VSDP: A MATLAB software package for Verified Semidefinite Programming

Christian Jansson

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

Abstract— VSDP is a MATLAB software package for solving rigorously semidefinite programming problems. Functions for computing verified forward error bounds of the true optimal value and verified certificates of feasibility and infeasibility are provided. All rounding errors due to floating point arithmetic are taken into account.

1. Introduction

Semidefinite Programming has emerged as a powerfool tool in many different areas ranging from control engineering to structural design, combinatorial optimization and global optimization (see the Handbook of Semidefinite Programming [1]). One reason is that there exists a kind of calculus of conic quadratic and semidefinite representable sets and functions, which offers a systematic way to recognize and reformulate a convex program as a semidefinite program. This calculus is applied for example in CVX [2], an optimization modelling language which is designed to support the formulation and construction of optimization problems that the user intends from the outset to be convex. On the other hand non-convex problems are frequently solved by using convex relaxations, where consequently also SDP-solvers can be used.

Many algorithms for solving semidefinite programming problems require that appropriate rank conditions are fulfilled, and that strictly feasible solutions of the primal and the dual problem exist, i.e Slater's constraint qualification holds. All these solvers do not provide a guaranteed accuracy or prove existence of optimal solutions. Nevertheless, appropriate warranties for computed results and rigorous forward error bounds can be useful in many situations, especially for ill-conditioned problems with dependencies in the input data, or ill-posed problems. It is well-known that for such problems (but not solely) rounding errors may affect the computation, and even many state-of-the-art solvers may produce erroneous approximations (cf. Neumaier and Shcherbina [3]).

Ill-conditioned and ill-posed problems are not rare in practice. In a paper of Ordóñez and Freund 2003 it is stated that 71% of the lp-instances in the NETLIB Linear Programming Library are ill-posed, and recently Freund, Ordóñez and Toh 2006 [4] have shown that 32 out of 85 problems of the SDPLIB are ill-posed.

VSDP is a software package which provides warranties by computing verified forward error bounds. *Verified*, or sometimes also called *rigorous*, means that the computed results are claimed to be valid with mathematical certainty even in the presence of rounding errors due to floating point arithmetic. VSDP [5] is written completely in MATLAB under use of INTLAB [6]. It is based on a rigorous post-processing applied to the output of semidefinite programming solvers. It is of particular importance that each solver can be used, and the solver need not to produce any error bounds, neither in the forward nor in the backward error sense. This package implements techniques described in [7] and [8], and has several features: it computes **verified lower and upper bounds** of the optimal value for semidefinite programs, proves **existence of feasible solutions**, also for LMI's, provides **rigorous certificates of infeasibility**, facilitates to **solve approximately** the problem by using different well-known semidefinite programming solvers, can handle **several formats**, and allows the use of **interval data**.

It is in the nature of verification methods that not every approximate solution can be verified, such as solvers normally cannot compute an approximate solution for each solvable problem. However, a good verification method should compute rigorous error bounds in almost all well-posed cases, whenever the used solver can compute a sufficiently close approximation. The numerical experiments of VSDP with the SDPLIB suite exhibit that at least for problems of middle size (up to thousands of constraints and millions of variables) rigorous lower (upper bounds) of the optimal value can be computed, provided the distance to dual infeasibility (primal infeasibility) is greater zero. But even if the distance to infeasibility is zero, i.e. the problem is ill-posed, VSDP allows rigorous results, if an a priori assumption about the existence of an optimal solution and its magnitude is known (c.f. [8]).

2. Quick Start

VSDP solves rigorously *semidefinite programming problems* in *block diagonal form*:

$$f_p^* := \min \sum_{j=1}^{n} \langle C_j, X_j \rangle \quad \text{s.t} \;\; \sum_{j=1}^{n} \langle A_{ij}, X_j \rangle = b_i, \;\; i = 1, \dots, m$$
$$X_j \succeq 0, \;\; j = 1, \dots, n, \qquad (1)$$

where $b \in \mathbf{R}^m$, and $C_j, A_{ij}, X_j \in S^{s_j}$, the linear space of real symmetric $s_j \times s_j$ matrices. The usual inner product on the linear space of symmetric matrices is denoted by $\langle .,. \rangle$, which is defined as the trace of the product of two matrices. $X \succeq 0$ means that X is positive semidefinite. Hence, $\succeq$ denotes the *Löwner partial order* on this linear space. It is $f_p^* := +\infty$ if the set of feasible solutions is empty, and $f_p^* := -\infty$ if the problem is unbounded.

If $s_j = 1$ for $j = 1, \ldots, n$ (i.e. C_j, A_{ij}, and X_j are real numbers), then (1) defines the standard linear programming problem.

The *Lagrangian dual* of (1) is

$$f_d^* := \max b^T y \text{ s.t. } Z_j = C_j - \sum_{i=1}^{m} y_i A_{ij} \succeq 0 \text{ for } j = 1, \ldots, n, \tag{2}$$

where $y \in \mathbf{R}^m$. It is $f_d^* := -\infty$, if the set of dual feasible solutions is empty, and $f_d^* := +\infty$ in the unbounded case. The constraints $\sum_{i=1}^m y_i A_{ij} \preceq C_j$ are called *linear matrix inequalities (LMI's)*.

Both problems are connected by *weak duality*

$$f_d^* \leq f_p^*, \tag{3}$$

but strong duality requires in contrast to linear programming additionally strict feasibility assumptions.

VSDP exploits the block-diagonal structure by an $n \times 2$ cell-array `blk`, n cell-arrays `C, X,` and an $m \times n$ cell-array `A` as follows: The j-th block `C{j}` and the blocks `A{i,j}` for $i = 1, \ldots, m$ are real symmetric matrices of common size s_j which is expressed by

`blk{j,1} = 's', blk{j,2} =` s_j.[1]

The block-matrices `C{j}` and `A{i,j}` may be symmetric floating-point or interval matrices, and can be defined in dense or sparse format.

For the purpose of illustration, we start with the following semidefinite program of dimension $m = 4$, $n = 1$, and $s_1 = 3$, i.e. the matrices consists of only one block. The problem depends on a fixed parameter `DELTA`:

```
>> DELTA = 1e-4;

>> C{1}    = [ 0    1/2    0;
              1/2 DELTA   0;
               0     0    DELTA ];
>> A{1,1} =  [ 0 -1/2 0;
              -1/2 0  0;
               0   0  0];
>> A{2,1} =  [ 1 0 0 ;
               0 0 0;
               0 0 0];
>> A{3,1} =  [ 0 0 1;
               0 0 0;
               1 0 0];
>> A{4,1} =  [ 0 0 0;
               0 0 1;
               0 1 0];
>> b = [1; 2*DELTA; 0; 0];
>> blk{1,1} = 's'; blk{1,2} = 3;
```

It is easy to prove that this problem has a zero duality gap with the optimal value −0.5 for every `DELTA > 0`. For `DELTA = 0` the problem is ill-posed with nonzero duality gap, and for negative `DELTA` it is primal and dual infeasible. Especially, it follows that the optimal value is not continuous in `DELTA = 0`.

At the moment, the two semidefinite solvers SDPT3 and SDPA are adapted in a VSDP routine called `MYSDPS`. Therefore, VSDP can be used for computing approximations with different solvers. The user can integrate also other solvers very easily. By default, the function `MYSDPS` calls the semidefinite programming solver SDPT3:

```
>> [objt,Xt,yt,Zt,info] = mysdps(blk,A,C,b);
```

The output consists of approximations of (i) the primal and dual optimal value both stored in `objt`, (ii) the primal and dual solutions `Xt, yt, Zt`, and (iii) information about termination and performance stored in `info`:

```
>> objt, termination = info(1),
objt =
   -5.0003225608e-001   -5.0000000622e-001
termination =
     0
```

For `termination = 0` we have normal termination without any warning. The first four decimal digits of the primal and dual optimal value are correct, but week 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. If we set the global variable `VSDP_CHOICE_SDP = 2` in the file `SDP_GLOBALPARAMETER`, then the solver SDPA is chosen via `MYSDPS`, and we obtain

```
>> [objt,Xt,yt,Zt,info] = mysdps(blk,A,C,b);
>> objt, termination = info(1)
objt =
   -8.4720539237e-001    6.9954120952e-001
termination =
     3
```

No decimal digit of the optimal value is correct, but a warning is given, which indicates that the problem is primal or dual infeasible. To obtain more reliability we can use the function `VSDPLOW` which computes a verified lower bound of the primal optimal value by using a previously computed approximation. This function is based on the following theorem:

[1] At the moment we have incorporated only symmetric matrices, which makes the first instruction redundant. But in future versions we want to distinguish also between other types of matrices. This structure is closely related to an older version of SDPT3.

Theorem 1 *Assume that the maximal eigenvalues of a primal optimal solution (X_j) are bounded by a nonnegative vector $(\overline{x}_j)$, where also infinite components are allowed. Let $\tilde{y} \in \mathbf{R}^m$ (a computed dual approximation). Let*

$$D_j = C_j - \sum_{i=1}^{m} \tilde{y}_i A_{ij}, \text{ and } \underline{d}_j <= \lambda_{\min}(D_j) \text{ for } j = 1, \ldots, n, \tag{4}$$

where $\lambda_{\min}$ denotes the smallest eigenvalue. Assume that D_j has at most l_j negative eigenvalues. Then the primal optimal value is bounded from below by

$$f_p^* \geq b^T \tilde{y} + \sum_{j=1}^{n} l_j \underline{d}_j^- \overline{x}_j =: \underline{f}_p^* \text{ where } \underline{d}_j^- := \min(0, \underline{d}_j). \tag{5}$$

Moreover, if

$$\underline{d}_j \geq 0 \quad for \quad \overline{x}_j = +\infty, \tag{6}$$

then the right hand side $\underline{f}_p^*$ is finite. If $\underline{d}_j \geq 0$ for $j = 1, \ldots, n$, then $\tilde{y}$ is dual feasible and $f_d^* \geq \underline{f}_p^*$, and if moreover $\tilde{y}$ is optimal, then $f_d^* = \underline{f}_p^*$.

There are no assumptions about the quality of $\tilde{y}$, but the last assertion implies that an approximation close to optimality should produce a rigorous lower bound with modest overestimation. The lower bound $\underline{f}_p^*$ sums up the approximate dual value $b^T\tilde{y}$ and the violations of dual feasibility by taking into account the signs and multiplying these violations with appropriate primal weights.

VSDPLOW uses as starting point the already computed approximations Xt,yt,Zt, and the call (all upper bounds $\overline{x}_j$ are assumed to be infinite) has the form

```
>> [fL, Y, dl] = vsdplow(blk,A,C,b,Xt,yt,Zt)
```

The output fL, Y, and dl corresponds to the lower bound $\underline{f}_p^*$, the certificate of dual feasibility Y, and the vector of eigenvalue bounds $\underline{d}$ where $\tilde{y}$ =Y, respectively. In the case where no certificate of feasibility could be computed we set Y = NaN.

With the SDPA approximations the rigorous lower bound is infinite, and dual feasibility is not verified, i.e. Y = NaN. But working with the SDPT3 approximations yields

```
fL =
  -5.0000000622e-001
```

With SDPT3 a finite rigorous lower bound close to the optimal value together with a certificate of dual feasibility Y is computed. Therefore, strong duality is verified. If finite upper bounds xu= $(\overline{x}_j)$ are known then the call of VSDPLOW is

```
>> [fL, Y, dl]=vsdplow(blk,A,C,b,Xt,yt,Zt,xu)
```

Similarly, with function VSDPUP we can compute a verified upper bound fU of the optimal value using the previously computed approximations. This function is based on the following theorem:

Theorem 2 *Assume that the absolute value of a dual optimal solution is bounded by a vector $\overline{y} > 0$, which may also have infinite components. Let $\tilde{X}_j \in S^{s_j}$ for $j = 1, \ldots, n$, and assume that each $\tilde{X}_j$ has at most k_j negative eigenvalues. Let for $i = 1, \ldots, m$ and $j = 1, \ldots, n$,*

$$r_i \geq |b_i - \sum_{j=1}^{n} \langle A_{ij}, \tilde{X}_j \rangle| \tag{7}$$

$$\underline{\lambda}_j \leq \lambda_{\min}(\tilde{X}_j), \text{ and} \tag{8}$$

$$\varrho_j \geq \sup\{\lambda_{\max}(C_j - \sum_{i=1}^{m} y_i A_{ij}) : \ -\overline{y} \leq y \leq \overline{y},\ C_j - \sum_{i=1}^{m} y_i A_{ij} \geq 0\}. \tag{9}$$

Then the dual optimal value satisfies

$$f_d^* \leq \sum_{j=1}^{n} \langle C_j, \tilde{X}_j \rangle - \sum_{j=1}^{n} k_j \underline{\lambda}_j^- \varrho_j + \sum_{i=1}^{m} r_i \overline{y}_i =: \overline{f}_d^*, \tag{10}$$

where $\underline{\lambda}_j^- := \min(0, \underline{\lambda}_j)$. Moreover, if

$$r_i = 0 \text{ for } \overline{y}_i = +\infty \text{ and } \underline{\lambda}_j \geq 0 \text{ for } \varrho_j = +\infty, \tag{11}$$

then the right hand side $\overline{f}_d^*$ is finite. If $\underline{\lambda}_j \geq 0$ and $r_i = 0$ for all i, j, then $(\tilde{X}_j)$ is primal feasible and $f_p^* \leq \overline{f}_d^*$. If moreover $(\tilde{X}_j)$ is optimal, then $f_p^* = \overline{f}_d^*$.

The bound $\overline{f}_d^*$ sums up the approximate primal objective value $\sum_{j=1}^{n} \langle C_j, \tilde{X}_j \rangle$ and the violations of primal feasibility (r_i and $\underline{\lambda}_j^-$) by taking into account the signs and multiplying these violations with appropriate weights ϱ_j and $\overline{y}_i$. The call of VSDPUP has the form

```
>> [fU, X, lb] = vsdpup(blk,A,C,b,Xt,yt,Zt);
fU =
  -4.9996776932e-001
```

The output fU, X and lb corresponds to the upper bound $\overline{f}_d^*$, the interval block-diagonal matrix (containing the rigorous certificate of primal feasibility), and the vector of eigenvalue bounds $\underline{\lambda}_j$.

Summarizing, by using the SDPT3 approximations we have verified the inequality

$$-5.0000000622e{-}001 = \underline{f}_p^* \leq \overline{f}_d^* = -4.9996776932e{-}001,$$

Certificates of strictly primal and strictly dual feasible solutions are computed. The Strong Duality Theorem implies that the primal and the dual problem have a nonempty compact set of optimal solutions. The upper and lower bounds of the optimal value show a modest overestimation, mainly due to the accuracy of SDPT3.

Table 1: Approximations $\tilde{f}_p^*$, $\tilde{f}_d^*$, $\tilde{r}$ of SDPT3

DELTA	$\tilde{f}_p^*$	$\tilde{f}_d^*$	$\tilde{r}$
1.0e-003	-5.00000e-001	-5.00000e-001	7.62771e-010
1.0e-004	-5.00032e-001	-5.00000e-001	1.29041e-008
1.0e-005	-7.69935e-001	-6.45752e-001	2.34663e-005

Table 2: Rigorous bounds fU, fL

DELTA	fU	fL
1.0e-003	-4.99997e-001	-5.00000e-001
1.0e-004	-4.98728e-001	-5.00000e-001
1.0e-005	1.57690e+000	-6.45752e-001

Further numerical results for different values DELTA are summarized in Tables 1 and 2. The approximate primal and dual optimal value computed by SDPT3 are denoted by $\tilde{f}_p^*$ and $\tilde{f}_d^*$, respectively. The value $\tilde{r}$ is the maximum of the relative gap and the measures for primal and dual infeasibility. In all cases the default values of SDPT3 are used, and normal termination without warning has occurred. SDPT3 is not backward stable, since in two cases $\tilde{f}_p^* < \tilde{f}_d^*$ violating the weak duality. For the smallest value of DELTA no decimal digit of $\tilde{f}_p^*$ or $\tilde{f}_d^*$ is correct. In all cases no warning was given. The approximate residual $\tilde{r}$ leads to the suspicion that at least five decimal digits are correct. The new rigorous bounds (which use the computed approximations of SDPT3) reflects much more the reliability of SDPT3, and the number of correct decimal digits for the computed result. The bounds fU and fL fulfill weak duality, and the true optimal value $-1/2$ is inside the bounds, which is not the case for the approximations $\tilde{f}_p^*$ and $\tilde{f}_d^*$ corresponding to the values DELTA $= 10^{-4}$ and DELTA $= 10^{-5}$.

3. Rigorous Error Bounds for the SDPLIB

The SDPLIB is a collection of semidefinite programming problems with different areas of applications. Freund, Ordóñez and Toh [4] have solved 85 problems of the SDPLIB with SDPT3. They have shown that 32 are ill-posed. VSDP could compute (by using SDPT3 as approximate solver) for all 85 problems a rigorous lower bound of the optimal value and verify the existence of strictly dual feasible solutions. This implies a zero duality gap for all these problems. A finite rigorous upper bound could be computed for all well-posed problems with one exception; this is hinf8 being ill-conditioned. For all 32 ill-posed problems VSDP has computed $\overline{f}_d^* = +\infty$, which reflects exactly that the distance to the next primal infeasible problem is zero as well as the infinite condition number.

Detailed numerical results can be found in [5]. For the 85 test problems, SDPT3 (with default values) gave 32 warnings, but 13 warnings were given for well-posed problems. No warning was given for 13 ill-posed problems. In other words, there is no correlation between warnings and the difficulty of the problem. What is the sense of warnings? I have no satisfactory answer. But rigorous bounds provide safety and are important, especially in the case where algorithms subsequently call other algorithms, as is done for example in branch-and-bound methods.

Some major characteristics of our numerical results for the SDPLIB are as follows: The median of the time ratio for computing the rigorous lower (upper) bound and the approximation is 0.045, (2.4), respectively. The median of the guaranteed accuracy for the problems with finite condition number is $4.9 \cdot 10^{-7}$. We have used here the median because there are some outliers. One of the largest problems which could be solved by VSDP is thetaG51 where the number of constraints is $m = 6910$, and the dimension of the primal symmetric matrix X is $s = 1001$ (implying 501501 variables). For this problem SDPT3 gave the message out of memory, and we used SDPA as approximate solver. The rigorous lower and upper bounds computed by VSDP are fL$= -3.4900 \cdot 10^2$, fU$= -3.4406 \cdot 10^2$, respectively. This is an outlier because the guaranteed relative accuracy is only 0.014, which may be sufficient in several applications, but is insufficient from a numerical point of view. However, existence of optimal solutions and strong duality is proved. The times in seconds for computing the approximations, the lower and the upper bound of the optimal value are t= 3687.95, tFL= 45.17, and tFU= 6592.52, respectively.

References

[1] H. Wolkowicz, R. Saigal, L. Vandenberghe, “Handbook of Semidefinite Programming ”, Kluwer Academic Publishers, 2000

[2] M. Grant, S. Boyd and Y. Ye, cvx User’s Guide, 2006, http://www.stanford.edu/~boyd/cvx

[3] A. Neumaier and O. Shcherbina, Safe bounds in linear and mixed-integer programming, *Math. Programming A* 99 (2004), 283-296.

[4] R. M. Freund, F. Ordóñez and K. Toh, Behavioral Measures and their Correlation with IPM Iteration on Semi-Definite Programming Problems, 2006, http://web.mit.edu/rfreund/www/sdpcond.pdf

[5] C. Jansson, VSDP: Verified SemiDefinite Programming, USER’S GUIDE, Beta Version 0.1 for MATLAB 7.0, http://www.ti3.tu-harburg.de/ to appear

[6] S. M . Rump, INTLAB - Interval Laboratory, the Matlab toolbox for verified computations, Version 5.3, 2006. http://www.ti3.tu-harburg.de/

[7] C. Jansson, D. Chaykin, C. Keil. Rigorous Error Bounds for the Optimal Value in Semidefinite Programming, *SIAM J. of Num. An.*, to appear, 2007

[8] C. Jansson, Termination and Verification for Ill-posed Semidefinite Programming Problems, submitted 2005.