Visualization Consulting

UA HPC offers consulting services in visualization. You can send an email to vislab-consult@list.arizona.edu for technical support from Data & Visualization consultant. To see a gallery of past projects please visit https://rtdatavis.github.io/.

Software

On this page:

- Visualization Consulting
- Software
  - Paraview
  - Blender
  - Visit
- GUIs Through Open OnDemand
- VPN - Virtual Private Network
- X11 Examples
  - Ansys Workbench
  - GLX - Useful for Visit

Within our visualization documentation:

- Creating Heightmaps ArcGIS Pro
- VisIt Workflows on UA HPC
- Visualization With Blender
  - Blender Command Line Rendering
  - Scaling up Blender rendering
- Visualization With ParaView
  - Essential Ideas for ParaView
  - Getting Started With ParaView
  - ParaView Cameras and Keyframes
  - Visualizing NetCDF Files
  - Getting Started With The Paraview Terminal (PvPython)
  - Graphs and Exporting Data
  - Paraview Headless Batch Rendering
  - Speeding up Filters & Rendering with Client Server configurations
  - Advanced Computational Geometry in Paraview with CGAL & Vespa
- VPN - Virtual Private Network
Paraview is available as a module. You will use those in server mode where the calculations are done on HPC nodes and the rendering is back on your workstation. View our material on Visualization With ParaView for more information.

- Essential Ideas for ParaView
- Getting Started With ParaView
- ParaView Cameras and Keyframes
- Visualizing NetCDF Files
- Getting Started With The Paraview Terminal (PvPython)
- Graphs and Exporting Data
- Paraview Headless Batch Rendering
- Speeding up Filters & Rendering with Client Server configurations
- Advanced Computational Geometry in Paraview with CGAL & Vespa

Blender

Blender is a tremendously useful open source 3D modeling program. Detailed instructions for Blender are available in several tutorials.

- Blender Command Line Rendering
- Scaling up Blender rendering

Visit

Step by step instructions for using Visit are on this page, and it is also available as a module on our system.

GUIs Through Open OnDemand

You can utilize the GUI version of an application using a Virtual Desktop or through one of our interactive applications using Open OnDemand.

When you choose an application from the Interactive Apps dropdown at the top of the page and enter the resources you need (like cores and wall time hours) you will have access to a session that will let you use a compute node interactively through a virtual desktop or software-specific GUI application.
A VPN service (Virtual Private Network) is available for HPC, primarily for applications that cannot navigate the bastion host for visualization, and because the performance is frequently better than tunneling through the Bastion host. This is separate from the UA VPN. We suggest you first try the Desktop application of OnDemand for GUI applications as shown for Matlab immediately above.

### X11 Examples

The alternative to using the Open OnDemand desktop is to use X11 tunneling, though it should be noted that in most cases the performance is not as good. The examples below use X11 forwarding to access the graphical interfaces for the listed software.

#### Ansys Workbench

In this example, we use the GUI for launching Ansys projects. Note: Ansys is also available as a GUI application through Open OnDemand. You can expect it to be much faster using OnDemand.

```bash
$ ssh -X netid@hpc.arizona.edu
$ shell -X
$ interactive -x -a your_group # starts an interactive session for one hour. The -x option will enable image forwarding
$ module load cudall # Provides the GLX functionality needed by some components like CFD Post
$ module load ansys
$ runwb2
```

#### GLX - Useful for VisIt

The functionality of GLX requires synchronization between your workstation and the compute nodes. Keep in mind that the login nodes do not have a graphics so you must request a compute using an interactive session.

These steps provide a practical example for using the Visualization software called VisIt.
$ ssh -X netid@hpc.arizona.edu
$ shell -X
$ interactive -a your_group -x # The -x flag provides the graphics tunneling
$ module load cudall # Provides support for GLX. A GPU node is not required
$ glxinfo # **See notes below if you get an error during this step
$ glxspheres64 # This is a good test of rendering back on the workstation
$ module load visit
$ visit

** You might get a "Bad Value" error on the Mac command line. This can be fixed by entering the following on the command line

$ defaults write org.macosforge.xquartz.X11 enable_iglx -bool true

On a Linux workstation, you might need to create a file /etc/X11/xorg.conf with the following section:

```
Section "ServerFlags"
  Option "AllowIndirectGLX" "on"
  Option "IndirectGLX" "on"
EndSection
```