The MOSEK frequently asked questions.

Version 5.0 (Revision 138).

www.mosek.com

Published by MOSEK ApS, Denmark.

Contact information

Phone	+45 3917 9907	
Fax	+45 3917 9823	
WEB	http://www.mosek.com	
Email	sales@mosek.com	Sales, pricing, and licensing.
	support@mosek.com	Technical support, questions and bug reports.
	info@mosek.com	Everything else.

Mail

MOSEK ApS
C/O Symbion Science Park
Fruebjergvej 3, Box 16
2100 Copenhagen Ø
Denmark

Contents

1 General **1**
1.1 What os the meaning when MOSEK reports “Out of space”? 1
1.1.1 The command line tool 1
1.1.2 Java, .NET, and Python 2
1.2 Does MOSEK support 64 bit operating systems? 2
1.3 Does MOSEK always report the same solution to a problem? 2
1.4 How come I find a different solution when I write a problem to disk, then read and solve it? 3
1.5 Does MOSEK solve non-convex problems? 3
1.5.1 Why does MOSEK not solve non-convex problems? 3
1.6 MOSEK seems to crash. Is that a feature? 3
1.7 How do I set up environment variables? 4
1.7.1 MAC OSX 4
1.7.2 Windows 4
1.7.3 Linux/Solaris/Unix 4

2 Optimizer **5**
2.1 I do not like the solution MOSEK reports. It it possible to get a different optimal solution? 5

3 Licensing **7**
3.1 FlexLM is cumbersome to install and maintain. Do I really need it? 7

4 Interfaces and language bindings **9**
4.1 How do I use MOSEK from Delphi? 9

5 Optimization toolbox for MATLAB **11**
5.1 ??? Undefined function or variable ’mosekopt’? 11
5.2 Unable to load mex file 11
5.3 How to make `mosekopt` silent? 13
5.3.1 How to report a MOSEK related problem? 13

5.4 Why is `mosekopt` not an m file? 14

Chapter 1

General

1.1 What os the meaning when MOSEK reports “Out of space”?

MOSEK requires a certain amount of memory to store a given optimization problem. The exact ammount is proportional to the number of variables, constraints and non-zero entries in the problem. Moreover, some additional memory is required to optimize it. How much depends on the structural properties of the problem.

In general it is not possible to predict how much memory MOSEK will require to solve a particular problem.

1.1.1 The command line tool

If the MOSEK command line tool runs out of space it will report

```
Return code - 1051  [MSK_RES_ERR_SPACE]
```

This means the operating system — for instance Windows — cannot supply the memory MOSEK requires. Depending on the operating system different options exist to circumvent the problem.

- On 32bit Windows (and other 32bit systems) each process can normally only allocate up to 2GB memory, even if more physical memory is available. Please read Microsoft’s explanation for details. Starting from MOSEK version 3.1 then it is possible to make MOSEK use 3GB of RAM in 32 bit Windows if you make some changes to the boot of operating system as described in Microsoft’s explanation.

  Installing more physical memory may be a solution up to the 2GB or 3GB limit. If the requirement go beyond that it may be necessary to switch to a 64bit platform.

- On 64bit operating systems the per-process memory limit is significantly higher, and the limitation is usually the amount of physical memory installed on the machine. See Section 1.2 for details.

- Make the optimization problem smaller. Many problems can be scaled down by sacrificing a level of detail.

1.1.2 Java, .NET, and Python

Java, .NET, amd Python applications runs under a virtual machine. MOSEK shares memeory with the virtual machine. This implies it might be necessary to force the virtual machine to free unused memory by explicitly calling the garbage collector (for example before optimization is performed) in order to make sufficient memory available to MOSEK.

Apart from this the interfaces are subject to the same limits as the command line tool.

1.2 Does MOSEK support 64 bit operating systems?

If you want MOSEK to exploit more than 2GB or 3GB of RAM you should run a 64 bit version of MOSEK on a 64 bit operating system. Both Windows, Linux, and the UNIXes come in 64 bit versions. Below is summary of the 64 bit operating systems provided.

- Windows 64 bit edition - X86: This platform has various names — 64-X86, AMD64, EM64T etc.
- Linux: Linux 64 bit is supported on the 64-X86 platforms.
- MAC OSX: Currently, 64 bit MAC OSX is not supported by MOSEK. Please contact MOSEK support if you are interested in a 64 bit MAC OSX version.
- Solaris: Solaris is available in a 64 bit version.

1.3 Does MOSEK always report the same solution to a problem?

The short answer is **no**. The reason is that the associative law

$$(x+y)+z = x+(y+z)$$

does not hold excatly when finite precision computations are performed. On any computer MOSEK runs all computations are performed n finite precision.

The implication of that the associative law does not hold exactly is a reordering of the computations can lead to slightly different results.

Now MOSEK adapts itself to different hardware in order to improve performance. This usually has the implication that the ordering is reordered and hende different results may the results.

1.4 How come I find a different solution when I write a problem to disk, then read and solve it?

Writing a file may change the ordering of non-zeros in the sparse columns of A, and in some cases the order of variables or constraints. This may lead to different rounding errors in subsequent optimizations and ultimately slightly different solutions.

1.5 Does MOSEK solve non-convex problems?

MOSEK cannot solve non-convex problems. If a problem is determined to be con-convex during optimization, the optimizer will terminate and report an error.

1.5.1 Why does MOSEK not solve non-convex problems?

MOSEK does not solve non-convex problems for the following reasons:

- The algorithms employed in MOSEK require convexity and cannot be applied to non-convex problems.
- It is in general impossible to measure the optimality or quality of a solution for a non-convex problem. This fits very badly into the general MOSEK framework.

In many cases a convex approximation of a non-convex problem will behave significantly better in several ways:

- The algorithms applied by MOSEK will (theoretically) always converge toward a global optimum.
- MOSEK will often be able to determine infeasibility or unboundedness of a problem.
- Global optimality of a solution can be proved within small margins.

1.6 MOSEK seems to crash. Is that a feature?

In some cases MOSEK may abort and emit an assertion message describing some internal unexpected situation, or it may simply crash.

In any case, please send the complete assertion message (if available) along with any relevant system information to

`support@mosek.com`

Relevant information includes:

- The MOSEK version, the operation system, service pack version and machine architecture.
- For MOSEK command line tool: The environment in which MOSEK was executed.
- For Java: The version of the Java Virtual machine.
- For .NET: The framework version.
- The assertion message printed to the MOSEK error stream (if available).
- The crash message on Windows systems.

1.7 How do I set up environment variables?

1.7.1 MAC OSX

Please read Apples instructions at http://developer.apple.com/qa/qa2001/qa1067.html

1.7.2 Windows

1. Right-click on **My Computer** (either on the desktop or in the start-menu) and select **properties**.
2. Click on the thumbnail **Advanced**.
3. Click the button **Environment Variables**.
4. If you have administrator priviledges you can either set/modify for the current user (in **User vaiables**) or globally for all users (in **System variables**).
5. Finally, click **OK** to close the dialog box, the click **OK** in the **Environment variables** window and close the **System properties window** window.

In general, environment variables defining lists of paths (for example `PATH` or `CLASSPATH`) contains multiple path strings separated by "`;`" (semi-colon).

Changes to environment variables may not take effect until next login.

1.7.3 Linux/Solaris/Unix

Environment variables are usually defined in a file called `.bashrc`, `.tcshrc` or similarly in the users home directory, depending on the user's shell.

The exact syntax is usually described in the `man`-page for the user's shell (`bash`, `tchs` etc.).

Chapter 2

Optimizer

2.1 I do not like the solution MOSEK reports. It it possible to get a different optimal solution?

It is not possible to directly make MOSEK to report a specific optimal solution among all possible optimal solutions.

If one an optimal solution is “better” than another, which should be reflected by the model, for example by adding penalties that will make the unwanted solutions non-optimal.

If this is not possible, the desired result may be obtained by solving a sequence of problems: First solve the original problem, then fix the objective function (within reasonable margins) on the optimal solution value, add a penalties to the model and reoptimize.

Chapter 3

Licensing

3.1 FlexLM is cumbersome to install and maintain. Do I really need it?

Yes. Currently MOSEK does not support any alternatives to FlexLM.

Chapter 4

Interfaces and language bindings

4.1 How do I use MOSEK from Delphi?

Simple language bindings for using the MOSEK DLL from Delphi on Windows is available. Please the C API manual for further details.

Chapter 5

Optimization toolbox for MATLAB

5.1 ??? Undefined function or variable 'mosekopt'?

Matlab was unable to locate the `mosekopt` library file.

Most likely you have forgotten to do

```
addpath <root>\mosek\<version>\tools\toolbox\14sp3
```

within MATLAB. Therefore, MATLAB cannot locate the `mosekopt` mex file. Remember to adapt `<root>` and `<version>` to your system.

Normally you will have to do `addpath` command every time MATLAB is started, however, it can be avoided if the addpath command is added to the file

```
<matlab>toolbox\local\startup.m
```

where ¡matlab¿ is the MATLAB root directory.

Alternatively the permanent modification of the MATLAB path can be performed using the Matlab menu item

```
\File\Set Path
```

5.2 Unable to load mex file

- For Windows:

  MATLAB reports something like

  ```
  DLL load failed for mex file
  c:\mosek\5\tools\toolbox\14sp3\mosekopt.dll The
  specified procedure could not be found. ??? Invalid MEX-file
  ```

Operating system	Name of OS variable for shared library search
Linux,Solaris	`LD_LIBRARY_PATH`
MAC OSX	`DYLD_LIBRARY_PATH`

By default MATLAB cannot locate the MOSEK optimization toolbox functions. Therefore you must execute the `addpath` command within MATLAB to change the so-called `matlabpath` appropriately. Indeed `matlabpath` should include a path to the MOSEK optimization toolbox functions. Pleas see the section “Locating the toolbox functions” in the MOSEK optimization toolbox for MATLAB manual for details on setting `matlabpath`.

If you have initialized `matlabpath` the problem is most likely that MOSEK cannot load the MOSEK DLL. The reason for this is that the variable `PATH` does not have the appropriate value.

Please check the Installation manual for instructions about how to setup the operating system variable `PATH`.

- For Linus/Solaris/UNIX: MATLAB reports something similar to

```
Unable to load mex file:
/usr/local/mosek/2/toolbox/14sp3/mosekopt.mexglx.
libmosek.so.2.5: cannot open shared object file: No such file or
directory ??? Invalid MEX-file
```

The cause of the problem is that the shared library

```
libmosek.so.2.5
```

cannot be loaded. This problem normally is caused by that the OS environment variable

```
LD_LIBRARY_PATH
```

is not appropriately setup. This environment variable is not called `LD_LIBRARY_PATH` on all platforms and Table 5.2 shows the appropriate name for different platforms. Note you can inspect the value of the variable inside MATLAB by executing the command

```
!echo $LD_LIBRARY_PATH
```

Finally, it is worthwhile to consult the MOSEK optimization tools installation manual for further details.

- MAC OSX:

  On MAC OSX you should setup the `DYLD_LIBRARY_PATH` variable appropriately. This can be tricky. Particularly if using MOSEK with another application such as MATLAB and is invoking MATLAB by clicking on the MATLAB icon. In such case you should either create or update the file

  ```
  $HOME/.MacOSX/environment.plist
  ```

  which should at least contain

  ```
  <?xml version="1.0" encoding="UTF-8"?>
  <!DOCTYPE plist PUBLIC
            "-//Apple Computer//DTD PLIST 1.0//EN"
            "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
  <plist version="1.0">
  <dict>
      <key>\mosek{}LM_LICENSE_FILE</key>
      <string>/home/eda/mosek/4/licenses/mosek.lic</string>
      <key>DYLD_LIBRARY_PATH</key>
      <string>/home/eda/mosek/4/tools/platform/osx32ppc/bin</string>
  </dict>
  </plist>
  ```

  You should of course input the relevant paths in string scopes for your installation.

  Before the setup in this file takes effect you should logout and login.

5.3 How to make `mosekopt` silent?

Use the command call

```
mosekopt('minimize echo(0)',prob)
```

The `echo` is the important addition in the above line.

5.3.1 How to report a MOSEK related problem?

If you have problem you would like to report and have help to solve, then please do following within MATLAB

```
diary mosek.txt
version
mosekopt('debug(10)')
```

```
%
% Whatever MATLAB commands that shows the error.
%
diary off
```

MATLAB will then produce a file named `mosek.txt` that contains the MATLAB dialogue including the messages MOSEK produces. Therefore, please include the file `mosek.txt` in your email with your problem report. It makes a lot easier for us diagnose the problem.

If your problem only occurs for some specific data then it might be useful to save the data you feed into MOSEK into file which can be include in your problem report. The MATLAB command example

```
save mosek x y
```

shows how to save the two variable `x` and `y` to the file `mosek.mat`. Please do not include `mosek.mat` in your problem report, if the file is very large.

The problem report you can mail to `support@mosek.com`.

5.4 Why is `mosekopt` not an m file?

`mosekopt` is not m file but a MATLAB MEX file. It is NOT a m file because it acts as a gateway to the native MOSEK library. This is simply the most convenient and efficient way for you to use a DLL or shared library like MOSEK from MATLAB.