The MOSEK Release Notes. Version 7.1 (Revision 31).

Mosek Support, support@mosek.com

www.mosek.com

- Published by MOSEK ApS, Denmark.

Contents

Chapter 1

Changes and new features in MOSEK

The section presents improvements and new features added to MOSEK in version 7.

1.1 Platform support

In Table 1.1 the supported platform and compiler used to build MOSEK shown. Although RedHat is explicitly mentioned as the supported Linux distribution then MOSEK will work on most other variants of Linux. However, the license manager tools requires Linux Standard Base 3 or newer is installed.

1.2 General changes

- The interior-point optimizer has been extended to semi-definite optimization problems. Hence, MOSEK can optimize over the positive semi-definite cone.
- The network detection has been completely redesigned. MOSEK no longer try detect partial networks. The problem must be a pure primal network for the network optimizer to be used.
- The parameter iparam.objective_sense has been removed.
- The parameter iparam.intpnt_num_threads has been removed. Use the parameter iparam.num_threads instead.
- MOSEK now automatically exploit multiple CPUs i.e. the parameter iparam.num_threads is set to 0 be default. Note the amount memory that MOSEK uses grows with the number of threads employed.

Platform	OS version	C compiler
linux32x86	Redhat 5 or newer (LSB 3+)	Intel C 13.0 (gcc 4.3, glibc 2.3.4)
linux64x86	RedHat 5 or newer (LSB 3+)	Intel C 13.0 (gcc 4.3, glibc 2.3.4)
osx64x86	OSX 10.7 Lion or newer	Intel C 13.0 (llvm-gcc-4.2)
win32x86	Windows Vista, Server 2003 or newer	Intel C 13.0 (VS 2008)
win64x86	Windows Vista, Server 2003 or newer	Intel C 13.0 (VS 2008)

Interface	Supported versions
Java	Sun Java 1.6+
Microsoft.NET	2.1+
Python 2	2.6+
Python 3	3.1+

Table 1.1: Supported platforms

- The MBT file format has been replaced by a new task format. The new format supports semi-definite optimization.
- the HTML version of the documentation is no longer included in the downloads to save space. It is still available online.
- MOSEK is more restrictive about the allowed names on variables etc. This is in particular the case when writing LP files.
- MOSEK no longer tries to detect the cache sizes and is in general less sensitive to the hardware.
- The parameter is set iparam.auto_update_sol_info is default off. In previous version it was by default on.
- The function relaxprimal has been deprecated and replaced by the function primalrepair.

1.3 Optimizers

1.3.1 Interior point optimizer

- The factorization routines employd by the interior-point optimizer for linear and conic optimization problems has been completely rewritten. In particular the dense column detection and handling is improved. The factorization routine will also exploit vendor tuned BLAS routines.

1.3.2 The simplex optimizers

- No major changes.

1.3.3 Mixed-integer optimizer

- A new mixed-integer for linear and conic problems has been introduced. It is from run-to-run determinitic and is parallelized. It is particular suitable for conic problems.

1.4 API changes

- Added support for semidefinite optimization.
- Some clean up has been performed implying some functions have been renamed.

1.5 Optimization toolbox for MATLAB

- A MOSEK equivalent of `bintprog` has been introduced.
- The functionality of the MOSEK version of `linprog` has been improved. It is now possible to employ the simplex optimizer in `linprog`.
- `mosekopt` now accepts a dense A matrix.
- An new method for specification of cones that is more efficient when the problem has many cones has introduced. The old method is still allowed but is deprecated.
- Support for semidefinite optimization problems has been added to the toolbox.

1.6 License system

- Flexlm has been upgraded to version 11.11.

1.7 Other changes

- The documentation has been improved.

1.8 Interfaces

- Semi-definite optimization capabilities have been add to the optimizer APIs.
- A major clean up have occured in the optimizer APIs. This should have little effect for most users.
- A new object orientated interface called Fusion has been added. Fusion is available Java, MATLAB, .NET and Python.
- The AMPL command line tool has been updated to the latest version.

1.9 Platform changes

- 32 bit MAC OSX on Intel x86 (osx32x86) is no longer supported.
- 32 and 64 bit Solaris on Intel x86 (solaris32x86,solaris64x86) is no longer supported.

Chapter 2

Limitations and known issues

- Operating system user names containing spaces can result in problems with the license system.
- We advise against simultaneously using multiple MOSEK environments within a single program as this might cause problems with the Flexlm licensing system.

Chapter 3

Libraries

MOSEK consists of several libraries. When packaging software using MOSEK for redistribution, only a subset of these are required, depending on the platform.

3.1 C/C++

When redistributing a program or library linked with MOSEK, following libraries must be included:

Linux	
64-bit	32-bit
`libmosek64.so.7.1`	`libmosek.so.7.1`
`libmosekglb64.so.7.1`	`libmosekglb.so.7.1`
`libiomp5.so`	`libiomp5.so`
Windows	
64-bit	32-bit
`mosek64_7_1.dll`	`mosek7_1.dll`
`mosekglb64_7_1.dll`	`mosekglb7_1.dll`
`libiomp5md.dll`	`libiomp5md.dll`
OS X	
64-bit	
`libmosek64.dylib.7.1`	
`libmosekglb64.dylib.7.1`	
`libiomp5.dylib`	

3.2 Java

When redistributing a Java application using with MOSEK, following libraries must be included:

Linux	
64-bit	32-bit
libmosek64.so.7.1	libmosek.so.7.1
libmosekglb64.so.7.1	libmosekglb.so.7.1
libiomp5.so	libiomp5.so
libmosekjava7_1.so	libmosekjava7_1.so
libmosekxx7_1.so	libmosekxx7_1.so
libmosekscopt7_1.so	libmosekscopt7_1.so
Windows	
64-bit	32-bit
mosek64_7_1.dll	mosek7_1.dll
mosekglb64_7_1.dll	mosekglb7_1.dll
libiomp5md.dll	libiomp5md.dll
mosekjava7_1.dll	mosekjava7_1.dll
mosekxx7_1.dll	mosekxx7_1.dll
mosekscopt7_1.dll	mosekscopt7_1.dll
OS X	
64-bit	
libmosek64.dylib.7.1	
libmosekglb64.dylib.7.1	
libiomp5.dylib	
libmosekjava7_1.dylib	
libmosekxx7_1.dylib	
libmosekscopt7_1.dylib	

By default the MOSEK/Java interface will look for the binaries in the same directory as the `.jar` file, so they should be placed in the same directory when redistributing.

3.3 .NET

When redistributing a .NET application using with MOSEK, following libraries must be included:

64-bit Windows	32-bit Windows
mosek64_7_1.dll	mosek7_1.dll
mosekglb64_7_1.dll	mosekglb7_1.dll
libiomp5md.dll	libiomp5md.dll
mosekxx7_1.dll	mosekxx7_1.dll
mosekscopt7_1.dll	mosekscopt7_1.dll
mosekdotnet.dll	mosekdotnet.dll

3.4 Python

When redistributing a Python application using with MOSEK, following libraries must be included:

Linux	
64-bit	32-bit
libmosek64.so.7.1	libmosek.so.7.1
libmosekglb64.so.7.1	libmosekglb.so.7.1
libiomp5.so	libiomp5.so
libmosekxx7_1.so	libmosekxx7_1.so
libmosekscopt7_1.so	libmosekscopt7_1.so
Windows	
64-bit	32-bit
mosek64_7_1.dll	mosek7_1.dll
mosekglb64_7_1.dll	mosekglb7_1.dll
libiomp5md.dll	libiomp5md.dll
mosekxx7_1.dll	mosekxx7_1.dll
mosekscopt7_1.dll	mosekscopt7_1.dll
OS X	
64-bit	
libmosek64.dylib.7.1	
libmosekglb64.dylib.7.1	
libiomp5.dylib	
libmosekxx7_1.dylib	
libmosekscopt7_1.dylib	

Furthermore, one (or both) of the directories

- `python/2/mosek` for Python 2.x applications, and
- `python/3/mosek` for Python 3.x applications.

must be included.

By default the MOSEK/Python API will look for the binary libraries in the MOSEK module directory, i.e. the directory containing `__init__.py`. Alternative, if the binary libraries reside in another directory, the application can pre-load the `mosekxx` library from another located before `mosek` is imported, e.g. like this

```
import ctypes ; ctypes.CDLL('my/path/to/mosekxx.dll')
```

Chapter 4

Third party libraries attributions

4.1 Intel MKL

MOSEK uses the Intel(R) Math Kernel Library. Please refer to the official MKL website for more information and details on the license agreement.

4.2 zlib

MOSEK includes the zlib library obtained from the zlib website. The license agreement for zlib is shown below.

```
zlib.h -- interface of the 'zlib' general purpose compression library
version 1.2.7, May 2nd, 2012

Copyright (C) 1995-2012 Jean-loup Gailly and Mark Adler

This software is provided 'as-is', without any express or implied
warranty.  In no event will the authors be held liable for any damages
arising from the use of this software.

Permission is granted to anyone to use this software for any purpose,
including commercial applications, and to alter it and redistribute it
freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not
   claim that you wrote the original software. If you use this software
   in a product, an acknowledgment in the product documentation would be
   appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be
   misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

Jean-loup Gailly        Mark Adler
jloup@gzip.org          madler@alumni.caltech.edu
```

4.3 fplib

MOSEK includes the floating point formating library developed by David M. Gay obtained from the netlib website. The license agreement for fplib is shown below.

```
/****************************************************************
 *
 * The author of this software is David M. Gay.
 *
 * Copyright (c) 1991, 2000, 2001 by Lucent Technologies.
 *
 * Permission to use, copy, modify, and distribute this software for any
 * purpose without fee is hereby granted, provided that this entire notice
 * is included in all copies of any software which is or includes a copy
 * or modification of this software and in all copies of the supporting
 * documentation for such software.
 *
 * THIS SOFTWARE IS BEING PROVIDED "AS IS", WITHOUT ANY EXPRESS OR IMPLIED
 * WARRANTY.  IN PARTICULAR, NEITHER THE AUTHOR NOR LUCENT MAKES ANY
 * REPRESENTATION OR WARRANTY OF ANY KIND CONCERNING THE MERCHANTABILITY
 * OF THIS SOFTWARE OR ITS FITNESS FOR ANY PARTICULAR PURPOSE.
 *
 ***************************************************************/
```

Chapter 5

Bug fixes and improvements

7.1.0.31

- Improved the conic mixed integer optimizer for badly scaled models.

7.1.0.30

- Fixed bug in the conic mixed integer optimizer were infeasibility was not detected correctly in the root node.

7.1.0.29

- Fixed bug in the conic mixed integer optimizer that could cause infeasible problems to not be detected correctly.

7.1.0.28

- In the optimizer API for Python3: fixed a length check error.

7.1.0.27

- Improved the conic mixed integer optimizer for some numerically unstable models.
- Fixed a segmentation fault that may happens when hot-starting the simplex algorithm.

7.1.0.26

- A bug has been fixed in the mixed integer optimizer that caused an infeasible solution to be reported for certain mixed integer conic problems.
- Fixed issue that prevent the OPF file reader to accept scalar values in a specific form.
- A bug has been fixed in CBF file reader related to OBJBCOORD.
- Fixed possible memory issue in automatic conic reformulation function.

7.1.0.25

- Fixed issue with automatic detection of number of CPU cores on windows machines with more than 32 cores.

7.1.0.24

- Fixed a bug in the Fusion API that may cause sparse matrix wrong behavior.
- In case of near dual infeasibility certificate the infeasibility reporting had a bug that has been fixed.
- A bug has been fixed in the mixed integer optimizer that in special case made MOSEK report an infeasible solution as optimal.

7.1.0.22

- Improved mixed integer optimizer for some numerically unstable models.
- Fixed possible memory overwrite in mixed integer optimizer.
- Fixed installation of token server on Windows.

7.1.0.20

- By default all messages originating from the response handler (i.e. error and warning messages) were nulled. They are now displayed.
- There is now a mosekdotnet.dll and a mosekdotnet_7_1.dll. Apart from the name they are identical. This means that both 7.0 and 7.1 .NET dlls can be included in the same project, but a single assembly can still only lin with one of them.

7.1.0.15

- Fixed memory leak in the mixed integer optimizer.

7.1.0.14

- Fine tuned the warning handling.

7.1.0.13

- Fixed a bug occuring in the presolve when the problem has linear dependencies that causes infeasibilities.

7.1.0.12

- Fixed a bug in the mixed integer optimizer that made the optimizer report an optimal but infeasible solution in certain cases.
- Fixed possible memory overwrite in the mixed integer optimizer.
- Improved documentation for Fusion API.

7.1.0.10

- Fixed Python setup script.

7.1.0.9

- Fixed some typos in the quick start guide.

7.1.0.7

- Fixed issue occurring when infeasible initial solutions was passed to the mixed-integer optimizer.

7.1.0.6

- Fixed a bug in the optimization toolbox for MATLAB that cold cause crash when a solution was specified for a conic problem.
- Fixed an bug that could cause a crash in the inteior-point optimizer for very large problems..

7.1.0.5

- Removed the unused parameter iparam.license_allow_overuse.
- Fixed an array type-checking bug in the optimizer API for Python.

7.1.0.4

- Fixed a segmentation fault that could occur on 32 bit platforms for very large problems.

7.1.0.0

- Updated mixed-integer conic optimizer with improved performance.
- Updated MPS reader with support for the CPLEX specific sections QMATRIX and QCMATRIX. The MPS reader has been improved.
- The default MPS format has been changed to the free MPS format.
- Exprimental support for reading the conic benchmark format. Semidefinite constraints and variables are not supported yet.
- A new experimental feature for reformulating certain quadratically constrained into a conic problem.
- In Fusion it is now possible to obtain optimizer statistics e.g. mixed integer gap etc.
- Added some of the BLAS and LAPACK functionality e.g. gemm and potrf functions.