

DSDP5 Installation Instructions

Steven J. Benson
Yinyu Ye

January 14, 2005

The compressed tar file `DSDP5.X.tar.gz` contains an implementation of the dual-scaling algorithm for conic programming optimization problems.

Create the `DSDP5.X` directory structure and enter it. For example,

```
gunzip DSDP5.X.tar.gz
tar -xvf DSDP5.X.tar
cd DSDP5.X
```

Several executables may have been provided. If not, it will have to be compiled. DSDP is written in the C programming language. It has been tested using several different compilers and architectures. In particular, it has been tested using gcc-2.96 and gcc-3.2 (C and C++), Intel-6.0, Intel-7.1, Intel-8.0, and Microsoft Visual Studio 2003.

Compiling DSDP:

Using Make:

DSDP was developed using Make – which is available on Linux and most Unix systems. To compile DSDP:

1. Enter the directory `DSDP5.X` and edit the file `make.include` to define the `DSDPROOT` variable as the full directory name.
2. In the same file, edit the compiler flags `CC`, `OPTFLAGS`, and `CFLAGS`.
3. If the MATLAB interface is required, also check and edit the `MEX` flag.
4. In the same file, edit the location of the BLAS and LAPACK libraries and include any other libraries required to link to them such as `-lg2c` or `-lm`.
5. Compile the source code to create the DSDP library and drivers.

```
make install
```

If problems persist, please send a copy of the compilation log to the developers.

Without Make:

DSDP can also be compiled by copying all of the source and header files into another directory, compiled, and linked with a driver routine (such as one of the files in `DSDPROOT/examples/`). This process is demonstrated in the `DSDPROOT/examples/allc/` subdirectory. This method neglects the directory structure of the source code and the Makefile system, but it works fine. If problems arise, please send a copy of the compilation log to the developers.

Compiler Flags:

BLAS and LAPACK:

DSDP uses BLAS and LAPACK for many of the underlying operations and must be linked to these libraries. These routines in BLAS and LAPACK are called from the C programming language under the assumption that the routine names are lower case and end with an underscore. The most common linking problem occurs when these assumptions are not true. Several compiler flags can be defined to change these assumptions. Define

- **CAPSBLAS** if the names of BLAS and LAPACK routine names use all capital letters.
- **NOUNDERBLAS** if no underscore is appended to the end of routine names. This flag was used to link DSDP to the reference BLAS available in Matlab using the Microsoft Compiler.
- **_DSDP_NONAMEMANGLING** if a C++ compiler is being used and the BLAS and LAPACK routine names should not be changed.

See the makefiles in the distribution for examples of use of these terms.

DSDP also assumes that the **integer** type in Fortran is the same size as a **long int** in C, and a **double precision** variable in Fortran is the same size as a **double** in C. If problems persist, the macros and type definitions in the file **dsdplapack.c**, located in the **DSDPROOT/include/** directory will have to be edited.

Timing:

DSDP can provide time profiles for several important operations. By default, the timing routine is not implemented due to portability issues among architectures. However, the following flags may be defined for the compiler to activate timing routines:

- **DSDP_MS_TIME** activates the timing utility from the Microsoft compiler
- **DSDP_TIME** activates the timing utility from the GCC and many other compilers.

The routines that call these two timing utilities can be found in the file **dsdptime.c** in **DSDPROOT/src/sys/** and edited.

MATLAB:

If the MATLAB interface is to be generated, one other flag may be of interest.

- **DSDPMATLAB** enables the use of the memory allocation and additional print statements available from MATLAB.

The standard memory allocation and print statements will be used if this flag is not defined.

Testing:

1. Run the executables by switching to directory **DSDPROOT/exec** and typing

```
> dsdp5   truss1.dat-s
> maxcut  graph1
> theta   graph1
```

Compare the output with the files `output.truss1`, `output.maxcut`, and `output.theta`. If the output from any of the tests differs significantly from the files, please report it to the developers.

2. DSDP can be called from MATLAB version 6.0 and higher. Run the sample problems by starting MATLAB in the **DSDPROOT/matlab** directory and typing

```
> check;
```

Compare the output with the output in `check.out`. For help using the package, type

```
> help dsdp;
```

Several example MATLAB files have been provided in **DSDPROOT/matlab** that create example problems, read data files, and verify solutions.