# FAQ-7 du manuel d’extraction des trajectoires / Soucis de mémoire avec Matlab

The following comes from:

<http://www.mathworks.fr/fr/help/matlab/matlab_prog/resolving-out-of-memory-errors.html>

## Resolving "Out of Memory" Errors

### General Suggestions for Reclaiming Memory

The MATLAB software generates an Out of Memory message whenever it requests a segment of memory from the operating system that is larger than what is currently available. When you see the Out of Memory message, use any of the techniques discussed under [Strategies for Efficient Use of Memory](http://www.mathworks.fr/fr/help/matlab/matlab_prog/strategies-for-efficient-use-of-memory.html) to help optimize the available memory. If the Out of Memory message still appears, you can try any of the following:

* Compress data to reduce memory fragmentation.
* If possible, break large matrices into several smaller matrices so that less memory is used at any one time.
* If possible, reduce the size of your data.
* Make sure that there are no external constraints on the memory accessible to MATLAB. (On The Open Group UNIX[[1](http://www.mathworks.fr/fr/help/matlab/matlab_prog/resolving-out-of-memory-errors.html%22%20%5Cl%20%22ftn.zmw57dd0e47303)] systems, use the limit command to check).
* Increase the size of the swap file. We recommend that you configure your system with twice as much swap space as you have RAM. See [Increasing System Swap Space](http://www.mathworks.fr/fr/help/matlab/matlab_prog/resolving-out-of-memory-errors.html#brh72ex-54), below.
* Add more memory to the system.

### Setting the Process Limit

The platforms and operating systems that MATLAB supports have different memory characteristics and limitations. In particular, the *process limit* is the maximum amount of virtual memory a single process (or application) can address. On 32-bit systems, this is the most important factor limiting data set size. The process limit must be large enough for MATLAB to store all of the data it is to process, any MATLAB program files, the MATLAB executable itself, and additional state information.

Where possible, choose an operating system that maximizes this number, that is, a 64-bit operating system. The following is a list of MATLAB supported operating systems and their process limits.

| **Operating System** | **Process Limit** |
| --- | --- |
| 32-bit MicrosoftWindows XP, Windows Vista™, Windows 7 | 2 GB |
| 32-bit Windows XP with 3 GB boot.ini switch or 32-bit Windows Vista or Windows 7 with increaseuserva set (see later)  | 3 GB |
| 32-bit Linux (Linux is a registered trademark of Linus Torvalds) | ~3 GB |
| 64-bit Windows or Linux running 32-bit MATLAB | ≤ 4 GB |
| 64-bit Windows, Apple Macintosh OS X, or Linux running 64-bit MATLAB | 8 TB |

To verify the current process limit of MATLAB on Windows systems, use the [memory](http://www.mathworks.fr/fr/help/matlab/ref/memory.html) function.

Maximum possible array: 583 MB (6.111e+008 bytes) \*

Memory available for all arrays: 1515 MB (1.588e+009 bytes) \*\*

Memory used by MATLAB: 386 MB (4.050e+008 bytes)

Physical Memory (RAM): 2014 MB (2.112e+009 bytes)

\* Limited by contiguous virtual address space available.

\*\* Limited by virtual address space available.

When called with one output variable, the memory function returns or displays the following values. See the function reference for [memory](http://www.mathworks.fr/fr/help/matlab/ref/memory.html) to find out how to use it with more than one output.

| **memory Return Value** | **Description** |
| --- | --- |
| MaxPossibleArrayBytes | Size of the largest single array MATLAB can currently create |
| MemAvailableAllArrays | Total size of the virtual address space available for data |
| MemUsedMATLAB | Total amount of memory used by the MATLAB process |

View the value against the Total entry in the Virtual Memory section. It is shown as 2 GB in the table, which is the default on Windows XP systems. On UNIX systems, see the ulimit command to view and set user limits including virtual memory.

### Disabling Java VM on Startup

On UNIX systems, you can increase the workspace size by approximately 400 MB if you start MATLAB without the Sun Java VM. To do this, use the command line option -nojvm to start MATLAB. This also increases the size of the largest contiguous block (and therefore the largest matrix) by about the same.

Using -nojvm comes with a penalty in that you will lose many features that rely on the Java software, including the entire development environment.

Starting MATLAB with the -nodesktop option does not save any substantial amount of memory.

Shutting down other applications and services (e.g., using msconfig on Windows systems) can help if total system memory is the limiting factor, but usually process limit (on 32-bit machines) is the main limiting factor.

### Increasing System Swap Space

The total memory available to applications on your computer is comprised of physical memory (RAM), plus a *page file*, or *swap file*, on disk. The swap file can be very large (e.g., 16 TB on 32-bit Windows, 512 TB on 64-bit Windows). The operating system allocates the virtual memory of each process to physical memory or to the swap file, depending on the needs of the system and other processes.

Most systems allow you to control the size of your swap file. The steps involved depend on the system you are running on.

|  |
| --- |
| **Note:**   There is no interface for directly controlling the swap space on Macintosh OS X systems. |

#### Windows Systems

Use the Windows Control Panel to change the size of the virtual memory paging file on your system. For more information, refer to the Windows help.

#### Linux Systems

You can change your swap space by using the mkswap and swapon commands. For more information on the above commands, type man followed by the command name at the Linux prompt.

### Using the 3GB Switch on Windows Systems

Microsoft Windows XP systems can allocate 3 GB (instead of the default 2 GB) to processes, if you set an appropriate switch in the boot.ini file of the system. MathWorks recommends that you only do this with Windows XP SP2 systems or later. This gives an extra 1 GB of virtual memory to MATLAB, not contiguous with the rest of the memory. This enables you to store more data, but not larger arrays, as these are limited by contiguous space. This is mostly beneficial if you have enough RAM (e.g., 3 or 4 GB) to use it.

After setting the switch, confirm the new value of the virtual memory after restarting your computer and using the [memory](http://www.mathworks.fr/fr/help/matlab/ref/memory.html) function.

[userview systemview] = memory;

systemview.VirtualAddressSpace

ans =

 Available: 1.6727e+009 % Virtual memory available to MATLAB.

 Total: 2.1474e+009 % Total virtual memory

For more documentation on this option, use the following URL:

<http://support.microsoft.com/kb/291988>

Similarly, on machines running Microsoft Windows Vista and Windows 7, you can achieve the same effect by using the command:

BCDEdit /set increaseuserva 3072

For more information on this option, go to the following website:

<http://msdn.microsoft.com>

### Freeing Up System Resources on Windows Systems

There are no functions implemented to manipulate the way MATLAB handles Microsoft Windows system resources. Windows systems use these resources to track fonts, windows, and screen objects. Resources can be depleted by using multiple figure windows, multiple fonts, or several UI controls. One way to free up system resources is to close all inactive windows. Windows system icons still use resources.

## Export an executable of the Matlab interface

You have created an interface using Matlab and you would like to use it as a standalone application, neither launching Matlab nor needing a licence? Do as follow:

* In the Matlab command window, type the command **deploytool**
* In the popup window, give a name to the project, choose the output directory and select the type **Windows Standalone Application**
* A new column appears on the right of the Matlab interface called Windows Standalone Application
* Click on **Add main file**  and select the main \*.m file of your interface
* Click on the cogwheel and select **Add MCR** and then **Embed the MCR in the package**
* Click on the cogwheel and select **Build**

That’s it, you know have the project in the output directory and the executable in the subfolder **distrib**. You now can use it without Matlab nor licence.