Licensed Software available on the HPC Cluster

Looking for one-on-one HPC help?
Email us to book an HPC Consultation
Spotted any errors or omissions in our new Wiki?
Tell us about it!
Getting an account
Getting started on Prince
Prince How-to Articles
Logging in
Windows Mac / Linux
Clusters and Storage
Prince (HPC)
Dumbo (Hadoop)
Brooklyn (OpenStack)
Dalma (NYU Abu Dhabi)
| Transfer data to/from the clusters | Transfer data to/from clusters using Globus submitting jobs with sotware available so
Licensed Software Available on the HP Cluster Building Software Slurm Tutorial Tutorials FAQs Scratch Area Cleanup Programming for Biologists
COMSOL

COMSOL is a problem-solving simulation environment, enforcing compatibility guarantees consistent multiphysics models. COMSOL Multiphysics is a general-purpose software platform, based on advanced numerical methods, for modeling and simulating physics-based problems. The package is cross-platform (Windows, Mac, Linux). The COMSOL Desktop helps you organize your simulation by presenting a clear overview of your model at any point. It uses functional form, structure, and aesthetics as the means to achieve simplicity for modeling complex realities.

NOTE: This license is for academic use only with Floating Network Licensing in nature i.e., authorized users are allowed to use the software on desktops. Please contact hpc@nyu.edu for the license. However, COMSOL is also available on NYU HPC cluster Prince.

Here is the list of modules available and concurrent users count i.e., authorized to use the software.

<table>
<thead>
<tr>
<th>Module Name</th>
<th>Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMSOL Multiphysics</td>
<td>3</td>
</tr>
<tr>
<td>AC/DC Module</td>
<td>1</td>
</tr>
<tr>
<td>CFD Module</td>
<td>2</td>
</tr>
<tr>
<td>Chemical Reaction Engineering Module</td>
<td>1</td>
</tr>
<tr>
<td>Heat Transfer Module</td>
<td>1</td>
</tr>
<tr>
<td>LiveLink for Excel</td>
<td>2</td>
</tr>
<tr>
<td>LiveLink for MATLAB</td>
<td>2</td>
</tr>
<tr>
<td>Material Library</td>
<td>2</td>
</tr>
<tr>
<td>Microfluidics Module</td>
<td>2</td>
</tr>
<tr>
<td>Molecular Flow Module</td>
<td>2</td>
</tr>
<tr>
<td>Particle Training Module</td>
<td>2</td>
</tr>
<tr>
<td>RF Module</td>
<td>1</td>
</tr>
<tr>
<td>Wave Optics Module</td>
<td>1</td>
</tr>
<tr>
<td>Structural Mechanics</td>
<td>1</td>
</tr>
</tbody>
</table>

License Availability on Prince:

Use command “comsol_licenses” at linux command prompt on hpc cluster (i.e., prince) to verify unused licenses from available list.

Using COMSOL on HPC Cluster:

Several versions of COMSOL are available on the HPC cluster. To use COMSOL on the HPC cluster, please load the relevant module in your batch job submission script:

```
module load comsol/5.3
```

Running a parallel COMSOL job on HPC cluster (Prince):

To submit a COMSOL job for running on multiple processing elements, follow below steps.

Step-1:
Create a directory on "scratch" as given below.

```
mkdir /scratch/<net_id>/example
cd /scratch/<net_id>/example
```

**Step-2:**

Copy example files to your newly created directory.

```
cp /share/apps/examples/comsol/run-comsol.sbatch /scratch/<net_id>/example/
cp /share/apps/examples/comsol/test-input.mph /scratch/<net_id>/example/
```

**Step-3:**

Edit the Sbatch script file (run-comsol.sbatch) accordingly as given below.

```
"cd /share/apps/examples/comsol" change to "cd /scratch/<net_id>/example/
```

**Step-4:**

Once the Sbatch script file is ready, it can be submitted to the job scheduler using sbatch. After successful completion of job, verify output log file for detail output information.

```
sbatch run-comsol.sbatch
```

**MATHEMATICA**

Mathematica is a general computing environment with organizing algorithmic, visualization, and user interface capabilities. The many mathematical algorithms included in Mathematica make computation easy and fast.

**Using Mathematica on HPC Cluster:**

To run Mathematica on the HPC cluster, please load the relevant module in your batch job submission script:

```
module load mathematica/11.0.1
```

Note: In the example below the module is loaded already in the sbatch script.

**Running a parallel Mathematica job on HPC cluster (Prince):**

To submit a Mathematica job for running on multiple processing elements, follow below steps.
Step-1:
Create a directory on "scratch" as given below.

```
Working on scratch
mkdir /scratch/<net_id>/example
cd /scratch/<net_id>/example
```

Step-2:
Copy example files to your newly created directory.

```
Copy Example
cp /share/apps/examples/mathematica/basic/example.m
   /scratch/<net_id>/example/
cp /share/apps/examples/mathematica/basic/run-mathematica.sbatch
   /scratch/<net_id>/example/
```

Step-3:
Edit the sbatch script file (run-mathematica.sbatch) accordingly as given below.

```
"cd /share/apps/examples/mathematica" change to "cd /scratch/<net_id>/example"
```

Step-4:
Once the sbatch script file is ready, it can be submitted to the job scheduler using sbatch. After successful completion of job, verify output log file generated.

```
sbatch
sbatch run-mathematica.sbatch
```

MATLAB
MATLAB is a technical computing environment for high performance numeric computation and visualization. MATLAB integrates numerical analysis, matrix computation, signal processing, and graphics in an easy to use environment without using traditional programming.

**MATLAB on personal computers and laptops**
NYU has a Total Academic Headcount (TAH) license which provides campus-wide access to MATLAB, Simulink, and a variety of add-on products. All faculty, researchers, and students (on any NYU campus) can use MATLAB on their personal computers and laptops and may go to the following site to download the NYU site license software free of charge.


MATLAB can be used for non-commercial, academic purposes.

**Using MATLAB on HPC Cluster:**
There are several versions of Matlab available on the cluster and the relevant version can be loaded.
Running a MATLAB job on a compute node:
Run the srun command below to get a compute node:

```
srun --pty /bin/bash
```

After getting access to a compute node navigate to the folder using “cd /share/apps/examples/matlab/basic/”. You can look at the sample basic sbatch script used to run the MATLAB job and create your own script. To run the script the below command can be used.

```
sbatch run-matlab.sbatch
```

SAS

SAS is a software package which enables programmers to perform information retrieval and data management, report writing and graphics, statistical analysis and data mining, business planning, forecasting, and decision support, operations research and project management, quality improvement, applications development, data warehousing (extract, transform, load), and platform independent and remote computing.

There are licenses for 2 CPUs on the HPC Cluster.

Using SAS on HPC Cluster:
To run SAS on the HPC cluster, please load the relevant module in your batch job submission script:

```
sas module
```

Note: In the example below the module is loaded already in the sbatch script.

Running a parallel SAS job on HPC cluster (Prince):
To submit a SAS job for running on multiple processing elements, follow below steps.

Step-1:
Create a directory on “scratch” as given below.
### Working on scratch

```bash
mkdir /scratch/<net_id>/example
cd /scratch/<net_id>/example
```

**Step-2:**

Copy example files to your newly created directory.

### Copy Example

```bash
cp /share/apps/examples/sas/test.sas /scratch/<net_id>/example/
cp /share/apps/examples/sas/test2.sas /scratch/<net_id>/example/
cp /share/apps/examples/sas/run-sas.sbatch /scratch/<net_id>/example/
```

**Step-3:**

Edit the sbatch script file (run-sas.sbatch) accordingly as given below.

```
"cd /share/apps/examples/sas" change to "cd /scratch/<net_id>/example/"
```

**Step-4:**

Once the sbatch script file is ready, it can be submitted to the job scheduler using sbatch. After successful completion of job, verify output log file generated.

```bash
sbatch run-sas.sbatch
```

### STATA

Stata is a command and menu-driven software package for statistical analysis. It is available for Windows, Mac, and Linux operating systems. Most of its users work in research. Stata’s capabilities include data management, statistical analysis, graphics, simulations, regression and custom programming.

**Using STATA on HPC Cluster:**

To run STATA on the HPC cluster, please load the relevant module in your batch job submission script:

```bash
module load stata/14.0
```

**Note:** In the example below the module is loaded already in the sbatch script.

**Running a parallel STATA job on HPC cluster (Prince):**

To submit a STATA job for running on multiple processing elements, follow below steps.
Step-1:
Create a directory on "scratch" as given below.

```
mkdir /scratch/<net_id>/example
cd /scratch/<net_id>/example
```

Step-2:
Copy example files to your newly created directory on scratch.

```
cp /share/apps/examples/stata/run-stata.sbatch /scratch/<net_id>/example/
cp /share/apps/examples/stata/stata-test.do /scratch/<net_id>/example/
```

Step-3:
Edit the sbatch script file (run-stata.sbatch) accordingly as given below.

```
"cd /share/apps/examples/stata" change to "cd /scratch/<net_id>/example/"
```

Step-4:
Once the sbatch script file is ready, it can be submitted to the job scheduler using sbatch. After successful completion of job, verify output log file generated.

```
sbatch run-stata.sbatch
```

GAUSSIAN

Gaussian uses basic quantum mechanic electronic structure programs. This software is capable of handling proteins and large molecules using semi-empirical, ab initio molecular orbital (MO), density functional, and molecular mechanics calculations.

**Using Gaussian on HPC Cluster:**

To run Gaussian on the HPC cluster, please load the relevant module in your batch job submission script:

```
module load gaussian/intel/g16a03
```

Note: In the example below the module is loaded already in the sbatch script.

**Running a parallel Gaussian job on HPC cluster (Prince):**

To submit a Gaussian job for running on multiple processing elements, follow below steps.
Step 1:

Create a directory on "scratch" as given below.

```
Working on scratch
mkdir /scratch/<net_id>/example
cd /scratch/<net_id>/example
```

Step 2:

Copy example files to your newly created directory.

```
Copy Example
cp /share/apps/examples/gaussian/test435.com /scratch/<net_id>/example/
cp /share/apps/examples/gaussian/run-gaussian.sbatch /scratch/<net_id>/example/
```

Step 3:

Edit the sbatch script file (run-gaussian.sbatch) accordingly as given below.

```
"cd /share/apps/examples/gaussian" change to "cd /scratch/<net_id>/example"
```

Step 4:

Once the sbatch script file is ready, it can be submitted to the job scheduler using sbatch. After successful completion of job, verify output log file generated.

```
sbatch
sbatch run-guassian.sbatch
```

---

### Rogue Wave (formerly Totalview)

Totalview from Rogue Wave Software provides a set of parallel debugging tools that give users control over program execution within a single thread or multiple groups of processes or threads. It comes with GUI and a command line interface. Job scripts on the Prince cluster can call a line as below to get the Totalview environment:

```
totalview module
module load totalview/8.15.4-6
```

For more information on the software, please visit its [product web page](#), and read its [documentation page](#).

---

### Knitro
**Knitro** is a commercial software package for solving large scale mathematical optimization problems. Knitro is specialized for nonlinear optimization, but also solves linear programming problems, quadratic programming problems, systems of nonlinear equations, and problems with equilibrium constraints. The unknowns in these problems must be continuous variables in continuous functions; however, functions can be convex or nonconvex. Knitro computes a numerical solution to the problem—it does not find a symbolic mathematical solution. Knitro versions 9.0.1 and 10.1.1 are available.

### totalview module

```
module load knitro/10.2.1
```

**Running a parallel Knitro job on HPC cluster (Prince):**

To submit a Knitro job for running on multiple processing elements, follow below steps.

**Step-1:**

Create a directory on "scratch" as given below.

```
mkdir /scratch/<net_id>/example
```

```
cd /scratch/<net_id>/example
```

**Step-2:**

Copy example files to your newly created directory.

```
cp /share/apps/knitro/10.2.1/examples/Python/knitro.py /scratch/<net_id>/example/
```

**Step-3:**

There is no sample sbatch script available for knitro.

**Step-4:**

After creating your own sbatch script you can execute it as follows:

```
sbatch
```

```
sbatch <script>.sbatch
```

**Working on scratch**

```
mkdir /scratch/<net_id>/example
```

```
cd /scratch/<net_id>/example
```

**Copy Example**

```
cp /share/apps/knitro/10.2.1/examples/Python/knitro.py /scratch/<net_id>/example/
```