

MATLAB



Derzeit ist die Nutzung der Matlab-Lizenzen auf Hilbert eingeschränkt, da es nicht genügend Lizenzen für den Campus gibt.

Auf dem HPC-System stehen mehrere MatLab-Versionen zu Verfügung. Die Lizenzierung findet über den zentralen Lizenzserver im ZIM statt.

Die Software kann z.B. über module load MATLAB/R2020b geladen werden.

Da es nur eine knappe Anzahl an Lizenzen gibt empfiehlt es sich beim Abschicken des Jobs direkt eine MatLab-Lizenz mit anzufordern. Dadurch wird der Job nur gestartet, wenn eine zentrale Lizenz verfügbar ist. Dies ist möglich über die Resource matlablicence im Jobfile

```
#PBS -l matlablicence=1
```

oder über eine explizite Angabe beim qsub:

```
qsub -l matlablicence=1 matlab.job
```

Zusätzliche Ressourcen für besondere Erweiterungen

Für die folgenden Toolboxes gibt es nur eine geringe Anzahl an Lizenzen, daher kann es sein, dass Jobs lange Warten müssen. Dies hat jedoch immer noch den Vorteil, dass Jobs nicht einfach abstürzen.

Toolbox	Lizenzen	Resource
Parallel Computing Toolbox	20	matlab_dct_licence
Image Toolbox	20	matlab_image_licence
Statistics Toolbox	50	matlab_stats_licence
Optimization Toolbox	50	matlab_opt_licence
Signal Toolbox	20	matlab_signal_licence

Diese Ressourcen können wie folgt angefordert werden:

```
#PBS -l matlab_image_licence=1
```

verfügbare Versionen

R2010a

R2011a

R2011b

R2014a

R2015a

R2015b

R2018a

R2020b

Beispiel Job-File für Matlab

Es ist gute Praxis, Parameter wie z.B. Input-Files und deren Verzeichnisse von außen zu setzen und nicht direkt in die Skripte einzutragen. Das gibt auch die Möglichkeit, MATLAB-Programme zu kompilieren und dann immer wieder mit anderen Parametern aufzurufen, ohne den Source ändern zu müssen.

myscript.m

```
% input_filename should be set in the call
if exist('input_filename') == 0
    fprintf("No input_filename given!\n");
    exit();
end

display(input_filename);

fprintf("DONE.\n");
```

Die Variable "input_filename" kann dann im JobScript beliebig gesetzt werden, was eine enorme Flexibilität und intrinsische Dokumentation ermöglicht.

matlab.job

```
#!/bin/bash

#PBS -l select=1:ncpus=###CORES###:mem=###MEMORY###
#PBS -l walltime=24:00:00
#PBS -r n
#PBS -N MATLAB
#PBS -A ###MYJAM PROJECT###
#PBS -l matlablicence=1

export LOGFILE=$PBS_O_WORKDIR/$PBS_JOBNAME"."$PBS_JOBID".log"

cd $PBS_O_WORKDIR

echo "$PBS_JOBID ($PBS_JOBNAME) @ `hostname` at `date` in "$RUNDIR" START"
> $LOGFILE
echo "`date +%d.%m.%Y-%T`" >> $LOGFILE

#load MATLAB Environment and set scratch directory
module load MATLAB/R2020b

echo >> $LOGFILE
echo "GLOBAL PARAMETERS" >> $LOGFILE
echo "-----" >> $LOGFILE
echo "Node      : "`hostname` >> $LOGFILE
echo "RunDir     : "$PBS_O_WORKDIR >> $LOGFILE
echo "MATLAB      : "$MATLAB_ROOT >> $LOGFILE
echo "# CPUs     : "$NCPUS >> $LOGFILE
echo >> $LOGFILE

input=/home/raub/run/MATLAB/test.input

#calling MATLAB
matlab -nosplash -nodesktop -nodisplay -r "input_filename='$input';
myscript"

#print the last known statistics of the job (memory usage, cpu time,
etc...)
echo >> $LOGFILE
qstat -f $PBS_JOBID >> $LOGFILE

echo "$PBS_JOBID ($PBS_JOBNAME) @ `hostname` at `date` in "$RUNDIR" END"
>> $LOGFILE
echo "`date +%d.%m.%Y-%T`" >> $LOGFILE
```

Die Platzhalter ###CORES###, ###MEMORY### und ###MYJAM PROJECT### müssen natürlich noch angepasst werden.

Der Output sollte dann wie folgt aussehen:

```
< M A T L A B (R) >
```

```
Copyright 1984-2020 The MathWorks, Inc.
```

```
R2020b
```

```
(9.9.0.1467703) 64-bit (glnxa64)
```

```
August 26, 2020
```

```
To get started, type doc.
```

```
For product information, visit www.mathworks.com.
```

```
input_filename =
```

```
    '/home/raub/run/MATLAB/test.input'
```

```
DONE.
```