

VSDP: A Matlab toolbox for verified
semidefinite-quadratic-linear programming

Version 2012

V. Härter, C. Jansson and M. Lange

Institute for Reliable Computing
Hamburg University of Technology

November 29, 2012

Abstract

VSDP is a software package that is designed for the computation of verified results in conic programming. The current version of VSDP supports the constraint cone consisting of the product of semidefinite cones, second-order cones and the nonnegative orthant. It provides functions for computing rigorous error bounds of the true optimal value, verified enclosures of ϵ -optimal solutions, and verified certificates of infeasibility. All rounding errors due to floating point arithmetic are taken into account.

VSDP is completely written in MATLAB. It uses INTLAB, and thus interval input data are supported as well. Via its interface, VSDP provides an easy access to the conic solvers CSDP, SeDuMi, SDPA, SDPT3, as well as LPSOLVE and LINPROG.

Numerical results for the NETLIB LP library, SDPLIB, DIMACS and Kocvara's test library for problems from structural optimization are presented in the second part of this manual. Many of these test problems are challenging real world problems of large scale.

The software and this manual can be downloaded from <http://www.ti3.tu-harburg.de/jansson/vsdp/>.

Copyright (C) 2012 Christian Jansson

Institute for Reliable Computing
Hamburg University of Technology
Schwarzenbergstr. 95, 21071 Hamburg, Germany
Email: jansson@tu-harburg.de

This program is free software for private and academic use. Commercial use or use in conjunction with a commercial program which requires VSDP or part of VSDP to function properly is prohibited. This program is distributed WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

Contents

1	Introduction	3
2	Installation and Notation	4
2.1	Installation	4
2.2	The Conic Programming Problem	5
3	Getting Started with VSDP	8
3.1	Linear Programming	8
3.2	Second Order Cone Programming	11
3.3	Semidefinite Programming	14
3.4	A Priori Upper Bounds for Optimal Solutions	19
3.5	Rigorous Certificates of Infeasibility	20
3.6	Handling Free Variables	23
4	Statistics of the Numerical Results	25
4.1	SDPLIB	25
4.2	NETLIB LP test library	26
4.3	DIMACS	27
4.4	Kovcara's Library of Structural Optimization Problems	27
5	Appendix A	28
5.1	A Quick Reference for VSDP	28
6	Appendix B - Numerical Results	32

1 Introduction

The extraordinary success and extensive growth of conic programming, especially of semidefinite programming, is due to the polynomial time solvability by interior point methods and the large amount of practical applications. Sometimes conic programming solvers fail to find a satisfactory approximation. Occasionally, they state that the problem is infeasible although it contains feasible solutions, or vice versa, see [15]. Verified error bounds, also called rigorous error bounds, means that, in the presence of rounding errors due to floating point arithmetic, the computed bounds are claimed to be valid with mathematical certainty. In other words, all rounding errors are taken into consideration. The major task of VSDP is to obtain a guaranteed accuracy by postprocessing the approximations produced by conic solvers. In our tests on 255 problems from different test libraries it turned out that the computed error bounds and enclosures depend heavily on the quality of the approximate solutions produced by the conic solvers. In particular, if the conic solver provides poor approximations, VSDP cannot compute satisfactory bounds. However, in this case the user is warned that possibly something went wrong, needs further attention, or requires to change to another conic solver.

Computational errors fall into three classes: intentional errors, like idealized models or discretization, unavoidable errors like uncertainties in parameters, and unintentional bugs and blunders in software and hardware. VSDP controls rigorously rounding errors. Please contact us, if you discover any unintentional bugs (jansson@tu-harburg.de, marko.lange@tu-harburg.de). In the latter case the version numbers of the used software together with an appropriate M-file should be provided. Any suggestions, comments, and criticisms are welcome. Many thanks in advance.

For theoretical details of the implemented algorithms in VSDP we refer to [12], [13], [14] and [15]. The reader is referred to [30] for verified results for other problems, such as nonlinear systems, eigenvalue problems or differential equations. The main differences to the first version of VSDP ([11]) are (i) the possibility to solve additional linear and second order cone programming problems, (ii) the extra robustness due to the ability to handle free variables, and (iii) the easy access to several conic solvers. All the numerical experiments in this paper are performed on a 2.80GHz Intel i7-2640M with 8 GB memory.

2 Installation and Notation

2.1 Installation

In order to run VSDP version 2012, the following requirements have to be fulfilled:

- VSDP runs on MATLAB R2009a or later [33].
- INTLAB, the toolbox for interval computation [29], is required.
- For full functionality at least one of the following approximate solvers has to be installed:

CSDP [6], SEDUMI [31], SDPA [10] or SDPT3 [34].

Optionally, LPSOLVE [1] or MATLAB's linprog [32] can be used, if only linear programs are solved.

VSDP consists of MATLAB m-files only. Thus it is platform-independent. VSDP runs on UNIX-like workstations as well as on Windows PCs. The software and this manual can be downloaded from <http://www.ti3.tu-harburg.de/jansson/vsdp/>. Once you downloaded the zip file, extract its contents to the directory where you want to install VSDP.

After the files have been extracted, add the VSDP directory and its sub-directories to the MATLAB search path. This can be done in MATLAB by using the set path dialog box or by the command `addpath`. Alternatively, the initialization function `vsdpinit` can be called to add the necessary paths. If all requirements are fulfilled and the corresponding directories have been added, VSDP is fully functional. The `vsdpinit`-function can also be used to initialize VSDP with some global settings. When typing `vsdpinit('sdpt3',1)` into the MATLAB command window the “welcome logo”

```
-----
***** Initialization of VSDP *****
*
*   VSDP v2012:   Verified Conic Programming
*               Christian Jansson, jansson@tu-harburg.de
*               Viktor Haerter, viktor.haerter@tu-harburg.de
*               Marko Lange, marko.lange@tu-harburg.de
*               Institute for Reliable Computing
*               Hamburg University of Technology,
*               Germany,  October 2012
*
***** VSDP is now ready for use *****
*****
```

confirms the successful initialization of the software package.

2.2 The Conic Programming Problem

Let $\mathbb{R}_+^n$ denote the nonnegative orthant, and let

$$\mathbb{L}^n := \{x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \in \mathbb{R}^n : x_1 \geq \|x_2\|_2\} \quad (2.1)$$

be the Lorentz cone. We denote by $\langle c, x \rangle := c^T x$ the usual Euclidean inner product of vectors in $\mathbb{R}^n$.

The set

$$\mathbb{S}_+^n := \{X \in \mathbb{R}^{n \times n} : X = X^T, v^T X v \geq 0, \forall v \in \mathbb{R}^n\} \quad (2.2)$$

denotes the cone of symmetric positive semidefinite $n \times n$ matrices. For symmetric matrices X, Y the inner product is given by

$$\langle X, Y \rangle := \text{trace}(XY). \quad (2.3)$$

Let A^f and A^l be a $m \times n_f$ and a $m \times n_l$ matrix, respectively, and let A_i^q be $m \times q_i$ matrices for $i = 1, \dots, n_q$. Let $x^f \in \mathbb{R}^{n_f}$, $x^l \in \mathbb{R}^{n_l}$, $x_i^q \in \mathbb{R}^{q_i}$ and $b \in \mathbb{R}^m$. Moreover, let $A_{1,j}^s, \dots, A_{m,j}^s, C_j^s, X_j^s$ be symmetric ($s_j \times s_j$) matrices for $j = 1, \dots, n_s$.

Now we can define the conic semidefinite-quadratic-linear programming problem in primal standard form:

$$\begin{aligned} \hat{f}_p := \min \quad & \langle c^f, x^f \rangle + \langle c^l, x^l \rangle + \sum_{i=1}^{n_q} \langle c_i^q, x_i^q \rangle + \sum_{j=1}^{n_s} \langle C_j^s, X_j^s \rangle \\ \text{s.t.} \quad & A^f x^f + A^l x^l + \sum_{i=1}^{n_q} A_i^q x_i^q + \sum_{j=1}^{n_s} \mathcal{A}_j^s(X_j^s) = b \\ & x^f \in \mathbb{R}^{n_f}, \quad \text{"free variables"} \\ & x^l \in \mathbb{R}_+^{n_l}, \quad \text{"nonnegative variables"} \\ & x_i^q \in \mathbb{L}^{q_i}, i = 1, \dots, n_q, \quad \text{"SOCP variables"} \\ & X_j^s \in \mathbb{S}_+^{s_j}, j = 1, \dots, n_s, \quad \text{"SDP variables"}. \end{aligned} \quad (\text{CP})$$

Here, the linear operator

$$\mathcal{A}_j^s(X_j^s) := (\langle A_{1,j}^s, X_j^s \rangle, \dots, \langle A_{m,j}^s, X_j^s \rangle)^T \quad (2.4)$$

maps the symmetric matrices X_j^s to $\mathbb{R}^m$.

By definition the vector x^f contains all unconstrained or free variables, whereas all other variables are bounded by conic constraints. In several applications some solvers (for example SDPA or CSDP) require that free variables are converted into the difference of nonnegative variables. Besides the mayor disadvantage that this transformation is numerical unstable, it also increases the number of variables of the particular problems. In VSDP version 2012 free variables can be handled in a numerical stable manner, see Section 3.6.

The objective function is a linear function and all equality constraints are linear, and thus conic programming is an extension of linear programming with additional conic constraints.

The adjoint operator $(\mathcal{A}_j^s)^*$ of the linear operator $\mathcal{A}_j^s$ is

$$(\mathcal{A}_j^s)^* y := \sum_{k=1}^{n_s} A_{k,j}^s y_k. \quad (2.5)$$

The dual problem associated with the primal problem (CP) is

$$\begin{aligned}
\hat{f}_d &:= \max b^T y \\
\text{s.t. } & (A^f)^T y + z^f = c^f, \quad z^f \in \mathbb{R}^{n_f}, \quad z^f = 0, \\
& (A^l)^T y + z^l = c^l, \quad z^l \in \mathbb{R}_+^{n_l}, \\
& (A_i^q)^T y + z_i^q = c_i^q, \quad z_i^q \in \mathbb{L}^{q_i}, \quad i = 1, \dots, n_q, \\
& (\mathcal{A}_j^s)^* y + Z_j^s = C_j^s, \quad Z_j^s \in \mathbb{S}_+^{s_j}, \quad j = 1, \dots, n_s.
\end{aligned} \tag{DCP}$$

For the conic programming problem weak duality can be proved, but not strong duality.

Occasionally, it is useful to represent the conic programming problem in a more compact form by using the vectorization operator vec . For any quadratic matrix X this operator is defined by

$$x = \text{vec}(X) = (X_{11}, \dots, X_{n1}, X_{12}, \dots, X_{n2}, \dots, X_{1n}, \dots, X_{nn})^T. \tag{2.6}$$

The inverse operation is denoted by $\text{mat}(x)$ such that $\text{mat}(\text{vec}(X)) = X$.

We condense the above quantities as follows:

$$\begin{aligned}
\bullet \quad x^s &:= (\text{vec}(X_1^s); \dots; \text{vec}(X_{n_s}^s)), \quad c^s := (\text{vec}(C_1^s); \dots; \text{vec}(C_{n_s}^s)), \\
\bullet \quad x^q &:= (x_1^q; \dots; x_{n_q}^q), \quad c^q := (c_1^q; \dots; c_{n_q}^q), \\
\bullet \quad x &:= (x^f; x^l; x^q; x^s), \quad c := (c^f; c^l; c^q; c^s).
\end{aligned} \tag{2.7}$$

Here c^q and x^q consist of $\bar{q} = \sum_{i=1}^{n_q} q_i$ components, and c^s, x^s are vectors of length $\bar{s} := \sum_{j=1}^{n_s} s_j^2$. The total length of x, c is equal to $n = n_f + n_l + \bar{q} + \bar{s}$. As in the syntax of MATLAB the separating column denotes the vertical concatenation of the corresponding vectors.

The matrices describing the linear equations are condensed as follows:

$$\begin{aligned}
\bullet \quad A^s &= (A_1^s, \dots, A_{n_s}^s), \quad \text{where } A_j^s = (\text{vec}(A_{1,j}^s), \dots, \text{vec}(A_{m,j}^s))^T, \\
\bullet \quad A^q &= (A_1^q, \dots, A_{n_q}^q), \\
\bullet \quad A &= (A^f, A^l, A^q, A^s).
\end{aligned} \tag{2.8}$$

Let the constraint cone K and its dual cone K^* be

$$K := \mathbb{R}^{n_f} \times \mathbb{R}_+^{n_l} \times \mathbb{L}^{q_1} \times \dots \times \mathbb{L}^{q_{n_q}} \times \mathbb{S}_+^{s_1} \times \dots \times \mathbb{S}_+^{s_{n_s}}, \tag{2.9}$$

$$K^* := \{0\}^{n_f} \times \mathbb{R}_+^{n_l} \times \mathbb{L}^{q_1} \times \dots \times \mathbb{L}^{q_{n_q}} \times \mathbb{S}_+^{s_1} \times \dots \times \mathbb{S}_+^{s_{n_s}}. \tag{2.10}$$

With these abbreviations we obtain the following blockform of the conic problem (CP)

$$\begin{aligned}
\hat{f}_p &:= \min c^T x \\
\text{s.t. } & Ax = b, \\
& x \in K.
\end{aligned} \tag{CP}$$

The corresponding dual problem has the form

$$\begin{aligned}
\hat{f}_d &:= \max b^T y \\
\text{s.t. } & z = c - (A)^T y \in K^*.
\end{aligned} \tag{DCP}$$

For a linear programming problem a vector $x \in \mathbb{R}^{n_l}$ is in the interior of the cone $K = \mathbb{R}_+^{n_l}$, if $x_i > 0$ for $i = 1, \dots, n_l$. For a vector $x \in \mathbb{R}^n$

let $\lambda_{\min}(x) := x_1 - \|x\|_2$ denote the smallest spectral value of x (see [2]). Then for second order cone programming problems a vector $x \in \mathbb{R}^{\bar{q}}$ is in the interior of the cone $K = \mathbb{L}^{q_1} \times \dots \times \mathbb{L}^{q_{n_q}}$, if $\lambda_{\min}(x_i) > 0$ for $i = 1, \dots, n_q$. Furthermore, for a symmetric matrix $X \in \mathbb{S}^n$ let $\lambda_{\min}(X)$ denote the smallest eigenvalue of X . Then for semidefinite programming problems a symmetric block matrix

$$X = \begin{pmatrix} X_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & X_{n_s} \end{pmatrix}$$

is in the interior of the cone $K = \mathbb{S}_+^{s_1} \times \dots \times \mathbb{S}_+^{s_{n_s}}$, if $\lambda_{\min}(X_j) > 0$ for $j = 1, \dots, n_s$.

It is well known that for linear programming problems strong duality $\hat{f}_p = \hat{f}_d$ holds without any constraint qualifications. General conic programs satisfy only the weak duality condition $\hat{f}_d \leq \hat{f}_p$. Strong duality requires additional constraint qualifications. For example *Slater's constraint qualifications* guarantee strong duality (see [23, 35]).

Theorem 2.1 (Strong Duality Theorem).

- If the primal problem is strictly feasible (i.e. there exists a primal feasible point x in the interior of K) and $\hat{f}_p$ is finite, then $\hat{f}_p = \hat{f}_d$ and the dual supremum is attained.
- If the dual problem is strictly feasible (i.e. there exists some y such that $z = c - (A)^T y$ is in the interior of K^*) and $\hat{f}_d$ is finite, then $\hat{f}_d = \hat{f}_p$, and the primal infimum is attained.

In general, one of the problems (CP) and (DCP) may have optimal solutions and its dual is infeasible, or the duality gap may be positive at optimality.

Duality theory is central to the study of optimization. Firstly, algorithms are frequently based on duality (like primal-dual interior point methods), secondly, they enable one to check whether or not a given feasible point is optimal, and thirdly, it allows one to compute verified results efficiently.

A knowledge of interval arithmetic is not required. Intervals are only used to specify error bounds. An interval, as usual, is defined by a lower and an upper bound. An interval vector or an interval matrix is defined as a set of vectors or matrices that vary between a lower and an upper vector or matrix, respectively. In other words, these are quantities with interval components. In INTLAB these interval quantities can be initialized with `infsup`. Equivalently, these quantities can be defined by a midpoint-radius representation. This representation is in INTLAB initialized by `midrad`.

3 Getting Started with VSDP

“...the routine can produce a computed result that is nonsensical and the application proceeds as if the results were correct. ... What should you do? The first defense is to adopt a sceptical attitude toward numerical results until you can verify them by independent methods.”

When Good Computers Make Bad Calculations, R. Meyer, President NAG, 2001,
www.nag.com/local/whitepapers/index.asp

This chapter provides a step-by-step introduction to VSDP. Basically, VSDP consists of four main functions: `mysdps`, `vsdpflow`, `vsdpup`, and `vsdpinfeas`. The function `mysdps` represents a simple interface to the conic solvers mentioned above. The functions `vsdpflow` and `vsdpup` compute rigorous enclosures of ϵ -optimal solutions as well as lower and upper bounds of the primal and dual optimal value, respectively. The function `vsdpinfeas` establishes a rigorous certificate of primal or dual infeasibility.

The VSDP data format coincides with the SeDuMi format. Semidefinite programs that are defined in the format of one of the supported solvers can be read by VSDP. More information about the corresponding transformation functions is given in the function reference in Appendix A.

Unless specified otherwise the approximate solvers are used in the following versions:

SDPT3 4.0, SeDuMi 1.3, SDPA 7.3.6, CSDP 6.0.1 – 2,
 LP-SOLVE 5.5.2.0 and LINPROG 0.9 – 0.

3.1 Linear Programming

In this section we describe how linear programming problems can be solved with VSDP. In particular, two linear programming examples are considered in detail. Each conic problem is fully described in VSDP by the four variables (A, b, c, K) . The first two quantities represent the affine constraints $Ax = b$. The third is the primal objective vector c , and the last describes the underlying cone. The cone K is a structure with four fields: $K.f$, $K.l$, $K.q$, $K.s$. The field $K.f$ stores the number of free variables n_f , the field $K.l$ stores the number of nonnegative variables n_l , the field $K.q$ stores the dimensions $q_1, \dots, q_{n_q}$ of the second order cones, and similarly $K.s$ stores the dimensions $s_1, \dots, s_{n_s}$ of the semidefinite cones. If a component of K is empty, then it is assumed that the corresponding cone do not occur.

Consider the linear programming problem

$$\begin{aligned} \min \quad & 2x_2 + 3x_4 + 5x_5 \\ \text{s.t.} \quad & \begin{pmatrix} -1 & 2 & 0 & 1 & 1 \\ 0 & 0 & -1 & 0 & 2 \end{pmatrix} x = \begin{pmatrix} 2 \\ 3 \end{pmatrix}, \\ & x \geq 0, \end{aligned} \tag{3.1}$$

with its corresponding dual problem

$$\begin{aligned} \max \quad & 2y_1 + 3y_2 \\ \text{s.t.} \quad & z := \begin{pmatrix} 0 \\ 2 \\ 0 \\ 3 \\ 5 \end{pmatrix} - \begin{pmatrix} -1 & 0 \\ 2 & 0 \\ 0 & -1 \\ 1 & 0 \\ 1 & 2 \end{pmatrix} y \geq 0. \end{aligned}$$

The unique exact optimal solution is given by

$$x^* = (0, 0.25, 0, 0, 1.5)^T, \quad y^* = (1, 2)^T \quad \text{with} \quad \hat{f}_p = \hat{f}_d = 12.$$

The input data of the problem in VSDP are

```
A = [-1 2 0 1 1; 0 0 -1 0 2];
b = [2 3]';
c = [0 2 0 3 5]';
K.1 = 5;
```

By using `vsdpinit` the approximate conic solver can be set globally for all VSDP functions. Here we choose the solver SDPT3:

```
vsdpinit('sdpt3');
```

If we call the `mysdps`-function for problem (3.1)

```
[objt xt yt zt info] = mysdps(A,b,c,K)
```

this problem will be solved approximately, yielding the output

```
objt =
    8.000000025993716    7.999999987362062
xt =
    0.000000009232368
    0.250000001445243
    0.000000004090490
    0.000000004292289
    1.500000002045272
yt =
    0.999999987362067
    2.000000004212642
zt =
    0.999999987432097
    0.000000025345895
    2.000000004282672
    2.000000012707963
    0.000000004282678
info =
    0
```

The returned variables have the following meaning: `objt` stores the approximate primal and dual optimal value, `xt` represents the approximate primal solution in vectorized form, and `(yt,zt)` is the dual solution pair. The last returned variable `info` gives information on the success of the conic solver:

- `info=0`: successful termination,
- `info=1`: the problem might be primal infeasible,

- `info=2`: the problem might be dual infeasible,
- `info=3`: the problem might be primal and dual infeasible.

With the approximate solution, a verified lower bound of the primal optimal value can be computed by the function `vsdpflow`:

```
[fL y dl] = vsdpflow(A,b,c,K,xt,yt,z)
fL =
    7.999999987362060
intval y =
    0.99999998736206
    2.00000000421264
dl =
    0.999999987362067
    0.000000025275865
    2.000000004212642
    2.000000012637933
    0.000000004212648
```

The output consists of (i) a verified lower bound for the optimal value stored in `fL`, (ii) a rigorous interval enclosure of a dual feasible solution `y`, and (iii) a componentwise lower bound of $z = c - A^T y$ stored in `dl`. Since `dl` is positive, the dual problem is strictly feasible, and the rigorous interval vector `y` contains a dual interior solution. Here only some significant digits of this interval vector are displayed. The upper and lower bounds of the interval `y` can be obtained by using the `sup`- and `inf`-routines of INTLAB. For more information about the `intval` data type see [29].

Next we compute an upper bound for the optimal value using the `vsdpup`-function:

```
[fU x lb] = vsdpup(A,b,c,K,xt,yt,z)
fU =
    8.000000025997951
intval x =
    0.00000000923236
    0.25000000144742
    0.00000000409049
    0.00000000429228
    1.50000000204525
lb =
    0.000000009232368
    0.250000001447414
    0.000000004090490
    0.000000004292289
    1.500000002045242
```

The returned variables have a similar meaning to those of `vsdpflow`: `fU` is a verified upper bound for the optimal value, `x` is a rigorous interval enclosure of a primal feasible solution, and `lb` is a componentwise lower bound for `x`. As for the interval vector `y` also for the interval vector `x` only some significant digits are displayed. Since `lb` is a positive vector, hence contained in the positive orthant, the primal problem is strictly feasible.

Summarizing, we have obtained a primal dual interval solution pair that contains a primal and dual strictly feasible ϵ -optimal solution, where

$$\epsilon = \frac{2(fU - fL)}{|fU| + |fL|} = 4.83e-9.$$

How a linear program with free variables can be solved with VSDP is demonstrated by the following example with one free variable x_3 :

$$\begin{aligned} \min \quad & x_1 + x_2 - 0.5x_3 \\ \text{s.t.} \quad & x_1 - x_2 + 2x_3 = 0.5, \\ & x_1 + x_2 - x_3 = 1, \\ & x_1 \geq 0, \quad x_2 \geq 0 \end{aligned}$$

The optimal solution pair of this problem is:

$$x^* = \left(\frac{5}{6}, 0, -\frac{1}{6}\right)^T, \quad y^* = \left(\frac{1}{6}, \frac{5}{6}\right)^T \quad \text{with} \quad \hat{f}_p = \hat{f}_d = \frac{11}{12}.$$

When entering a problem the order of the variables is important: Firstly free variables, secondly nonnegative variables, thirdly second order cone variables, and last semidefinite variables. This order must be maintained in the matrix **A** as well as in the primal objective **c**. In the given example, the free variable is x_3 , the nonnegative variables are x_1, x_2 . Second order cone variables and semidefinite variables are not present. Therefore, the problem data are

```
K.f = 1; % number of free variables
K.l = 2; % number of nonnegative variables
A = [2 1 -1; % first column corresponds to free var. x3
     -1 1 1]; % second and third to bounded x1, x2
c = [-0.5 1 1]'; % the same applies to c
b = [0.5 1]';
```

Rigorous bounds for the optimal value can be obtained with:

```
[objt xt yt zt info] = mysdps(A,b,c,K);
fL = vsdpLOW(A,b,c,K,xt,yt,zt)
fU = vsdpUP(A,b,c,K,xt,yt,zt)
fL =
    0.916666666222149
fU =
    0.916666666922786
```

3.2 Second Order Cone Programming

This section explains how to work with second order cone problems. Due to (CP) the second order cone problem in standard primal form is given by

$$\begin{aligned} \hat{f}_p &:= \min \langle c^f, x^f \rangle + \sum_{i=1}^{n_q} \langle c_i^q, x_i^q \rangle \\ \text{s.t.} \quad & A^f x^f + \sum_{i=1}^{n_q} A_i^q x_i^q = b, \\ & x^f \in \mathbb{R}^{n_f}, x_i^q \in \mathbb{L}^{q_i}, i = 1, \dots, n_q. \end{aligned} \tag{SOCP}$$

The corresponding dual maximization problem is

$$\begin{aligned} \hat{f}_d &:= \max b^T y \\ \text{s.t.} \quad & (A_i^q)^T y + z_i^q = c_i^q, \quad z_i^q \in \mathbb{L}^{q_i}, \quad i = 1, \dots, n_q, \\ & (A^f)^T y + z^f = c^f, \quad z^f \in \mathbb{R}^{n_f}, \\ & z^f = 0. \end{aligned} \tag{DSOCP}$$

Let us consider the total least squares problem taken from [8]:

$$\begin{aligned} \max & -y_1 - y_2 \\ \text{s.t.} \quad & y_1 \geq \| (q - Py_{3:5}) \|_2, \\ & y_2 \geq \left\| \begin{pmatrix} 1 \\ y_{3:5} \end{pmatrix} \right\|_2, \end{aligned} \quad (3.2)$$

where

$$P = \begin{pmatrix} 3 & 1 & 4 \\ 0 & 1 & 1 \\ -2 & 5 & 3 \\ 1 & 4 & 5 \end{pmatrix}, \quad q = \begin{pmatrix} 0 \\ 2 \\ 1 \\ 3 \end{pmatrix}, \quad y \in \mathbb{R}^5.$$

We want to transform this problem to the dual form (DSOCP). The two inequalities can be written in the form

$$\begin{pmatrix} y_1 \\ q - Py_{3:5} \end{pmatrix} \in \mathbb{L}^5 \quad \text{and} \quad \begin{pmatrix} y_2 \\ 1 \\ y_{3:5} \end{pmatrix} \in \mathbb{L}^5.$$

Since

$$\begin{pmatrix} y_1 \\ q - Py_{3:5} \end{pmatrix} = \underbrace{\begin{pmatrix} 0 \\ 0 \\ 2 \\ 1 \\ 3 \end{pmatrix}}_{=c_1^q} - \underbrace{\begin{pmatrix} -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3 & 1 & 4 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & -2 & 5 & 3 \\ 0 & 0 & 1 & 4 & 5 \end{pmatrix}}_{=(A_1^q)^T} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \end{pmatrix},$$

and

$$\begin{pmatrix} y_2 \\ 1 \\ y_{3:5} \end{pmatrix} = \underbrace{\begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}}_{=c_2^q} - \underbrace{\begin{pmatrix} 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 \end{pmatrix}}_{=(A_2^q)^T} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \end{pmatrix},$$

our dual problem (DSOCP) takes the form

$$\begin{aligned} \max \quad & \underbrace{\begin{pmatrix} -1 & -1 & 0 & 0 & 0 \end{pmatrix}}_{=b^T} y \\ \text{s.t.} \quad & z = \underbrace{\begin{pmatrix} 0 \\ 0 \\ 2 \\ 1 \\ 3 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}}_{=c} - \underbrace{\begin{pmatrix} -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3 & 1 & 4 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & -2 & 5 & 3 \\ 0 & 0 & 1 & 4 & 5 \\ 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 \end{pmatrix}}_{=A^T} y \in K^*, \end{aligned} \quad (3.3)$$

where $K^* = \mathbb{L}^5 \times \mathbb{L}^5$.

We want to solve this problem with SeDuMi and enter the problem data of the primal problem.

```

vsdpinit('sedumi');
c = [0 0 2 1 3 0 1 0 0 0]';
A = [ -1  0  0  0  0  0  0  0  0  0;
      0  0  0  0  0 -1  0  0  0  0;
      0  3  0 -2  1  0  0 -1  0  0;
      0  1  1  5  4  0  0  0 -1  0;
      0  4  1  3  5  0  0  0  0 -1];
b = [-1 -1 0 0 0]';

```

Apart from the data (A,b,c), the vector $q=[5;5]$ of the second order cone block sizes must be forwarded to the structure K:

```
K = struct('q',[5;5]);
```

Now we compute approximate solutions by using `mysdps` and then verified error bounds by using `vsdpflow` and `vsdpup`:

```

[objt xt yt zt info] = mysdps(A,b,c,K);
[fL y dl] = vsdpflow(A,b,c,K,xt,yt,zt);
[fU x lb] = vsdpup(A,b,c,K,xt,yt,zt);

```

The approximate primal and dual optimal values and the rigorous lower and upper bounds are

```

objt =
-3.332908595087726  -3.332908596319013
fL =
-3.332908600178669
fU =
-3.332908594014274

```

The quantities x and y are not displayed here. The two output vectors lb and dl provide rigorous lower bounds for the spectral values of these of these variables. Since both vectors are positive

```

lb =
1.0e-009 *
0.000637934149950
0.224786633751251
dl =
1.0e-011 *
0.275468536869994
0.181410442223751

```

the primal and the dual set of feasible solutions have a non empty relative interior. Thus Slater's constraint qualifications (see Theorem 2.1) imply that the duality gap is zero. The intervals x and y contain interior feasible ϵ -optimal solutions, where $\epsilon = \frac{2(fL-fU)}{|fL|+|fU|} = 1.85e-9$.

The conic program (CP) allows to mix constraints of different types. Let us, for instance, add the linear inequality $\sum_{i=1}^5 y_i \leq 3.5$ to the previous dual problem. By using the standard form (CP), (DCP), and the condensed quantities (2.7) it follows that we must add to the matrix A the column consisting of ones and to c the value 3.5. We extend the input data as follows:

```

A = [[1 1 1 1 1]' A];
c = [3.5; c];
K.1 = 1;
[objt xt yt zt info] = mysdps(A,b,c,K);
[fL y dl] = vsdpflow(A,b,c,K,xt,yt,zt);

```

```
[fU x lb] = vsdpup(A,b,c,K,xt,yt,zt);
```

Then we obtain

```
[fL fU]
ans =
-3.572766612944500 -3.572766405153391
```

3.3 Semidefinite Programming

Let the SDP program be given in the standard form (CP)

$$\min \sum_{j=1}^{n_s} \langle C_j^s, X_j^s \rangle \quad \text{s.t.} \quad \sum_{j=1}^{n_s} \langle A_{i,j}^s, X_j^s \rangle = b_i, \quad i = 1, \dots, m, \\ X_j^s \in \mathbb{S}_+^{s_j}, \quad j = 1, \dots, n_s. \quad (3.4)$$

Its dual (DCP) is

$$\max b^T y \quad \text{s.t.} \quad Z_j^s = C_j^s - \sum_{i=1}^m y_i A_{i,j}^s \in \mathbb{S}_+^{s_j}, \quad j = 1, \dots, n_s. \quad (3.5)$$

The matrices $A_{i,j}^s, C_j^s, X_j^s$ are assumed to be symmetric $s_j \times s_j$ matrices. We store this problem in our condensed format (2.6), (2.7), and (2.8).

We consider the example (see [6]):

$$\min \sum_{j=1}^3 \langle C_j^s, X_j^s \rangle \quad \text{s.t.} \quad \sum_{j=1}^3 \langle A_{i,j}^s, X_j^s \rangle = b_i, \quad i = 1, 2, \\ X_1^s \in \mathbb{S}_+^2, \quad X_2^s \in \mathbb{S}_+^3, \quad X_3^s \in \mathbb{S}_+^2,$$

where

$$A_{1,1}^s = \begin{pmatrix} 3 & 1 \\ 1 & 3 \end{pmatrix}, \quad A_{1,2}^s = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad A_{1,3}^s = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, \\ A_{2,1}^s = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}, \quad A_{2,2}^s = \begin{pmatrix} 3 & 0 & 1 \\ 0 & 4 & 0 \\ 1 & 0 & 5 \end{pmatrix}, \quad A_{2,3}^s = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}, \\ C_1^s = \begin{pmatrix} -2 & -1 \\ -1 & -2 \end{pmatrix}, \quad C_2^s = \begin{pmatrix} -3 & 0 & -1 \\ 0 & -2 & 0 \\ -1 & 0 & -3 \end{pmatrix}, \quad C_3^s = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}, \\ b = \begin{pmatrix} 1 \\ 2 \end{pmatrix}.$$

In the condensed format the corresponding coefficient matrix and the primal objective are

$$A = \begin{pmatrix} 3 & 1 & 1 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3 & 0 & 1 & 0 & 4 & 0 & 1 & 0 & 5 & 0 & 0 & 0 & 1 \end{pmatrix}, \\ c = (-2 \quad -1 \quad -1 \quad -2 \quad -3 \quad 0 \quad -1 \quad 0 \quad -2 \quad 0 \quad -1 \quad 0 \quad -3 \quad 0 \quad 0 \quad 0 \quad 0)^T.$$

We enter the problem data

```
A = [3 1 1 3 0 0 0 0 0 0 0 0 0 1 0 0 0;...
      0 0 0 0 3 0 1 0 4 0 1 0 5 0 0 0 1];
c = [-2 -1 -1 -2 -3 0 -1 0 -2 0 -1 0 -3 0 0 0 0]';
b = [1;2];
```

We define the structure K for the PSD-cone

```
K = struct('s',[2;3;2]);
```

And we call mysdps

```
% we select the SDP-solver SDPT3
vsdpinit('sdpt3');
[objt xt yt zt info] = mysdps(A,b,c,K);
objt =
-2.749999966056180 -2.750000014595577
% other quantities are not displayed for brevity
```

We get verified error bounds by calling vsdpflow and vsdpup

```
[fL y dL] = vsdpflow(A,b,c,K,xt,yt,zt)
fL =
-2.750000014595577
intval y =
-0.75000000723431
-1.00000000368063
dL =
0.0000000000000001
0.0000000000000000
0.750000007234317

[fU x lB] = vsdpup(A,b,c,K,xt,yt,zt)
fU =
-2.749999966061938
intval x =
0.12500000226309
0.12499998868775
% only the first two componensts displayed for brevity
lB =
1.0e-08 *
0.000000040245585
0.00000000587181
0.678710305518414
```

The components $lB(j)$ are lower bounds of the smallest eigenvalues $\lambda_{\min}([X_j^s])$ for $j = 1, 2, 3$. Hence $lB > 0$ proves that all real matrices X_j^s , that are contained in the interval matrix $[X_j^s]$ are in the interior of the cone $\mathbb{S}_+^2 \times \mathbb{S}_+^3 \times \mathbb{S}_+^2$, where the interval matrices $[X_j^s]$ are obtained by applying the `mat`-operator to `intval x`. Analogously, $dL(j)$ are lower bounds for the smallest eigenvalues of the dual interval matrices $[Z_j^s]$ that correspond to the dual solution y . Since for this example, both dL and lB are positive, strict feasibility is proved for the primal and the dual problem, and strong duality holds valid.

Now we consider the following example (see [11]):

$$\begin{aligned} \min \langle C_1, X \rangle \quad \text{s.t.} \quad & \langle A_{1,1}, X \rangle = 1, & \langle A_{2,1}, X \rangle = 2\delta \\ & \langle A_{3,1}, X \rangle = 0, & \langle A_{4,1}, X \rangle = 0, \\ & X \in \mathbb{S}_+^3, \end{aligned} \quad (3.6)$$

where

$$\begin{aligned}
C_1 &= \begin{pmatrix} 0 & \frac{1}{2} & 0 \\ \frac{1}{2} & \delta & 0 \\ 0 & 0 & \delta \end{pmatrix}, & b &= \begin{pmatrix} 1 \\ 2\delta \\ 0 \\ 0 \end{pmatrix}, \\
A_{1,1} &= \begin{pmatrix} 0 & -\frac{1}{2} & 0 \\ -\frac{1}{2} & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, & A_{2,1} &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \\
A_{3,1} &= \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}, & A_{4,1} &= \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}.
\end{aligned}$$

It is easy to prove that for

$\delta > 0$: the problem is feasible with $\hat{f}_p = \hat{f}_d = -0.5$ (zero duality gap),

$\delta = 0$: the problem is feasible but ill-posed with nonzero duality gap,

$\delta < 0$: the problem is infeasible.

For $\delta > 0$ the primal optimal solution of the problem is given by the matrix

$$X^* = \begin{pmatrix} 2\delta & -1 & 0 \\ -1 & \frac{1}{2\delta} & 0 \\ 0 & 0 & 0 \end{pmatrix}. \quad (3.7)$$

The corresponding dual optimal vector is $y^* = (0 \quad -1/(4\delta) \quad 0 \quad 0)^T$. We choose $\delta = 10^{-4}$ and enter the problem.

```

% define delta parameter
d = 1e-4;
% define the constraint matrices
A1 = [0  -.5  0;
      -.5  0  0;
      0   0  0];
A2 = [1  0  0 ;
      0  0  0;
      0  0  0];
A3 = [0  0  1;
      0  0  0;
      1  0  0];
A4 = [0  0  0;
      0  0  1;
      0  1  0];
% enter the right-hand side b of Ax=b
b = [1; 2*d; 0; 0];
% define the primal objective C
C = [0  .5  0;
      .5  d  0;
      0   0  d];
% define the structure K
K = struct('s',3);
% vectorize the matrices Ai and C
A = [A1(:),A2(:),A3(:),A4(:)]';
c = C(:);

```

```

% select the SDP-solver SDPT3
vsdpinit('sdpt3');
% call mysdps
[objt,xt,yt,zt,info] = mysdps(A,b,c,K);

```

The SDPT3-solver provides the following results:

```

objt, xt, yt, info
objt =
-4.999999947920115e-001    -5.000000061296884e-001
xt =
 1.999999999974619e-004
-1.000000000011298e+000
 4.478536229147952e-030
-1.000000000011298e+000
 5.000000014419667e+003
 1.593322005301818e-026
 4.478536229147952e-030
 1.593322005301818e-026
 3.777320067161448e-005
yt =
 3.313820644761502e-005
-2.500165721680681e+003
 5.507005918385401e-016
 1.101156370546606e-019
info =
 0
% zt: not displayed for brevity

```

The value of the return parameter `info` confirms successful termination of the solver. The first eight decimal digits of the primal and dual optimal values are correct, since $\hat{f}_p = \hat{f}_d = -0.5$, all components of the approximate solutions `xt` and `yt` have at least five correct decimal digits. Nevertheless, successful termination reported by a solver gives no guarantee on the quality of the computed solution. For instance, if we apply SeDuMi to the same problem we obtain:

```

vsdpinit('sedumi');
[objt,xt,yt,zt,info] = mysdps(A,b,c,K)
objt =
-4.999996777820868e-001    -4.999988671612469e-001
xt =
 2.000001356937796e-004
-9.999999999321320e-001
 2.028164302578316e-018
-9.999999999321317e-001
 5.000001113166548e+003
-1.024571528811840e-014
 2.028164302578319e-018
-1.024571528811840e-014
 2.108333901738334e-003
yt =
 1.296216294214629e-003
-2.506475417277308e+003
-6.075900257956433e-016
-1.216853433177092e-019
% zt: hidden for brevity
info =

```

0

SeDuMi terminates without any warning, but some results are poor. For instance, weak duality is not satisfied since the approximate primal optimal value is smaller than the dual one. In other words, the algorithm is not backward stable for this example. The CSDP-solver gives similar results:

```

vsdpinit('csdp')
[objt,xt,yt,zt,info] = mysdps(A,b,c,K)
objt =
-4.999954437796610e-001    -4.999827919759200e-001
xt =
  2.0000000000000001e-004
 -1.0000000000000001e+000
           0
 -1.0000000000000001e+000
  5.000022781101705e+003
           0
           0
           0
  2.278110169488024e-002
yt =
  3.441604678409375e-005
 -2.500086040113521e+003
           0
           0
info =
  0

```

A good deal worse are the results that can be derived with older versions of these solvers, including SDPT3 and SDPA.

Reliable results can be obtained by the functions `vsdpflow` and `vsdpup`. Firstly, we consider `vsdpflow` and the approximate solver SDPT3.

```

vsdpinit('sdpt3');
[objt,xt,yt,zt,info] = mysdps(A,b,c,K);
[fL, y, dl] = vsdpflow(A,b,c,K,xt,yt,zt)
fL =
-5.000000060522118e-01
intval y =
  3.247706111272894e-05
 -2.500162415566622e+03
  5.566885486605820e-16
  1.113135550464241e-19
dl =
  8.882597348622539e-16

```

the vector `y` is a rigorous interior dual ϵ -optimal solution where we shall see that $\epsilon = 2.27e-8$. The positivity of `dl` verifies that `y` contains a dual strictly feasible solution. In particular, strong duality holds. By using SeDuMi similar rigorous results are obtained. But for the SDPA-solver we get

```

vsdpinit('sdpa');
[objt,xt,yt,zt,info] = mysdps(A,b,c,K);
[fL, y, dl] = vsdpflow(A,b,c,K,xt,yt,zt)
fL =

```

```

-Inf
y =
NaN
dl =
NaN

```

Thus an infinite lower bound for the primal optimal value is obtained and dual feasibility is not verified. The rigorous lower bound strongly depends on the computed approximate solution and therefore on the used approximate conic solver.

Similarly, a verified upper bound and a rigorous enclosure of a primal ϵ -optimal solution can be computed by using the `vsdpup`-function together with SDPT3:

```

vsdpinit('sdpt3');
[objt,xt,yt,zt,info] = mysdps(A,b,c,K);
[fU, x, lb] = vsdpup(A,b,c,K,xt,yt,zt)
fU =
-4.999999947944040e-01
intval x =
2.000000000000000e-04
-1.000000000000000e+00
0.000000000000000e+00
-1.000000000000000e+00
5.000000014584274e+03
0.000000000000000e+00
0.000000000000000e+00
0.000000000000000e+00
3.747168382082697e-05
lb =
1.255214736404359e-13

```

The output `fU` is close to the dual optimal value $\hat{f}_d = -0.5$. The interval vector `x` contains a primal strictly feasible solution, see (3.7), and the variable `lb` is a lower bound for the smallest eigenvalue of `x`. Because `lb` is positive, Slater's condition is fulfilled and strong duality is verified once more.

Summarizing, by using SDPT3 for the considered example with parameter $\delta = 10^{-4}$, VSDP verified strong duality with rigorous bounds for the optimal value

$$-0.500000007 \leq \hat{f}_p = \hat{f}_d \leq -0.499999994.$$

The rigorous upper and lower error bounds of the optimal value show only modest overestimation. Strictly primal and dual feasible solutions are obtained. Strong duality is verified. Moreover, we have seen that the quality of the rigorous results depends strongly on the quality of the computed approximations.

3.4 A Priori Upper Bounds for Optimal Solutions

In many practical applications the order of the magnitude of a primal or dual optimal solution is known a priori. This is the case in many combinatorial optimization problems, or, for instance, in truss topology design where the design variables such as bar volumes can be roughly

bounded. If such bounds are available they can speed up the computation of guaranteed error bounds for the optimal value substantially, see ¹.

For linear programming problems the upper bound for the variable x^l is a vector $\bar{x}$ such that $x^l \leq \bar{x}$. For second order cone programming the upper bounds for block variables x_i^q can be entered as a vector of upper bounds $\bar{\lambda}_i$ of the largest spectral values $\lambda_{\max}(x_i^q) = (x_i^q)_1 + \|(x_i^q)\|_2$, $i = 1, \dots, n_q$. Similarly, for semidefinite programs upper bounds for the primal variables X_j^s can be entered as a vector of upper bounds of the largest eigenvalues $\lambda_{\max}(X_j^s)$, $j = 1, \dots, n_s$. An upper bound $\bar{y}$ for the dual optimal solution y is a vector which is componentwise larger then $|y|$. Analogously, for conic programs with free variables the upper bound can be entered as a vector $\bar{x}$ such that $|x^f| \leq \bar{x}$.

As an example, we consider the previous SDP problem (3.6) with an upper bound $\mathbf{xu} = 10^5$ for $\lambda_{\max}(X)$.

```
vsdpinit('sedumi');
[objt,xt,yt,zt,info] = mysdps(A,b,c,K);
xu = 1e5;
fL = vsdpflow(A,b,C,K,xt,yt,zt,xu)
fL =
-5.000382413476973e-001
```

Now, we suppose the existence of dual upper bounds

```
yu = 1e5 * [1 1 1 1]';
fU = vsdpup(A,b,C,K,xt,yt,zt,yu)
fU =
-4.999793324037286e-001
```

yielding also a reasonable bound.

3.5 Rigorous Certificates of Infeasibility

The functions **vsdpflow** and **vsdpup** prove strict feasibility and compute rigorous error bounds. For the verification of infeasibility the function **vsdpinfeas** can be applied. In this section we show how to use this function.

We consider the second order cone problem ([22], Example 1.7.2)

$$\begin{aligned} \min \quad & 0^T x \\ \text{s.t.} \quad & \begin{pmatrix} 1 & 0 & 0.5 \\ 0 & 1 & 0 \end{pmatrix} x = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \\ & x \in \mathbb{L}^3 \end{aligned}$$

with its dual problem

$$\begin{aligned} \max \quad & -y_2 \\ \text{s.t.} \quad & \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} - \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0.5 & 0 \end{pmatrix} y \in \mathbb{L}^3. \end{aligned}$$

Both, the primal and the dual problem, are infeasible. We can easily prove this fact by assuming that there exists a primal feasible point x . This point has to satisfy $x_3 = -2x_1$, and therefore $x_1 \geq \sqrt{x_2^2 + (-2x_1)^2}$. From the second equality constraint we get $x_2 = 1$ yielding the contradiction

¹ references

$x_1 \geq \sqrt{1+4x_1^2}$. Thus, the primal problem has no feasible solution. A certificate of infeasibility of the primal problem is given by the dual unbounded ray $y = \alpha(-2, 1)^T$.

The input data are

```
A = [1 0 .5;
      0 1 0];
b = [0;1];
c = [0;0;0];
K = struct('q',3);
```

Using the approximate solver SDPT3 we obtain a rigorous certificate of infeasibility with the routine `vsdpinfeas`:

```
vsdpinit('sdpt3');
[objt,xt,yt,zt,info] = mysdps(A,b,c,K);
[isinf x y] = vsdpinfeas(A,b,c,K,'p',xt,yt,zt);
```

Apart from the problem data parameters (A,b,c,K) and the approximations (xt,yt,zt) an additional parameter is required, namely

- 'p' if primal infeasibility should be verified,
- 'd' if dual infeasibility should be verified.

The function `vsdpinfeas` tries to verify the existence of an improving ray using the approximations that are computed by the approximate solver. If the return value `isinf` is equal to one then `vsdpinfeas` has verified primal or dual infeasibility depending on the fifth parameter `p` or `d`, respectively. If no infeasibility certificates could be found `vsdpinfeas` returns zero.

For the considered example `vsdpinfeas` returns

```
isinf =
    1
x =
    NaN
y =
   -1.154887880531688e+00
    1.000000000000000e+00
```

Hence primal infeasibility is verified. The return parameter `y` provides a rigorous dual improving ray. The return parameter `x` must be empty, since we did not check dual infeasibility.

Now we try to solve the problem (3.6) for $\delta = -10^4 < 0$. We know that in this case the problem is primal and dual infeasible.

```
d = -1e-4;
C = [ 0  0.5  0;
      0.5  d  0;
      0   0  d];
c = C(:);
b = [1; 2*d; 0; 0];
vsdpinit('sedumi');
[objt,xt,yt,zt,info] = mysdps(A,b,c,K);
```

SeDuMi terminates the computation with the termination code `info=3` and gives the warnings

...

```

Dual infeasible, primal improving direction found.
...
Primal infeasible, dual improving direction found.
...

```

If we apply the routines `vsdlow` and `vsdpup`

```

[fL, y, dL] = vsdpow(A,b,c,K,xt,yt,z);
[fU, x, lb] = vsdpup(A,b,c,K,xt,yt,z);

```

then the bounds `fL`, `fU` are infinite, as expected. By applying `vsdpinfeas` we obtain

```

[isinf, x, y] = vsdpinfeas(A,b,c,K,'p',xt,yt,z)
isinf =
    0
x =
    NaN
y =
    NaN

```

Since `isinf=0`, primal infeasibility could not be verified, and therefore the certificates `x`, `y` are set to `NaN`. The reason is that all dual improving rays y must satisfy

$$-\sum_{i=1}^4 y_i A_{i,1} = \begin{pmatrix} -y_2 & y_1/2 & -y_3 \\ y_1/2 & 0 & -y_4 \\ -y_3 & -y_4 & 0 \end{pmatrix} \in \mathbb{S}_+^3. \quad (3.8)$$

This is only possible for $y_1 = y_3 = y_4 = 0$. Hence, for each improving ray the matrix (3.8) has a zero eigenvalue. In VSDP we verify positive semidefiniteness by computing enclosures of the eigenvalues. If all enclosures are non-negative positive semidefiniteness is proved. If one eigenvalue is zero then, except in special cases, the corresponding enclosure has a negative component implying that positive semidefiniteness cannot be proved and primal infeasibility cannot be verified.

Now we try to verify dual infeasibility by using `vsdpinfeas` with the parameter `'d'`.

```

[isinf, x, y] = vsdpinfeas(A,b,c,K,'d',xt,yt,z)
isinf =
    1
intval x =
    0.000000000000000e+00
    0.000000000000000e+00
    0.000000000000000e+00
    0.000000000000000e+00
    5.338038498740059e+03
    0.000000000000000e+00
    0.000000000000000e+00
    0.000000000000000e+00
    4.661961482647223e+03
y =
    NaN

```

Dual infeasibility can be proved. An easy calculation shows that the primal improving ray must satisfy

$$X = \begin{pmatrix} 0 & 0 & 0 \\ 0 & x_{22} & 0 \\ 0 & 0 & x_{33} \end{pmatrix} \in \mathbb{S}_+^3.$$

Again the improving ray has an eigenvalue that is equal to zero. Nevertheless, in this special case VSDP is able to verify dual infeasibility due to an included mechanism that will be explained in another paper.

3.6 Handling Free Variables

Free variables occur often in practice. Handling free variables in interior point algorithms is a pending issue (see for example [3, 4, 21]). Frequently users convert a problem with free variables into one with restricted variables by representing the free variables as a difference of two nonnegative variables. This approach increases the problem size and introduces ill-posedness, which may lead to numerical difficulties.

For an example we consider the test problem **nb_L1** from the DIMACS test library [27]. The problem originates from side lobe minimization in antenna engineering. This is a second order cone programming problem with 915 equality constraints, 793 SOCP blocks each of size 3, and 797 nonnegative variables. Moreover, the problem has two free variables that are described as the difference of four nonnegative variables. This problem can be loaded from the examples directory of VSDP.

```
vsdpinit('sdpt3');
load nb_L1.mat;
[objt xt yt zt info] = mysdps(A,b,c,K);
objt =
    -1.301227061044399e+01    -1.301227074897884e+01
```

SDPT3 solves the problem without warnings, although it is ill-posed according to Renegar's definition [28].

Now we try to get rigorous bounds using the approximation of SDPT3.

```
fL = vsdp_low(A,b,c,K,xt,yt,zt)
fL =
    -Inf

fU = vsdp_up(A,b,c,K,xt,yt,zt)
fU =
    -1.301227023586709e+01
```

These results reflect that the interior of the dual feasible solution set is empty. An ill-posed problem has the property that the distance to primal or dual infeasibility is zero. If as above the distance to dual infeasibility is zero, then there are sequences of dual infeasible problems with input data converging to the input data of the original problem. Each problem of the sequence is dual infeasible and thus has the dual optimal solution $-\infty$. Hence, the result $-\infty$ of `vsdp_low` is exactly the limit of the optimal values of the dual infeasible problems and reflects the fact that the distance to dual infeasibility is zero. This demonstrates that the infinite bound computed by VSDP is sharp, when viewed as the limit of a sequence of infeasible problems. Similarly is the situation if the distance to primal infeasibility is zero.

If the free variables are not converted into restricted ones then the problem is well-posed and a rigorous finite lower bound can be computed.

```
load nb_L1free.mat;
[objt xt yt zt info] = mysdps(A,b,c,K);
objt =
```

-1.301227061378058e+01 -1.301227082231597e+01

By using the computed approximations we obtain the following rigorous bounds:

fL = vsdp_{low}(A,b,c,K,xt,yt,zt)
fL =
-1.301227082244206e+01

fU = vsdp_{up}(A,b,c,K,xt,yt,zt)
fU =
-1.301227061210429e+01

Therefore, without splitting the free variables, we get rigorous finite lower and upper bounds of the exact optimal value with an accuracy of about 8 decimal digits. Moreover, verified interior solutions are computed for both the primal and the dual problem, proving strong duality.

In Table 3.1 we display rigorous bounds for the optimal value of eight problems contained in the DIMACS test library that have free variables (see [4, 19]). We modified these problems reversing the substitution of the free variables. We have listed the results for the problems with free variables and for the same problems when representing the free variables as the difference of two nonnegative variables. The table contains the rigorous upper bounds fU , the rigorous lower bounds fL , and the computing times measured in seconds for the approximate solution t_s , the lower bound t_u , and the upper bound t_l , respectively. The table demonstrates the drastic improvement if free variables are not split.

Table 3.1: DIMACS test problems with free variables.

Problem	fU	fL	t_s	t_u	t_l
nb	-5.07030901e-02	-Inf	3.51	3.55	2.69
nbfree	-5.07030854e-02	-5.07030948e-02	2.38	0.16	0.06
nb_L1	-13.0122697	-Inf	4.67	3.53	4.53
nb_L1free	-13.0122706	-13.0122708	5.33	10.1	9.85
nb_L2	-1.52319900	-Inf	3.59	2.32	33.8
nb_L2free	-1.62897194	-1.62897198	3.30	0.14	0.09
nql30	Inf	-Inf	1.98	0.67	21.3
nql30free	Inf	-0.94604707	1.04	0.35	4.36
nql60	Inf	-Inf	5.26	1.93	110
nql60free	Inf	-0.93942524	3.28	1.57	15.1
nql180	Inf	-Inf	47.0	46.2	950
nql180free	Inf	-28.1554324	41.9	27.9	76.1
qssp30	-6.49666899	-Inf	1.05	21.7	17.5
qssp30free	-6.49667562	-6.49667575	0.97	1.84	0.63
qssp60	-6.56269779	-Inf	4.01	7.01	68.4
qssp60free	-6.56270378	-6.56270652	3.67	12.5	2.44
qssp180	-Inf	-Inf	52.5	14.5	642
qssp180free	-5.91218713	-6.63961091	52.3	115	15.3

Independent of the transformation of the free variables the primal problem of the **nql** instances are ill-posed. The weak error bound of optimal solution for the **qssp180** instance is due to the large number of equality constraints. A system with 130,141 equality constraints and 261,365 variables has to be solved rigorously. In the next version of VSDP the accuracy for such large problems will be improved.

4 Statistics of the Numerical Results

“If error analysis could be automated, the mysteries of floating-point arithmetic could remain hidden; the computer would append to every displayed numerical result a modest over-estimate of its uncertainty due to imprecise data and approximate arithmetic, or else explain what went wrong, and all at a cost not much higher than if no error analysis had been performed. So far, every attempt to achieve this ideal has been thwarted.”

William M. Kahan, The Regrettable Failure of Automated Error Analysis, 1989.

In this section, we present statistics for the numerical results obtained by VSDP for conic programming problems. The tests were performed using approximations computed by the four conic solvers: CSDP, SEDUMI, SDPT3, and SDPA. For second order cone programming problems only SEDUMI and SDPT3 were used. The solvers have been called with their default parameters. Almost all of the problems that could not be solved with a guaranteed accuracy about 10^{-7} are known to be ill-posed (cf. [26], [25]).

We measure the difference between two numbers by the frequently used quantity

$$\mu(a, b) := \frac{a - b}{\max\{1.0, (|a| + |b|)/2\}}. \quad (4.1)$$

Notice that we do not use the absolute value of $a - b$. Hence, a negative sign implies that $a < b$.

4.1 SDPLIB

In the following, we describe the numerical results for problems from the SDPLIB suite of Borchers [7]. In [9] it is shown that four problems are infeasible, and 32 problems are ill-posed.

VSDP could compute rigorous bounds of the optimal values for all feasible well-posed problems and verify the existence of strictly primal and dual feasible solutions. Hence strong duality is proved. For the 32 ill-posed problems VSDP has computed the upper bound $fU = \infty$, which reflects the fact that the distance to the next primal infeasible problem is zero. Detailed numerical results can be found in the tables in Appendix B. Table 6.1 displays for the SDPLIB the rigorous upper bound fU , the rigorous lower bound fL , and the rigorous error bound $\mu(fU, fL)$. We have set $\mu(fU, fL) = NaN$ if the upper or the lower bound is infinite. Table 6.1 also contains computational times in seconds, where t_s is the time for computing the approximations, and t_u and t_l are the times for computing the upper and the lower bounds, respectively.

Some major characteristics of our numerical results for the SDPLIB are summarized in the Table 4.1. It displays in the first row the median $\text{med } \mu(\mathbf{fU}, \mathbf{fL})$ of the computed error bounds, in the second row the largest error bound $\max \mu(\mathbf{fU}, \mathbf{fL})$, and in the third row the minimal error bound

$\min \mu(\mathbf{fU}, \mathbf{fL})$. For this statistic only the well-posed problems has been taken into account. In the two last rows the medians of time ratios t_u/t_s and t_l/t_s are displayed.

Table 4.1: Statistic of numerical results for the SDPLIB.

	SDPT3	SEDUMI	SDPA	CSDP
med $\mu(\mathbf{fU}, \mathbf{fL})$	1.67e-8	4.63e-7	5.36e-8	1.02e-8
max $\mu(\mathbf{fU}, \mathbf{fL})$	7.41e-4	1.11	6.99e-05	4.15e-4
min $\mu(\mathbf{fU}, \mathbf{fL})$	5.79e-10	1.67e-9	1.12e-08	9.32e-10
med(t_u/t_s)	2.04	1.67	1.07	1.00
med(t_l/t_s)	0.01	1.26	0.01	1.55

The median of $\mu(\mathbf{fU}, \mathbf{fL})$ shows that for all conic solvers rigorous error bounds with about 7 decimal digits could be computed for most problems.

Furthermore, the table shows that the error bounds as well as the time ratios depend significantly on the used conic solver. In particular the resulting time ratios indicate that the conic solvers CSDP and SeDuMi aim to compute approximate primal interior ϵ -optimal solutions. In contrast SDPA and SDPT3 aim to compute dual interior ϵ -optimal solutions.

Even the largest problem **MaxG60** with about 24 million variables and 7000 constraints can be solved by VSDP, with high accuracy and in a reasonable time.

4.2 NETLIB LP test library

Here we describe some numerical results for the NETLIB linear programming library. This is a well known test suite containing many difficult to solve, real-world examples from a variety of sources [24].

For this test set Ordóñez and Freund [26] have shown that 71% of the problems are ill-posed. This statement is well reflected by our results: for the ill-posed problems VSDP computed infinite lower or infinite upper bounds. This happens if the distance to the next dual infeasible or primal infeasible problem is zero, respectively.

For computing approximations we used the solvers LINPROG, LPSOLVE, SEDUMI, and SDPT3. In the following table we display the same quantities as in Section 4.1. Again only the well-posed problems has been taken into account.

Table 4.2: Statistic of numerical results for the NETLIB LP test library.

	SDPT3	SEDUMI	LINPROG	LPSOLVE
med $\mu(\mathbf{fU}, \mathbf{fL})$	3.02e-8	1.08e-5	2.46e-7	2.46e-7
max $\mu(\mathbf{fU}, \mathbf{fL})$	6.60e-3	0.63	1.99	6.60e-3
min $\mu(\mathbf{fU}, \mathbf{fL})$	2.34e-10	2.05e-11	6.02e-10	5.22e-10
med(t_u/t_s)	0.95	1.29	1.49	5.59
med(t_l/t_s)	0.31	2.33	1.13	3.86

Here we would like to mention also the numerical results of the C++ software package LURUPA [16], [18]. In [17] comparisons with other software packages for the computation of rigorous errors bounds are described.

4.3 DIMACS

We present some statistics of numerical results for the DIMACS test library of semidefinite-quadratic-linear programs. This library was assembled for the purposes of the 7-th DIMACS Implementation Challenge. There are about 50 challenging problems that are divided into 12 groups. For details see [27]. In each group there are about 5 instances, from routinely solvable ones reaching to those at or beyond the capabilities of current solvers. Due to limited available memory some problems of the DIMACS test library have been omitted in our test. These are: `coppo68`, `industry2`, `fap09`, `fap25`, `fap36` `hamming112` and `qssp180old`.

One of the largest problems which could be solved by VSDP is the problem `hamming102`, with 23,041 equality constraints and 1,048,576 variables. The approximate solvers SDPA and CSDP are designed for SDP

Table 4.3: Statistic of numerical results for the DIMACS test library.

	SDPT3	SEDUMI	SDPA (only SDP)	CSDP (only SDP)
med $\mu(\mathbf{fU}, \mathbf{fL})$	8.90e-9	2.31e-8	5.48e-8	4.02e-9
max $\mu(\mathbf{fU}, \mathbf{fL})$	1.15	1.98	4.02e-7	1.11e-8
min $\mu(\mathbf{fU}, \mathbf{fL})$	5.92e-10	2.11e-9	1.32e-8	1.24e-9
med(t_u/t_s)	0.78	1.03	0.03	0.03
med(t_l/t_s)	0.35	2.25	0.06	1.16

problems only. Hence, in Table 4.3 we have displayed for these solvers only the statistic of the SDP problems.

4.4 Kovcara's Library of Structural Optimization Problems

In this section numerical results for problems from structural and topological optimization are presented. Structural and especially free material optimization gains more and more interest in the recent years. The most prominent example is the design of ribs in the leading edge of the new Airbus A380. We performed tests on problems from the test library collected by Kovcara. More details on these problems can be found in [5, 20, 36]. This is a collection of 26 sparse semidefinite programming problems. For 24 problems out of this collection VSDP could compute a rigorous primal and dual ϵ -optimal solution. Caused by the limited available memory and the great computational times the largest problems `mater-5`, `mater-6`, `shmup5`, `trto5` and `vibra5` has been tested only with the solver SDPT3. The largest problem that was rigorously solved by VSDP is `shmup5`. This problem has 1800 equality constraints and 13,849,441 variables.

A statistic of these numerical experiments can be found in Table 4.4.

Table 4.4: Statistic of numerical results for Kovcara's test library

	SDPT3	SEDUMI	SDPA	CSDP
med $\mu(\mathbf{fU}, \mathbf{fL})$	1.90e-6	1.92e-7	2.25e-5	3.43e-7
max $\mu(\mathbf{fU}, \mathbf{fL})$	6.38e-4	8.80e-4	1.90e-3	6.30e-5
min $\mu(\mathbf{fU}, \mathbf{fL})$	1.38e-9	7.74e-9	1.05e-8	1.62e-9
med(t_u/t_s)	2.11	3.22	1.03	0.38
med(t_l/t_s)	0.01	2.58	0.01	0.92

5 Appendix A

5.1 A Quick Reference for VSDP

Table 5.1: Main functions of VSDP.

Filename	Description
<code>vsdpinit.m</code>	Initialization function that adds the corresponding VSDP directories to the MATLAB search path. Moreover, <code>vsdpinit</code> can be used to setup global options. The command <code>vsdpinit('sdpt3')</code> , for instance, sets SDPT3 as the default approximate solver. For more details type <code>help vsdpinit</code> into the MATLAB command window.
<code>mysdps.m</code>	Interface to several conic solvers. Calls a preselected solver to get an approximate solution of a conic programming problem given in the condensed form (CP, DCP). Calling sequence: <code>[objt,xt,yt,zt,info] = mysdps(A,b,c,K,[x0,y0,z0,opts])</code> Input: <code>A</code> the constraint matrix <code>b</code> the dual objective <code>c</code> the primal objective <code>K.f</code> number of free variables <code>K.l</code> number of nonnegative variables <code>K.q</code> vector of dimensions of SOCP cones <code>K.s</code> vector of dimensions of SDP cones <code>x0,y0,z0</code> initial starting point <code>opts</code> option structure for local settings (see <code>vsdpinit</code>) Output: <code>objt(1)</code> the approximate primal optimal value <code>objt(2)</code> the approximate dual optimal value <code>xt</code> the approximate primal solution <code>yt, zt</code> the approximate dual solution pair <code>info</code> information on the success of the conic solver <code>info=0</code> — successful termination <code>info=1</code> — the problem might be primal infeasible <code>info=2</code> — the problem might be dual infeasible <code>info=3</code> — the problem might be primal and dual infeasible

vsdplow.m	Provides a rigorous lower bound of the primal optimal value as well as a rigorous dual ϵ-optimal solution. Calling sequence: <code>[fL,y,d1] = vsdplow(A,b,c,K,[xt],yt,[zt,xu,opts])</code> Input: <code>A,b,c,K</code> the problem data (see <code>mysdps</code>) <code>xt</code> the approximate primal solution <code>yt,zt</code> the approximate dual solution pair <code>xu</code> upper bounds of the optimal solution – for linear programming problems $x^l \leq \text{xu}(1:\text{n1})$ – for socp problems $(x_i^q)_1 + \ x_i^q\ _2 \leq \text{xu}(\text{n1}+\text{i})$ – for sdp problems $\lambda_{\max}(X_j^s) \leq \text{xu}(\text{n1}+\text{nq}+\text{j})$ <code>opts</code> option structure for local settings (see <code>vsdpinit</code>) Output: <code>fL</code> a verified lower bound for the optimal value <code>y</code> a rigorous dual feasible solution <code>d1</code> a rigorous lower bound of $z=c-A'y$ – for linear programming problems $z^l \geq \text{d1}(1:\text{n1})$ – for SOCP problems $(z_i^q)_1 - \ z_i^q\ _2 \geq \text{d1}(\text{n1}+\text{i})$ – for SDP problems $\lambda_{\min}(Z_j^s) \geq \text{d1}(\text{n1}+\text{nq}+\text{j})$
vsdpup.m	Provides a rigorous upper bound of the dual optimal value as well as an interval enclosure of a rigorous primal ϵ-optimal solution. Calling sequence: <code>[fU,x,lb] = vsdpup(A,b,c,K,xt,[yt,zt,yu,opts])</code> Input: <code>A,b,c,K</code> the problem data (see <code>mysdps</code>) <code>xt,yt,zt</code> the approximate primal-dual solution <code>yu</code> an upper bound of the dual optimal solution $y \in \mathbb{R}^m$ <code>opts</code> option structure for local settings (see <code>vsdpinit</code>) Output: <code>fU</code> a verified upper bound for the optimal value <code>x</code> a rigorous interval enclosure of a primal strictly feasible solution <code>lb</code> a rigorous lower bound of the primal solution x – for linear programming problems $x^l \geq \text{lb}(1:\text{n1})$ – for SOCP problems $(x_i^q)_1 - \ x_i^q\ _2 \geq \text{lb}(\text{n1}+\text{i})$ – for SDP problems $\lambda_{\min}(X_j^s) \geq \text{lb}(\text{n1}+\text{nq}+\text{j})$
vsdpinfeas.m	Implements algorithms that compute rigorous infeasibility certificates. Calling sequence: <code>[infeas,x,y] = vsdpinfeas(A,b,c,K,pd,xt,yt,[zt,opts])</code> Input: <code>A,b,c,K</code> the problem data (see <code>mysdps</code>) <code>pdinf</code> 'p' - to compute a certificate of primal infeasibility 'd' - to compute a certificate of dual infeasibility <code>xt,yt,zt</code> the approximate primal-dual solution <code>opts</code> option structure for local settings (see <code>vsdpinit</code>) Output: <code>infeas</code> 1 – primal infeasibility is proved 0 – could not prove infeasibility -1 – dual infeasibility is proved <code>x</code> a rigorous dual infeasibility certificate (primal unbounded ray) <code>y</code> a rigorous primal infeasibility certificate (dual unbounded ray)

Table 5.2: Data transformation functions in VSDP.

Filename	Description
<code>mps2vsdp.m</code>	Reads and transforms problem data in the MPS format into VSDP format. Calling sequence: <code>[A,b,c,K,pd] = mps2vsdp(problem)</code> Input: <code>problem</code> can be a string of the name of MPS data file, or the problem structure created by <code>read_mps</code> Output: <code>A,b,c,K</code> the problem data in the VSDP / SeDuMi format <code>pd</code> 'p' – problem was converted into primal form 'd' – problem was converted into dual form
<code>sdpa2vsdp.m</code>	Reads and converts data from an input data file in SDPA format. Calling sequence: <code>[A,b,c,K] = sdpa2vsdp(filename)</code> Input: <code>filename</code> a string of the name of the input data file Output: <code>A,b,c,K</code> the problem data in the VSDP / SeDuMi format
<code>sdpam2vsdp.m</code>	Transforms problem data in SDPAM format into VSDP format. Calling sequence: <code>[A,b,c,K,xt,yt,z] = sdpam2vsdp(bLOCKSTRUCT,c,F,[x0,X0,Y0])</code> Input: <code>bLOCKSTRUCT</code> vector containing the block structure of the problem <code>c</code> primal objective <code>F</code> cell vector of coefficient matrices and dual objective <code>x0,X0,Y0</code> cell arrays for the initial primal/dual solution vectors For more information about the format of SDPAM, see [10]. Output: <code>A,b,c,K,xt,yt,z</code> the problem data in the VSDP / SeDuMi format
<code>vsdp2sdpam.m</code>	Transforms problem data in VSDP format into SDPAM format (see <code>sdpam2vsdp</code>). Calling sequence: <code>[mDIM,nBLOCK,bLOCKSTRUCT,c,F,x0,X0,Y0] = vsdp2sdpam(A,b,c,K,[xt,yt,z])</code>
<code>sdpt2vsdp.m</code>	Converts problem data in SDPT3 format into VSDP format. Calling sequence: <code>[K,A,c,x0,z0] = sdpt2vsdp(blk,A,C,[X,Z])</code> Input: <code>blk</code> a cell array describing the structure of the cone K – <code>blk{i,1}</code> can be 's', 'q', 'l', 'u' depending on the i-th cone in K (more on the data format used by SDPT3 see [34]) – <code>blk{i,2}</code> the dimension of the i-th cone K <code>A</code> a cell array storing the constraint matrices of dimensions corresponding to the cone dimensions given by the array <code>blk</code> <code>C</code> a cell array containing vectors of the primal objective <code>X</code> a cell array containing vectors of the primal approximate solution <code>Z</code> a cell array containing vectors of the dual approximate solution Output: <code>K,A,c,xt,z</code> the problem data in the VSDP format Since the parameters <code>b</code> and <code>yt</code> are equal in both formats, they are omitted.
<code>vsdp2sdpt3.m</code>	Converts problem data in VSDP format into SDPT3 format (see <code>sdpt2vsdp</code>). Calling sequence: <code>[blk,A,C,X,Z] = vsdp2sdpt3(K,A,c,[x0,z0,opts])</code>

<code>vsdp12vsdp.m</code>	Converts problem data from the old VSDP 2006 format into the current VSDP 2012 format. Calling sequence: <code>[K,A,c,xt,zt] = vsdp12vsdp(blk,At,C,Xt,Zt)</code> Input: <code>blk</code> $n \times 2$ cell array with dimensions of the semidefinite cone – <code>blk{j,1}</code> allways 's', – <code>blk{j,2}</code> the dimension s_j of the j-th SDP block K <code>A</code> $m \times n$ cell array, where <code>A{i,j}</code> stores the $s_j \times s_j$ constraint matrix $A_{i,j}^s$ of the j-th semidefinite block and the i-th equality <code>C</code> n cell array containing the primal objective <code>Xt</code> n cell array containing the initial primal solution <code>Zt</code> n cell array containing the initial dual solution Output: <code>K,A,c,xt,zt</code> problem data in the current VSDP 2012 format Since the parameters <code>b</code> and <code>yt</code> are equal in both formats, they are omitted.
---------------------------	--

Table 5.3: Additional functions for verification in VSDP.

Filename	Description
<code>vsdpneig.m</code>	Computes a rigorous eigenvalue enclosure for normal matrices. Calling sequence: <code>lambda = vsdpneig(A)</code> Input: <code>A</code> normal real or complex matrix Output: <code>lambda</code> <code>intval</code> vector enclosing all eigenvalues of <code>A</code>
<code>bnd4sd.m</code>	Computes a lower verified bound for the eigenvalues of symmetric matrices. Calling sequence: <code>[lambdaMin,trneg] = bnd4sd(A)</code> Input: <code>A</code> symmetric matrix Output: <code>lambdaMin</code> rigorous lower bound for the minimum eigenvalue of <code>A</code> <code>trneg</code> rigorous lower bound for sum of all negative eigenvalues of <code>A</code>

6 Appendix B - Numerical Results

Table 6.1: Rigorous error bounds for the SDPLIB

Solver	fU	fL	$\mu(fU, fL)$	t_s	t_u	t_l
arch0 equations: 174 variables: 13215						
sdpt3	-5.66517029e-01	-5.66517274e-01	2.45e-07	1.88e+00	4.11e+00	7.73e-03
sedumi	-5.66513308e-01	-5.66517273e-01	3.97e-06	2.13e+00	5.83e+00	2.79e+00
sdpa	-5.66517052e-01	-5.66517308e-01	2.56e-07	1.59e+00	7.58e-02	4.11e-03
csdp	-5.66517230e-01	-5.66517274e-01	4.38e-08	2.16e+00	2.33e+00	4.22e+00
arch2 equations: 174 variables: 13215						
sdpt3	-6.71515330e-01	-6.71515409e-01	7.89e-08	1.88e+00	8.03e+00	4.33e-03
sedumi	-6.71503644e-01	-6.71515408e-01	1.18e-05	2.02e+00	1.11e+01	5.18e-03
sdpa	-6.71514992e-01	-6.71515452e-01	4.60e-07	1.73e+00	7.79e-02	4.10e-03
csdp	-6.71515334e-01	-6.71515410e-01	7.62e-08	2.02e+00	8.80e-02	4.25e+00
arch4 equations: 174 variables: 13215						
sdpt3	-9.72625949e-01	-9.72627419e-01	1.47e-06	1.80e+00	1.94e+00	4.33e-03
sedumi	-9.72572248e-01	-9.72627418e-01	5.52e-05	2.05e+00	5.82e+00	2.69e+00
sdpa	-9.72626536e-01	-9.72627443e-01	9.07e-07	1.94e+00	2.05e+00	3.97e-03
csdp	-9.72626704e-01	-9.72627417e-01	7.13e-07	2.15e+00	4.75e+00	2.07e+00
arch8 equations: 174 variables: 13215						
sdpt3	-7.05694732e+00	-7.05698005e+00	4.64e-06	1.85e+00	2.01e+00	4.65e-03
sedumi	-7.05676796e+00	-7.05698004e+00	3.01e-05	2.16e+00	5.91e+00	2.84e+00
sdpa	-7.05685762e+00	-7.05698004e+00	1.73e-05	2.06e+00	2.16e+00	2.78e-02
csdp	-7.05697977e+00	-7.05698005e+00	3.94e-08	2.32e+00	8.27e-02	2.29e+00
control1 equations: 21 variables: 70						
sdpt3	-1.77846267e+01	-1.77846267e+01	9.05e-10	3.56e-01	3.51e-01	2.03e-03
sedumi	-1.77846266e+01	-1.77846267e+01	9.01e-09	2.82e-01	1.40e+00	2.13e-03
sdpa	-1.77846260e+01	-1.77846271e+01	6.15e-08	1.74e-01	1.98e-01	2.04e-03
csdp	-1.77846266e+01	-1.77846267e+01	6.22e-09	2.45e-02	1.28e-01	3.16e-02
control2 equations: 66 variables: 265						
sdpt3	-8.29999610e+00	-8.30000019e+00	4.93e-07	8.85e-01	3.69e+00	2.07e-03
sedumi	-8.29999697e+00	-8.30000000e+00	3.65e-07	3.81e-01	4.54e-01	1.82e-03
sdpa	-8.29998887e+00	-8.30000021e+00	1.37e-06	2.23e-01	6.01e-01	1.84e-03
csdp	-8.2999989e+00	-8.29999999e+00	1.25e-08	7.81e-02	1.31e-01	2.93e-01
control3 equations: 136 variables: 585						
sdpt3	-1.36332516e+01	-1.36332695e+01	1.31e-06	1.19e+00	4.34e+00	3.26e-03
sedumi	-1.36332458e+01	-1.36332673e+01	1.58e-06	1.04e+00	6.44e+00	1.98e-03
sdpa	-1.36332473e+01	-1.36332673e+01	1.47e-06	6.99e-01	2.93e+00	1.95e-03
csdp	-1.36331222e+01	-1.36332663e+01	1.06e-05	3.79e-01	2.10e+00	3.49e-01
control4 equations: 231 variables: 1030						
sdpt3	-1.97939020e+01	-1.97942360e+01	1.69e-05	2.67e+00	5.64e+00	2.43e-03
sedumi	-1.97941479e+01	-1.97942361e+01	4.45e-06	1.21e+00	7.52e+00	2.18e-03
sdpa	-1.97941576e+01	-1.97942368e+01	4.00e-06	1.97e+00	8.22e+00	2.16e-03
csdp	-1.97941621e+01	-1.97942304e+01	3.45e-06	7.42e-01	3.23e+00	2.77e+00
control5 equations: 351 variables: 1600						
sdpt3	-1.68898955e+01	-1.68836479e+01	2.22e-04	4.98e+00	3.30e+01	2.68e-03
sedumi	-1.68832376e+01	-1.68836132e+01	2.22e-05	2.60e+00	8.16e+00	2.76e-03
sdpa	-1.68835702e+01	-1.68836189e+01	2.88e-06	3.55e+00	3.85e+00	2.77e-03
csdp	-1.68829119e+01	-1.68835994e+01	4.07e-05	2.94e+00	5.73e+00	5.55e+00
control6 equations: 496 variables: 2295						
sdpt3	-3.72962827e+01	-3.73044633e+01	2.19e-04	1.17e+01	2.14e+01	2.97e-03
sedumi	-3.73017345e+01	-3.73045159e+01	7.46e-05	4.85e+00	1.49e+01	2.84e-03
sdpa	-3.73041322e+01	-3.73044816e+01	9.37e-06	7.84e+00	1.80e-01	2.85e-03
csdp	-3.73043700e+01	-3.73044136e+01	1.17e-06	6.04e+00	2.95e+01	1.03e+01
control7 equations: 666 variables: 3115						
sdpt3	-2.06248376e+01	-2.06251438e+01	1.48e-05	1.72e+01	2.72e-01	3.38e-03
sedumi	-2.06236683e+01	-2.06251113e+01	7.00e-05	9.39e+00	2.55e+01	3.43e-03
sdpa	-2.06249269e+01	-2.06251029e+01	8.53e-06	1.60e+01	2.89e-01	3.23e-03
csdp	-2.06250571e+01	-2.06250727e+01	7.57e-07	3.69e+01	1.87e+02	1.55e+02

Solver	fU	fL	$\mu(fU, fL)$	t_s	t_u	t_l
control8 equations: 861 variables: 4060						
sdpt3	-2.02820429e+01	-2.02863712e+01	2.13e-04	1.30e+01	4.00e+01	3.92e-03
sedumi	-2.02489101e+01	-2.02863848e+01	1.85e-03	1.28e+01	5.98e+01	3.85e-03
sdpa	-2.02862322e+01	-2.02863737e+01	6.97e-06	2.96e+01	9.68e+01	3.64e-03
csdp	-2.02863455e+01	-2.02863607e+01	7.48e-07	1.35e+02	2.05e+02	6.40e+01
control9 equations: 1081 variables: 5130						
sdpt3	-1.46645939e+01	-1.46754678e+01	7.41e-04	1.77e+01	7.50e+01	4.62e-03
sedumi	-1.46446427e+01	-1.46754348e+01	2.10e-03	2.05e+01	4.13e+01	4.43e-03
sdpa	-1.46754052e+01	-1.46754314e+01	1.78e-06	5.38e+01	1.12e+02	4.22e-03
csdp	-1.46754127e+01	-1.46754250e+01	8.35e-07	9.72e+01	2.11e+02	9.71e+01
control10 equations: 1326 variables: 6325						
sdpt3	Inf	-3.85331048e+01	NaN	2.97e+01	4.13e+01	8.79e-03
sedumi	-3.82027938e+01	-3.85331589e+01	8.61e-03	4.43e+01	1.81e+02	4.93e-03
sdpa	-3.85324462e+01	-3.85330937e+01	1.68e-05	1.00e+02	1.05e+02	5.15e-03
csdp	-3.85329716e+01	-3.85330351e+01	1.65e-06	1.49e+02	6.61e+02	2.95e+02
control11 equations: 1596 variables: 7645						
sdpt3	-3.19552153e+01	-3.19587678e+01	1.11e-04	4.55e+01	4.58e+01	6.31e-03
sedumi	-3.17422256e+01	-3.19587148e+01	6.80e-03	7.17e+01	3.07e+02	5.93e-03
sdpa	-3.19580047e+01	-3.19587765e+01	2.42e-05	1.62e+02	1.38e+00	5.93e-03
csdp	-3.19454079e+01	-3.19586702e+01	4.15e-04	2.22e+02	2.01e+03	6.72e+02
equalg11 equations: 801 variables: 321201						
sdpt3	-6.29150881e+02	-6.29155301e+02	7.03e-06	1.82e+01	3.90e+01	6.18e-01
sedumi	-6.26864929e+02	-6.29155913e+02	3.65e-03	7.37e+01	2.17e+02	1.41e+02
sdpa	-6.29146256e+02	-6.29155297e+02	1.44e-05	4.09e+02	1.48e+02	6.10e-01
csdp	-6.29120114e+02	-6.29155294e+02	5.59e-05	5.49e+01	4.30e+01	1.29e+02
equalg51 equations: 1001 variables: 501501						
sdpt3	-4.00558939e+03	-4.00560148e+03	3.02e-06	3.21e+01	3.11e+00	1.20e+00
sedumi	-3.89871913e+03	-4.00561728e+03	2.70e-02	1.66e+02	4.94e+02	2.02e+02
sdpa	-4.00554016e+03	-4.00560138e+03	1.53e-05	1.64e+02	1.78e+00	1.16e+00
csdp	-4.00558813e+03	-4.00560132e+03	3.29e-06	1.05e+02	7.65e+01	1.72e+02
gpp100 equations: 101 variables: 5050						
sdpt3	Inf	4.49435488e+01	NaN	7.23e-01	7.15e+00	2.69e-02
sedumi	Inf	4.49421174e+01	NaN	1.12e+00	1.43e+00	1.79e+00
sdpa	Inf	4.49435483e+01	NaN	3.05e+00	5.46e+00	2.49e-02
csdp	Inf	4.49435488e+01	NaN	1.24e+00	8.52e+00	9.88e-01
gpp124-1 equations: 125 variables: 7750						
sdpt3	Inf	7.34307384e+00	NaN	9.57e-01	1.22e+01	3.17e-02
sedumi	Inf	-1.37517251e+01	NaN	1.68e+00	1.98e+00	3.80e+00
sdpa	Inf	7.34307496e+00	NaN	3.75e+00	8.47e+00	2.58e-02
csdp	Inf	7.34307523e+00	NaN	2.33e+00	2.40e+00	3.67e+00
gpp124-2 equations: 125 variables: 7750						
sdpt3	Inf	4.68622938e+01	NaN	6.48e-01	8.07e+00	2.90e-02
sedumi	Inf	4.68611683e+01	NaN	1.51e+00	1.92e+00	1.90e+00
sdpa	Inf	4.68622931e+01	NaN	3.37e+00	6.75e+00	2.61e-02
csdp	Inf	4.68622940e+01	NaN	2.10e+00	1.78e+00	1.94e+00
gpp124-3 equations: 125 variables: 7750						
sdpt3	Inf	1.53014124e+02	NaN	6.66e-01	7.60e+00	2.66e-02
sedumi	Inf	1.53011575e+02	NaN	1.43e+00	1.59e+00	2.58e+00
sdpa	Inf	1.53014123e+02	NaN	3.18e+00	5.97e+00	2.36e-02
csdp	Inf	1.53014126e+02	NaN	1.53e+00	3.83e+00	2.23e+00
gpp124-4 equations: 125 variables: 7750						
sdpt3	Inf	4.18987486e+02	NaN	7.19e-01	8.88e+00	2.34e-02
sedumi	Inf	4.18983669e+02	NaN	1.51e+00	1.99e+00	2.47e+00
sdpa	Inf	4.18987583e+02	NaN	7.89e-01	5.12e+00	3.38e-02
csdp	Inf	4.18987621e+02	NaN	1.96e+00	4.00e+01	2.09e+00
gpp250-1 equations: 251 variables: 31375						
sdpt3	Inf	1.54449140e+01	NaN	2.21e+00	2.49e+01	5.88e-02
sedumi	Inf	5.69147712e+00	NaN	5.73e+00	5.93e+00	1.05e+01
sdpa	Inf	1.54449133e+01	NaN	1.52e+01	2.37e+01	6.33e-02
csdp	Inf	1.54448983e+01	NaN	6.51e+00	6.38e+00	5.11e+00
gpp250-2 equations: 251 variables: 31375						
sdpt3	Inf	8.18686781e+01	NaN	1.61e+00	1.83e+01	6.07e-02
sedumi	Inf	8.18164705e+01	NaN	4.88e+00	8.97e+00	9.65e+00
sdpa	Inf	8.18689536e+01	NaN	1.52e+01	2.69e+01	6.84e-02
csdp	Inf	8.18505416e+01	NaN	4.60e+00	6.10e+00	2.69e+01

Solver	fU	fL	$\mu(fU, fL)$	t_s	t_u	t_l
gpp250-3 equations: 251 variables: 31375						
sdpt3	Inf	3.03537191e+02	NaN	1.74e+00	1.96e+01	6.57e-02
sedumi	Inf	3.03515408e+02	NaN	7.35e+00	5.74e+00	1.25e+01
sdpa	Inf	3.03539306e+02	NaN	1.61e+01	2.33e+01	7.04e-02
csdp	Inf	3.03539259e+02	NaN	6.41e+00	5.19e+00	5.25e+00
gpp250-4 equations: 251 variables: 31375						
sdpt3	Inf	7.47326112e+02	NaN	1.55e+00	1.69e+01	5.94e-02
sedumi	Inf	7.47291829e+02	NaN	5.90e+00	9.81e+00	1.21e+01
sdpa	Inf	7.47328274e+02	NaN	1.64e+01	2.34e+01	5.92e-02
csdp	Inf	7.47328290e+02	NaN	5.87e+00	2.34e+01	4.91e+00
gpp500-1 equations: 501 variables: 125250						
sdpt3	Inf	2.53205086e+01	NaN	1.11e+01	1.79e+02	2.14e-01
sedumi	Inf	-4.87116465e+02	NaN	3.17e+01	2.32e+01	8.58e+01
sdpa	Inf	2.53205228e+01	NaN	1.21e+02	1.53e+02	2.04e-01
csdp	Inf	2.53189557e+01	NaN	3.40e+01	2.99e+01	7.34e+01
gpp500-2 equations: 501 variables: 125250						
sdpt3	Inf	1.56060093e+02	NaN	4.64e+00	5.73e+01	2.24e-01
sedumi	Inf	1.33242528e+02	NaN	2.95e+01	6.07e+01	4.14e+01
sdpa	Inf	1.56060371e+02	NaN	1.01e+02	1.63e+02	2.10e-01
csdp	Inf	1.56060372e+02	NaN	2.36e+01	2.72e+01	2.41e+01
gpp500-3 equations: 501 variables: 125250						
sdpt3	Inf	5.13011157e+02	NaN	4.47e+00	1.01e+02	2.11e-01
sedumi	Inf	5.12972060e+02	NaN	3.08e+01	2.81e+01	5.37e+01
sdpa	Inf	5.13017598e+02	NaN	9.89e+01	1.30e+02	2.09e-01
csdp	Inf	5.13017540e+02	NaN	1.96e+01	2.92e+01	2.42e+01
gpp500-4 equations: 501 variables: 125250						
sdpt3	Inf	1.56700960e+03	NaN	4.63e+00	5.54e+01	2.14e-01
sedumi	Inf	1.56699533e+03	NaN	2.90e+01	3.26e+01	3.41e+01
sdpa	Inf	1.56701878e+03	NaN	1.23e+02	1.41e+02	2.04e-01
csdp	Inf	1.56701878e+03	NaN	2.21e+01	4.16e+01	1.98e+01
hinfl equations: 13 variables: 41						
sdpt3	Inf	-2.03272802e+00	NaN	3.33e-01	8.28e-01	3.81e-03
sedumi	Inf	-2.03268132e+00	NaN	4.33e-01	1.44e+00	3.74e-03
sdpa	Inf	-2.03261833e+00	NaN	1.64e-01	2.05e+00	7.26e-03
csdp	Inf	-2.03266062e+00	NaN	3.87e-02	5.34e-01	3.81e-03
hinf2 equations: 13 variables: 51						
sdpt3	Inf	-1.09681526e+01	NaN	3.70e-01	4.56e+00	4.38e-03
sedumi	-1.08392247e+01	-1.09683670e+01	1.18e-02	4.36e-01	1.41e+00	7.85e-01
sdpa	-1.09670019e+01	-1.09670633e+01	5.60e-06	3.09e-01	2.76e-01	2.13e-03
csdp	Inf	-1.09671373e+01	NaN	4.27e-02	6.90e-01	1.16e-01
hinf3 equations: 13 variables: 51						
sdpt3	Inf	-5.69544040e+01	NaN	5.24e-01	5.37e+00	3.51e-03
sedumi	Inf	-5.69480045e+01	NaN	2.71e-01	1.94e+00	1.03e+00
sdpa	Inf	-5.69505167e+01	NaN	2.24e-01	2.38e+00	3.28e-01
csdp	Inf	-5.69455865e+01	NaN	3.03e-02	5.53e-01	1.25e-01
hinf4 equations: 13 variables: 51						
sdpt3	Inf	-2.74764875e+02	NaN	6.03e-01	3.46e+00	5.46e-03
sedumi	Inf	-2.74764017e+02	NaN	4.90e-01	4.00e+00	3.16e-03
sdpa	Inf	-2.74764087e+02	NaN	1.88e-01	2.70e+00	4.06e-03
csdp	Inf	-2.74764066e+02	NaN	2.72e-02	6.07e-01	1.02e-01
hinf5 equations: 13 variables: 51						
sdpt3	Inf	-3.62579765e+02	NaN	6.06e-01	3.07e+00	3.32e-03
sedumi	Inf	-3.62951132e+02	NaN	3.11e-01	2.02e+00	2.88e-01
sdpa	Inf	-3.62889308e+02	NaN	1.35e-01	2.46e+00	4.04e-03
csdp	Inf	-Inf	NaN	4.70e-02	5.16e-02	4.15e-02
hinf6 equations: 13 variables: 51						
sdpt3	Inf	-4.48950850e+02	NaN	4.69e-01	1.21e+00	3.47e-03
sedumi	Inf	-4.48951180e+02	NaN	3.41e-01	1.43e+00	9.71e-01
sdpa	Inf	-4.48971541e+02	NaN	2.36e-01	1.88e+00	4.52e-03
csdp	Inf	-4.48942508e+02	NaN	3.12e-02	5.77e-01	1.55e-01
hinf7 equations: 13 variables: 51						
sdpt3	Inf	-3.90819828e+02	NaN	5.54e-01	4.55e+00	3.31e-03
sedumi	Inf	-4.03624423e+02	NaN	2.96e-01	2.23e+00	6.54e-01
sdpa	Inf	-3.90817395e+02	NaN	2.23e-01	2.01e+00	3.78e-03
csdp	Inf	-3.90825577e+02	NaN	3.27e-02	3.47e-01	7.08e-02

Solver	fU	fL	$\mu(fU, fL)$	t_s	t_u	t_l
hinf8 equations: 13 variables: 51						
sdpt3	Inf	-1.16170520e+02	NaN	5.07e-01	1.41e+00	3.51e-03
sedumi	Inf	-1.16151122e+02	NaN	4.00e-01	6.76e-01	5.22e-01
sdpa	Inf	-1.16161924e+02	NaN	1.70e-01	1.81e+00	6.06e-03
csdp	Inf	-1.16170832e+02	NaN	3.62e-02	1.13e-01	1.94e-01
hinf9 equations: 13 variables: 51						
sdpt3	-2.36248985e+02	-2.36249258e+02	1.16e-06	5.76e-01	1.12e+00	3.21e-01
sedumi	-2.36247782e+02	-2.36252218e+02	1.88e-05	2.73e-01	1.06e-02	2.40e+00
sdpa	-2.36232735e+02	-2.36249258e+02	6.99e-05	5.45e-01	3.02e+00	2.56e+00
csdp	-2.36249212e+02	-2.36249288e+02	3.19e-07	3.48e-02	1.17e-02	3.14e-01
hinf10 equations: 21 variables: 66						
sdpt3	Inf	-1.08813953e+02	NaN	8.71e-01	1.27e+00	6.08e-01
sedumi	Inf	-1.08775372e+02	NaN	4.33e-01	2.78e-01	3.33e-03
sdpa	Inf	-1.08869700e+02	NaN	4.48e-01	1.88e+00	7.94e-03
csdp	Inf	-1.08752572e+02	NaN	5.44e-02	2.57e-01	4.91e-03
hinf11 equations: 31 variables: 97						
sdpt3	Inf	-6.59301753e+01	NaN	6.52e-01	1.65e+00	4.08e-03
sedumi	Inf	-6.58859105e+01	NaN	6.43e-01	7.06e-01	4.49e-03
sdpa	Inf	-6.59010123e+01	NaN	5.73e-01	2.05e+00	4.01e-03
csdp	Inf	-6.58874368e+01	NaN	5.83e-02	4.48e-01	4.27e-03
hinf12 equations: 43 variables: 120						
sdpt3	Inf	-1.88512622e-05	NaN	1.92e+00	3.62e+00	4.56e-03
sedumi	Inf	-1.02543992e-01	NaN	4.80e-01	4.70e-01	3.84e-03
sdpa	Inf	-3.21323790e+00	NaN	2.67e-01	1.80e+00	3.62e-03
csdp	Inf	-1.42731095e-01	NaN	6.34e-02	1.37e-01	5.88e-03
hinf13 equations: 57 variables: 178						
sdpt3	Inf	-4.43598679e+01	NaN	1.35e+00	3.52e+00	3.00e+00
sedumi	Inf	-4.68274032e+01	NaN	4.59e-01	8.08e-01	2.39e+00
sdpa	Inf	-4.51586573e+01	NaN	2.07e-01	1.64e+00	4.82e-03
csdp	Inf	-4.47709309e+01	NaN	1.30e-01	1.91e-01	4.43e-01
hinf14 equations: 73 variables: 227						
sdpt3	Inf	-1.29932110e+01	NaN	9.89e-01	4.27e+00	3.08e-03
sedumi	Inf	-1.29947389e+01	NaN	1.19e+00	2.19e+00	5.26e-03
sdpa	Inf	-1.30006109e+01	NaN	1.47e-01	1.82e+00	5.85e-03
csdp	Inf	-1.29920331e+01	NaN	1.90e-01	1.22e+00	6.75e-03
hinf15 equations: 91 variables: 273						
sdpt3	Inf	-2.39958457e+01	NaN	1.49e+00	1.39e+01	3.57e+00
sedumi	Inf	-2.65870483e+01	NaN	7.25e-01	5.71e-01	2.03e+00
sdpa	Inf	-2.57797730e+01	NaN	1.68e-01	1.89e+00	5.87e-03
csdp	Inf	-2.44671077e+01	NaN	2.58e-01	2.33e-01	5.38e-01
infd1 equations: 10 variables: 465						
sdpt3	Inf	-Inf	NaN	3.33e-01	1.65e-01	2.22e+00
sedumi	Inf	-Inf	NaN	8.45e-02	8.00e-02	9.63e-01
sdpa	Inf	1.20723065e+06	NaN	1.12e-01	1.21e+00	2.63e-03
csdp	Inf	-Inf	NaN	5.94e-02	7.64e-02	8.40e-01
infd2 equations: 10 variables: 465						
sdpt3	Inf	-Inf	NaN	1.97e-01	1.46e-01	2.31e+00
sedumi	Inf	-Inf	NaN	7.56e-02	8.51e-02	8.70e-01
sdpa	Inf	2.81458878e+06	NaN	1.18e-01	1.26e+00	2.72e-03
csdp	Inf	-Inf	NaN	8.44e-02	6.69e-02	8.04e-01
infp1 equations: 10 variables: 465						
sdpt3	Inf	-Inf	NaN	5.86e-01	5.30e+00	4.15e-01
sedumi	-3.58596958e+02	-Inf	NaN	3.50e-02	3.36e-01	6.69e-02
sdpa	-8.30762452e+07	-Inf	NaN	1.03e-01	1.76e-02	1.01e+00
csdp	Inf	-Inf	NaN	5.61e-02	8.10e-01	5.25e-02
infp2 equations: 10 variables: 465						
sdpt3	-2.94672354e+02	-Inf	NaN	4.76e-01	3.00e+00	4.59e-01
sedumi	-6.75033823e+01	-Inf	NaN	3.11e-02	3.72e-01	9.04e-02
sdpa	-1.48749913e+08	-Inf	NaN	1.17e-01	1.20e-02	1.02e+00
csdp	-3.19719604e+02	-Inf	NaN	6.26e-02	5.06e-01	8.14e-02
maxg11 equations: 800 variables: 320400						
sdpt3	-6.29164777e+02	-6.29164783e+02	9.75e-09	6.72e+00	8.90e-01	5.22e-02
sedumi	-6.29164779e+02	-6.29164783e+02	6.78e-09	4.39e+01	8.95e-01	5.36e+01
sdpa	-6.29164761e+02	-6.29164783e+02	3.47e-08	3.31e+01	8.85e-01	5.10e-02
csdp	-6.29164781e+02	-6.29164783e+02	3.63e-09	2.07e+01	9.19e-01	1.96e+01

Solver	fU	fL	$\mu(fU, fL)$	t_s	t_u	t_l
maxg32 equations: 2000 variables: 2001000						
sdpt3	-1.56763961e+03	-1.56763964e+03	1.98e-08	5.73e+01	1.08e+01	5.25e+00
sedumi	-1.56763964e+03	-1.56763964e+03	3.19e-09	7.98e+02	1.05e+01	9.28e+02
sdpa	-1.56763961e+03	-1.56763964e+03	2.00e-08	6.11e+02	1.04e+01	5.05e+00
csdp	-1.56763963e+03	-1.56763964e+03	1.03e-08	1.91e+02	1.07e+01	2.13e+02
maxg51 equations: 1000 variables: 500500						
sdpt3	-4.00625552e+03	-4.00625552e+03	5.79e-10	1.91e+01	1.68e+00	3.72e+00
sedumi	-4.00625546e+03	-4.00625552e+03	1.54e-08	1.11e+02	1.61e+00	2.59e+02
sdpa	-4.00625535e+03	-4.00625552e+03	4.24e-08	7.29e+01	1.73e+00	2.72e+00
csdp	-4.00625550e+03	-4.00625552e+03	4.63e-09	5.71e+01	1.75e+00	1.16e+02
maxg55 equations: 5000 variables: 12502500						
sdpt3	-1.28698665e+04	-1.28698667e+04	1.37e-08	1.45e+03	3.27e+02	1.74e+02
sedumi	-1.28586134e+04	-4.22582451e+04	1.07e+00	1.53e+04	3.28e+04	5.12e+04
sdpa	-1.28698660e+04	-1.28698667e+04	5.37e-08	7.86e+03	1.45e+02	1.48e+02
csdp	-1.28698665e+04	-1.28698667e+04	1.03e-08	2.30e+03	1.45e+02	2.65e+03
maxg60 equations: 7000 variables: 24503500						
sdpt3	-1.52222680e+04	-1.52222680e+04	1.46e-09	3.36e+03	8.67e+02	3.73e+02
sedumi	-1.52222680e+04	-2.14325762e+04	3.39e-01	5.40e+04	4.15e+02	5.80e+04
sdpa	-1.52222676e+04	-1.52222680e+04	2.98e-08	2.10e+04	4.02e+02	3.99e+02
csdp	-1.52222680e+04	-1.52222680e+04	2.35e-09	5.96e+03	3.90e+02	6.44e+03
mcp100 equations: 100 variables: 5050						
sdpt3	-2.26157351e+02	-2.26157352e+02	3.98e-09	2.61e-01	2.87e-02	3.60e-03
sedumi	-2.26157351e+02	-2.26157352e+02	1.86e-09	4.07e-01	1.04e+00	4.74e-01
sdpa	-2.26157348e+02	-2.26157352e+02	1.71e-08	2.51e-01	3.85e-02	4.74e-03
csdp	-2.26157350e+02	-2.26157352e+02	6.62e-09	3.73e-01	2.87e-02	4.17e-01
mcp124-1 equations: 124 variables: 7750						
sdpt3	-1.41990475e+02	-1.41990477e+02	1.50e-08	4.22e-01	6.92e-02	4.86e-03
sedumi	-1.41990476e+02	-2.83533791e+02	6.65e-01	5.40e-01	3.63e-02	2.03e+00
sdpa	-1.41990476e+02	-1.41990477e+02	1.13e-08	3.55e-01	3.32e-02	3.27e-03
csdp	-1.41990476e+02	-1.41990477e+02	4.90e-09	5.66e-01	3.13e-02	6.46e-01
mcp124-2 equations: 124 variables: 7750						
sdpt3	-2.69880170e+02	-2.69880171e+02	8.23e-10	4.16e-01	3.62e-02	4.78e-03
sedumi	-2.69880169e+02	-2.69880171e+02	4.51e-09	5.71e-01	3.87e-02	6.99e-01
sdpa	-2.69880163e+02	-2.69880171e+02	3.12e-08	3.20e-01	3.78e-02	4.74e-03
csdp	-2.69880166e+02	-2.69880171e+02	1.68e-08	8.41e-01	3.32e-02	1.69e+00
mcp124-3 equations: 124 variables: 7750						
sdpt3	-4.67750110e+02	-4.67750114e+02	9.90e-09	3.55e-01	3.55e-02	6.20e-03
sedumi	-4.67750112e+02	-4.67750114e+02	4.44e-09	5.78e-01	4.63e-02	7.09e-01
sdpa	-4.67750103e+02	-4.67750116e+02	2.76e-08	3.05e-01	4.51e-02	6.21e-03
csdp	-4.67750113e+02	-4.67750114e+02	3.04e-09	7.00e-01	3.41e-02	1.07e+00
mcp124-4 equations: 124 variables: 7750						
sdpt3	-8.64411863e+02	-8.64411864e+02	8.10e-10	3.85e-01	3.60e-02	3.27e-03
sedumi	-8.64411863e+02	-8.64411864e+02	1.67e-09	5.75e-01	2.31e+00	7.40e-01
sdpa	-8.64411847e+02	-8.64411865e+02	2.14e-08	3.15e-01	3.50e-02	2.94e-03
csdp	-8.64411863e+02	-8.64411864e+02	1.27e-09	7.30e-01	3.37e-02	6.37e-01
mcp250-1 equations: 250 variables: 31375						
sdpt3	-3.17264340e+02	-3.17264340e+02	1.95e-09	8.12e-01	8.63e-02	8.39e-03
sedumi	-3.17264339e+02	-5.60214445e+02	5.54e-01	2.40e+00	8.33e-02	6.03e+00
sdpa	-3.17264324e+02	-3.17264343e+02	6.02e-08	9.87e-01	7.85e-02	8.44e-03
csdp	-3.17264339e+02	-3.17264340e+02	4.30e-09	2.31e+00	7.49e-02	2.32e+00
mcp250-2 equations: 250 variables: 31375						
sdpt3	-5.31930081e+02	-5.31930084e+02	5.25e-09	9.97e-01	9.25e-02	1.47e-02
sedumi	-5.31930082e+02	-7.32552597e+02	3.17e-01	2.27e+00	8.32e-02	2.92e+00
sdpa	-5.31930042e+02	-5.31930086e+02	8.36e-08	1.03e+00	8.25e-02	1.45e-02
csdp	-5.31930081e+02	-5.31930084e+02	5.15e-09	2.09e+00	8.01e-02	4.45e+00
mcp250-3 equations: 250 variables: 31375						
sdpt3	-9.81172566e+02	-9.81172572e+02	5.61e-09	9.13e-01	8.42e-02	2.39e-02
sedumi	-9.81172568e+02	-9.81172572e+02	3.45e-09	2.23e+00	8.41e-02	3.48e+00
sdpa	-9.81172528e+02	-9.81172574e+02	4.66e-08	1.10e+00	8.53e-02	2.41e-02
csdp	-9.81172568e+02	-9.81172572e+02	3.31e-09	2.24e+00	8.38e-02	4.49e+00
mcp250-4 equations: 250 variables: 31375						
sdpt3	-1.68196010e+03	-1.68196011e+03	9.09e-09	9.48e-01	9.08e-02	9.61e-03
sedumi	-1.68196011e+03	-1.68196011e+03	3.78e-09	2.24e+00	8.70e-02	2.83e+00
sdpa	-1.68196002e+03	-1.68196012e+03	5.57e-08	1.06e+00	8.92e-02	7.16e-03
csdp	-1.68196010e+03	-1.68196011e+03	1.02e-08	2.25e+00	8.12e-02	2.15e+00

Solver	fU	fL	$\mu(fU, fL)$	t_s	t_u	t_l
mcp500-1 equations: 500 variables: 125250						
sdpt3	-5.98148516e+02	-5.98148517e+02	1.11e-09	2.21e+00	3.79e-01	2.92e-02
sedumi	-5.98148516e+02	-2.08470467e+03	1.11e+00	1.48e+01	3.25e-01	5.46e+01
sdpa	-5.98148510e+02	-5.98148519e+02	1.43e-08	5.73e+00	2.80e-01	2.88e-02
csdp	-5.98148516e+02	-5.98148517e+02	1.98e-09	6.49e+00	3.99e-01	6.57e+00
mcp500-2 equations: 500 variables: 125250						
sdpt3	-1.07005676e+03	-1.07005677e+03	2.38e-09	3.15e+00	3.88e-01	7.26e-02
sedumi	-1.07005676e+03	-1.85041952e+03	5.34e-01	1.40e+01	3.36e-01	3.63e+01
sdpa	-1.07005675e+03	-1.07005677e+03	1.84e-08	7.22e+00	3.14e-01	7.37e-02
csdp	-1.07005676e+03	-1.07005677e+03	3.68e-09	9.14e+00	2.94e-01	9.34e+00
mcp500-3 equations: 500 variables: 125250						
sdpt3	-1.84796999e+03	-1.84797002e+03	1.84e-08	3.23e+00	3.06e-01	1.37e-01
sedumi	-1.84797002e+03	-1.84797002e+03	1.94e-09	1.40e+01	3.03e-01	1.77e+01
sdpa	-1.84796998e+03	-1.84797002e+03	2.24e-08	6.46e+00	3.09e-01	1.44e-01
csdp	-1.84797001e+03	-1.84797002e+03	4.66e-09	9.02e+00	3.08e-01	1.83e+01
mcp500-4 equations: 500 variables: 125250						
sdpt3	-3.56673799e+03	-3.56673805e+03	1.75e-08	2.82e+00	3.14e-01	2.42e-01
sedumi	-3.56673804e+03	-3.56673805e+03	2.98e-09	1.32e+01	3.06e-01	1.66e+01
sdpa	-3.56673801e+03	-3.56673805e+03	1.16e-08	6.85e+00	2.97e-01	1.95e-01
csdp	-3.56673803e+03	-3.56673805e+03	6.73e-09	8.49e+00	3.00e-01	1.78e+01
qap5 equations: 136 variables: 351						
sdpt3	Inf	4.36000000e+02	NaN	5.10e-01	3.51e+00	3.57e-03
sedumi	Inf	4.36000000e+02	NaN	2.06e-01	1.25e+00	2.65e-03
sdpa	Inf	4.35999916e+02	NaN	1.75e-01	1.63e+00	3.38e-03
csdp	Inf	4.36000000e+02	NaN	8.81e-02	5.97e-01	1.02e-01
qap6 equations: 229 variables: 703						
sdpt3	Inf	3.81416019e+02	NaN	1.40e+00	3.84e+00	5.16e-03
sedumi	Inf	3.81421123e+02	NaN	6.05e-01	7.00e-01	4.08e-03
sdpa	Inf	3.81425579e+02	NaN	2.75e-01	1.96e+00	3.64e-03
csdp	Inf	3.81432762e+02	NaN	1.48e-01	1.26e+00	1.68e-01
qap7 equations: 358 variables: 1275						
sdpt3	Inf	4.24804016e+02	NaN	2.27e+00	5.99e+00	2.89e-03
sedumi	Inf	4.24806105e+02	NaN	8.23e-01	1.83e+00	2.38e-03
sdpa	Inf	4.24811292e+02	NaN	3.72e-01	3.10e+00	3.97e-03
csdp	Inf	4.24817776e+02	NaN	4.09e-01	3.53e+00	4.46e-01
qap8 equations: 529 variables: 2145						
sdpt3	Inf	7.56898325e+02	NaN	1.22e+00	1.83e+00	3.78e-03
sedumi	Inf	7.56926102e+02	NaN	1.82e+00	3.03e+00	3.03e-03
sdpa	Inf	7.56944360e+02	NaN	8.46e-01	5.60e+00	3.56e-03
csdp	Inf	7.56938172e+02	NaN	1.45e+00	8.04e+00	2.74e-03
qap9 equations: 748 variables: 3403						
sdpt3	Inf	1.40992991e+03	NaN	1.74e+00	3.77e+00	6.05e-03
sedumi	Inf	1.40991041e+03	NaN	2.65e+00	2.75e+00	3.46e-03
sdpa	Inf	1.40993182e+03	NaN	1.72e+00	7.85e+00	4.06e-03
csdp	Inf	1.40993834e+03	NaN	2.31e+00	1.68e+01	7.29e+00
qap10 equations: 1021 variables: 5151						
sdpt3	Inf	1.09257367e+03	NaN	3.02e+00	5.99e+00	5.22e-03
sedumi	Inf	1.09257230e+03	NaN	5.16e+00	5.53e+00	6.61e-03
sdpa	Inf	1.09259410e+03	NaN	3.08e+00	1.27e+01	4.18e-03
csdp	Inf	1.09259851e+03	NaN	4.34e+00	2.80e+01	8.37e+00
qpg11 equations: 800 variables: 1280800						
sdpt3	-2.44865909e+03	-2.44865913e+03	1.69e-08	6.93e+00	1.72e+00	1.30e-01
sedumi	-2.44865911e+03	-2.44865913e+03	1.05e-08	3.81e+02	4.65e+02	4.60e+02
sdpa	-2.44865905e+03	-2.44865913e+03	3.40e-08	1.23e+02	1.33e+00	1.28e-01
csdp	-2.44865910e+03	-2.44865913e+03	1.45e-08	8.05e+01	1.30e+00	8.26e+01
qpg51 equations: 1000 variables: 2001000						
sdpt3	-1.18179999e+04	-1.18180000e+04	4.41e-09	1.76e+01	2.31e+01	1.89e+00
sedumi	-1.18180000e+04	-1.18180000e+04	2.70e-09	1.10e+03	1.29e+03	1.34e+03
sdpa	-1.18179996e+04	-1.18180000e+04	3.70e-08	2.56e+02	2.50e+00	1.98e+00
csdp	-1.18179999e+04	-1.18180000e+04	1.07e-08	1.87e+02	2.83e+00	1.98e+02
ss30 equations: 132 variables: 43497						
sdpt3	-2.02395010e+01	-2.02395106e+01	4.76e-07	9.24e+00	1.20e+01	9.67e-03
sedumi	-2.02375061e+01	-2.02395106e+01	9.90e-05	1.05e+01	2.50e+01	3.75e+01
sdpa	-2.02395019e+01	-2.02395106e+01	4.31e-07	5.32e+00	1.17e+01	9.99e-03
csdp	-2.02395040e+01	-2.02395107e+01	3.32e-07	6.40e+00	6.30e+00	1.91e+01

Solver	fU	fL	$\mu(fU, fL)$	t_s	t_u	t_l
theta1 equations: 104 variables: 1275						
sdpt3	-2.29999997e+01	-2.30000001e+01	1.68e-08	7.07e-01	1.49e-02	1.99e-03
sedumi	-2.29999971e+01	-2.30000000e+01	1.26e-07	2.12e-01	2.09e-01	1.91e-01
sdpa	-2.29999991e+01	-2.30000003e+01	5.29e-08	5.62e-01	2.23e-02	2.99e-03
csdp	-2.29999997e+01	-2.30000000e+01	1.56e-08	9.78e-02	1.23e-02	1.95e-01
theta2 equations: 498 variables: 5050						
sdpt3	-3.28791689e+01	-3.28791690e+01	2.89e-09	5.04e-01	7.89e-02	3.31e-03
sedumi	-3.28791621e+01	-3.28791690e+01	2.11e-07	8.02e-01	1.12e+00	3.27e-03
sdpa	-3.28791686e+01	-3.28791691e+01	1.45e-08	1.06e+00	9.94e-02	4.02e-03
csdp	-3.28791689e+01	-3.28791690e+01	3.77e-09	1.46e+00	1.18e-01	2.66e+00
theta3 equations: 1106 variables: 11325						
sdpt3	-4.21669813e+01	-4.21669815e+01	5.27e-09	1.51e+00	4.09e-01	4.08e-03
sedumi	-4.21669492e+01	-4.21669815e+01	7.67e-07	3.60e+00	4.63e+00	3.92e-03
sdpa	-4.21669809e+01	-4.21669816e+01	1.59e-08	4.39e+00	4.11e-01	3.69e-03
csdp	-4.21669812e+01	-4.21669815e+01	8.19e-09	4.13e+00	4.02e-01	8.25e+00
theta4 equations: 1949 variables: 20100						
sdpt3	-5.03212213e+01	-5.03212220e+01	1.42e-08	4.28e+00	1.77e+00	5.49e-03
sedumi	-5.03211986e+01	-5.03212220e+01	4.63e-07	1.98e+01	4.76e+01	5.84e-03
sdpa	-5.03212209e+01	-5.03212221e+01	2.25e-08	2.12e+01	1.71e+00	5.55e-03
csdp	-5.03212217e+01	-5.03212220e+01	4.31e-09	1.04e+01	1.72e+00	2.10e+01
theta5 equations: 3028 variables: 31375						
sdpt3	-5.72323069e+01	-5.72323073e+01	7.06e-09	9.80e+00	5.54e+00	7.83e-03
sedumi	-5.72323070e+01	-5.72323073e+01	6.60e-09	7.33e+01	5.52e+00	7.81e-03
sdpa	-5.72323061e+01	-5.72323075e+01	2.48e-08	8.51e+01	5.54e+00	7.76e-03
csdp	-5.72323068e+01	-5.72323073e+01	9.43e-09	2.53e+01	5.77e+00	5.14e+01
theta6 equations: 4375 variables: 45150						
sdpt3	-6.34770870e+01	-6.34770872e+01	3.11e-09	2.50e+01	1.55e+01	1.07e-02
sedumi	-6.34770863e+01	-6.34770872e+01	1.57e-08	2.24e+02	1.56e+01	1.06e-02
sdpa	-6.34770859e+01	-6.34770876e+01	2.55e-08	2.44e+02	1.60e+01	1.06e-02
csdp	-6.34770865e+01	-6.34770873e+01	1.22e-08	6.45e+01	1.56e+01	1.57e+02
thetagl1 equations: 2401 variables: 321201						
sdpt3	-3.99999995e+02	-4.00000000e+02	1.27e-08	2.11e+01	3.77e+00	5.51e-02
sedumi	Inf	-6.10085467e+04	NaN	8.87e+01	1.83e+02	2.58e+02
sdpa	-3.99999991e+02	-4.00000000e+02	2.31e-08	9.82e+01	3.75e+00	5.47e-02
csdp	-3.99999998e+02	-4.00000000e+02	4.91e-09	4.41e+01	3.79e+00	8.73e+01
thetags1 equations: 6910 variables: 501501						
sdpt3	-3.48934237e+02	-3.49000003e+02	1.88e-04	4.51e+02	1.59e+03	2.52e+00
sedumi	-3.48996886e+02	-3.49000007e+02	8.94e-06	1.40e+03	3.90e+03	4.74e+03
sdpa	-3.48991260e+02	-3.49000047e+02	2.52e-05	1.87e+03	2.34e+03	3.81e+00
csdp	-3.48999993e+02	-3.49000010e+02	4.86e-08	7.11e+02	5.98e+01	2.83e+03
truss1 equations: 6 variables: 19						
sdpt3	8.99999651e+00	8.99999629e+00	2.54e-08	2.26e-01	1.15e-02	3.33e-03
sedumi	8.99999637e+00	8.99999630e+00	7.92e-09	1.94e-01	3.29e-01	3.30e-01
sdpa	8.99999651e+00	8.99999626e+00	2.78e-08	3.17e-01	9.62e-03	3.36e-03
csdp	8.99999632e+00	8.99999631e+00	9.32e-10	1.87e-02	9.46e-03	3.74e-03
truss2 equations: 58 variables: 331						
sdpt3	1.23380357e+02	1.23380356e+02	9.58e-09	3.52e-01	1.36e+00	1.52e-02
sedumi	1.23380357e+02	1.23380356e+02	5.04e-09	9.12e-01	2.83e+00	2.66e+00
sdpa	1.23380360e+02	1.23380356e+02	3.71e-08	2.25e-01	3.32e-02	1.41e-02
csdp	1.23380356e+02	1.23380356e+02	1.18e-09	1.99e-01	4.64e-02	2.75e-02
truss3 equations: 27 variables: 91						
sdpt3	9.10999627e+00	9.10999613e+00	1.51e-08	2.60e-01	1.23e-02	3.49e-03
sedumi	9.10999624e+00	9.10999618e+00	5.93e-09	1.78e-01	4.79e-01	1.15e+00
sdpa	9.10999639e+00	9.10999602e+00	4.14e-08	2.63e-01	1.19e-02	4.15e-03
csdp	9.10999622e+00	9.10999620e+00	1.53e-09	5.70e-02	1.22e-02	2.34e-01
truss4 equations: 12 variables: 37						
sdpt3	9.00999645e+00	9.00999629e+00	1.77e-08	2.86e-01	1.77e-02	6.41e-03
sedumi	9.00999636e+00	9.00999625e+00	1.24e-08	2.95e-01	6.17e-01	3.00e-01
sdpa	9.00999656e+00	9.00999606e+00	5.49e-08	2.40e-01	1.98e-02	4.17e-03
csdp	9.00999630e+00	9.00999629e+00	9.85e-10	2.45e-02	1.72e-02	5.01e-02
truss5 equations: 208 variables: 1816						
sdpt3	1.32635678e+02	1.32635678e+02	1.39e-09	5.64e-01	6.17e+00	1.56e-02
sedumi	1.32635678e+02	1.32635677e+02	6.33e-09	1.10e+00	2.36e+00	6.71e+00
sdpa	1.32635679e+02	1.32635675e+02	3.34e-08	6.08e-01	5.74e-02	1.52e-02
csdp	1.32635678e+02	1.32635678e+02	1.80e-09	7.28e-01	5.09e-02	1.53e+00

Solver	fU	fL	$\mu(fU, fL)$	t_s	t_u	t_l
truss6 equations: 172 variables: 901						
sdpt3	9.01002700e+02	9.01000997e+02	1.89e-06	1.08e+00	9.15e+00	7.14e-02
sedumi	9.01001411e+02	9.01001389e+02	2.43e-08	1.32e+01	1.19e+01	7.37e-02
sdpa	Inf	1.66669318e+02	NaN	8.78e-01	8.73e+00	8.84e-02
csdp	Inf	9.01001394e+02	NaN	1.85e+00	2.20e+01	2.02e+00
truss7 equations: 86 variables: 451						
sdpt3	9.00005565e+02	9.00001296e+02	4.74e-06	1.18e+00	3.26e+00	7.47e-02
sedumi	9.00001445e+02	9.00001384e+02	6.75e-08	1.99e+01	7.92e+01	7.27e-02
sdpa	9.00019739e+02	9.00001372e+02	2.04e-05	5.75e-01	1.69e+00	1.89e+00
csdp	9.00001590e+02	9.00001400e+02	2.11e-07	9.12e-01	4.43e+00	7.50e-02
truss8 equations: 496 variables: 6271						
sdpt3	Inf	1.33114589e+02	NaN	1.74e+00	2.17e+01	6.51e-02
sedumi	1.33114589e+02	1.33114589e+02	5.47e-09	2.61e+00	1.96e+01	1.36e+01
sdpa	1.33114595e+02	1.33114566e+02	2.12e-07	2.13e+00	1.30e-01	1.15e+01
csdp	1.33114589e+02	-Inf	NaN	3.13e+00	1.41e-01	3.18e+01

Table 6.2: Rigorous error bounds for the NETLIB LP library

Solver	fU	fL	$\mu(fU, fL)$	t_s	t_u	t_l
25fv47 equations: 821 variables: 1876						
sdpt3	Inf	-Inf	NaN	1.53e+00	9.50e-01	2.29e+00
sedumi	Inf	-Inf	NaN	1.64e+00	9.15e-02	6.03e+00
linprog	Inf	-Inf	NaN	9.83e-01	7.84e-02	2.76e+00
lpsolve	Inf	-Inf	NaN	5.84e-01	7.13e-02	4.89e+01
80bau3b equations: 9799 variables: 15047						
sdpt3	Inf	-Inf	NaN	1.18e+01	5.13e+00	1.62e+01
sedumi	Inf	-Inf	NaN	6.86e+00	1.22e+02	1.24e+02
linprog	Inf	-Inf	NaN	9.81e+01	1.54e+02	7.95e-01
lpsolve	Inf	-Inf	NaN	1.07e+01	1.19e+02	3.84e+01
adlittle equations: 56 variables: 138						
sdpt3	Inf	2.25494963e+05	NaN	2.43e-01	3.00e-01	2.26e-02
sedumi	Inf	2.25494963e+05	NaN	5.94e-02	5.92e-01	8.30e-02
linprog	Inf	2.25494963e+05	NaN	1.43e-02	2.97e-01	3.24e-02
lpsolve	Inf	2.25494963e+05	NaN	3.37e-03	3.19e-01	4.26e-02
afiro equations: 27 variables: 51						
sdpt3	-4.64753142e+02	-4.64753159e+02	3.73e-08	9.26e-02	8.87e-03	2.10e-01
sedumi	-4.64740663e+02	-4.64753233e+02	2.70e-05	3.12e-02	1.95e-01	3.93e-01
linprog	-4.64753142e+02	-4.64753144e+02	3.68e-09	8.69e-03	7.63e-03	3.13e-03
lpsolve	-4.64418145e+02	-4.64753147e+02	7.21e-04	1.46e-03	2.32e-01	3.30e-02
agg equations: 488 variables: 615						
sdpt3	Inf	-3.59917673e+07	NaN	4.48e-01	8.83e-01	2.56e-02
sedumi	Inf	-3.59917680e+07	NaN	3.18e-01	4.95e-01	1.11e+00
linprog	Inf	-3.59917673e+07	NaN	7.68e-02	5.56e-01	2.70e-02
lpsolve	Inf	-3.59917680e+07	NaN	2.35e-02	5.41e-01	1.30e-01
agg2 equations: 516 variables: 758						
sdpt3	Inf	-2.02392524e+07	NaN	3.79e-01	8.77e-01	3.57e-02
sedumi	Inf	-2.02392524e+07	NaN	2.43e-01	8.97e-01	6.19e-01
linprog	Inf	-2.02392524e+07	NaN	9.23e-02	9.24e-01	3.55e-02
lpsolve	Inf	-2.02392524e+07	NaN	3.24e-02	4.64e-01	1.64e-01
agg3 equations: 516 variables: 758						
sdpt3	Inf	1.03121159e+07	NaN	3.50e-01	4.86e-01	3.48e-02
sedumi	Inf	1.03121159e+07	NaN	3.04e-01	4.91e-01	3.82e-01
linprog	Inf	1.03121159e+07	NaN	9.27e-02	5.12e-01	1.62e-01
lpsolve	Inf	1.03121159e+07	NaN	3.21e-02	5.29e-01	1.01e-01
bandm equations: 305 variables: 472						
sdpt3	Inf	-1.58628019e+02	NaN	2.96e-01	2.56e-01	2.84e-02
sedumi	Inf	-1.58628018e+02	NaN	1.19e-01	3.21e-01	4.93e-01
linprog	Inf	-1.58628019e+02	NaN	3.67e-02	2.61e-01	8.92e-02
lpsolve	Inf	-1.58631002e+02	NaN	3.35e-02	3.15e-01	4.05e+00
beaconfd equations: 173 variables: 295						
sdpt3	Inf	-Inf	NaN	1.56e-01	1.82e-01	1.80e+00
sedumi	Inf	-Inf	NaN	1.35e-01	1.74e-01	2.01e+00
linprog	Inf	-Inf	NaN	2.74e-02	1.82e-01	9.66e-01
lpsolve	Inf	-Inf	NaN	9.85e-03	1.84e-01	2.83e-02
blend equations: 74 variables: 114						
sdpt3	-3.08121496e+01	-3.08121502e+01	1.80e-08	1.08e-01	1.19e-02	7.10e-03
sedumi	-3.08114297e+01	-3.08121498e+01	2.34e-05	5.73e-02	2.30e-01	7.28e-02
linprog	-3.03867641e+01	-3.08121498e+01	1.39e-02	1.64e-02	2.74e-01	6.45e-03
lpsolve	-3.08114661e+01	-3.08121501e+01	2.22e-05	3.65e-03	2.39e-01	4.93e-02
bn11 equations: 643 variables: 1586						
sdpt3	Inf	-Inf	NaN	6.47e-02	6.92e-02	1.48e+00
sedumi	Inf	1.97762934e+03	NaN	5.24e-01	5.28e-02	8.99e-02
linprog	Inf	1.97762956e+03	NaN	8.66e-02	5.11e-02	8.60e-02
lpsolve	Inf	1.97762956e+03	NaN	1.03e-01	4.48e-02	7.04e-01

Solver	fU	fL	$\mu(fU, fL)$	t_s	t_u	t_l
bnl2 equations: 2324 variables: 4486						
sdpt3	Inf	-Inf	NaN	2.81e+00	5.18e-01	9.88e-01
sedumi	Inf	-Inf	NaN	1.45e+00	4.25e+00	1.13e+01
linprog	Inf	-Inf	NaN	7.78e-01	1.80e+00	1.98e+00
lpsolve	Inf	-Inf	NaN	8.93e-01	3.33e+00	1.48e+00
boeing1 equations: 384 variables: 980						
sdpt3	6.66766444e+04	-Inf	NaN	5.54e-01	1.24e+00	5.90e-01
sedumi	3.44899308e+02	-Inf	NaN	4.37e-01	1.68e+00	7.26e+00
linprog	3.35225009e+02	-Inf	NaN	6.30e-01	1.64e+00	6.63e-02
lpsolve	3.35582522e+02	-Inf	NaN	4.98e-02	1.99e+00	1.30e+00
boeing2 equations: 143 variables: 382						
sdpt3	1.07044726e+06	-Inf	NaN	2.31e-01	3.01e-01	1.80e-01
sedumi	3.15018751e+02	-Inf	NaN	2.47e-01	1.86e-02	2.36e+00
linprog	3.15019252e+02	-Inf	NaN	6.11e-02	2.97e-01	2.66e-02
lpsolve	3.15018729e+02	-Inf	NaN	8.65e-03	2.89e-01	3.07e-02
bore3d equations: 315 variables: 559						
sdpt3	-1.16859273e+03	-Inf	NaN	1.26e+00	4.15e-01	1.87e-02
sedumi	-1.37308039e+03	-Inf	NaN	4.14e-01	4.49e-01	1.92e-02
linprog	-1.37307982e+03	-Inf	NaN	5.36e-02	1.30e+00	1.84e-02
lpsolve	-1.37308031e+03	-Inf	NaN	2.97e-02	1.28e+00	1.59e-02
brandy equations: 220 variables: 303						
sdpt3	Inf	-0.00000000e+00	NaN	3.91e+00	2.50e-02	1.69e-02
sedumi	Inf	-Inf	NaN	1.45e-01	2.14e-02	2.34e+00
linprog	Inf	-Inf	NaN	3.54e-02	2.26e-02	7.67e-01
lpsolve	Inf	-Inf	NaN	1.69e-02	2.01e-02	3.80e-02
capri equations: 353 variables: 741						
sdpt3	-2.69001282e+03	-2.69163881e+03	6.04e-04	8.00e-01	6.68e-01	9.79e-01
sedumi	-2.69001283e+03	-2.69167096e+03	6.16e-04	2.70e-01	3.45e-01	5.01e-01
linprog	-2.69001260e+03	-2.69158994e+03	5.86e-04	8.00e-02	3.51e-01	1.93e-01
lpsolve	-2.69001289e+03	-2.69163689e+03	6.04e-04	4.79e-02	3.53e-01	3.81e-01
cycle equations: 2857 variables: 4830						
sdpt3	Inf	-Inf	NaN	2.07e+00	1.03e+00	1.74e-02
sedumi	Inf	-Inf	NaN	1.98e+00	1.43e+00	1.35e-01
linprog	Inf	-Inf	NaN	1.34e+01	1.15e+00	1.49e-01
lpsolve	Inf	-Inf	NaN	1.23e+00	3.59e+00	9.59e-02
czprob equations: 1158 variables: 3562						
sdpt3	Inf	2.18519368e+06	NaN	1.07e+00	4.52e+00	3.32e-01
sedumi	Inf	2.18519670e+06	NaN	5.64e-01	2.90e+00	1.21e+00
linprog	Inf	2.18519670e+06	NaN	1.91e-01	3.37e+00	1.34e+00
lpsolve	Inf	2.18519670e+06	NaN	5.36e-01	4.86e+00	1.11e+00
d2q06c equations: 2171 variables: 5831						
sdpt3	Inf	-Inf	NaN	3.49e+00	9.95e-01	3.67e+00
sedumi	Inf	-Inf	NaN	2.42e+00	1.07e+00	1.07e+01
linprog	Inf	-Inf	NaN	1.84e+00	1.10e+00	3.39e+00
lpsolve	Inf	-Inf	NaN	3.71e+00	2.12e+00	5.78e+01
d6cube equations: 6184 variables: 6599						
sdpt3	-3.15489540e+02	-Inf	NaN	9.82e+01	3.76e+01	1.44e-01
sedumi	-3.15489942e+02	-Inf	NaN	3.78e+01	5.30e-01	1.47e-01
linprog	-3.15489307e+02	-Inf	NaN	6.26e+01	3.63e+01	1.56e-01
lpsolve	-3.15489549e+02	-Inf	NaN	1.06e+03	3.65e+01	1.29e-01
degen2 equations: 444 variables: 757						
sdpt3	Inf	-1.43517800e+03	NaN	3.08e-01	4.41e-02	4.10e-02
sedumi	Inf	-1.43517800e+03	NaN	1.27e-01	3.09e-02	4.18e-01
linprog	Inf	-1.43517800e+03	NaN	6.13e-02	3.99e-02	4.10e-02
lpsolve	Inf	-Inf	NaN	8.63e-02	2.63e-02	6.46e-01
degen3 equations: 1503 variables: 2604						
sdpt3	Inf	-9.87311019e+02	NaN	1.71e+00	1.58e-01	4.54e+00
sedumi	Inf	-9.87294000e+02	NaN	6.11e-01	1.46e-01	1.59e+00
linprog	Inf	-9.87294001e+02	NaN	7.31e-01	1.80e-01	1.40e-01

Solver	fU	fL	$\mu(fU, fL)$	t_s	t_u	t_l
df001 equations: 12230 variables: 18314						
sdpt3	-4.28188731e+06	-Inf	NaN	2.80e+02	2.98e+02	5.17e-01
sedumi	-1.10842629e+07	-Inf	NaN	4.68e+02	4.80e+02	5.17e-01
linprog	-1.12638667e+07	-Inf	NaN	1.49e+03	4.58e+03	4.95e-01
lpsolve	Inf	-Inf	NaN	6.50e+03	1.65e+04	3.41e-01
e226 equations: 223 variables: 472						
sdpt3	Inf	-Inf	NaN	6.56e-01	3.72e-01	4.21e-01
sedumi	Inf	-Inf	NaN	1.45e-01	2.22e-01	1.61e+00
linprog	Inf	-Inf	NaN	4.64e-02	8.10e-02	1.38e-01
lpsolve	Inf	-Inf	NaN	2.25e-02	3.76e-02	1.11e+00
etamacro equations: 688 variables: 1223						
sdpt3	4.82458742e+03	-Inf	NaN	2.10e+00	3.97e+00	2.08e-02
sedumi	7.56277517e+02	-Inf	NaN	4.73e-01	5.67e-02	1.99e-02
linprog	2.75779525e+03	-Inf	NaN	1.03e+00	8.42e-01	2.03e-02
lpsolve	7.55715292e+02	-Inf	NaN	8.58e-02	7.58e-01	1.94e-02
ffff800 equations: 524 variables: 1028						
sdpt3	Inf	1.00000000e+00	NaN	4.08e-01	4.56e-01	6.82e-02
sedumi	Inf	1.00000000e+00	NaN	2.00e-01	1.56e-01	5.31e-02
linprog	Inf	5.55677247e+05	NaN	1.06e-01	5.49e-02	2.02e-01
lpsolve	Inf	5.55679458e+05	NaN	6.34e-02	9.05e-02	6.61e-01
finnis equations: 614 variables: 1147						
sdpt3	Inf	-Inf	NaN	1.84e+00	8.09e-01	8.54e-01
sedumi	Inf	-Inf	NaN	7.65e-01	3.10e+00	7.13e+00
linprog	Inf	-Inf	NaN	1.20e-01	2.84e-01	3.43e-01
lpsolve	Inf	-Inf	NaN	7.56e-02	7.33e-01	2.64e-01
fit1d equations: 1026 variables: 2076						
sdpt3	9.14639025e+03	9.14525894e+03	1.24e-04	7.86e-01	9.98e-02	1.25e-01
sedumi	9.14637844e+03	9.14637809e+03	3.83e-08	6.17e-01	1.55e+00	8.46e-02
linprog	9.14637810e+03	9.14637496e+03	3.43e-07	2.26e-01	9.17e-02	3.84e-01
lpsolve	9.14637825e+03	9.14637624e+03	2.19e-07	2.02e-01	3.83e-01	9.83e-01
fit1p equations: 1677 variables: 2703						
sdpt3	-9.14637809e+03	-9.14637809e+03	2.34e-10	7.25e-01	9.15e-02	1.65e-01
sedumi	-9.14637356e+03	-9.14638651e+03	1.42e-06	7.38e-01	1.14e-01	1.08e+00
linprog	-9.14623589e+03	-9.14637809e+03	1.55e-05	1.71e-01	3.49e-01	5.53e-01
lpsolve	-9.14637746e+03	-9.14637810e+03	7.00e-08	5.50e-01	3.85e+02	1.15e+00
fit2d equations: 10500 variables: 21025						
sdpt3	6.84655987e+04	6.84642932e+04	1.91e-05	2.49e+00	4.05e+00	4.21e+00
sedumi	6.84642960e+04	6.84642847e+04	1.65e-07	5.17e+00	9.32e-01	1.76e+01
linprog	6.84642933e+04	6.84642932e+04	7.16e-10	1.38e+01	9.20e-01	8.12e-01
lpsolve	6.84642938e+04	6.84642409e+04	7.72e-07	2.42e+01	2.56e+01	1.59e+02
fit2p equations: 13525 variables: 24025						
sdpt3	-6.84642931e+04	-6.84642933e+04	3.80e-09	4.07e+00	5.56e+00	1.14e+00
sedumi	-6.84641164e+04	-6.84647361e+04	9.05e-06	2.48e+00	7.65e-01	4.82e+00
linprog	-6.84543674e+04	-6.84642933e+04	1.45e-04	4.29e+00	5.53e+00	1.14e+00
ganges equations: 1681 variables: 3387						
sdpt3	1.09585756e+05	-Inf	NaN	1.38e+00	5.37e+00	8.43e-01
sedumi	1.09816456e+05	-Inf	NaN	5.65e-01	1.03e+00	3.05e-01
linprog	1.09585739e+05	-Inf	NaN	7.78e-01	1.35e+00	7.22e-01
lpsolve	Inf	-Inf	NaN	3.20e+00	1.36e-04	8.33e-05
gfrd-pnc equations: 1092 variables: 1966						
sdpt3	-6.90218472e+06	-Inf	NaN	5.87e-01	2.64e+00	7.70e+00
sedumi	-6.90223568e+06	-Inf	NaN	4.72e-01	7.88e-02	8.86e+00
linprog	Inf	-Inf	NaN	5.65e-01	6.88e-01	1.64e+00
lpsolve	-6.90223600e+06	-Inf	NaN	1.75e-01	5.41e-01	3.98e-01
greenbea equations: 5405 variables: 8087						
sdpt3	Inf	-Inf	NaN	2.45e+01	3.50e+01	2.26e-01
sedumi	Inf	-Inf	NaN	1.45e+01	1.00e+01	2.27e-01
linprog	Inf	-Inf	NaN	2.94e+01	1.16e+01	2.72e-01
lpsolve	Inf	-Inf	NaN	7.03e+01	1.46e+02	1.52e-01
greenbeb equations: 5405 variables: 8084						
sdpt3	Inf	-Inf	NaN	2.91e+01	1.26e+01	2.65e-01
sedumi	Inf	-Inf	NaN	1.69e+01	1.44e+02	2.27e-01
linprog	Inf	-Inf	NaN	2.83e+01	3.32e+01	2.59e-01
lpsolve	Inf	-Inf	NaN	6.52e+01	7.21e+02	1.51e-01

Solver	fU	fL	$\mu(fU, fL)$	t_s	t_u	t_l
grow15 equations: 645 variables: 1545						
sdpt3	1.06870941e+08	1.06870941e+08	9.59e-10	4.59e-01	5.61e-02	9.98e-02
sedumi	1.06871404e+08	1.06869807e+08	1.49e-05	5.49e-01	5.46e-01	5.96e-01
linprog	Inf	1.06870936e+08	NaN	7.87e-01	1.47e+00	1.25e-01
lpsolve	1.06870942e+08	-Inf	NaN	1.04e-01	3.09e-01	2.35e+00
grow22 equations: 946 variables: 2266						
sdpt3	1.60834337e+08	1.60834336e+08	8.24e-09	5.08e-01	6.97e-02	1.54e-01
sedumi	1.60834481e+08	1.60833572e+08	5.65e-06	4.79e-01	6.03e-01	7.53e-01
linprog	1.60902195e+08	1.60834333e+08	4.22e-04	3.51e+00	3.43e+00	1.42e-01
lpsolve	1.60834337e+08	-Inf	NaN	1.75e-01	5.39e-01	7.13e+00
grow7 equations: 301 variables: 721						
sdpt3	4.77878119e+07	4.77878118e+07	1.52e-09	2.94e-01	2.80e-02	5.25e-02
sedumi	4.77878125e+07	4.77878071e+07	1.13e-07	1.42e-01	1.94e-01	2.42e-01
linprog	6.21351105e+10	4.56377707e+07	2.00e+00	2.46e-01	1.50e-01	4.92e-02
lpsolve	4.77878118e+07	-Inf	NaN	2.17e-02	1.63e-01	8.43e-01
israel equations: 142 variables: 316						
sdpt3	8.96644825e+05	8.96644819e+05	6.76e-09	2.97e-01	1.80e-02	1.60e-02
sedumi	8.96654011e+05	8.96644331e+05	1.08e-05	1.28e-01	1.77e-01	1.57e-01
linprog	8.96644825e+05	8.96644815e+05	1.10e-08	5.02e-02	2.81e-01	1.55e-02
lpsolve	8.96644825e+05	8.96644821e+05	4.85e-09	8.17e-03	6.34e-02	8.66e-02
kb2 equations: 41 variables: 93						
sdpt3	1.74990013e+03	1.74989929e+03	4.78e-07	2.20e-01	8.59e-03	2.36e-01
sedumi	1.74990155e+03	1.74989929e+03	1.29e-06	8.47e-02	1.04e-01	1.10e-01
linprog	4.87012456e+06	1.74807342e+03	2.00e+00	8.03e-02	7.53e-02	1.04e-01
lpsolve	1.74990013e+03	1.74989934e+03	4.53e-07	2.47e-03	1.65e-02	2.00e-02
maros-r7 equations: 3136 variables: 9408						
sdpt3	1.49718517e+06	1.49718517e+06	8.29e-10	4.08e+00	3.32e-01	4.95e-01
sedumi	1.49718517e+06	1.49718517e+06	2.06e-11	3.76e+00	4.39e+00	8.82e+00
linprog	1.49718517e+06	1.49718517e+06	1.02e-09	4.98e+00	3.13e-01	5.24e-01
lpsolve	1.49718517e+06	1.49718247e+06	1.80e-06	5.72e+00	1.23e+01	1.93e+01
maros equations: 1443 variables: 2289						
sdpt3	Inf	-Inf	NaN	2.25e+00	3.64e+00	3.51e-02
sedumi	Inf	-Inf	NaN	7.10e-01	6.15e-01	2.78e-02
linprog	Inf	-Inf	NaN	1.05e+00	2.02e+00	3.08e-02
lpsolve	Inf	-Inf	NaN	3.47e-01	4.52e-01	2.25e-02
modszk1 equations: 687 variables: 1620						
sdpt3	Inf	-0.00000000e+00	NaN	6.39e-02	5.68e-02	6.25e-02
sedumi	Inf	2.43995916e+02	NaN	4.31e-01	5.36e-02	6.24e-01
linprog	Inf	3.05141973e+02	NaN	7.03e-02	5.37e-02	4.07e-01
lpsolve	Inf	3.20615972e+02	NaN	4.62e-02	4.43e-02	2.23e-01
nesm equations: 2923 variables: 5181						
sdpt3	-1.40748735e+07	-1.40760535e+07	8.38e-05	4.64e+00	5.36e+00	2.65e-01
sedumi	-1.40736863e+07	-Inf	NaN	1.06e+01	1.09e+01	1.92e+01
linprog	-1.40760364e+07	-1.40760369e+07	3.51e-08	2.42e+01	2.60e-01	2.73e+01
lpsolve	Inf	-Inf	NaN	1.02e+01	1.42e-04	8.18e-05
perold equations: 1376 variables: 2179						
sdpt3	1.94140610e+04	-Inf	NaN	2.28e+00	8.06e+00	8.05e-01
sedumi	9.38982915e+03	-Inf	NaN	1.59e+00	1.08e-01	3.92e-01
linprog	Inf	-Inf	NaN	1.76e+00	3.55e+00	3.52e-01
lpsolve	Inf	-Inf	NaN	2.41e+03	1.77e-04	8.37e-05
pilot-ja equations: 1988 variables: 3179						
sdpt3	1.96752903e+05	-Inf	NaN	2.16e+01	1.56e-01	7.76e-02
sedumi	Inf	-Inf	NaN	2.07e+00	5.95e+00	8.81e-02
linprog	Inf	-Inf	NaN	9.54e+00	8.10e+00	8.49e-02
lpsolve	Inf	-Inf	NaN	2.22e+04	2.30e-04	9.17e-05
pilot-we equations: 2789 variables: 3725						
sdpt3	1.80284579e+11	-Inf	NaN	1.02e+01	1.24e+01	1.82e+00
sedumi	Inf	-Inf	NaN	8.99e+00	1.68e+01	5.41e+00
linprog	1.33427101e+10	-Inf	NaN	4.57e+00	1.46e+01	1.22e+00
lpsolve	2.72011490e+06	-Inf	NaN	1.24e+00	3.94e+03	8.72e+01
pilot equations: 3652 variables: 6133						
sdpt3	4.35768326e+03	-Inf	NaN	3.65e+01	4.74e+01	4.03e+01
sedumi	5.57922306e+02	-Inf	NaN	3.76e+01	3.18e-01	2.36e+01
linprog	Inf	-Inf	NaN	2.03e+01	1.57e+01	7.62e+00
lpsolve	5.57489946e+02	-Inf	NaN	1.16e+02	9.96e+02	4.27e+03

Solver	fU	fL	$\mu(fU, fL)$	t_s	t_u	t_l
pilot4 equations: 1000 variables: 1569						
sdpt3	2.06228909e+04	-Inf	NaN	1.21e+00	3.43e+00	4.86e-01
sedumi	2.58873935e+03	-Inf	NaN	2.39e+00	7.98e-02	3.80e-01
linprog	2.51190491e+11	-Inf	NaN	1.01e-01	4.96e+00	2.31e+00
lpsolve	Inf	-Inf	NaN	3.58e-01	3.60e+00	2.06e-01
pilot87 equations: 4883 variables: 8491						
sdpt3	Inf	-Inf	NaN	1.37e+02	6.47e+01	2.15e+01
sedumi	Inf	-Inf	NaN	3.34e+02	1.83e+03	2.73e+02
linprog	Inf	-Inf	NaN	3.66e+01	1.32e+01	1.57e+01
lpsolve	Inf	-Inf	NaN	8.05e+02	1.97e-04	8.92e-05
pilotnov equations: 2172 variables: 3487						
sdpt3	1.01761677e+07	-Inf	NaN	8.92e+00	1.70e-01	6.55e-02
sedumi	8.38270298e+04	-Inf	NaN	1.16e+00	2.53e+00	6.41e-02
linprog	4.33128534e+10	-Inf	NaN	7.38e+00	7.56e+00	7.07e-02
lpsolve	Inf	-Inf	NaN	3.14e+01	2.20e+01	6.34e-02
qap8 equations: 912 variables: 1632						
sdpt3	Inf	2.03500000e+02	NaN	7.77e-01	6.15e-01	1.01e-01
sedumi	Inf	2.03500000e+02	NaN	4.64e-01	1.37e-01	5.36e-01
linprog	Inf	2.03500000e+02	NaN	1.22e+00	1.08e-01	9.24e-01
qap12 equations: 3192 variables: 8856						
sdpt3	Inf	5.22894349e+02	NaN	2.23e+01	2.00e+00	4.92e-01
sedumi	Inf	5.22894278e+02	NaN	2.27e+01	1.83e+00	4.71e+01
linprog	Inf	5.22869098e+02	NaN	7.37e+01	2.05e+00	2.58e+02
qap15 equations: 6330 variables: 22275						
sdpt3	Inf	1.04099400e+03	NaN	1.55e+02	1.58e+01	6.22e+02
sedumi	Inf	1.04099392e+03	NaN	2.15e+02	1.55e+01	4.22e+02
linprog	Inf	1.04098646e+03	NaN	8.53e+02	1.53e+01	9.61e+02
recipelp equations: 180 variables: 340						
sdpt3	Inf	-Inf	NaN	9.62e-01	3.70e+00	1.79e-02
sedumi	Inf	-Inf	NaN	3.30e-01	1.15e+00	8.61e-03
linprog	Inf	-Inf	NaN	5.87e-01	9.03e-02	8.44e-03
lpsolve	Inf	-Inf	NaN	2.12e-02	6.40e-02	6.43e-03
sc105 equations: 105 variables: 163						
sdpt3	Inf	-5.22020620e+01	NaN	1.15e-01	2.71e-01	1.20e-01
sedumi	Inf	-5.22020612e+01	NaN	4.37e-02	2.73e-01	6.36e-02
linprog	Inf	-5.22021214e+01	NaN	1.21e-02	3.33e-02	2.74e-02
lpsolve	Inf	-Inf	NaN	3.83e-03	3.50e-02	1.06e-01
sc205 equations: 205 variables: 317						
sdpt3	Inf	-5.22020925e+01	NaN	1.32e-01	1.71e-01	1.81e-01
sedumi	Inf	-5.22198266e+01	NaN	6.66e-02	6.34e-02	1.17e-01
linprog	Inf	-5.22020615e+01	NaN	1.61e-02	5.74e-02	1.41e-02
lpsolve	Inf	-Inf	NaN	8.59e-03	2.00e-02	2.22e-01
sc50a equations: 50 variables: 78						
sdpt3	Inf	-6.45750771e+01	NaN	8.54e-02	7.93e-01	4.23e-03
sedumi	Inf	-6.45750771e+01	NaN	4.72e-02	2.15e-01	1.10e-01
linprog	Inf	-6.45750771e+01	NaN	1.00e-02	1.00e-02	6.06e-03
lpsolve	Inf	-6.45750803e+01	NaN	2.21e-03	8.51e-02	4.98e-02
sc50b equations: 50 variables: 78						
sdpt3	Inf	-7.00000002e+01	NaN	8.84e-02	7.13e-01	9.45e-02
sedumi	Inf	-7.00000000e+01	NaN	3.04e-02	1.75e-01	8.32e-02
linprog	Inf	-7.00000000e+01	NaN	9.46e-03	1.12e-02	4.70e-03
lpsolve	Inf	-7.00000000e+01	NaN	2.04e-03	8.65e-02	3.88e-02
scagr25 equations: 471 variables: 671						
sdpt3	-1.47534330e+07	-1.47534332e+07	1.13e-08	2.86e-01	3.58e-01	5.53e-02
sedumi	-1.47475618e+07	-1.47534331e+07	3.98e-04	1.25e-01	5.17e-01	1.80e-01
linprog	-1.47534330e+07	-1.47534331e+07	8.20e-10	2.50e-02	9.32e-02	3.17e-02
lpsolve	-1.47534330e+07	-Inf	NaN	3.68e-02	9.31e-02	1.79e-01
scagr7 equations: 129 variables: 185						
sdpt3	-2.33138982e+06	-2.33138983e+06	4.79e-09	2.15e-01	1.90e-01	9.93e-03
sedumi	-2.33138982e+06	-2.33138982e+06	5.60e-10	8.92e-02	1.04e-01	2.08e-01
linprog	-2.33138982e+06	-2.33138982e+06	6.02e-10	1.42e-02	3.73e-02	9.30e-03
lpsolve	-2.33138982e+06	-2.33138982e+06	5.22e-10	4.84e-03	2.72e-02	4.17e-02

Solver	fU	fL	$\mu(fU, fL)$	t_s	t_u	t_l
scfxm1 equations: 330 variables: 600						
sdpt3	Inf	-Inf	NaN	7.43e-01	3.22e-01	3.41e+00
sedumi	Inf	-Inf	NaN	1.51e-01	3.31e-01	3.16e+00
linprog	Inf	-Inf	NaN	4.31e-02	1.20e-01	1.10e+00
lpsolve	Inf	-Inf	NaN	3.04e-02	5.45e-02	5.51e-02
scfxm2 equations: 660 variables: 1200						
sdpt3	Inf	-Inf	NaN	7.81e-01	2.98e-01	3.08e+00
sedumi	Inf	-Inf	NaN	2.52e-01	6.25e-01	4.77e+00
linprog	Inf	-Inf	NaN	6.29e-02	1.93e-01	1.36e+00
lpsolve	Inf	-Inf	NaN	9.14e-02	1.48e-01	1.63e-01
scfxm3 equations: 990 variables: 1800						
sdpt3	Inf	-Inf	NaN	8.35e-01	3.61e-01	3.08e+00
sedumi	Inf	-Inf	NaN	2.67e-01	7.85e-01	5.06e+00
linprog	Inf	-Inf	NaN	8.87e-02	2.59e-01	3.45e+00
lpsolve	Inf	-Inf	NaN	2.26e-01	2.69e-01	2.91e-01
scorpion equations: 388 variables: 466						
sdpt3	Inf	1.87812482e+03	NaN	2.12e-01	3.46e-02	2.66e-02
sedumi	Inf	1.87812482e+03	NaN	7.59e-02	2.57e-02	2.38e-01
linprog	Inf	1.87812482e+03	NaN	2.28e-02	2.86e-02	7.10e-02
lpsolve	Inf	1.87812482e+03	NaN	1.32e-02	2.25e-02	1.77e-01
scrs8 equations: 490 variables: 1275						
sdpt3	Inf	-Inf	NaN	4.94e-01	4.24e-01	4.87e-01
sedumi	Inf	-Inf	NaN	2.45e-01	2.91e-01	1.93e+00
linprog	Inf	-Inf	NaN	5.48e-02	4.51e-02	1.68e+00
lpsolve	Inf	-Inf	NaN	3.84e-02	8.26e-02	1.19e-01
scsd1 equations: 77 variables: 760						
sdpt3	8.66666683e+00	8.66666666e+00	1.87e-08	8.57e-02	1.33e-02	4.28e-02
sedumi	8.66669054e+00	8.66666667e+00	2.75e-06	5.62e-02	1.96e-01	2.46e-01
linprog	8.66666668e+00	8.66666667e+00	8.78e-10	1.56e-02	4.19e-02	4.32e-02
lpsolve	8.6666758e+00	8.66666667e+00	1.04e-07	7.30e-03	1.04e-01	3.37e-01
scsd6 equations: 147 variables: 1350						
sdpt3	5.05000016e+01	5.05000001e+01	3.04e-08	1.48e-01	1.91e-01	9.17e-02
sedumi	5.05442697e+01	5.04999995e+01	8.76e-04	6.53e-02	3.62e-01	1.50e+00
linprog	5.05000001e+01	5.05000001e+01	1.02e-09	2.24e-02	6.02e-02	7.78e-02
lpsolve	Inf	5.05000000e+01	NaN	2.37e-02	4.72e-01	1.04e+00
scsd8 equations: 397 variables: 2750						
sdpt3	9.05000101e+02	9.04999996e+02	1.15e-07	1.88e-01	3.22e-01	1.51e-01
sedumi	9.05137268e+02	9.05000000e+02	1.52e-04	9.49e-02	4.11e-01	6.86e-01
linprog	9.05000054e+02	9.05000000e+02	5.96e-08	3.86e-02	1.29e-01	1.71e-01
lpsolve	9.05000085e+02	9.04999998e+02	9.67e-08	1.25e-01	4.96e-01	1.72e+00
sctap1 equations: 300 variables: 660						
sdpt3	1.41225000e+03	1.41224999e+03	6.91e-09	2.10e-01	3.63e-02	3.16e-02
sedumi	1.41233484e+03	1.41224998e+03	6.01e-05	9.73e-02	3.50e-01	7.04e-01
linprog	1.41226058e+03	1.41225000e+03	7.49e-06	2.65e-02	2.85e-01	3.01e-02
lpsolve	1.41225001e+03	1.41224999e+03	1.43e-08	1.40e-02	1.82e-01	2.80e-01
sctap2 equations: 1090 variables: 2500						
sdpt3	1.72480718e+03	1.72480713e+03	2.99e-08	2.58e-01	8.41e-02	1.12e-01
sedumi	2.00650164e+03	1.72480708e+03	1.51e-01	9.81e-02	8.27e-01	1.57e+00
linprog	1.72480715e+03	1.72480714e+03	3.46e-09	6.89e-02	7.89e-02	1.16e-01
lpsolve	1.72480716e+03	1.72480701e+03	9.04e-08	1.15e-01	4.77e-01	1.43e+00
sctap3 equations: 1480 variables: 3340						
sdpt3	1.42400002e+03	1.42399999e+03	2.33e-08	3.52e-01	1.11e-01	1.55e-01
sedumi	2.73874749e+03	1.42399978e+03	6.32e-01	1.35e-01	1.06e+00	2.38e+00
linprog	1.42401487e+03	1.42400000e+03	1.04e-05	1.10e-01	1.25e+00	1.52e-01
lpsolve	1.42400031e+03	1.42399996e+03	2.46e-07	1.86e-01	1.04e+00	1.40e+00
seba equations: 1028 variables: 2057						
sdpt3	-1.57115887e+004	-Inf	NaN	6.23e-01	6.40e-01	6.84e+00
sedumi	-1.57115840e+004	-Inf	NaN	3.59e-01	5.10e-01	6.45e+00
linprog	-1.57115038e+004	-Inf	NaN	9.25e-02	2.23e-01	3.07e+00
lpsolve	-1.57115996e+004	-Inf	NaN	1.19e-01	6.32e-01	2.76e-01
share1b equations: 117 variables: 253						
sdpt3	-7.60856944e+04	-7.65893186e+04	6.60e-03	2.06e-01	2.32e-01	1.46e-02
sedumi	-7.61180988e+04	-7.65893186e+04	6.17e-03	1.48e-01	1.87e-01	3.68e-01
linprog	-7.60674745e+04	-7.65893186e+04	6.84e-03	2.52e-02	5.62e-02	5.29e-02
lpsolve	-7.60890377e+04	-7.65893524e+04	6.55e-03	8.27e-03	3.60e-02	6.05e-02

Solver	fU	fL	$\mu(fU, fL)$	t_s	t_u	t_l
share2b equations: 96 variables: 162						
sdpt3	-4.15730719e+02	-4.15732242e+02	3.66e-06	1.31e-01	1.44e-01	8.34e-03
sedumi	-4.15730640e+02	-4.15732241e+02	3.85e-06	8.57e-02	1.36e-01	2.39e-01
linprog	-4.15730847e+02	-4.15732241e+02	3.35e-06	1.56e-02	4.52e-02	7.98e-03
lpsolve	-4.15730798e+02	-Inf	NaN	4.65e-03	4.84e-02	2.64e-02
shell equations: 1775 variables: 2428						
sdpt3	-1.20882532e+09	-Inf	NaN	1.94e+00	2.03e+00	5.45e-02
sedumi	-1.20882534e+09	-Inf	NaN	3.65e-01	9.90e-01	6.00e-02
linprog	Inf	-Inf	NaN	2.40e+00	4.94e-01	7.32e-02
lpsolve	-1.20882535e+09	-Inf	NaN	3.92e-01	3.63e+00	6.51e-02
ship04l equations: 402 variables: 2166						
sdpt3	Inf	-0.00000000e+00	NaN	5.61e+00	4.05e-02	1.02e-01
sedumi	Inf	1.79332454e+06	NaN	1.07e-01	2.20e-02	6.26e-01
linprog	Inf	1.79332454e+06	NaN	3.08e-02	3.18e-02	2.90e-01
lpsolve	Inf	1.79332454e+06	NaN	4.75e-02	2.15e-02	6.60e-01
ship04s equations: 402 variables: 1506						
sdpt3	Inf	-0.00000000e+00	NaN	5.64e+00	3.24e-02	7.67e-02
sedumi	Inf	1.79871470e+06	NaN	1.18e-01	2.12e-02	8.71e-01
linprog	Inf	1.79871470e+06	NaN	2.58e-02	2.94e-02	2.05e-01
lpsolve	Inf	1.79871470e+06	NaN	3.29e-02	1.92e-02	5.72e-01
ship08l equations: 778 variables: 4363						
sdpt3	Inf	-0.00000000e+00	NaN	5.46e-02	7.54e-02	1.57e-01
sedumi	Inf	1.90905521e+06	NaN	2.13e-01	4.55e-02	7.34e-01
linprog	Inf	1.90905521e+06	NaN	7.74e-02	7.70e-02	6.15e-01
lpsolve	Inf	1.90905521e+06	NaN	1.35e-01	3.95e-02	1.48e+00
ship08s equations: 778 variables: 2467						
sdpt3	Inf	-0.00000000e+00	NaN	4.65e-02	5.35e-02	9.03e-02
sedumi	Inf	1.92009821e+06	NaN	1.21e-01	3.72e-02	4.10e-01
linprog	Inf	1.92009821e+06	NaN	3.58e-02	4.32e-02	3.16e-01
lpsolve	Inf	1.92009821e+06	NaN	7.14e-02	3.51e-02	5.82e-01
ship12l equations: 1151 variables: 5533						
sdpt3	Inf	-0.00000000e+00	NaN	7.40e-02	9.58e-02	2.00e-01
sedumi	Inf	1.47018792e+06	NaN	3.03e-01	5.67e-02	1.61e+00
linprog	Inf	1.47018792e+06	NaN	8.41e-02	8.60e-02	7.69e-01
lpsolve	Inf	1.47018792e+06	NaN	2.59e-01	5.24e-02	2.13e+00
ship12s equations: 1151 variables: 2869						
sdpt3	Inf	-0.00000000e+00	NaN	5.39e-02	6.67e-02	1.04e-01
sedumi	Inf	1.48923613e+06	NaN	1.75e-01	5.13e-02	5.38e-01
linprog	Inf	1.48923613e+06	NaN	3.98e-02	5.92e-02	4.08e-01
lpsolve	Inf	1.48923613e+06	NaN	1.35e-01	4.75e-02	7.74e-01
sierra equations: 2036 variables: 5279						
sdpt3	-1.53929040e+07	-Inf	NaN	2.12e+00	2.36e+00	3.57e-02
sedumi	-1.53943566e+07	-Inf	NaN	1.47e+00	2.20e-01	3.79e-02
linprog	Inf	-Inf	NaN	6.48e+00	1.08e+01	4.75e-02
lpsolve	-1.53943618e+07	-Inf	NaN	5.47e-01	1.50e+00	2.72e-02
stair equations: 467 variables: 823						
sdpt3	Inf	2.51266719e+02	NaN	8.28e-01	9.97e+00	1.07e+00
sedumi	Inf	2.51266833e+02	NaN	2.08e-01	4.42e+00	3.31e-01
linprog	Inf	-Inf	NaN	3.42e-01	1.69e-01	8.20e-01
lpsolve	Inf	2.51266945e+02	NaN	1.14e-01	6.12e-01	3.01e-01
standata equations: 1075 variables: 1538						
sdpt3	-1.25630495e+03	-Inf	NaN	1.09e+00	2.27e+00	1.93e+00
sedumi	-1.25769950e+03	-Inf	NaN	1.93e-01	3.81e-01	1.98e+00
linprog	Inf	-Inf	NaN	1.14e+00	2.34e-01	3.18e-01
lpsolve	Inf	-Inf	NaN	1.72e-01	2.21e-01	2.75e-01
standgub equations: 1184 variables: 1649						
sdpt3	Inf	-Inf	NaN	1.08e+00	7.33e-01	1.82e-02
sedumi	Inf	-Inf	NaN	2.56e-01	4.16e+00	1.74e-02
linprog	Inf	-Inf	NaN	2.78e-01	3.39e-01	1.70e-02
lpsolve	Inf	-Inf	NaN	1.84e-01	2.14e-01	1.79e-02
standmps equations: 1075 variables: 1646						
sdpt3	8.56476559e+05	-Inf	NaN	2.06e+00	1.31e+00	9.39e-01
sedumi	-1.40601750e+03	-Inf	NaN	3.74e-01	8.77e-02	2.06e+00
linprog	Inf	-Inf	NaN	1.20e+00	3.45e+00	4.44e-01
lpsolve	Inf	-Inf	NaN	2.05e-01	1.40e+02	3.39e-01

Solver	fU	fL	$\mu(fU, fL)$	t_s	t_u	t_l
stocfor1 equations: 117 variables: 165						
sdpt3	-4.11295915e+04	-4.11319764e+04	5.80e-05	1.14e-01	1.36e-01	1.20e-01
sedumi	-4.11292420e+04	-4.11319762e+04	6.65e-05	9.23e-02	1.27e-01	2.01e-01
linprog	-4.11296249e+04	-4.11319762e+04	5.72e-05	1.96e-02	4.77e-02	3.82e-02
lpsolve	-4.11292406e+04	-4.11319762e+04	6.65e-05	4.37e-03	2.67e-02	7.27e-02
stocfor2 equations: 2157 variables: 3045						
sdpt3	-3.90244069e+04	-3.90244086e+04	4.17e-08	4.08e-01	6.46e-01	1.36e-01
sedumi	-3.90243736e+04	-3.90244085e+04	8.94e-07	6.25e-01	1.10e+00	1.75e+00
linprog	Inf	-3.90244085e+04	NaN	1.63e-01	6.04e-01	4.38e-01
stocfor3 equations: 16675 variables: 23541						
sdpt3	-3.99767624e+04	-3.99767852e+04	5.70e-07	3.55e+00	5.51e+00	1.07e+00
sedumi	-3.99751046e+04	-3.99853396e+04	2.56e-04	7.15e+00	1.05e+00	1.46e+01
linprog	Inf	-3.99767840e+04	NaN	2.16e+00	9.98e+00	3.93e+00
truss equations: 1000 variables: 8806						
sdpt3	4.58816100e+05	4.58815847e+05	5.51e-07	7.46e-01	1.85e+00	4.49e-01
sedumi	4.58816729e+05	4.58815847e+05	1.92e-06	5.41e-01	2.10e+00	2.13e+00
linprog	4.58816391e+05	4.58815847e+05	1.18e-06	7.31e-01	5.54e-01	4.44e-01
lpsolve	4.58816050e+05	-Inf	NaN	1.85e+00	1.41e+01	2.53e+01
tuff equations: 587 variables: 944						
sdpt3	Inf	-Inf	NaN	3.09e-01	4.11e+00	2.39e-02
sedumi	4.38910588e+02	-Inf	NaN	7.45e-01	3.31e+00	2.32e-02
linprog	Inf	-Inf	NaN	1.49e-01	5.53e+00	3.26e-02
lpsolve	-2.91909428e-01	-Inf	NaN	1.08e-01	4.32e+01	1.66e-02
vtp-base equations: 203 variables: 465						
sdpt3	-7.05484826e+04	-Inf	NaN	4.11e-01	2.27e-02	7.57e-01
sedumi	-1.29829460e+05	-Inf	NaN	5.75e-01	1.15e+00	2.12e-01
linprog	-1.29829213e+05	-Inf	NaN	9.81e-02	1.33e-01	1.21e-01
lpsolve	-1.29828312e+05	-Inf	NaN	2.30e-02	1.23e-01	5.14e-02
woodlp equations: 244 variables: 2595						
sdpt3	Inf	1.44284190e+00	NaN	7.95e-01	1.39e-01	5.22e-02
sedumi	Inf	1.44290241e+00	NaN	3.60e-01	1.44e-01	1.28e+00
linprog	Inf	1.44290241e+00	NaN	3.20e-01	1.50e-01	4.19e-02
lpsolve	Inf	-Inf	NaN	9.84e-02	1.34e-01	2.55e+00
woodw equations: 1098 variables: 8418						
sdpt3	Inf	1.30393357e+00	NaN	8.34e-01	7.89e-01	5.09e-01
sedumi	Inf	1.30447633e+00	NaN	6.65e-01	2.58e-01	2.90e+00
linprog	Inf	1.30447633e+00	NaN	3.71e-01	1.19e+00	5.20e-01
lpsolve	Inf	-Inf	NaN	1.30e+00	9.26e-01	3.90e+01

Table 6.3: Rigorous error bounds for the DIMACS test suite

Solver	fU	fL	$\mu(fU, fL)$	t_s	t_u	t_l
nb equations: 123 variables: 2383						
sdpt3	-5.07030901e-02	-Inf	NaN	3.51e+00	3.55e+00	2.69e+01
sedumi	-5.07030946e-02	-Inf	NaN	7.10e-01	4.40e+00	1.11e+01
nbl1 equations: 915 variables: 3176						
sdpt3	-1.30122697e+01	-Inf	NaN	4.67e+00	3.53e+00	4.53e+01
sedumi	-1.30122700e+01	-Inf	NaN	2.79e+00	2.23e-01	3.56e+01
nbl2 equations: 123 variables: 4195						
sdpt3	-1.52319900e+00	-Inf	NaN	3.59e+00	2.32e+00	3.38e+01
sedumi	-1.54735962e+00	-Inf	NaN	9.70e-01	1.22e+00	8.99e-01
nbl2bessel equations: 123 variables: 2641						
sdpt3	-1.02569495e-01	-Inf	NaN	2.63e+00	9.16e-01	2.13e+01
sedumi	-1.02569498e-01	-Inf	NaN	7.46e-01	2.57e+00	9.24e+00
biomedp equations: 6515 variables: 42432196						
sdpt3	Inf	-8.61125822e+06	NaN	3.27e+03	1.06e+05	1.04e+03
sedumi	Inf	-1.76473106e+02	NaN	9.35e+04	2.49e+04	9.70e+04
sdpa	Inf	3.35735719e+01	NaN	3.14e+04	6.12e+04	1.70e+03
csdp	Inf	-Inf	NaN	9.61e+04	4.38e-01	7.01e-02
bm1 equations: 883 variables: 777924						
sdpt3	Inf	2.31115401e+01	NaN	1.16e+02	1.61e+02	3.48e+01
sedumi	Inf	-8.06015054e+00	NaN	1.39e+02	1.67e+02	3.54e+02
sdpa	Inf	2.34385900e+01	NaN	8.52e+01	1.72e+02	1.61e+01
csdp	Inf	-Inf	NaN	2.71e+02	1.53e-04	1.17e-04
copo14 equations: 1275 variables: 3108						
sdpt3	3.14543425e-09	-8.22602268e-10	3.97e-09	2.24e+00	3.95e+00	1.36e-01
sedumi	2.18114470e-09	-4.08543080e-09	6.27e-09	1.35e+00	4.12e+00	1.34e-01
sdpa	1.26527788e-08	-4.68096997e-08	5.95e-08	2.03e+00	8.19e-02	1.32e-01
csdp	2.04751347e-09	-4.38230803e-09	6.43e-09	4.40e+00	8.26e-02	4.45e+00
copo23 equations: 5820 variables: 13938						
sdpt3	1.72375824e-08	-5.34634911e-10	1.78e-08	4.11e+01	1.54e+02	5.69e-01
sedumi	1.48608536e-08	-1.88135782e-08	3.37e-08	7.84e+01	1.59e+02	6.30e-01
sdpa	2.99830014e-09	-1.15213843e-08	1.45e-08	1.01e+02	3.31e-01	6.45e-01
csdp	3.85051826e-09	-7.21557680e-09	1.11e-08	1.77e+02	3.49e-01	1.87e+02
filter48sopc equations: 969 variables: 3284						
sdpt3	Inf	1.41612865e+00	NaN	2.05e+01	1.02e+02	1.05e-01
sedumi	1.41613043e+00	1.41612873e+00	1.20e-06	2.30e+00	8.53e+00	2.85e+00
minphase equations: 48 variables: 2304						
sdpt3	Inf	5.98100315e+00	NaN	8.53e-01	1.40e+00	5.71e-02
sedumi	Inf	5.98087746e+00	NaN	6.26e-01	4.47e-01	3.87e-02
sdpa	Inf	5.97906292e+00	NaN	2.93e-01	1.06e+00	7.50e-02
csdp	Inf	5.98222919e+00	NaN	2.19e-01	1.78e+00	4.30e-01
hamming756 equations: 1793 variables: 16384						
sdpt3	-4.26666661e+01	-4.26666668e+01	1.69e-08	2.72e+00	7.43e-01	3.31e-01
sedumi	-1.00000004e+00	-4.26666667e+01	1.91e+00	5.80e+00	4.64e+00	3.31e-01
sdpa	-4.26666647e+01	-4.26666677e+01	7.22e-08	1.29e+01	8.59e-02	3.32e-01
csdp	-4.26666665e+01	-4.26666667e+01	3.83e-09	6.04e+00	9.07e-02	6.60e+00
hamming834 equations: 16129 variables: 65536						
sdpt3	-2.55999998e+01	-2.56000001e+01	9.92e-09	4.60e+02	7.35e-01	4.54e+00
sedumi	-1.00000000e+00	-2.56000002e+01	1.85e+00	4.41e+03	4.59e+03	1.55e+00
sdpa	-2.55999940e+01	-2.56000043e+01	4.02e-07	1.03e+04	7.80e-01	1.42e+00
csdp	-2.55999999e+01	-2.56000000e+01	5.10e-09	2.19e+03	6.90e-01	6.83e+03
hamming98 equations: 2305 variables: 262144						
sdpt3	-2.23999998e+02	-2.24000001e+02	1.22e-08	5.86e+00	3.84e-01	5.74e+00
sedumi	-1.00000000e+00	-2.24000000e+02	1.98e+00	2.20e+01	2.87e+01	5.46e+00
sdpa	-2.23999997e+02	-2.24000002e+02	2.40e-08	4.21e+01	3.93e-01	5.48e+00
csdp	-2.23999999e+02	-2.24000000e+02	4.53e-09	2.48e+01	3.82e-01	3.55e+01
hamming102 equations: 23041 variables: 1048576						
sdpt3	-1.02399999e+02	-1.02400000e+02	8.90e-09	1.32e+03	1.66e+01	2.07e+01
sdpa	-1.02399992e+02	-1.02400018e+02	2.50e-07	3.50e+04	6.12e+00	2.08e+01
csdp	-1.02400000e+02	-1.02400000e+02	2.60e-09	7.55e+03	6.01e+00	7.59e+03

Solver	fU	fL	$\mu(fU, fL)$	t_s	t_u	t_l
hinf12 equations: 43 variables: 216						
sdpt3	Inf	-6.52534931e-05	NaN	1.70e+00	1.25e+00	1.82e-02
sedumi	Inf	-1.02542956e-01	NaN	3.61e-01	3.41e-01	4.61e-03
sdpa	Inf	-3.21198653e+00	NaN	2.43e-01	1.92e-01	7.48e-03
csdp	Inf	-1.42234319e-01	NaN	1.03e-01	3.40e-01	9.38e-03
hinf13 equations: 57 variables: 326						
sdpt3	Inf	-4.43703741e+01	NaN	9.31e-01	2.32e-01	3.23e+00
sedumi	Inf	-4.67090598e+01	NaN	4.67e-01	6.01e-01	2.01e+00
sdpa	Inf	-4.49057517e+01	NaN	1.62e-01	2.72e-01	1.10e-02
csdp	Inf	-4.47883302e+01	NaN	1.60e-01	8.47e-01	2.73e-01
nql30 equations: 3601 variables: 8260						
sdpt3	9.48927666e-01	-Inf	NaN	3.28e+00	3.01e+01	2.52e+01
sedumi	9.47902691e-01	-Inf	NaN	1.51e+01	5.19e+01	1.24e+02
nql30new equations: 3680 variables: 6302						
sdpt3	Inf	-Inf	NaN	1.98e+00	6.72e-01	2.13e+01
sedumi	Inf	-Inf	NaN	1.05e+01	4.14e-01	1.50e+02
nql60 equations: 14401 variables: 32720						
sdpt3	2.49482341e+00	-Inf	NaN	1.24e+01	3.74e+01	3.69e+02
sedumi	1.92474133e+00	-Inf	NaN	6.04e+02	6.04e+02	4.30e+03
nql60new equations: 14560 variables: 25202						
sdpt3	Inf	-Inf	NaN	5.26e+00	1.93e+00	8.86e+01
sedumi	Inf	-Inf	NaN	4.11e+02	3.28e+00	4.25e+03
nql180 equations: 129601 variables: 292560						
sdpt3	Inf	-Inf	NaN	5.29e+02	1.02e+03	4.58e+03
nql180new equations: 130080 variables: 226802						
sdpt3	Inf	-Inf	NaN	4.70e+01	4.62e+01	9.50e+02
qssp30 equations: 5674 variables: 11164						
sdpt3	6.72793666e+00	-Inf	NaN	1.03e+01	8.14e+01	1.34e+02
sedumi	6.70808799e+00	-Inf	NaN	5.18e+01	6.25e+01	4.22e+02
qssp30new equations: 3691 variables: 7566						
sdpt3	-6.49666899e+00	-Inf	NaN	1.05e+00	2.17e+01	1.75e+01
sedumi	-6.49666253e+00	-Inf	NaN	2.13e+01	6.41e+01	2.89e+02
qssp60 equations: 22144 variables: 43924						
sdpt3	8.16687038e+00	-Inf	NaN	1.54e+02	5.69e+02	1.13e+03
sedumi	8.16695054e+00	-Inf	NaN	2.13e+03	1.85e+03	2.03e+03
qssp60new equations: 14581 variables: 29526						
sdpt3	-6.56269453e+00	-Inf	NaN	4.01e+00	7.01e+00	6.84e+01
sedumi	-6.56173100e+00	-Inf	NaN	1.24e+03	6.05e+03	1.57e+04
qssp180new equations: 130141 variables: 261366						
sdpt3	Inf	-Inf	NaN	5.05e+01	9.90e+01	6.48e+02
sched5050orig equations: 2527 variables: 4979						
sdpt3	Inf	2.66729886e+04	NaN	1.40e+00	8.83e-01	3.28e+00
sedumi	Inf	2.66730010e+04	NaN	5.99e+00	2.21e+00	1.88e+01
sched5050scaled equations: 2526 variables: 4977						
sdpt3	2.93202502e+01	7.85203844e+00	1.16e+00	1.12e+00	2.25e+00	1.97e+00
sedumi	2.90607910e+01	7.85203844e+00	1.15e+00	6.07e-01	1.48e+00	2.11e+00
sched100100orig equations: 8338 variables: 18240						
sdpt3	Inf	7.17364665e+05	NaN	4.74e+00	3.71e+00	1.18e+00
sedumi	Inf	7.17367782e+05	NaN	1.57e+02	3.06e+01	3.06e+02
sched100100scaled equations: 8337 variables: 18238						
sdpt3	Inf	2.73307849e+01	NaN	3.75e+00	2.23e+00	1.45e+00
sedumi	Inf	2.73307769e+01	NaN	3.31e+00	1.49e+00	1.00e+01
sched100500orig equations: 4844 variables: 9746						
sdpt3	Inf	1.81889214e+05	NaN	2.48e+00	1.42e+00	6.08e-01
sedumi	Inf	1.81889939e+05	NaN	1.97e+01	7.68e+00	2.16e+01
sched100500scaled equations: 4843 variables: 9744						
sdpt3	Inf	6.71650307e+01	NaN	2.20e+00	2.70e+00	7.71e-01
sedumi	Inf	6.71650173e+01	NaN	1.63e+00	1.04e+00	2.70e+00
sched200100orig equations: 18087 variables: 37889						
sdpt3	Inf	1.41359220e+05	NaN	1.32e+01	6.10e+00	2.41e+00
sedumi	Inf	-Inf	NaN	1.10e+03	2.08e+02	1.12e+03

Solver	fU	fL	$\mu(fU, fL)$	t_s	t_u	t_l
sched200100scaled				equations: 18086 variables: 37887		
sdpt3	Inf	5.18119603e+01	NaN	1.21e+01	5.01e+00	3.03e+00
sedumi	Inf	5.18118660e+01	NaN	1.13e+01	4.09e+00	2.94e+01
torusg3-15				equations: 3375 variables: 11390625		
sdpt3	-3.18810927e+08	-3.18810927e+08	5.92e-10	4.36e+02	4.72e+01	7.02e+01
sedumi	-3.18810924e+08	-3.18810928e+08	1.24e-08	4.74e+03	5.09e+01	5.15e+03
sdpa	-3.18810922e+08	-3.18810927e+08	1.65e-08	3.69e+03	3.93e+01	6.30e+01
csdp	-3.18810927e+08	-3.18810927e+08	1.24e-09	7.05e+02	4.00e+01	8.35e+02
torusg3-8				equations: 512 variables: 262144		
sdpt3	-4.83409458e+07	-4.83409459e+07	2.74e-09	3.27e+00	3.09e-01	8.71e-01
sedumi	-4.83409457e+07	-4.83409460e+07	5.36e-09	1.90e+01	2.16e+01	2.52e+01
sdpa	-4.83409453e+07	-4.83409460e+07	1.32e-08	1.12e+01	3.22e-01	8.75e-01
csdp	-4.83409456e+07	-4.83409460e+07	8.23e-09	9.35e+00	3.06e-01	2.21e+01
toruspm3-15-50				equations: 3375 variables: 11390625		
sdpt3	-3.47479406e+03	-3.47479408e+03	6.82e-09	3.24e+02	4.02e+01	5.71e+01
sedumi	-3.47479408e+03	-3.47479408e+03	2.11e-09	1.73e+04	8.14e+01	1.66e+04
sdpa	-3.47479389e+03	-3.47479408e+03	5.42e-08	2.24e+03	4.01e+01	5.79e+01
csdp	-3.47479408e+03	-3.47479408e+03	1.56e-09	6.75e+02	4.01e+01	7.86e+02
toruspm3-8-50				equations: 512 variables: 262144		
sdpt3	-5.27808661e+02	-5.27808663e+02	4.39e-09	3.10e+00	3.19e-01	8.73e-01
sedumi	-5.27808661e+02	-5.27808663e+02	3.34e-09	3.32e+01	3.42e+01	7.47e+01
sdpa	-5.27808635e+02	-5.27808664e+02	5.55e-08	7.05e+00	3.15e-01	8.66e-01
csdp	-5.27808661e+02	-5.27808663e+02	4.02e-09	9.61e+00	3.09e-01	1.12e+01
truss5				equations: 208 variables: 3301		
sdpt3	1.32635678e+02	1.32635678e+02	2.37e-09	5.19e-01	6.21e-01	1.52e-01
sedumi	1.32635679e+02	1.32635677e+02	8.42e-09	1.35e+00	1.13e+01	7.74e+00
csdp	1.32635678e+02	1.32635678e+02	1.80e-09	7.50e-01	7.28e-02	1.99e+00
truss8				equations: 496 variables: 11914		
sdpt3	1.33114592e+02	1.33114589e+02	2.07e-08	1.89e+00	5.76e+00	5.11e-01
sedumi	1.33114591e+02	-Inf	NaN	2.62e+00	2.68e+01	3.42e+01
sdpa	Inf	1.33114566e+02	NaN	2.28e+00	4.11e+00	1.41e+01
csdp	1.33114589e+02	-Inf	NaN	2.87e+00	3.44e+00	3.85e+01

Table 6.4: Rigorous error bounds for structural optimization problems.

Solver	fU	fL	$\mu(fU, fL)$	t_s	t_u	t_l
buck1 equations: 36 variables: 661						
sdpt3	-1.46419152e+02	-1.46419152e+02	1.39e-09	1.35e+00	1.09e+00	5.47e-03
sedumi	-1.46419150e+02	-1.46419153e+02	1.80e-08	1.03e+00	2.97e-01	1.16e+00
sdpa	-1.46419150e+02	-1.46419152e+02	1.05e-08	1.85e-01	1.30e-02	2.39e-03
csdp	-1.46419150e+02	-1.46419152e+02	1.39e-08	9.13e-02	1.95e-02	2.67e-01
buck2 equations: 144 variables: 9553						
sdpt3	-2.92368073e+02	-2.92368296e+02	7.62e-07	1.13e+00	2.28e+00	4.77e-03
sedumi	-2.92368288e+02	-2.92368297e+02	3.26e-08	2.57e+00	5.91e+00	5.80e+00
sdpa	-2.92367551e+02	-2.92368296e+02	2.55e-06	7.09e-01	1.67e+00	5.27e-03
csdp	-2.92368287e+02	-2.92368295e+02	2.79e-08	1.67e+00	1.67e+00	1.80e+00
buck3 equations: 544 variables: 103585						
sdpt3	-6.07590310e+02	-6.07607438e+02	2.82e-05	8.57e+00	9.12e+00	1.67e-02
sedumi	-6.07573101e+02	-6.07703046e+02	2.14e-04	6.11e+01	3.20e-01	1.68e+02
sdpa	-6.07322486e+02	-6.07611001e+02	4.75e-04	9.34e+00	9.80e+00	1.74e-02
csdp	-6.07568490e+02	-6.07606804e+02	6.31e-05	4.40e+01	3.46e+01	6.81e+01
buck4 equations: 1200 variables: 454129						
sdpt3	-4.86141647e+02	-4.86141983e+02	6.89e-07	4.94e+01	5.15e+01	4.88e+01
sedumi	Inf	-4.86142027e+02	NaN	4.05e+02	5.58e+03	1.01e+03
sdpa	-4.86131851e+02	-4.86142461e+02	2.18e-05	1.18e+02	2.26e+02	1.14e+02
csdp	-4.86139598e+02	-4.86142023e+02	4.99e-06	2.24e+02	5.46e+02	2.46e+02
buck5 equations: 3280 variables: 3104401						
sdpt3	-4.36193708e+02	-4.36263569e+02	1.60e-04	4.86e+02	4.97e+02	4.16e-01
sedumi	-4.36082406e+02	-4.36466403e+02	8.80e-04	1.23e+04	4.13e+04	4.30e+04
sdpa	-4.36105250e+02	-4.36262654e+02	3.61e-04	1.92e+03	1.93e+03	4.17e-01
csdp	-4.36235832e+02	-4.36240107e+02	9.80e-06	2.54e+03	5.14e+03	9.60e+03
mater-1 equations: 103 variables: 1322						
sdpt3	1.43465441e+02	1.43465438e+02	2.12e-08	4.18e-01	4.01e+00	1.08e-02
sedumi	1.43465440e+02	1.43465438e+02	1.34e-08	9.90e-01	5.34e+00	9.98e-03
csdp	1.43465440e+02	1.43465438e+02	1.70e-08	2.81e-01	3.13e-02	8.01e-01
mater-2 equations: 423 variables: 6074						
sdpt3	1.41591867e+02	1.41591866e+02	6.70e-09	1.56e+00	1.24e+01	4.54e-02
sedumi	1.41591887e+02	1.41591865e+02	1.56e-07	7.38e+00	7.38e+01	1.43e+01
csdp	1.41591867e+02	1.41591866e+02	1.63e-09	3.29e+00	1.61e-01	3.70e+00
mater-3 equations: 1439 variables: 21518						
sdpt3	Inf	1.33916256e+02	NaN	4.26e+00	5.97e+01	1.54e-01
sedumi	1.33920907e+02	1.33916241e+02	3.48e-05	5.64e+01	3.45e+02	4.54e+01
csdp	1.33916257e+02	1.33916256e+02	7.56e-09	3.32e+01	1.04e+02	3.37e+01
mater-4 equations: 4807 variables: 74978						
sdpt3	Inf	1.34262716e+02	NaN	1.61e+01	4.23e+02	4.91e-01
sedumi	Inf	1.34262687e+02	NaN	2.25e+02	2.72e+03	2.19e+02
csdp	Inf	1.34262716e+02	NaN	4.32e+02	5.01e+03	4.37e+02
mater-5 equations: 20463 variables: 327758						
sdpt3	Inf	1.33538715e+02	NaN	1.18e+02	9.53e+03	2.16e+00
mater-6 equations: 20463 variables: 327758						
sdpt3	Inf	1.33538715e+02	NaN	1.41e+02	1.77e+03	4.23e+01
shmup1 equations: 16 variables: 1713						
sdpt3	-1.88414829e+02	-1.88414833e+02	1.90e-08	1.15e+00	1.69e+01	1.14e-02
sedumi	-1.88414829e+02	-1.88414833e+02	2.27e-08	1.46e+00	4.69e+00	2.39e+00
sdpa	-1.88414828e+02	-1.88414833e+02	2.34e-08	3.54e-01	1.42e-02	3.66e-03
csdp	-1.88414777e+02	-1.88414842e+02	3.44e-07	8.86e-01	9.65e-03	5.98e-01
shmup2 equations: 200 variables: 194881						
sdpt3	-3.46242547e+03	-3.46242683e+03	3.93e-07	1.88e+01	1.25e+02	3.45e-01
sedumi	-3.46242547e+03	-3.46242707e+03	4.63e-07	9.82e+01	3.79e+02	3.90e+02
sdpa	-3.46241800e+03	-3.46242686e+03	2.56e-06	3.22e+01	3.66e+01	1.00e+02
csdp	-3.46241265e+03	-3.46242909e+03	4.75e-06	4.00e+01	4.27e-01	1.19e+02
shmup3 equations: 420 variables: 812641						
sdpt3	-2.09883443e+03	-2.09883789e+03	1.64e-06	1.24e+02	1.18e+03	9.24e-01
sedumi	-2.09883146e+03	-2.09883931e+03	3.74e-06	9.69e+02	2.03e+00	2.50e+03
sdpa	-2.09863743e+03	-2.09883792e+03	9.55e-05	2.87e+02	8.51e+02	7.73e-01
csdp	-2.09882493e+03	-2.09884835e+03	1.12e-05	3.85e+02	5.04e+03	1.30e+03

Solver	fU	fL	$\mu(fU, fL)$	t_s	t_u	t_l
shmup4 equations: 800 variables: 2827361						
sdpt3	-7.99251994e+03	-7.99255155e+03	3.95e-06	5.78e+02	5.36e+03	3.88e+00
sedumi	-7.99253351e+03	-7.99256893e+03	4.43e-06	7.88e+03	9.62e+00	2.08e+04
sdpa	-7.99230976e+03	-7.99255241e+03	3.04e-05	1.55e+03	1.47e+03	6.58e-01
csdp	-7.99229951e+03	-7.99259347e+03	3.68e-05	1.46e+03	9.63e+00	2.80e+03
shmup5 equations: 1800 variables: 13849441						
sdpt3	-2.38579087e+04	-2.38589269e+04	4.27e-05	4.22e+03	4.23e+04	2.39e+04
trto1 equations: 36 variables: 361						
sdpt3	-1.10450000e+03	-1.10450000e+03	2.90e-09	3.07e-01	3.40e-01	2.49e-03
sedumi	-1.10449999e+03	-1.10450000e+03	7.75e-09	2.18e-01	1.15e-02	8.58e-01
sdpa	-1.10449998e+03	-1.10450001e+03	3.45e-08	2.44e-01	1.83e-02	1.45e-03
csdp	-1.10449998e+03	-1.10450001e+03	2.64e-08	3.08e-02	1.16e-02	1.81e-01
trto2 equations: 144 variables: 4897						
sdpt3	-1.27999722e+04	-1.28000000e+04	2.17e-06	7.00e-01	2.08e+00	2.63e-03
sedumi	-1.27999980e+04	-1.28000005e+04	1.93e-07	2.10e+00	1.06e+01	2.27e+00
sdpa	-1.27997282e+04	-1.28000170e+04	2.26e-05	3.99e-01	4.40e-02	2.54e-03
csdp	-1.27999978e+04	-1.28000000e+04	1.77e-07	1.44e+00	2.59e+00	4.25e+00
trto3 equations: 544 variables: 52225						
sdpt3	-1.27996731e+04	-1.28000051e+04	2.59e-05	3.65e+00	4.15e+00	9.16e-03
sedumi	-1.27999995e+04	-1.28000003e+04	6.01e-08	2.23e+01	1.46e+02	3.13e+01
sdpa	-1.27967378e+04	-1.28000090e+04	2.56e-04	4.39e+00	4.68e+00	9.22e-03
csdp	-1.27999934e+04	-1.28000005e+04	5.52e-07	1.61e+01	6.44e+01	6.32e+01
trto4 equations: 1200 variables: 228001						
sdpt3	-1.27657010e+04	-1.27658311e+04	1.02e-05	2.33e+01	2.42e+01	3.37e-02
sedumi	-1.27658078e+04	-1.27658501e+04	3.31e-06	2.12e+02	1.39e+03	8.64e+02
sdpa	-1.27418262e+04	-1.27659080e+04	1.89e-03	3.84e+01	4.26e+01	3.44e-02
csdp	Inf	-Inf	NaN	9.67e+01	4.41e-04	3.30e-04
trto5 equations: 3280 variables: 1554721						
sdpt3	-1.27918614e+04	-1.28000269e+04	6.38e-04	1.49e+02	1.88e+02	7.83e+00
vibra1 equations: 36 variables: 661						
sdpt3	-4.08190123e+01	-4.08190124e+01	3.63e-09	4.97e-01	1.38e-02	1.99e-03
sedumi	-4.08190097e+01	-4.08190130e+01	7.96e-08	2.77e-01	3.00e-01	1.22e+00
sdpa	-4.08190114e+01	-4.08190126e+01	2.95e-08	1.99e-01	2.51e-02	3.73e-03
csdp	-4.08190121e+01	-4.08190125e+01	1.09e-08	7.82e-02	1.86e-02	2.82e-01
vibra2 equations: 144 variables: 9553						
sdpt3	-1.66015334e+02	-1.66015364e+02	1.85e-07	1.32e+00	2.71e+00	4.01e-03
sedumi	-1.66015349e+02	-1.66015365e+02	9.44e-08	2.42e+00	3.04e+00	1.48e+01
sdpa	-1.66014156e+02	-1.66015363e+02	7.27e-06	1.01e+00	1.19e+00	4.11e-03
csdp	-1.66015343e+02	-1.66015363e+02	1.21e-07	4.15e+00	3.38e+00	3.26e+00
vibra3 equations: 544 variables: 103585						
sdpt3	-1.72611848e+02	-1.72613197e+02	7.81e-06	1.01e+01	1.87e+01	1.73e-02
sedumi	-1.72612631e+02	-1.72613123e+02	2.85e-06	5.16e+01	3.17e-01	4.11e+02
sdpa	-1.72554938e+02	-1.72614442e+02	3.45e-04	1.36e+01	1.28e+01	6.74e+01
csdp	-1.72612963e+02	-1.72613108e+02	8.41e-07	4.71e+01	3.02e-01	1.31e+02
vibra4 equations: 1200 variables: 454129						
sdpt3	-1.65612143e+02	-1.65613347e+02	7.27e-06	6.04e+01	1.31e+02	6.77e-02
sedumi	-1.65612112e+02	-1.65613493e+02	8.34e-06	5.10e+02	1.54e+00	2.57e+03
sdpa	-1.65568646e+02	-1.65613342e+02	2.70e-04	1.08e+02	1.06e+02	6.61e-02
csdp	-1.65609715e+02	-1.65615700e+02	3.61e-05	2.60e+02	1.55e+00	2.54e+02
vibra5 equations: 3280 variables: 3104401						
sdpt3	-1.65893152e+02	-1.65903300e+02	6.12e-05	5.32e+02	5.69e+02	1.52e+01

References

- [1] Lp_solve reference guide. <http://lpsolve.sourceforge.net/5.5/index.htm>.
- [2] A. Alizadeh and D. Goldfarb. Second-order cone programming. *Mathematical Programming*, xxx:1–30, 1999.
- [3] E. D. Andersen. Handling free variables in primal dual interior-point methods using a quadratic cone, May 2002. Talk given at SIAM Conference on Optimization, Toronto.
- [4] M. F. Anjos and S. Burer. On handling free variables in interior-point methods for conic linear optimization. *SIAM J. Optimization (SIOPT)*, 18(4):1310–1325, 2007.
- [5] A. Ben-Tal, A. N. M. Kocvara, and J. Zowe. Free material optimization via semidefinite programming: the multiload case with contact conditions. *SIAM J. Optimization*, 9:813–832, 1999.
- [6] B. Borchers. *CSDP User’s Guide*, 2009. <https://projects.coin-or.org/Csdp/>.
- [7] B. Borchers. SDPLIB 1.2: A collection of semidefinite programming test problems. Technical report, Socorro, NM, USA, June, 1998.
- [8] L. El Ghaoui and H. Lebret. Robust solution to least squares problems with uncertain data. In *Proceedings of the second international workshop on Recent advances in total least squares techniques and errors-in-variables modeling*, pages 161–170, Philadelphia, PA, USA, 1997. Society for Industrial and Applied Mathematics.
- [9] R. M. Freund, F. Ordóñez, and K.-C. Toh. Behavioral measures and their correlation with ipm iteration counts on semi-definite programming problems. *Math. Program.*, 109:445–475, January 2007.
- [10] K. Fujisawa, M. Fukuda, K. Kobayashi, M. Kojima, K. Nakata, M. Nakata, and M. Yamashita. *SDPA (SemiDefinite Programming Algorithm), User’s Manual — Version 7.1.2*, June 2008.
- [11] C. Jansson. Vsdp: Verified semidefinite programming.
- [12] C. Jansson. Rigorous lower und upper bounds in linear programming. *SIAM J. Optimization (SIOPT)*, 14(3):914–935, 2004.
- [13] C. Jansson. Guaranteed accuracy for conic programming problems in vector lattices. 2007. <http://arxiv.org/abs/0707.4366v1>.
- [14] C. Jansson. On verified numerical computations in convex programming. *Japan Journal of Industrial and Applied Mathemantics*, 26(2-3):337–363, October 2009.
- [15] C. Jansson, D. Chaykin, and C. Keil. Rigorous error bounds for the optimal value in semidefinite programming. *SIAM Journal on Numerical Analysis*, 46(1):180–200, 2007.
- [16] C. Keil. Lurupa - rigorous error bounds in linear programming. In B. Buchberger, S. Oishi, M. Plum, and S. M. Rump, editors, *Algebraic and Numerical Algorithms and Computer-assisted Proofs*, number 05391 in Dagstuhl Seminar Proceedings, 2006.
- [17] C. Keil. A comparison of software packages for verified linear programming, 2008. http://www.optimization-online.org/DB_HTML/2008/06/2007.html.
- [18] C. Keil. *Rigorous Error Bounds for Finite Dimensional Linear Programming Problems*. Books on Demand GmbH, 2009.
- [19] K. Kobayashi, K. Nakata, and M. Kojima. A conversion of an sdp having free variables into the standard form sdp. *Computational Optimization and Applications*, 36:289–307, 2007.

- [20] M. Kočvara. On the modelling and solving of the truss design problem with global stability constraints. *Structural and Multidisciplinary Optimization*, 23:189–203, 2002.
- [21] C. Mézáros. On free variables in interior point methods. *Optimization Methods and Software*, 9(1):121–139, 1998.
- [22] A. Nemirovski. *Lectures on Modern Convex Optimization*. preprint, Israel Institute of Technology, Georgia Institute of Technology, Haifa, Israel and Atlanta, USA, 2005.
- [23] Y. E. Nesterov and A. S. Nemirovsky. A general approach to polynomial-time algorithms design for convex programming. Technical report, Central Economical and Mathematical Institute, USSR Academy of Sciences, Moscow, 1988.
- [24] NETLIB. Netlib linear programming library. World Wide Web. URL <http://www.netlib.org>.
- [25] F. Ordóñez and R. Freund. Computational experience and the explanatory value of condition measures for linear optimization. *SIAM J. Optimization (SIOPT)*, 14(2):307–333, 2003.
- [26] F. Ordóñez and R. M. Freund. Computational experience and the explanatory value of condition measures for linear optimization. *SIAM J. Optimization*, 14:307–333, 2003.
- [27] G. Pataki and S. Schmieta. The dimacs library of mixed semidefinite-quadratic-linear programs. Technical report, Computational Optimization Research Center, Columbia University, July, 2002. <http://dimacs.rutgers.edu/Challenges/Seventh/Instances/>.
- [28] J. Renegar. Some perturbation theory for linear programming. *Mathematical Programming*, 65(1(A)):79–91, 1995.
- [29] S. M. Rump. Intlab - interval laboratory. In T. Csendes, editor, *Developments in Reliable Computing*, pages 77–104. Kluwer Academic Publishers, Dordrecht, 1999. <http://www.ti3.tu-harburg.de/rump/>.
- [30] S. M. Rump. Verification methods: Rigorous results using floating-point arithmetic. *Acta Numerica*, 19:287–449, 2010.
- [31] J. F. Sturm. *Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones*. Department of Econometrics, Tilburg University, 1998-2001.
- [32] I. The MathWorks. Documentation of linprog.
- [33] I. The MathWorks. *Getting Started with MATLAB*, 2012. <http://www.mathworks.com/help/matlab/getting-started-with-matlab.html>.
- [34] K. C. Toh, M. J. Todd, and R. H. Tutuncu. *On the implementation and usage of SDPT3 – a Matlab software package for semidefinite-quadratic-linear programming, version 4.0*, Juli 2006. <http://www.math.nus.edu.sg/~mattohkc/sdpt3.html>.
- [35] L. Vandenberghe and S. Boyd. Semidefinite programming. *SIAM Review*, 38:49–95, 1994.
- [36] J. Zowe, M. Kočvara, and M. Bendsøe. Free material optimization via mathematical programming. *Mathematical Programming*, 79:445–466, 1997.