Z, X
1.0e-1	11	3.6281e-12	-4.6478e-12
1.0e-3	14	1.9319e-13	-2.1580e-13
1.0e-5	16	2.0917e-11	-2.2839e-11
1.0e-7	18	1.8293e-13	-1.5546e-13
1.0e-9	20	7.2534e-10	6.7006e-12
1.0e-11	20	7.2544e-10	6.7004e-12

Table 4.1: Size of Q, $n = 25$; Requested Accuracy in relative duality gap: 1.0e-10; Density is $\frac{1}{25}$; Optimal Value is 5.7041e+01.

5 Conclusion

We have presented an alternative approach to solving SDPs and applied it to the SDP relaxation of the Max-Cut problem. The approach is based on the strong/robust primal-dual

Dimension of Q n	total cpu seconds	nmbr of major iterations	norm(ZX) at end	violation of eig. pos. in Z, X
55	6.6000e+01	11	9.0884e-09	-1.7515e-08
65	1.1234e+02	11	7.2081e-10	-2.5326e-09
75	1.9134e+02	13	2.1361e-11	-1.7416e-10

Table 4.2: Requested Accuracy in relative duality gap: 1.0e-10; Density is $\frac{1}{n}$; crossover tolerance 1.0e-01; optimal values of order 150; SUN computer.

Dimension of Q n	total cpu seconds	nmbr of major iterations	norm(ZX) at end	violation of eig. pos. in Z, X
165	2635	14	7.7056e-14	-2.775e-14
175	6778.6	15	8.0369e-14	-2.5371e-14
185	6908.8	15	9.0987e-14	-4.011e-14
195	11397	15	1.7324e-12	-3.2036e-11
205	8958.5	15	1.0029e-13	-3.7858e-14

Table 4.3: Requested Accuracy in relative duality gap: 1.0e-14; Density is $\frac{1}{n}$; crossover tolerance 1.0e-01; optimal values of order 1000; SUN computer.

Dimension of Q n	total cpu seconds	nmbr of major iterations	norm(ZX) at end	violation of eig. pos. in Z, X
15	7.75	9	1.7535e-014	-3.0863e-015
25	17.36	10	1.9291e-013	-1.5217e-013
35	30.43	10	1.1044e-012	-8.4977e-013
45	59.87	11	2.7926e-011	-2.763e-011
55	86.89	12	4.0912e-011	-3.7568e-011
65	131.11	11	4.9924e-012	-5.0519e-012
75	275.89	12	1.0448e-008	-1.4504e-008
85	468.46	14	8.6208e-011	-3.4404e-010
95	414.03	11	7.4253e-011	-3.0935e-010
105	878.64	15	7.5993e-010	-1.4946e-008

Table 4.4: Requested Accuracy in relative duality gap: 1.0e-10; Density is $\frac{3}{n}$; crossover tolerance 1.0e-01; optimal values of order 150; PII computer.

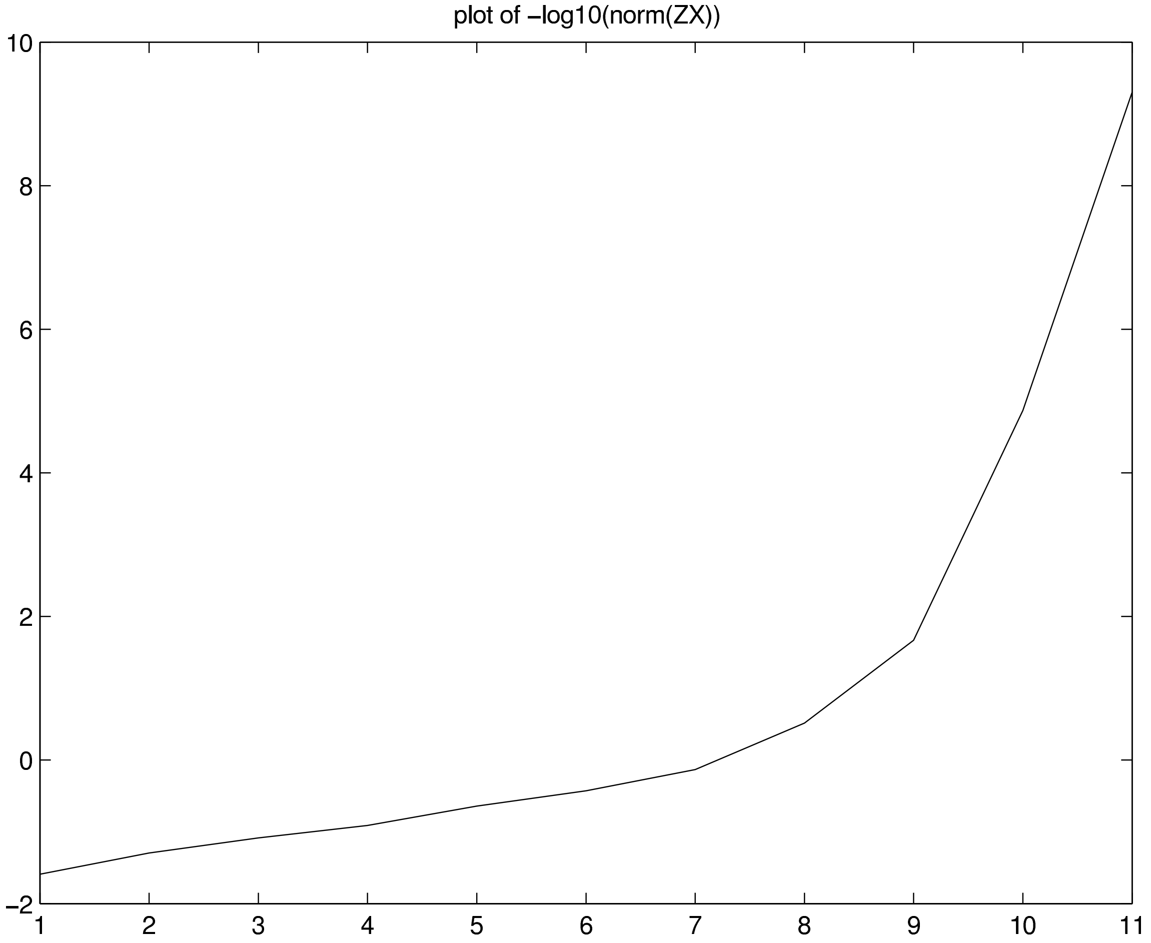

Figure 4.1: $n = 55$. Crossover at iteration 7.

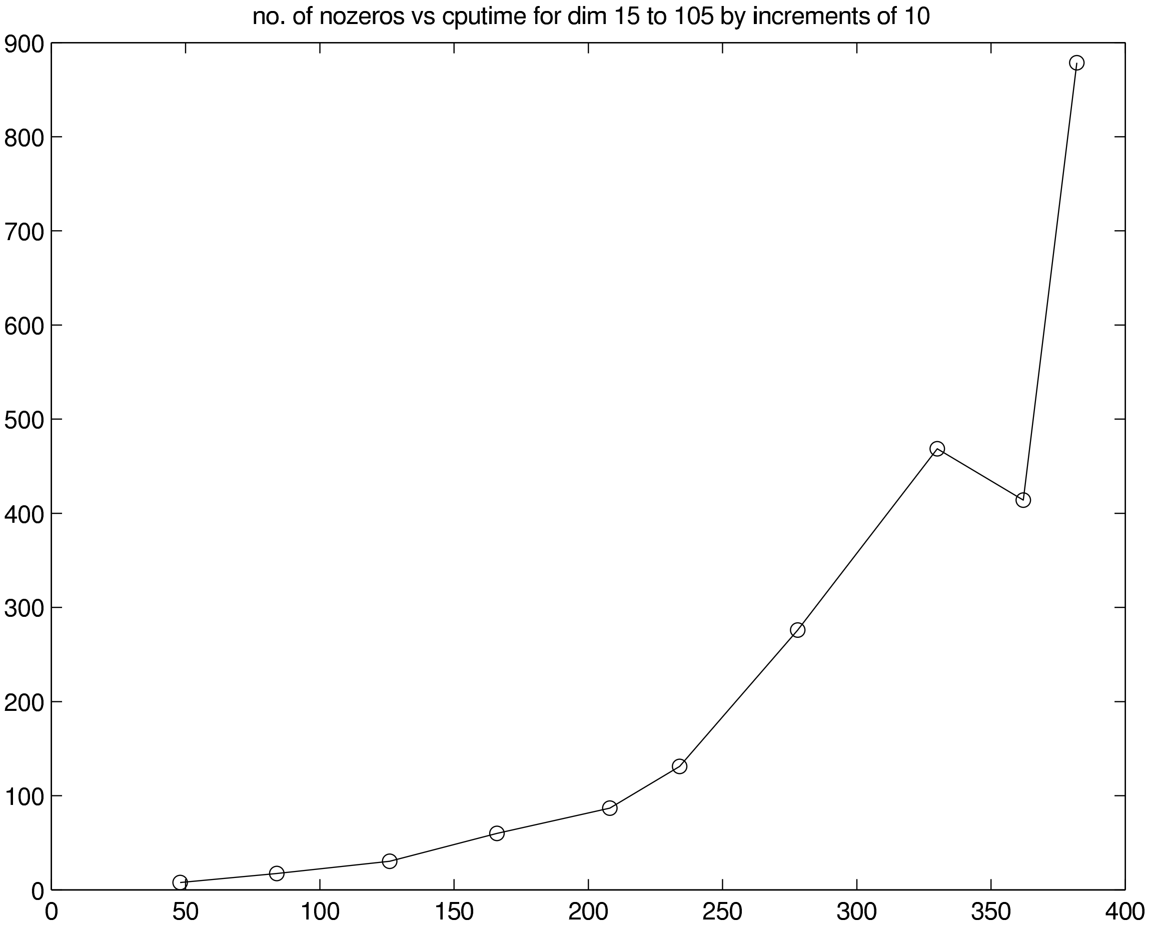

Figure 4.2: Nonzeros vs cputime. $n = 15 : 10 : 105$.

Dimension of Q n	total cpu seconds	nmbr of major iterations	norm(ZX) at end	violation of eig. pos. in Z, X
15	30.1	11	1.9039e-011	-3.5098e-012
25	134.13	15	1.9742e-009	-2.7465e-010
35	117.27	12	2.5534e-010	-8.8532e-011
45	174.34	11	1.3799e-008	-1.9154e-008
55	373.6	14	7.3741e-009	-3.1854e-009
65	1388.1	19	9.6109e-009	-2.7634e-009
75	507.46	13	3.5887e-009	-3.3886e-009
85	1856.1	16	6.4806e-011	-4.7964e-011

Table 4.5: **Without preconditioning**; Requested Accuracy in relative duality gap: 1.0e-10; Density is $\frac{3}{n}$; crossover tolerance 1.0e-01; optimal values of order 150; PII computer.

Dimension of Q n	total cpu seconds	nmbr of major iterations	norm(ZX) at end	violation of eig. pos. in Z, X
155	5493.6	14	6.068e-014	-4.3061e-013
165	4595.4	13	3.8625e-014	-2.8454e-013

Table 4.6: Requested Accuracy in relative duality gap: 1.0e-14; Density is $\frac{1}{n}$; crossover tolerance 1.0e-01; optimal values of order 150; PII computer.

path-following interior-point framework. But, it can still exploit sparsity in the data. Neither the symmetrization step nor the Schur complement system are used. The method uses basic tools that are successful for solving an overdetermined system of nonlinear equations with zero residual, i.e. PCG applied to the GN method. We have shown how to derive preconditioners for this approach and, in particular, derived the optimal diagonal column scaling preconditioner. The total cost of an iteration lies in the solution of a linear least squares problem. This least squares problem is solved using the (preconditioned) conjugate gradient type method of Paige and Saunders [22]. The cost of each CG iteration is a sparse matrix multiplication (essentially $Z\Delta X$) and a matrix scaling (essentially $\mathrm{Diag}\,(\Delta y)X$).

Numerical tests are ongoing as are extensions to more general SDPs. This includes a comparison between using first order methods and an inexact GN framework, i.e. solving the least squares problem to a desired accuracy so that the work is of the same order as one iteration of first order methods.

Ongoing research involves: further testing on the SDPLIB problem set; developing improved block diagonal and partial Cholesky preconditioners; parallelization of the PCG approach; purification techniques to recover positive semi-definiteness; and proving convergence for the crossover technique.

Acknowledgments

The author is indebted to Michael Saunders (Department of Management Science and Engineering, Stanford University) for providing the *LSQR* MATLAB code for the PCG-like

method. The author would also like to thank Nick Higham (Department of Mathematics, University of Manchester) and Levent Tunçel (Department of Combinatorics and Optimization, Univerity of Waterloo) for many helpful discussions and comments.

References

[1] M.F. ANJOS, N. HIGHAM, and H. WOLKOWICZ. A semidefinite programming approach for the closest correlation matrix problem. Technical Report CORR 2001-in progress, University of Waterloo, Waterloo, Ontario, 2001.

[2] M.F. ANJOS and H. WOLKOWICZ. A strengthened SDP relaxation via a second lifting for the Max-Cut problem. *Special Issue of Discrete Applied Mathematics devoted to Foundations of Heuristics in Combinatorial Optimization*, to appear.

[3] S. J. BENSON, Y. YE, and X. ZHANG. Solving large-scale sparse semidefinite programs for combinatorial optimization. *SIAM J. Optim.*, 10(2):443–461 (electronic), 2000.

[4] Å. BJÖRCK. Methods for sparse least squares problems. In J.R. Bunch and D. J. Rose, editors, *Sparse Matrix Computations*, pages 177–199. Academic Press, New York, 1976.

[5] S. BURER and R.D.C. MONTEIRO. A projected gradient algorithm for solving the maxcut SDP relaxation. *Optimization Methods and Software*, 15(3-4):175–200, 2001.

[6] E. de KLERK. *Interior point methods for semidefinite programming.* PhD thesis, Delft University, 1997.

[7] E. de KLERK, C. ROOS, and T. TERLAKY. Initialization in semidefinite programming via a self-dual skew-symmetric embedding. *Oper. Res. Lett.*, 20(5):213–221, 1997.

[8] R.S DEMBO, S.C. EISENSTAT, and T. STEIHAUG. Inexact Newton methods. *SIAM J. Numer. Anal.*, 19:400–408, 1982.

[9] J.E. DENNIS Jr. and R.B. SCHNABEL. *Numerical methods for unconstrained optimization and nonlinear equations*, volume 16 of *Classics in Applied Mathematics.* Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1996. Corrected reprint of the 1983 original.

[10] J.E. DENNIS Jr. and H. WOLKOWICZ. Sizing and least-change secant methods. *SIAM J. Numer. Anal.*, 30(5):1291–1314, 1993.

[11] M.M. DEZA and M. LAURENT. *Geometry of cuts and metrics.* Springer-Verlag, Berlin, 1997.

[12] A.V. FIACCO and G.P. McCORMICK. *Nonlinear programming.* Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, second (first 1968) edition, 1990. Sequential unconstrained minimization techniques.

[13] M.X. GOEMANS and F. RENDL. Combinatorial optimization. In H. Wolkowicz, R. Saigal, and L. Vandenberghe, editors, *HANDBOOK OF SEMIDEFINITE PROGRAMMING: Theory, Algorithms, and Applications*. Kluwer Academic Publishers, Boston, MA, 2000.

[14] G. H. GOLUB and V. PEREYRA. The differentiation of pseudoinverses and nonlinear least squares problems whose variables separate. *SIAM J. Numer. Anal.*, 10:413–432, 1973.

[15] M. GONZALEZ-LIMA and H. WOLKOWICZ. Primal-dual methods using least squares for linear programming. Technical Report CORR 2001-in progress, University of Waterloo, Waterloo, Ontario, 2001.

[16] A. GREENBAUM. *Iterative methods for solving linear systems*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1997.

[17] C. HELMBERG and F. RENDL. A spectral bundle method for semidefinite programming. *SIAM J. Optim.*, 10(3):673 – 696, 2000.

[18] C. T. KELLEY. *Iterative methods for linear and nonlinear equations*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1995. With separately available software.

[19] S. KRUK, M. MURAMATSU, F. RENDL, R.J. VANDERBEI, and H. WOLKOWICZ. The Gauss-Newton direction in linear and semidefinite programming. *Optimization Methods and Software*, 15(1):1–27, 2001.

[20] P. MATSTOMS. *The Multifrontal Solution of Sparse Linear Least Squares Problems*. Licentiat thesis, Department of Mathematics, Linköping University, Sweden, 1991.

[21] P. MATSTOMS. Sparse QR factorization in MATLAB. *ACM Trans. Math. Software*, 20:136–159, 1994.

[22] C.C. PAIGE and M.A. SAUNDERS. LSQR: an algorithm for sparse linear equations and sparse least squares. *ACM Trans. Math. Software*, 8(1):43–71, 1982.

[23] M.V. RAMANA, L. TUNÇEL, and H. WOLKOWICZ. Strong duality for semidefinite programming. *SIAM J. Optim.*, 7(3):641–662, 1997.

[24] F. RENDL, R. SOTIROV, and H. WOLKOWICZ. Exploiting sparsity in interior point methods: applications to SDP and QAP. Technical Report in progress, University of Waterloo, Waterloo, Canada, 2001.

[25] M.J. TODD. A study of search directions in primal-dual interior-point methods for semidefinite programming. *Optim. Methods Softw.*, 11&12:1–46, 1999.

[26] A. VAN der SLUIS. Condition numbers and equilibration of matrices. *Numer. Math.*, 14:14–23, 1969/1970.

[27] H. WOLKOWICZ, R. SAIGAL, and L. VANDENBERGHE, editors. *HANDBOOK OF SEMIDEFINITE PROGRAMMING: Theory, Algorithms, and Applications*. Kluwer Academic Publishers, Boston, MA, 2000. xxvi+654 pages.

[28] S. WRIGHT. *Primal-Dual Interior-Point Methods*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, Pa, 1996.

A Current Numerical Results

The matlab codes have been polished up since the paper was submitted. Though the code is still just in matlab (no mex files), the times have improved dramatically. In addition, the SDPLIB test problems have been solved. This page contains the latest numerical results.

Dim Q	cpu sec	major	rel. opt.	rel. norm(ZX)	rel. viol.	nnz
n		iter.	at end	at end	eig. pos. Z, X	in Q
165	1120	14	869.61	8.9414e-17	-3.3566e-17	476
175	1891	15	928.97	8.5746e-17	-3.2607e-17	504
185	3079.6	15	1041.8	8.9598e-17	-3.7512e-17	538
195	3251.9	14	1045.7	8.2733e-17	-4.2463e-17	572
205	5874.7	17	1207.7	1.0142e-16	-3.5923e-17	602

Table A.1: Random problems; Requested Accuracy in relative duality gap: 1.0e-14; Density is $\frac{1}{n}$; crossover tolerance 1.0e-01; optimal values of order 1000; SUN computer.

Dim Q	cpu sec	cpu sec	cpu sec	major	rel. opt.	rel. norm(ZX)	rel. viol.	nnz
n	to 1 dec	to 7 dec	to 14 dec	iter.	at end	at end	eig. pos. Z, X	in Q
100	39.66	127.23	192.36	13	226.16	2.5926e-17	-5.0771e-17	638
124	70.93	351.73	502.57	14	141.99	3.3609e-17	-2.8961e-16	410
124	71.15	241.03	342.4	14	269.88	4.4706e-17	-1.0508e-16	760
124	56.32	392.78	547.51	14	467.75	2.3752e-17	-3.3513e-17	1364
124	82.79	370.25	577.79	14	864.41	1.6515e-16	-8.7386e-16	2666
250	648.9	2339.9	3087.2	15	317.26	2.1095e-16	-4.7591e-15	892
250	537.79	2384	2805.6	14	531.93	2.8552e-16	-5.5422e-16	1472
250	620.82	3844.2	5620	13	981.17	2.7276e-17	-2.6909e-16	2816
250	947.5	7834.3	10893	15	1682	6.5743e-16	-2.2708e-15	5092
500	6729.4	32472	42094	20	598.15	2.9931e-17	-1.5378e-16	1701
500	5422.5	38232	59391	18	1070.1	7.2114e-16	-9.7197e-15	2939
500	7282.3	42758	58360	16	1848	1.2577e-15	-2.0375e-14	5210
500	11548	80822	1.3137e+05	17	3566.7	6.1352e-15	-5.7 509e-14	10740

Table A.2: Requested Accuracy in relative duality gap: 1.0e-14; SDPLIB test problems; crossover tolerance 1.0e-01; SUN computer.

Dim Q	cpu sec	major	rel. opt.	rel. norm(ZX)	rel. viol.	nnz
n		iter.	at end	at end	eig. pos. Z, X	in Q
100	96.34	12	226.16	1.2399e-13	-1.2062e-12	638
124	252.57	13	141.99	1.2238e-14	-2.3655e-13	410
124	171.99	13	269.88	2.0918e-11	-1.1324e-11	760
124	292.48	13	467.75	7.0013e-13	-5.9152e-13	1364
124	289.21	13	864.41	5.7229e-11	-1.035e-11	2666
250	1134.9	14	317.26	2.9025e-11	-5.5891e-11	892
250	1058.3	14	531.93	2.8552e-16	-5.5638e-16	1472
250	1633.6	12	981.17	8.9829e-12	-5.8956e-12	2816
250	3036	14	1682	3.481e-11	-2.1537e-11	5092
500	14669	19	598.15	4.12e-13	-2.9424e-12	1701
500	21489	17	1070.1	1.0229e-11	-2.7013e-11	2939
500	20691	15	1848	3.6924e-11	-1.364e-11	5210
500	46752	16	3566.7	3.3634e-11	-1.5e-11	10740

Table A.3: Requested Accuracy in relative duality gap: 1.0e-10; SDPLIB test problems; crossover tolerance 1.0e-01; SGI (Irix6.5_64-Mips) computer.