SDPB 3.0.0

October 29, 2024

Contents

1 Introduction

SDPB is an arbitrary-precision semidefinite program solver, specialized for "polynomial matrix programs" (defined below). This document describes SDPB's usage and input/output. Much more detail about its design is given in [1]. The reader is encouraged to look there for a better understanding of SDPB's parameters and internal operation.

1.1 Installation and Requirements

See `Install.md` for up-to-date instructions on getting pre-made binaries or building from source.

2 Polynomial Matrix Programs

`SDPB` solves the following type of problem, which we call a *polynomial matrix program* (PMP). Consider a collection of symmetric polynomial matrices

$$M_j^n(x) = \begin{pmatrix} P_{j,11}^n(x) & \dots & P_{j,1m_j}^n(x) \\ \vdots & \ddots & \vdots \\ P_{j,m_j1}^n(x) & \dots & P_{j,m_jm_j}^n(x) \end{pmatrix} \tag{2.1}$$

labeled by $0 \leq n \leq N$ and $1 \leq j \leq J$, where each element $P_{j,rs}^n(x)$ is a polynomial in x. Given $b \in \mathbb{R}^N$, we would like to

$$\begin{array}{ll} \text{maximize} & b_0 + b \cdot y \quad \text{over} \quad y \in \mathbb{R}^N, \\ \text{such that} & M_j^0(x) + \sum_{n=1}^N y_n M_j^n(x) \succeq 0 \quad \text{for all } x \geq 0 \text{ and } 1 \leq j \leq J. \end{array} \tag{2.2}$$

The notation $M \succeq 0$ means "M is positive semidefinite."

The constant b_0 in formulation (2.2) is completely irrelevant to the solution algorithm, but is included for convenience.

3 Input to `SDPB`

You will normally start with a Polynomial Matrix Program (PMP) described in a file (or several files) in JSON, Mathematica, or XML format. PMP must first be converted, using `pmp2sdp` program, into an internal SDP format that SDPB can quickly load. SDP format is described in `docs/SDPB_input_format.md`.

`pmp2sdp` uses a PMP definition that is slightly different but equivalent to (2.2):

$$\begin{array}{ll} \text{maximize} & a \cdot z \quad \text{over} \quad z \in \mathbb{R}^{N+1}, \\ \text{such that} & \sum_{n=0}^N z_n W_j^n(x) \succeq 0 \quad \text{for all } x \geq 0 \text{ and } 1 \leq j \leq J, \\ & n \cdot z = 1. \end{array} \tag{3.1}$$

where $W_j^n(x)$ are matrix polynomials. The normalization condition $n \cdot z = 1$ can be used to solve for one of the components of z in terms of the others. For numerical stability, SDPB eliminates the component z_k that has the largest absolute value of $|n_k|$. Calling the remaining components $y \in \mathbb{R}^N$, we arrive at (2.2), where $M_j^n(x)$ are linear combinations of $W_j^n(x)$ and b_0, b_n are linear combinations of the a_n. This difference in convention is for convenient use in the conformal bootstrap. Note that conversion to (2.2) is trivial, if you choose normalization vector $n = (1, 0, \ldots, 0)$: in that case $(b_0 \ldots b_N) = (a_0 \ldots a_N)$, $(y_1 \ldots y_N) = (z_1 \ldots z_N)$, $M_j^n(x) = W_j^n(x)$.

To construct an SDP, `pmp2sdp` needs the following information:

- for each $j = 1, \ldots, J$:
  - polynomial matrices $W_j^0(x), \ldots, W_j^N(x)$,
  - bilinear bases $q_m^{(j1)}(x)$ $(m = 0, \ldots, \delta_{j1})$ and $q_m^{(j2)}(x)$ $(m = 0, \ldots, \delta_{j2})$ used to construct matrices $Q_{\delta_{j1}}$ and $Q_{\delta_{j2}}$ (see eqs. (2.6) and (2.11) in [1]),
  - sample points $x_k^{(j)}$ $(k = 0, \ldots, d_j)$,
  - sample scalings $s_k^{(j)}$ $(k = 0, \ldots, d_j)$,
- an objective function $a \in \mathbb{R}^{N+1}$.
- a normalization vector $n \in \mathbb{R}^{N+1}$.

By default, d_j equals to the maximum degree of polynomials in $W_j^0(x), \ldots, W_j^N(x)$. To improve performance, one may use a lower value (e.g. by specifying `"reducedPrefactor"` or choosing sample points explicitly, see below). A bilinear basis is a collection of polynomials $q_m^{(j)}(x)$ such that $\deg q_m^{(j)} = m$, for example monomials $q_m^{(j)}(x) = x^m$. (A better choice for numerical stability is usually orthogonal polynomials on the positive real line.) The sample points, sample scalings and bilinear bases determine how the PMP is represented internally as an SDP. In principle, they do not affect the solution of the PMP, but in practice they can affect numerical stability. See Section 3.3 in [1] for details.

`pmp2sdp` can parse input in `JSON`, `Mathematica` and `XML` formats, described below. We recommend using `JSON`; the two other formats are left for backward compatibility.

For convenience and/or for better efficiency when running `pmp2sdp` in parallel, you may split PMP into several files. In that case, you should provide an `.nsv` file with a null-separated list of PMP files. See details in `docs/Usage.md`.

3.1 `JSON` format

Listing 1: JSON format for PMP

```
PMP ≡ {
        "objective": ["a_0", ...,"a_N"],
        (optional) "normalization": ["n_0", ...,"n_N"],
        "PositiveMatrixWithPrefactorArray": [
                ⟨positive matrix with prefactor 1⟩,
                ...
                ⟨positive matrix with prefactor J⟩,
        ]
}
```

positive matrix with prefactor j $\equiv$ `{`
    (optional) `"prefactor" | "DampedRational":` $\langle$*damped-rational prefactor*$\rangle$`,`
    (optional) `"reducedPrefactor":` $\langle$*damped-rational prefactor*$\rangle$`,`
    `"polynomials": [`
        `[`$\langle$*polynomial vector* $Q^n_{j,11}(x)\rangle$`, ...` $\langle$*polynomial vector* $Q^n_{j,m_j 1}(x)\rangle$`],`
        `...`
        `[`$\langle$*polynomial vector* $Q^n_{j,1m_j}(x)\rangle$`, ...,` $\langle$*polynomial vector* $Q^n_{j,m_j m_j}(x)\rangle$`]`
    `],`
    (optional) `"samplePoints": ["`$x^{(j)}_0$`", ...,"`$x^{(j)}_{d_j}$`"],`
    (optional) `"sampleScalings": ["`$s^{(j)}_0$`", ...,"`$s^{(j)}_{d_j}$`"],`
    (optional) `"reducedSampleScalings": ["`$s'^{(j)}_0$`", ...,"`$s'^{(j)}_{d_j}$`"],`
    (optional) `"bilinearBasis_0": [`
        $\langle$*polynomial* $q^{(j1)}_0(x)\rangle$`,`
        `...`
        $\langle$*polynomial* $q^{(j1)}_{\lfloor d_j/2\rfloor}(x)\rangle$
    `],`
    (optional) `"bilinearBasis_1": [`
        $\langle$*polynomial* $q^{(j2)}_0(x)\rangle$`,`
        `...`
        $\langle$*polynomial* $q^{(j2)}_{\lfloor (d_j-1)/2\rfloor}(x)\rangle$
    `],`
    (optional, obsolete) `"bilinearBasis": [`
        $\langle$*polynomial* $q^{(j)}_0(x)\rangle$`,`
        `...`
        $\langle$*polynomial* $q^{(j)}_{\lfloor d_j/2\rfloor}(x)\rangle$
    `]`
`}`

polynomial vector $Q^n_{j,rs}(x)$ $\equiv$ `[`$\langle$*polynomial* $Q^0_{j,rs}(x)\rangle$`, ...`$\langle$*polynomial* $Q^N_{j,rs}(x)\rangle$`]`

polynomial $c_0 + c_1 x + \cdots + c_d x^d$ $\equiv$ `["`c_0`", ...,"`c_d`"]`

damped-rational prefactor $c\frac{b^x}{\prod_{i=1}^k (x-p_i)}$ $\equiv$ `{`
    `"constant": "`c`"`
    `"base": "`b`"`
    `"poles": ["`p_1`", ...,"`p_k`"]`
`}`

Some fields in `JSON` format are optional.

If `"normalization"` is not specified, the default value $n = (1, 0, \ldots, 0)$ is assumed, effectively reducing formulation (3.1) to (2.2).

If `"samplePoints"`, `"sampleScalings"`, `"reducedSampleScalings"`, `"bilinearBasis_0"` and/or "bilinearBasis_1" are not specified, they are calculated automatically from the "prefactor" and "reducedPrefactor". Only damped-rational prefactors are supported. Such prefactors are relevant to the conformal bootstrap problems. These stand for

$$\texttt{DampedRational[}c\texttt{, }\{p_1,\ldots,p_k\}\texttt{, }b\texttt{, }x\texttt{]} \quad \rightarrow \quad c\frac{b^x}{\prod_{i=1}^{k}(x-p_i)}. \tag{3.2}$$

Sample points $x_0 \ldots x_{d_j}$ are chosen to minimize interpolation errors on the positive real line. d_j is determined as maximum degree of polynomials in $W_j^0(x), \ldots, W_j^N(x)$ plus the number of poles in the reduced prefactor minus the number of poles in the prefactor. Sample scalings are $s_k = \chi(x_k)$ and reduced sample scalings are $s'_k = \chi'(x_k)$, where $\chi(x)$ is the prefactor and $\chi'(x)$ is the reduced prefactor. "bilinearBasis_0" is a set of orthogonal polynomials with respect to measure $\sum_0^{d_j} s'_k \delta(x-x_k)dx$ on the positive real line, and "bilinearBasis_1" is a set of orthogonal polynomials with respect to measure $x\sum_0^{d_j} s'_k \delta(x-x_k)dx$.

If "prefactor" is not specified, default sample points, sample scalings and bilinear bases are calculated assuming the prefactor $\chi(x) = e^{-x}$. For constant constraints (degree-0 polynomials), default prefactor is $\chi(x) = 1$. Note that in SDPB 3.0.0 and earlier, "prefactor" was called "DampedRational"; this name is left for backward compatibility.

If "reducedPrefactor" is not specified, it is set to the same value as "prefactor".

If "bilinearBasis" is specified, it is used for both $q_m^{(j1)}(x)$ and $q_m^{(j2)}(x)$. This field is left for backward compatibility with SDPB 3.0.0 and earlier.

The JSON format is also described by the schema in docs/json_schema/pmp_schema.json.

3.2 Mathematica interface

JSON can be generated with a WritePmpJson function in the included Mathematica package mathematica/SDPB.m. Here file is the JSON file to be written to, and sdp has a form described in Listing 2. You can also provide a custom function getSampleDataFn generating sample points, sample scalings and bilinear bases. An example of such function is getAnalyticSampleData in SDPB.m. It implements the same sampling algorithm as in pmp2sdp.

As an example bootstrap application, the included notebook Bootstrap2dExample.m computes a single-correlator dimension bound for 2d CFTs with a $\mathbb{Z}_2$ symmetry, as in [2].

Listing 2: Usage of WritePmpJson in SDPB.m

function call ≡
WritePmpJson[file, ⟨*sdp*⟩, prec]
or
WritePmpJson[file, ⟨*sdp*⟩, prec, ⟨*getSampleDataFn*⟩]

sdp ≡ SDP[⟨*objective*⟩, ⟨*normalization*⟩, ⟨*positive matrices with prefactors*⟩]

objective ≡ {a_0, ..., a_N}

normalization $\equiv$ {n_0, ..., n_N}

positive matrices with prefactors $\equiv$ {
    ⟨*positive matrix with prefactor 1*⟩,
    ...
    ⟨*positive matrix with prefactor J*⟩,
}

positive matrix with prefactor j $\equiv$
PositiveMatrixWithPrefactor[<|
(optional) "prefactor" -> ⟨*prefactor*⟩,
(optional) "reducedPrefactor" -> ⟨*prefactor*⟩,
"polynomials" -> {
    {
        {$Q^0_{j,11}(x)$, ..., $Q^N_{j,11}(x)$}, ..., {$Q^0_{j,m_j1}(x)$, ..., $Q^N_{j,m_j1}(x)$}
    },
    ...
    {
        {$Q^0_{j,1m_j}(x)$, ..., $Q^N_{j,1m_j}(x)$}, ..., {$Q^0_{j,m_jm_j}(x)$, ..., $Q^N_{j,m_jm_j}(x)$}
    },
}
|>]

prefactor $\equiv$
DampedRational[c, {$p_1, \ldots, p_k$}, b, x]
or
const

getSampleDataFn[PositiveMatrixWithPrefactor, prec] $\equiv$ function returning a map
<|
(optional) "samplePoints" ->...
(optional) "sampleScalings" -> ...
(optional) "bilinearBasis_0" -> ...
(optional) "bilinearBasis_1" ->...
|>

3.3 XML format

pmp2sdp can read XML files describing PMP in form (2.2). Note that we recommend using JSON as more versatile, compact and efficient.

Listing 3: XML format for PMP

PMP $\equiv$
<sdp>
⟨*xml for objective*⟩

⟨*xml for polynomial vector matrices*⟩
</sdp>

xml for objective ≡
<objective>
<elt>b_0</elt>
...
<elt>b_N</elt>
</objective>

xml for polynomial vector matrices ≡
<polynomialVectorMatrices>
⟨*xml for polynomial vector matrix* $M_1^n(x)$⟩
...
⟨*xml for polynomial vector matrix* $M_J^n(x)$⟩
</polynomialVectorMatrices>

xml for polynomial vector matrix $M_j^n(x)$ ≡
<polynomialVectorMatrix>
<rows>m_j</rows>
<cols>m_j</cols>
<elements>
⟨*xml for polynomial vector* $P_{j,11}^n(x)$⟩
...
⟨*xml for polynomial vector* $P_{j,m_j1}^n(x)$⟩
...
⟨*xml for polynomial vector* $P_{j,1m_j}^n(x)$⟩
...
⟨*xml for polynomial vector* $P_{j,m_jm_j}^n(x)$⟩
</elements>
<samplePoints>
<elt>$x_0^{(j)}$</elt>
...
<elt>$x_{d_j}^{(j)}$</elt>
</samplePoints>
<sampleScalings>
<elt>$s_0^{(j)}$</elt>
...
<elt>$s_{d_j}^{(j)}$</elt>
</sampleScalings>
<bilinearBasis>
⟨*xml for polynomial* $q_0^{(j)}(x)$⟩
...
⟨*xml for polynomial* $q_{\lfloor d_j/2 \rfloor}^{(j)}(x)$⟩
</bilinearBasis>
</polynomialVectorMatrix>

xml for polynomial vector $P^n_{j,rs}(x)$ $\equiv$
<polynomialVector>
$\langle$*xml for polynomial* $P^0_{j,rs}(x)\rangle$
...
$\langle$*xml for polynomial* $P^N_{j,rs}(x)\rangle$
</polynomialVector>

xml for polynomial $a_0 + a_1 x + \ldots a_d x^d$ $\equiv$
<polynomial>
<coeff>a_0</coeff>
...
<coeff>a_d</coeff>
</polynomial>

3.4 An Example

Let's look at an example. Consider the following problem: maximize $-y$ such that

$$1 + x^4 + y\left(\frac{x^4}{12} + x^2\right) \quad \geq \quad 0 \qquad \text{for all } x \geq 0 \tag{3.3}$$

This is an PMP with 1×1 positive-semidefiniteness constraints. We will arbitrarily choose a prefactor of e^{-x} = `DampedRational[1,{}, 1/E,x]`. The `Mathematica` code for this example (see `mathematica/Tests.m`) is

Listing 4: Mathematica input for the example (3.3)

```
Module[
  {
    pols = {
      PositiveMatrixWithPrefactor[<|
        "prefactor" -> DampedRational[1,{}, 1/E,x],
        "polynomials" -> {{{1 + x^4, x^4/12 + x^2}}}
      >|]
    },
    norm = {1, 0},
    obj = {0, -1},
    prec = 200
  },
  WritePmpJson["pmp.json", SDP[obj, norm, pols], prec];
];
```

It produces the following JSON file

Listing 5: JSON file `pmp.json` produced by listing 4. Decimals are truncated at 12 digits.

```
{
```

```
  "objective": [
    "0",
    "-1"
  ],
  "normalization": [
    "1",
    "0"
  ],
  "PositiveMatrixWithPrefactorArray": [
    {
      "DampedRational": {
        "base": "0.3678794411",
        "constant": "1",
        "poles": []
      },
      "polynomials": [
        [
          [
            [
              "1",
              "0",
              "0",
              "0",
              "1"
            ],
            [
              "0",
              "0",
              "1",
              "0",
              "0.08333333333"
            ]
          ]
        ]
      ]
    }
  ]
}
```


4 Internal SDP

To understand the output of SDPB, we need a rough understanding of its internal representation of the above PMP as a semidefinite program (SDP). Much more detail is given in [1]. The PMP (2.2) is translated into a dual pair of SDPs of the following form:

$$\begin{array}{lll} \mathcal{D}: & \text{maximize} & \mathrm{Tr}(CY) + b_0 + b \cdot y \quad \text{over} \quad y \in \mathbb{R}^N,\ Y \in \mathcal{S}^K, \\ & \text{such that} & \mathrm{Tr}(A_* Y) + By = c,\ \text{and} \\ & Y \succeq 0. & \end{array} \tag{4.1}$$

$$
\begin{array}{rll}
\mathcal{P}: & \text{minimize} & b_0 + c\cdot x \quad \text{over} \quad x \in \mathbb{R}^P,\ X \in \mathcal{S}^K, \\
& \text{such that} & X = \sum_{p=1}^P A_p x_p - C, \\
& & B^T x = b, \\
& & X \succeq 0,
\end{array}
\tag{4.2}
$$

where "$\succeq 0$" means "is positive-semidefinite" and

$$
\begin{aligned}
c &\in \mathbb{R}^P, \\
B &\in \mathbb{R}^{P\times N}, \\
A_1, \ldots, A_P, C &\in \mathcal{S}^K.
\end{aligned}
\tag{4.3}
$$

Here, $\mathcal{S}^K$ is the space of $K \times K$ symmetric real matrices, and $\mathrm{Tr}(A_* Y)$ denotes the vector $(\mathrm{Tr}(A_1 Y), \ldots, \mathrm{Tr}(A_P Y)) \in \mathbb{R}^P$. An optimal solution to (4.1) and (4.2) is characterized by $XY = 0$ and also equality of the primal and dual objective functions $\mathrm{Tr}(CY) + b_0 + b\cdot y = b_0 + c\cdot x$.

The residues

$$
\begin{aligned}
P &\equiv \sum_i A_i x_i - X - C, \\
p &\equiv b - B^T x, \\
d &\equiv c - \mathrm{Tr}(A_* Y) - By,
\end{aligned}
\tag{4.4}
$$

measure the failure of x, X, y, Y to satisfy their constraints. We say a point $q = (x, X, y, Y)$ is "primal feasible" or "dual feasible" if the residues are sufficiently small,

$$
\begin{array}{rrcccl}
\text{primal feasible:} & \texttt{primalError} & \equiv & \max_{i,j}\{|p_i|, |P_{ij}|\} & < & \texttt{primalErrorThreshold}; \\
\text{dual feasible:} & \texttt{dualError} & \equiv & \max_i\{|d_i|\} & < & \texttt{dualErrorThreshold},
\end{array}
$$

where `primalErrorThreshold` $\ll 1$ and `dualErrorThreshold` $\ll 1$ are parameters chosen by the user.

An optimal point should be both primal and dual feasible, and have (nearly) equal primal and dual objective values. Specifically, let us define `dualityGap` as the normalized difference between the primal and dual objective functions

$$
\begin{aligned}
\texttt{dualityGap} &\equiv \frac{|\texttt{primalObjective} - \texttt{dualObjective}|}{\max\{1, |\texttt{primalObjective} + \texttt{dualObjective}|\}}, \\
\texttt{primalObjective} &\equiv b_0 + c\cdot x, \\
\texttt{dualObjective} &\equiv \mathrm{Tr}(CY) + b_0 + b\cdot y.
\end{aligned}
\tag{4.5}
$$

A point is considered "optimal" if

$$
\texttt{dualityGap} < \texttt{dualityGapThreshold},
\tag{4.6}
$$

where `dualityGapThreshold` $\ll 1$ is chosen by the user.

5 Output of SDPB

5.1 Terminal Output

The output from running SDPB on the example problem in section 3.4 is in listing 6. The input, output, and checkpoint files are listed first, followed by various parameters. After each iteration, SDPB prints the following:

time: The current solver runtime in seconds.

mu: The value of the complementarity $\mathrm{Tr}(XY)/K$.

P-obj: The primal objective value $b_0 + c \cdot x$.

D-obj: The dual objective value $\mathrm{Tr}(CY) + b_0 + b \cdot y$.

gap: The value of dualityGap.

P-err: The primal error $\max_{i,j}\{|P_{ij}|\}$.

p-err: The primal error $\max_i\{|p_i||\}$.

D-err: The dual error $\max_i\{|d_i|\}$.

P-step: The primal step length $\alpha_{\mathcal{P}}$ described in [1].

D-step: The dual step length $\alpha_{\mathcal{D}}$ described in [1].

beta: The corrector centering parameter β_c described in [1].

These values are also written to iterations.json in the output folder (see Section 5.3).

If an optimal solution exists, the primal and dual error will decrease until the problem becomes primal and dual feasible. Then the primal and dual objective functions start to converge, and the complementarity μ decreases until the duality gap becomes smaller than dualityGapThreshold.

The terminal output ends with the final values of the primal/dual objectives, primal/dual errors and duality gap.

Listing 6: Output of SDPB for the input file in listing 5

```
$ ./build/pmp2sdp  -i test/data/end-to-end_tests/1d/input/pmp.json -o test/out/sdp --precision 664
Processed 1 SDP blocks in 0.01 seconds, output: test/out/sdp

$ ./build/sdpb --noFinalCheckpoint -s test/out/sdp -o test/out/sdpb --precision=664
2024-Oct-29 20:04:14 Start SDPB
SDPB version: 3.0.0-82-g4552e479
MPI processes: 1, nodes: 1
SDP directory   : "test/out/sdp"
out directory   : "test/out/sdpb"

Parameters:
```

```
maxIterations                = 500
maxRuntime                   = 9223372036854775807
checkpointInterval           = 3600
findPrimalFeasible           = false
findDualFeasible             = false
detectPrimalFeasibleJump     = false
detectDualFeasibleJump       = false
precision(actual)            = 400(448)
dualityGapThreshold          = 1e-30
primalErrorThreshold         = 1e-30
dualErrorThreshold           = 1e-30
initialMatrixScalePrimal     = 1e+20
initialMatrixScaleDual       = 1e+20
feasibleCenteringParameter   = 0.1
infeasibleCenteringParameter = 0.3
stepLengthReduction          = 0.7
maxComplementarity           = 1e+100
initialCheckpointDir         = "test/out/sdp.ck"
checkpointDir                = "test/out/sdp.ck"
noFinalCheckpoint            = true
writeSolution                = x,y
procGranularity              = 1
verbosity                    = 1

Initialize SDP solver
        primal dimension: 5
        dual dimension: 1
        SDP blocks: 1
2024-Jan-04 20:04:14 Start solver iterations

          time    mu      P-obj       D-obj      gap         P-err       p-err       D-err      P-step   D-step   beta
-----------------------------------------------------------------------------------------------------------------------
1            0 1.0e+40  +0.00       +0.00       0.00       +1.00e+20   +1.00       +2.88e+20   0.631    0.647    0.300
2            0 5.0e+39  +9.49e+19   -1.64e+20   1.00       +3.69e+19   +0.369      +1.02e+20   0.653    0.639    0.300
3            0 2.5e+39  +1.04e+20   -2.92e+20   1.00       +1.28e+19   +0.128      +3.68e+19   0.660    0.639    0.300
4            0 1.2e+39  +1.43e+20   -4.30e+20   1.00       +4.35e+18   +0.0435     +1.33e+19   0.652    0.638    0.300
5            0 5.8e+38  +1.91e+20   -6.13e+20   1.00       +1.51e+18   +0.0151     +4.80e+18   0.645    0.636    0.300
6            0 2.8e+38  +2.52e+20   -8.65e+20   1.00       +5.38e+17   +0.00538    +1.75e+18   0.640    0.634    0.300
7            0 1.4e+38  +3.36e+20   -1.21e+21   1.00       +1.94e+17   +0.00194    +6.38e+17   0.636    0.633    0.300
8            0 7.2e+37  +4.52e+20   -1.69e+21   1.00       +7.05e+16   +0.000705   +2.34e+17   0.635    0.633    0.300
9            0 3.6e+37  +6.16e+20   -2.34e+21   1.00       +2.57e+16   +0.000257   +8.58e+16   0.634    0.633    0.300
10           0 1.8e+37  +8.43e+20   -3.23e+21   1.00       +9.43e+15   +9.43e-05   +3.15e+16   0.633    0.633    0.300
11           0 9.3e+36  +1.16e+21   -4.46e+21   1.00       +3.46e+15   +3.46e-05   +1.16e+16   0.633    0.633    0.300
12           0 4.7e+36  +1.59e+21   -6.16e+21   1.00       +1.27e+15   +1.27e-05   +4.24e+15   0.633    0.633    0.300
13           0 2.4e+36  +2.20e+21   -8.49e+21   1.00       +4.65e+14   +4.65e-06   +1.56e+15   0.633    0.633    0.300
14           0 1.2e+36  +3.03e+21   -1.17e+22   1.00       +1.71e+14   +1.71e-06   +5.71e+14   0.633    0.633    0.300
15           0 6.1e+35  +4.18e+21   -1.62e+22   1.00       +6.26e+13   +6.26e-07   +2.09e+14   0.633    0.633    0.300
16           0 3.1e+35  +5.77e+21   -2.23e+22   1.00       +2.30e+13   +2.30e-07   +7.68e+13   0.633    0.633    0.300
17           0 1.6e+35  +7.96e+21   -3.08e+22   1.00       +8.42e+12   +8.42e-08   +2.82e+13   0.633    0.633    0.300
18           0 7.9e+34  +1.10e+22   -4.25e+22   1.00       +3.09e+12   +3.09e-08   +1.03e+13   0.633    0.633    0.300
19           0 4.0e+34  +1.52e+22   -5.86e+22   1.00       +1.13e+12   +1.13e-08   +3.79e+12   0.633    0.633    0.300
20           0 2.0e+34  +2.09e+22   -8.09e+22   1.00       +4.15e+11   +4.15e-09   +1.39e+12   0.633    0.633    0.300
21           0 1.0e+34  +2.89e+22   -1.12e+23   1.00       +1.52e+11   +1.52e-09   +5.09e+11   0.633    0.633    0.300
22           0 5.2e+33  +3.98e+22   -1.54e+23   1.00       +5.58e+10   +5.58e-10   +1.87e+11   0.633    0.633    0.300
23           0 2.6e+33  +5.50e+22   -2.13e+23   1.00       +2.05e+10   +2.05e-10   +6.85e+10   0.633    0.633    0.300
24           0 1.3e+33  +7.59e+22   -2.93e+23   1.00       +7.51e+09   +7.51e-11   +2.51e+10   0.633    0.633    0.300
25           0 6.7e+32  +1.05e+23   -4.05e+23   1.00       +2.75e+09   +2.75e-11   +9.21e+09   0.633    0.633    0.300
26           0 3.4e+32  +1.44e+23   -5.59e+23   1.00       +1.01e+09   +1.01e-11   +3.38e+09   0.633    0.633    0.300
27           0 1.7e+32  +1.99e+23   -7.71e+23   1.00       +3.70e+08   +3.70e-12   +1.24e+09   0.633    0.633    0.300
28           0 8.7e+31  +2.75e+23   -1.06e+24   1.00       +1.36e+08   +1.36e-12   +4.54e+08   0.633    0.633    0.300
29           0 4.4e+31  +3.80e+23   -1.47e+24   1.00       +4.98e+07   +4.98e-13   +1.67e+08   0.633    0.633    0.300
30           0 2.2e+31  +5.24e+23   -2.03e+24   1.00       +1.83e+07   +1.83e-13   +6.11e+07   0.633    0.633    0.300
31           0 1.1e+31  +7.23e+23   -2.80e+24   1.00       +6.70e+06   +6.70e-14   +2.24e+07   0.633    0.633    0.300
32           0 5.7e+30  +9.98e+23   -3.86e+24   1.00       +2.46e+06   +2.46e-14   +8.22e+06   0.633    0.633    0.300
33           0 2.9e+30  +1.38e+24   -5.32e+24   1.00       +9.01e+05   +9.01e-15   +3.01e+06   0.633    0.633    0.300
34           0 1.5e+30  +1.90e+24   -7.35e+24   1.00       +3.30e+05   +3.30e-15   +1.11e+06   0.633    0.633    0.300
35           0 7.4e+29  +2.62e+24   -1.01e+25   1.00       +1.21e+05   +1.21e-15   +4.05e+05   0.633    0.633    0.300
36           0 3.7e+29  +3.62e+24   -1.40e+25   1.00       +4.44e+04   +4.44e-16   +1.49e+05   0.633    0.633    0.300
```

37	0 1.9e+29	+4.99e+24	−1.93e+25	1.00	+1.63e+04	+1.63e-16	+5.45e+04	0.633	0.633	0.300
38	0 9.6e+28	+6.89e+24	−2.66e+25	1.00	+5.98e+03	+5.98e-17	+2.00e+04	0.633	0.633	0.300
39	0 4.8e+28	+9.51e+24	−3.68e+25	1.00	+2.19e+03	+2.19e-17	+7.33e+03	0.633	0.633	0.300
40	0 2.4e+28	+1.31e+25	−5.07e+25	1.00	+803.	+8.03e-18	+2.69e+03	0.634	0.634	0.300
41	0 1.2e+28	+1.81e+25	−7.00e+25	1.00	+294.	+2.94e-18	+985.	0.634	0.634	0.300
42	0 6.3e+27	+2.50e+25	−9.64e+25	1.00	+108.	+1.08e-18	+360.	0.636	0.635	0.300
43	0 3.2e+27	+3.44e+25	−1.32e+26	1.00	+39.1	+3.91e-19	+131.	0.642	0.639	0.300
44	0 1.6e+27	+4.72e+25	−1.81e+26	1.00	+14.0	+1.40e-19	+47.5	0.657	0.649	0.300
45	0 7.8e+26	+6.42e+25	−2.41e+26	1.00	+4.80	+4.80e-20	+16.7	0.699	0.677	0.300
46	0 3.6e+26	+8.51e+25	−3.04e+26	1.00	+1.44	+1.44e-20	+5.40	0.785	0.768	0.300
47	0 1.6e+26	+1.02e+26	−3.19e+26	1.00	+0.310	+3.10e-21	+1.25	1.00	1.00	0.300
48	0 5.3e+25	+8.01e+25	−1.83e+26	1.00	+7.61e-116	+2.88e-118	+5.79e-90	0.734	0.734	0.100
49	0 1.8e+25	+2.73e+25	−6.20e+25	1.00	+9.24e-116	+2.94e-117	+1.08e-85	0.736	0.736	0.100
50	0 6.0e+24	+9.21e+24	−2.09e+25	1.00	+1.56e-115	+4.24e-117	+4.71e-86	0.736	0.736	0.100
51	0 2.0e+24	+3.10e+24	−7.08e+24	1.00	+1.09e-115	+1.74e-117	+1.41e-86	0.736	0.736	0.100
52	0 6.9e+23	+1.04e+24	−2.39e+24	1.00	+1.07e-115	+6.29e-117	+3.94e-87	0.737	0.737	0.100
53	0 2.3e+23	+3.51e+23	−8.08e+23	1.00	+1.33e-115	+4.65e-118	+1.05e-87	0.737	0.737	0.100
54	0 7.8e+22	+1.18e+23	−2.72e+23	1.00	+1.02e-115	+9.43e-117	+2.79e-88	0.738	0.738	0.100
55	0 2.6e+22	+3.95e+22	−9.14e+22	1.00	+7.61e-116	+9.56e-117	+7.36e-89	0.738	0.738	0.100
56	0 8.8e+21	+1.32e+22	−3.07e+22	1.00	+1.02e-115	+5.74e-117	+1.93e-89	0.739	0.739	0.100
57	0 2.9e+21	+4.43e+21	−1.03e+22	1.00	+1.15e-115	+1.19e-116	+5.03e-90	0.739	0.739	0.100
58	0 9.8e+20	+1.48e+21	−3.44e+21	1.00	+8.15e-116	+1.77e-116	+1.31e-90	0.739	0.739	0.100
59	0 3.3e+20	+4.95e+20	−1.15e+21	1.00	+7.61e-116	+3.24e-117	+3.42e-91	0.740	0.740	0.100
60	0 1.1e+20	+1.66e+20	−3.85e+20	1.00	+1.27e-115	+8.39e-118	+8.91e-92	0.740	0.740	0.100
61	0 3.7e+19	+5.54e+19	−1.29e+20	1.00	+1.20e-115	+3.30e-118	+2.32e-92	0.740	0.740	0.100
62	0 1.2e+19	+1.85e+19	−4.30e+19	1.00	+1.02e-115	+7.83e-119	+6.03e-93	0.740	0.740	0.100
63	0 4.1e+18	+6.18e+18	−1.44e+19	1.00	+6.16e-116	+1.44e-119	+1.57e-93	0.740	0.740	0.100
64	0 1.4e+18	+2.06e+18	−4.80e+18	1.00	+1.60e-116	+3.76e-120	+4.08e-94	0.740	0.740	0.100
65	0 4.6e+17	+6.89e+17	−1.60e+18	1.00	+4.17e-117	+9.76e-121	+1.06e-94	0.740	0.740	0.100
66	0 1.5e+17	+2.30e+17	−5.36e+17	1.00	+1.08e-117	+2.54e-121	+2.75e-95	0.740	0.740	0.100
67	0 5.1e+16	+7.68e+16	−1.79e+17	1.00	+2.81e-118	+6.60e-122	+7.16e-96	0.740	0.740	0.100
68	0 1.7e+16	+2.56e+16	−5.97e+16	1.00	+7.31e-119	+1.72e-122	+1.86e-96	0.740	0.740	0.100
69	0 5.7e+15	+8.56e+15	−1.99e+16	1.00	+1.90e-119	+4.58e-123	+4.83e-97	0.740	0.740	0.100
70	0 1.9e+15	+2.86e+15	−6.65e+15	1.00	+4.93e-120	+1.38e-123	+1.25e-97	0.740	0.740	0.100
71	0 6.3e+14	+9.53e+14	−2.22e+15	1.00	+1.28e-120	+7.26e-124	+3.26e-98	0.740	0.740	0.100
72	0 2.1e+14	+3.18e+14	−7.41e+14	1.00	+3.33e-121	+4.71e-124	+8.47e-99	0.740	0.740	0.100
73	0 7.1e+13	+1.06e+14	−2.47e+14	1.00	+8.64e-122	+1.22e-124	+2.20e-99	0.740	0.740	0.100
74	0 2.4e+13	+3.54e+13	−8.26e+13	1.00	+2.24e-122	+3.18e-125	+5.71e-100	0.740	0.740	0.100
75	0 7.9e+12	+1.18e+13	−2.76e+13	1.00	+5.83e-123	+8.26e-126	+1.48e-100	0.740	0.740	0.100
76	0 2.6e+12	+3.94e+12	−9.20e+12	1.00	+1.51e-123	+2.14e-126	+3.85e-101	0.740	0.740	0.100
77	0 8.8e+11	+1.32e+12	−3.07e+12	1.00	+3.93e-124	+5.57e-127	+1.00e-101	0.740	0.740	0.100
78	0 2.9e+11	+4.39e+11	−1.02e+12	1.00	+1.02e-124	+1.45e-127	+2.60e-102	0.740	0.740	0.100
79	0 9.8e+10	+1.47e+11	−3.42e+11	1.00	+2.65e-125	+3.75e-128	+6.74e-103	0.740	0.740	0.100
80	0 3.3e+10	+4.89e+10	−1.14e+11	1.00	+6.88e-126	+9.74e-129	+1.75e-103	0.740	0.740	0.100
81	0 1.1e+10	+1.63e+10	−3.80e+10	1.00	+1.79e-126	+2.53e-129	+4.54e-104	0.740	0.740	0.100
82	0 3.6e+09	+5.44e+09	−1.27e+10	1.00	+4.64e-127	+6.57e-130	+1.18e-104	0.740	0.740	0.100
83	0 1.2e+09	+1.82e+09	−4.24e+09	1.00	+1.20e-127	+1.71e-130	+3.06e-105	0.740	0.740	0.100
84	0 4.0e+08	+6.06e+08	−1.41e+09	1.00	+3.13e-128	+4.43e-131	+7.95e-106	0.740	0.740	0.100
85	0 1.3e+08	+2.02e+08	−4.72e+08	1.00	+8.12e-129	+1.15e-131	+2.06e-106	0.740	0.740	0.100
86	0 4.5e+07	+6.75e+07	−1.57e+08	1.00	+2.11e-129	+2.99e-132	+5.36e-107	0.740	0.740	0.100
87	0 1.5e+07	+2.25e+07	−5.25e+07	1.00	+5.47e-130	+7.74e-133	+1.39e-107	0.740	0.740	0.100
88	0 5.0e+06	+7.51e+06	−1.75e+07	1.00	+1.42e-130	+2.02e-133	+3.61e-108	0.740	0.740	0.100
89	0 1.7e+06	+2.51e+06	−5.85e+06	1.00	+3.69e-131	+5.17e-134	+9.38e-109	0.740	0.740	0.100
90	0 5.6e+05	+8.36e+05	−1.95e+06	1.00	+9.57e-132	+1.32e-134	+2.44e-109	0.740	0.740	0.100
91	0 1.9e+05	+2.79e+05	−6.51e+05	1.00	+2.48e-132	+4.07e-135	+6.32e-110	0.740	0.740	0.100
92	0 6.2e+04	+9.31e+04	−2.17e+05	1.00	+6.41e-133	+3.42e-136	+1.64e-110	0.740	0.740	0.100
93	0 2.1e+04	+3.11e+04	−7.24e+04	1.00	+1.65e-133	+3.73e-136	+4.26e-111	0.740	0.740	0.100
94	0 6.9e+03	+1.04e+04	−2.41e+04	1.00	+4.13e-134	+7.06e-137	+1.11e-111	0.741	0.741	0.100
95	0 2.3e+03	+3.47e+03	−8.04e+03	1.00	+5.50e-135	+1.24e-135	+2.87e-112	0.741	0.741	0.100
96	0 7.7e+02	+1.17e+03	−2.67e+03	1.00	+5.50e-135	+2.24e-136	+7.45e-113	0.741	0.741	0.100
97	0 2.6e+02	+394.	−886.	1.00	+5.50e-135	+1.33e-135	+1.93e-113	0.741	0.741	0.100
98	0 85.	+136.	−291.	1.00	+6.88e-135	+1.38e-135	+5.01e-114	0.738	0.738	0.100
99	0 29.	+49.8	−93.6	1.00	+8.25e-135	+0.00	+1.31e-114	0.722	0.722	0.100
100	0 10.	+21.0	−29.2	1.00	+8.25e-135	+0.00	+3.65e-115	0.694	0.694	0.100
101	0 3.8	+10.5	−8.33	1.00	+6.88e-135	+0.00	+1.11e-115	0.707	0.707	0.100
102	0 1.4	+5.45	−1.41	1.00	+8.73e-135	+7.64e-136	+3.27e-116	0.765	0.765	0.100
103	0 0.43	+2.99	+0.854	0.555	+2.75e-135	+6.29e-136	+7.67e-117	0.768	0.768	0.100
104	0 0.13	+2.20	+1.54	0.176	+2.02e-135	+1.55e-136	+1.78e-117	0.764	0.764	0.100

```
105          0 0.041    +1.95      +1.75      0.0557     +3.92e-136  +0.00      +4.21e-118  0.764   0.764   0.100
106          0 0.013    +1.88      +1.81      0.0175     +3.57e-136  +0.00      +9.94e-119  0.766   0.766   0.100
107          0 0.0040   +1.85      +1.83      0.00543    +1.70e-136  +1.38e-135 +2.33e-119  0.768   0.768   0.100
108          0 0.0012   +1.84      +1.84      0.00168    +5.13e-136  +0.00      +5.39e-120  0.770   0.770   0.100
109          0 0.00038  +1.84      +1.84      0.000514   +6.37e-136  +0.00      +1.24e-120  0.772   0.772   0.100
110          0 0.00012  +1.84      +1.84      0.000157   +9.57e-136  +1.86e-136 +2.82e-121  0.773   0.773   0.100
111          0 3.5e-05  +1.84      +1.84      4.77e-05   +9.95e-137  +1.08e-136 +6.39e-122  0.774   0.774   0.100
112          0 1.1e-05  +1.84      +1.84      1.44e-05   +5.12e-136  +2.90e-136 +1.44e-122  0.775   0.775   0.100
113          0 3.2e-06  +1.84      +1.84      4.37e-06   +8.62e-136  +3.56e-136 +3.24e-123  0.776   0.776   0.100
114          0 9.7e-07  +1.84      +1.84      1.32e-06   +7.51e-136  +4.08e-136 +7.26e-124  0.776   0.776   0.100
115          0 2.9e-07  +1.84      +1.84      3.97e-07   +5.08e-136  +2.73e-136 +1.62e-124  0.777   0.777   0.100
116          0 8.8e-08  +1.84      +1.84      1.19e-07   +5.77e-136  +3.84e-136 +3.62e-125  0.777   0.777   0.100
117          0 2.6e-08  +1.84      +1.84      3.59e-08   +9.94e-136  +4.95e-136 +8.08e-126  0.777   0.777   0.100
118          0 7.9e-09  +1.84      +1.84      1.08e-08   +5.35e-136  +2.62e-136 +1.80e-126  0.777   0.777   0.100
119          0 2.4e-09  +1.84      +1.84      3.24e-09   +2.19e-136  +1.90e-136 +4.01e-127  0.777   0.777   0.100
120          0 7.2e-10  +1.84      +1.84      9.73e-10   +7.67e-136  +4.85e-136 +8.92e-128  0.778   0.778   0.100
121          0 2.2e-10  +1.84      +1.84      2.92e-10   +8.89e-136  +5.16e-136 +1.99e-128  0.778   0.778   0.100
122          0 6.5e-11  +1.84      +1.84      8.77e-11   +5.98e-136  +3.68e-136 +4.41e-129  0.778   0.778   0.100
123          0 1.9e-11  +1.84      +1.84      2.63e-11   +7.13e-136  +2.89e-136 +9.82e-130  0.778   0.778   0.100
124          0 5.8e-12  +1.84      +1.84      7.90e-12   +3.06e-136  +7.54e-137 +2.18e-130  0.778   0.778   0.100
125          0 1.7e-12  +1.84      +1.84      2.37e-12   +9.92e-136  +4.60e-136 +4.85e-131  0.778   0.778   0.100
126          0 5.2e-13  +1.84      +1.84      7.11e-13   +5.20e-136  +3.37e-136 +1.08e-131  0.778   0.778   0.100
127          0 1.6e-13  +1.84      +1.84      2.13e-13   +2.32e-136  +3.97e-136 +2.40e-132  0.778   0.778   0.100
128          0 4.7e-14  +1.84      +1.84      6.40e-14   +2.24e-136  +2.86e-136 +5.43e-133  0.778   0.778   0.100
129          0 1.4e-14  +1.84      +1.84      1.92e-14   +2.54e-136  +9.40e-137 +1.17e-133  0.778   0.778   0.100
130          0 4.2e-15  +1.84      +1.84      5.76e-15   +4.15e-136  +6.62e-137 +3.99e-134  0.778   0.778   0.100
131          0 1.3e-15  +1.84      +1.84      1.73e-15   +6.32e-136  +2.25e-136 +1.79e-134  0.778   0.778   0.100
132          0 3.8e-16  +1.84      +1.84      5.19e-16   +2.14e-136  +7.38e-137 +8.25e-134  0.778   0.778   0.100
133          0 1.1e-16  +1.84      +1.84      1.56e-16   +3.37e-136  +2.46e-136 +2.20e-134  0.778   0.778   0.100
134          0 3.4e-17  +1.84      +1.84      4.67e-17   +1.12e-136  +6.53e-137 +3.85e-134  0.778   0.778   0.100
135          0 1.0e-17  +1.84      +1.84      1.40e-17   +4.30e-136  +6.07e-137 +8.25e-134  0.778   0.778   0.100
136          0 3.1e-18  +1.84      +1.84      4.20e-18   +5.18e-136  +3.61e-136 +1.99e-133  0.778   0.778   0.100
137          0 9.3e-19  +1.84      +1.84      1.26e-18   +2.71e-136  +1.17e-136 +1.35e-133  0.778   0.778   0.100
138          0 2.8e-19  +1.84      +1.84      3.78e-19   +2.81e-136  +1.63e-136 +6.49e-133  0.778   0.778   0.100
139          0 8.4e-20  +1.84      +1.84      1.13e-19   +3.12e-136  +1.75e-136 +2.82e-133  0.778   0.778   0.100
140          0 2.5e-20  +1.84      +1.84      3.40e-20   +7.02e-136  +5.05e-136 +8.94e-133  0.778   0.778   0.100
141          0 7.5e-21  +1.84      +1.84      1.02e-20   +7.15e-136  +5.85e-137 +1.22e-132  0.778   0.778   0.100
142          0 2.3e-21  +1.84      +1.84      3.06e-21   +8.79e-136  +4.70e-137 +8.71e-133  0.778   0.778   0.100
143          0 6.8e-22  +1.84      +1.84      9.19e-22   +1.15e-135  +2.78e-136 +2.74e-133  0.778   0.778   0.100
144          0 2.0e-22  +1.84      +1.84      2.76e-22   +5.67e-136  +4.45e-136 +5.94e-133  0.778   0.778   0.100
145          0 6.1e-23  +1.84      +1.84      8.27e-23   +6.16e-136  +3.91e-136 +3.77e-132  0.778   0.778   0.100
146          0 1.8e-23  +1.84      +1.84      2.48e-23   +1.14e-135  +4.40e-136 +1.11e-132  0.778   0.778   0.100
147          0 5.5e-24  +1.84      +1.84      7.44e-24   +6.65e-136  +4.59e-136 +1.32e-132  0.778   0.778   0.100
148          0 1.6e-24  +1.84      +1.84      2.23e-24   +3.93e-136  +2.56e-136 +7.97e-133  0.778   0.778   0.100
149          0 4.9e-25  +1.84      +1.84      6.70e-25   +7.85e-136  +3.06e-136 +1.77e-132  0.778   0.778   0.100
150          0 1.5e-25  +1.84      +1.84      2.01e-25   +4.17e-136  +2.62e-136 +4.13e-133  0.778   0.778   0.100
151          0 4.4e-26  +1.84      +1.84      6.03e-26   +5.77e-136  +3.80e-136 +3.89e-132  0.778   0.778   0.100
152          0 1.3e-26  +1.84      +1.84      1.81e-26   +4.39e-136  +1.70e-136 +6.24e-132  0.778   0.778   0.100
153          0 4.0e-27  +1.84      +1.84      5.43e-27   +7.21e-136  +4.33e-136 +2.04e-132  0.778   0.778   0.100
154          0 1.2e-27  +1.84      +1.84      1.63e-27   +5.54e-136  +3.51e-136 +4.22e-132  0.778   0.778   0.100
155          0 3.6e-28  +1.84      +1.84      4.88e-28   +5.55e-136  +3.99e-136 +7.05e-132  0.778   0.778   0.100
156          0 1.1e-28  +1.84      +1.84      1.47e-28   +2.16e-136  +2.71e-136 +1.73e-131  0.778   0.778   0.100
157          0 3.2e-29  +1.84      +1.84      4.40e-29   +8.40e-136  +3.49e-136 +1.20e-131  0.778   0.778   0.100
158          0 9.7e-30  +1.84      +1.84      1.32e-29   +2.15e-136  +4.09e-136 +3.41e-131  0.778   0.778   0.100
159          0 2.9e-30  +1.84      +1.84      3.96e-30   +6.19e-136  +4.88e-136 +3.44e-131  0.778   0.778   0.100
160          0 8.7e-31  +1.84      +1.84      1.19e-30   +1.88e-136  +1.63e-136 +1.42e-131  0.778   0.778   0.100
-----found primal-dual optimal solution-----

primalObjective = 1.84026576313204924668804017173055420056358532030282556465761906133430166726537336826049865612094019
0211160188629478172102847851400971545
dualObjective   = 1.84026576313204924668804017172924388084784907020307957926406455972756967820389551729116356865203683
7213248476950467408148728442213127677
dualityGap      = 3.56013718775636270149999059635335050723442743109168831293885607041894974620853522385695676694435617
2630595608074240236952679818626742854e-31
primalError     = 8.00472462794652854216192415154691251952338782833660449388851026694624386819579233719251965820695076
6138417240341523554115275581567613244e-136
dualError       = 3.67963333625613903189297919337418392740579208646582684485331852877210732905668816990827126132539445
5152562009003409010941981912150248860e-131
```

```
Saving solution to      : "test/out/sdpb"
```

For larger, parallel runs, the exact solution that SDPB computes can vary between different machines, or even different configurations on the same run. For example, the step size is computed by adding up many other numbers. The order in which these numbers are added can change the final sum in the lowest significant bits. This does not make the solver unstable, but it does introduce a minor amount of variation. The solver still converges to the same answer up to the error thresholds and duality gap, but the actual value of the residual errors and gap will be different.

5.2 Termination

The possible termination reasons for SDPB are as follows

`found primal-dual optimal solution`
: Found a solution for x, X, y, Y that is simultaneously primal feasible, dual feasible, and optimal.

`found primal feasible solution`
: Found a solution for x, X that is primal feasible. SDPB will only terminate with this result if the option `--findPrimalFeasible` is specified.

`found dual feasible solution`
: Found a solution for y, Y that is dual feasible. SDPB will only terminate with this result if the option `--findDualFeasible` is specified.

`primal feasible jump detected`
: A Newton step with primal step length $\alpha_{\mathcal{P}}$ just occurred, without resulting in a primal feasible solution. (Usually this means one should increase `precision`.)

`dual feasible jump detected`
: A Newton step with dual step length $\alpha_{\mathcal{D}}$ just occurred, without resulting in a dual feasible solution. (Usually this means one should increase `precision`.)

`maxIterations exceeded`
: SDPB has run for more iterations than specified by the option `--maxIterations`.

`maxRuntime exceeded`
: SDPB has run for longer than specified by the option `--maxRuntime`.

`maxComplementarity exceeded`
: $\mu = \mathrm{Tr}(XY)/\dim(X)$ exceeded the value specified by `--maxComplementarity`. This might indicate that the problem is unbounded and no optimal solution will be found.

When using SDPB to determine primal or dual feasibility, one can specify the options `--findPrimalFeasible` or `--findDualFeasible`. This will cause the solver to terminate immediately once the primal or dual errors are sufficiently small. This often occurs immediately after the primal or dual step lengths become equal to 1. A step length of 1 means that the solver has found a Newton step that exactly solves the primal or dual constraints, while preserving positive-semidefiniteness of X, Y. Sometimes a step length of 1 does not result in sufficiently small primal/dual errors. This is indicative of numerical instabilities and usually means `precision` should be increased. The options `--detectPrimalFeasibleJump` and `--detectPrimalFeasibleJump` cause `SDPB` to terminate if a step length of 1 occurs without resulting in primal/dual feasibility. If desired, one can then restart the solver with a higher value of `precision`.

5.3 Output File

Listing 7: Contents of the output file `test/out/sdpb/out.txt` corresponding to listing 5. Decimal expansions have been truncated for brevity. `Mathematica` uses `*^` instead of the character `e` for scientific notation. Thus, the output format is not quite suitable for import into `Mathematica` without modification. This could be changed in future versions.

```
terminateReason = "found primal-dual optimal solution";
primalObjective = 1.84026576313204924668804017173055420056358532030282556 46...;
dualObjective   = 1.84026576313204924668804017172924388084784907020307957 92...;
dualityGap      = 3.56013718775636270149999059635335050723442743109168831 29...e-31;
primalError     = 8.00472462794652854216192415154691251952338782833660449 38...e-136;
dualError       = 3.67963333625613903189297919337418392740579208646582684 48...e-131;
Solver runtime  = 0;
```

The output file `test/out/sdpb/out.txt` corresponding to listing 5 is shown in listing 7. It includes the reason for termination, the final primal/dual objective values, the final duality gap, the final primal/dual errors, and the total runtime.

The output file `test/out/sdpb/iterations.json` contains information about each iteration in machine-readable `JSON` format. It includes all fields printed to terminal (see Section 5.1), as well as the quantity `R-err` $= ||\mu I - XY||_{\max}$, condition number for matrix Q, the worst condition number across all SDP blocks and the name of the corresponding block.

The vector y is saved in `test/out/sdpb/y.txt`, and the two blocks of the x vector are saved in `test/out/sdpb/x_0.txt` and `test/out/sdpb/x_1.txt`. For JSON and Mathematica, you can also restore the vector z from y using normalization condition $n \cdot z = 1$ (3.1). If you call SDPB with `--writeSolution="x,y,z"` option, z is saved in `test/out/sdpb/z.txt`. To write additional files for the matrices X and Y, add the option `--writeSolution="x,y,z,X,Y"`.

The output file `test/out/sdpb/c_minus_By/c_minus_By.json` contains the functional $c - B.y$. To plot this functional against x, you will need to know the sample points of your SDP. They can be read from `test/out/sdp/pmp_info.json` generated by `pmp2dsp`.

The value of y gives the solution to our optimization problem. The function

$$1 + x^4 + (-1.840265763084)\left(\frac{x^4}{12} + x^2\right) \tag{5.1}$$

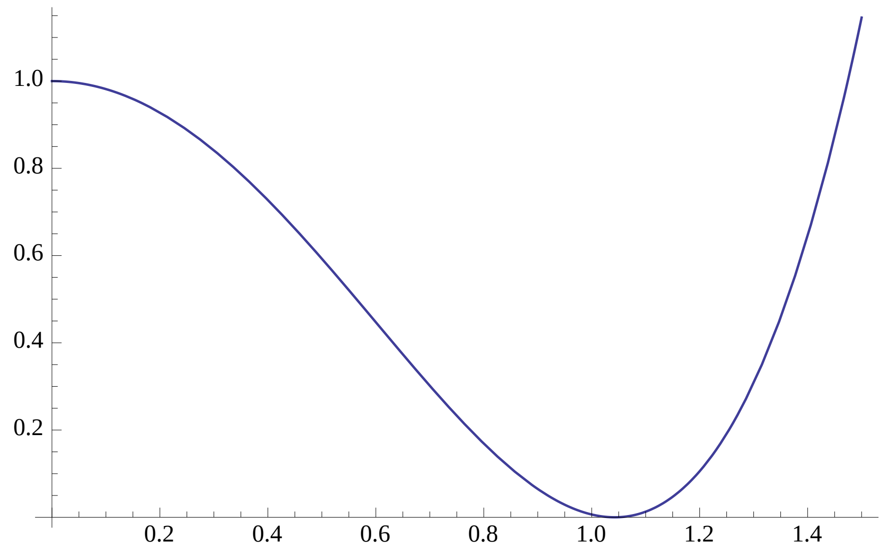

Figure 1: A plot of $1 + x^4 + y\left(\frac{x^4}{12} + x^2\right)$ with $y = -1.840265763084$ equal to its optimal value. The zero near $x = 1$ shows that $-y$ cannot be further increased without violating the positivity constraint.

is plotted in figure 1. The zero near $x = 1$ shows that y is optimal.

5.4 Checkpoints

Every `checkpointInterval`, `SDPB` saves a new checkpoint in a directory with the `.ck` extension. `SDPB` also saves a checkpoint after termination, provided the option `--noFinalCheckpoint` is not specified.

A checkpoint file encodes the values of x, X, y, Y. If `SDPB` detects an existing checkpoint file on startup, it will use those values of x, X, y, Y as initial conditions in the solver. Thus, `SDPB` can be stopped and started at will without losing progress.

A typical workflow for long-running computations on shared machines is to specify a moderate `checkpointInterval` (e.g. one hour) and a somewhat larger `maxRuntime` (e.g. 12 hours). `SDPB` will terminate after 12 hours and can then be restarted without losing progress. If `SDPB` is killed prematurely, then at most 1 hour of progress will be lost. This pattern of restarting gives other users chances to run their processes. It can be sustained indefinitely, allowing extremely long computations.

Checkpoints are written in binary format to conserve space and speed up loading and unloading. If you specify the `--writeSolution="x,y,X,Y"` option, the output directory can also be used to restart a computation with the `-i` option. It will not be bitwise identical to restarting from a binary checkpoint, but it should be very, very, very close.

Text checkpoints can be useful if you want to solve a different system by starting closer to previously solved system. You can also use it to continue a calculation with a different number of cores, or even on a different machine. Using the previous input as an example,

```
$ ./build/sdpb -s test/out/sdp -o test/out/sdpb --noFinalCheckpoint --writeSolution="x,y,X,Y"
$ ./build/sdpb -s test/out/sdp -o test/out/sdpb --noFinalCheckpoint -i test/out/sdpb
```

the second calculation will start from the end of the first calculation.

6 Attribution

If you use SDPB in work that results in publication, please cite [1]. Depending on how SDPB is used, the following sources might also be relevant:

- The first use of semidefinite programming in the bootstrap [3].
- The generalization of semidefinite programming methods to arbitrary spacetime dimension [4].
- The generalization of semidefinite programming methods to arbitrary systems of correlation functions [5].

7 Acknowledgements

SDPB makes extensive use of the parallel linear algebra library Elemental [11], the Boost C++ libraries [9], the libxml2 library [10], and the multiprecision libraries GMP [12], and MPFR [13].

SDPB was partially based on the solvers SDPA and SDPA-GMP [6–8], which were essential sources of inspiration and examples.

Thanks to Filip Kos, David Poland, and Alessandro Vichi for collaboration in developing semidefinite programming methods for the conformal bootstrap and assistance testing SDPB. Thanks to Amir Ali Ahmadi, Hande Benson, Pablo Parrilo, and Robert Vanderbei for advice and discussions about semidefinite programming.

I am supported by DOE grant number DE-SC0009988 and a William D. Loughlin Membership at the Institute for Advanced Study.

References

[1] David Simmons-Duffin, “A Semidefinite Program Solver for the Conformal Bootstrap,” arXiv:1502.02033 [hep-th].

[2] V. S. Rychkov and A. Vichi, “Universal Constraints on Conformal Operator Dimensions,” Phys. Rev. D **80**, 045006 (2009) arXiv:0905.2211 [hep-th].

[3] D. Poland, D. Simmons-Duffin and A. Vichi, "Carving Out the Space of 4D CFTs," JHEP **1205**, 110 (2012) arXiv:1109.5176 [hep-th].

[4] F. Kos, D. Poland and D. Simmons-Duffin, "Bootstrapping the $O(N)$ vector models," JHEP **1406**, 091 (2014) arXiv:1307.6856 [hep-th].

[5] F. Kos, D. Poland and D. Simmons-Duffin, "Bootstrapping Mixed Correlators in the 3D Ising Model," JHEP **1411**, 109 (2014) arXiv:1406.4858 [hep-th].

[6] M. Yamashita, K. Fujisawa, M. Fukuda, K. Nakata, and M. Nakata, "A high-performance software package for semidefinite programs: SDPA 7," Research Report B-463, Dept. of Mathematical and Computing Science, Tokyo Institute of Technology, Tokyo, Japan (2010).

[7] M. Yamashita, K. Fujisawa, and M. Kojima, "Implementation and evaluation of SDPA 6.0 (SemiDefinite Programming Algorithm 6.0)," Optimization Methods and Software" 18 491-505 (2003).

[8] M. Nakata, "A numerical evaluation of highly accurate multiple-precision arithmetic version of semidefinite programming solver: SDPA-GMP, -QD and -DD.," 2010 IEEE International Symposium on Computer-Aided Control System Design (CACSD), 29-34 Sept 2010.

[9] C++ Standards Committee Library Working Group and other contributors, "BOOST C++ Libraries," http://www.boost.org.

[10] Gnome Project, Libxml2, http://www.xmlsoft.org/

[11] J. Poulson, B. Marker, R. van de Geijn, J. Hammond, and N. Romero, "Elemental: A new framework for distributed memory dense matrix computations, ACM Transactions on Mathematical Software," ACM Trans. Math. Softw. 39 2 13:1-24 (2013), doi:10.1145/2427023.2427030

[12] The GNU Multiprecision Library, https://gmplib.org/

[13] The GNU MPFR Library, https://www.mpfr.org/