

OpenFOAM-12 tutorial with Desktop App

Using the Desktop App is one way to run OpenFOAM interactively. This can be a very useful tool during the early stages of design and development. It is recommended to run large simulations in batch mode.

Design your simulation

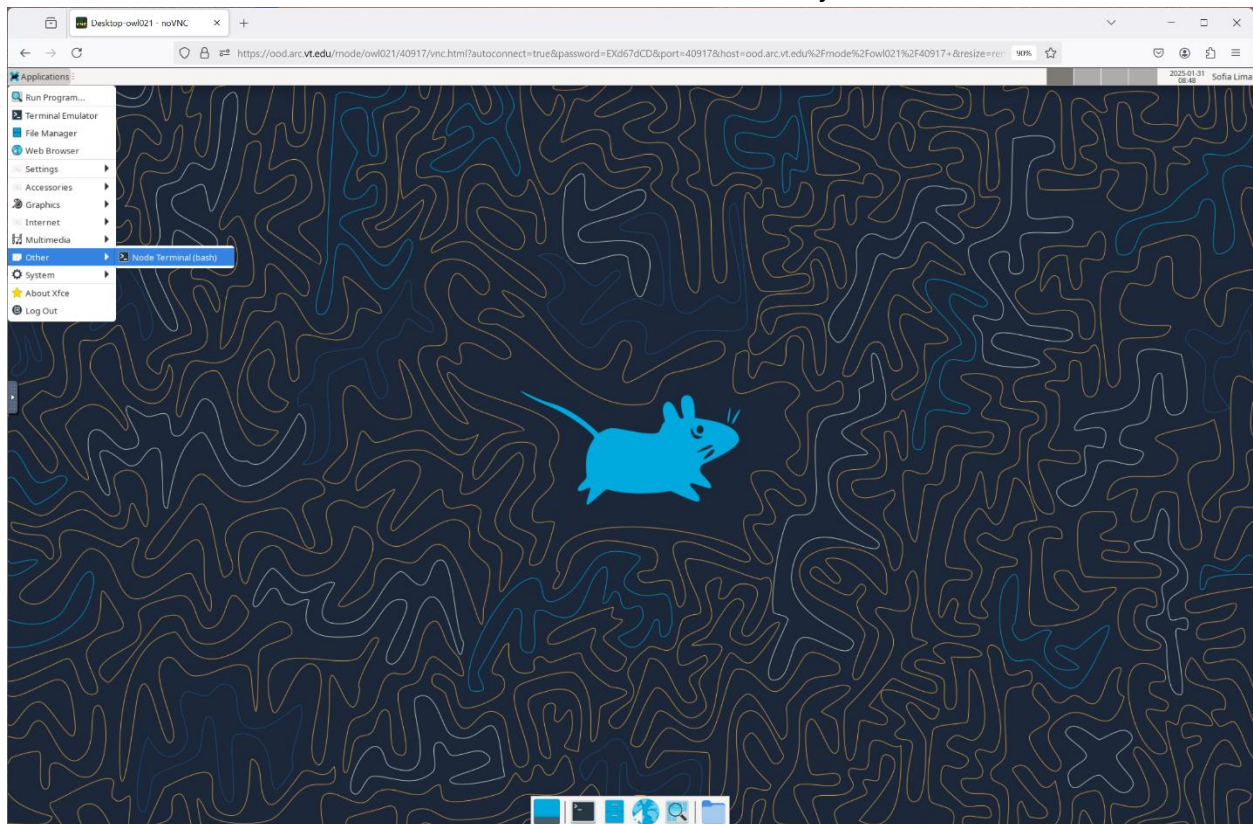
We will be using a simple incompressible Fluid example provided with the installation. We will copy the necessary data into our working directory. This example is designed to run with 4 processes in parallel; thus we will request 4 CPU processors (aka “cores” or “chips”).

The screenshot shows the ARC Open OnDemand web interface. The browser address bar displays the URL: https://ood.arc.vt.edu/jun/sys/dashboard/batch_connect/sys/bc_vt_desktop_alpine/session_contexts/new. The top navigation bar includes links for Files, Jobs, Clusters, Interactive Apps, Legacy Apps, and My Interactive Sessions. The user is logged in as 'slima2' and can click on 'Log Out'.

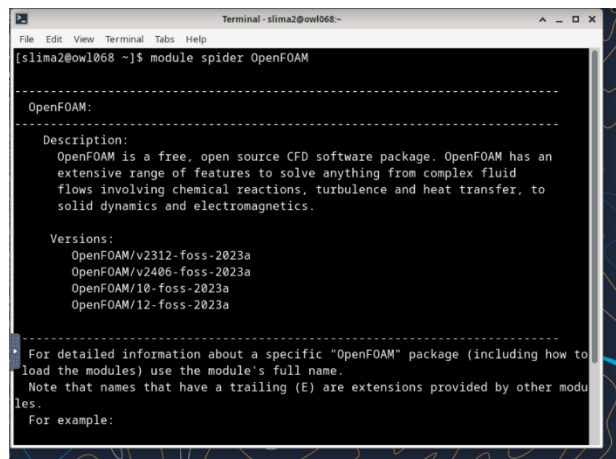
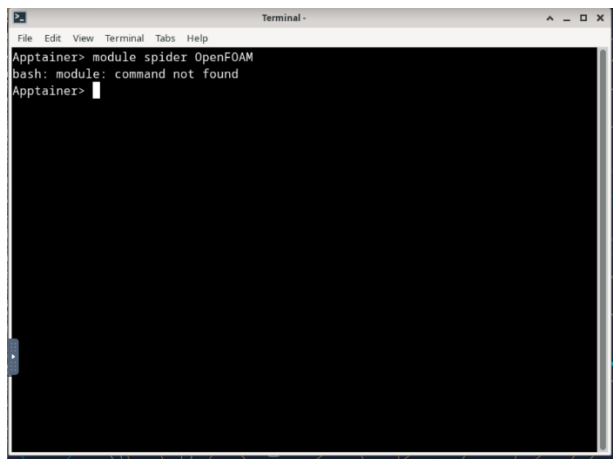
The main content area is titled 'Owl Desktop' and includes the following sections:

- Interactive Apps:** A list of available applications including Jupyter Notebook for Python and Julia, RStudio, Spyder, JupyterLab, MATLAB, and Owl Desktop (which is selected).
- Legacy Apps:** A list of legacy applications including IDE, Code Server, Eclipse, Nvidia-Nsight-Eclipse, PyCharm, AMDuProf, Ansys Workbench, DeepLabCut, Desktop, GAMS Studio, IGV, and Jupyter Notebook -- test conda.
- Owl Desktop Configuration:**
 - Description:** This app will launch a VNC Server in a container on one node.
 - Account:** A text input field containing 'arcadm'. Below it, a note states: "The allocation you would like to use for SLURM."
 - Partition:** A dropdown menu set to 'normal_q'.
 - Number of hours (min-1, max-48):** A text input field containing '1'.
 - Number of cores per node (min-1, max-128):** A text input field containing '4'.
 - Launch:** A blue button to start the session.
 - Footer Note:** * The Owl Desktop session data for this session can be accessed under the [data root directory](#).

We will use the Node Terminal to execute commands interactively.

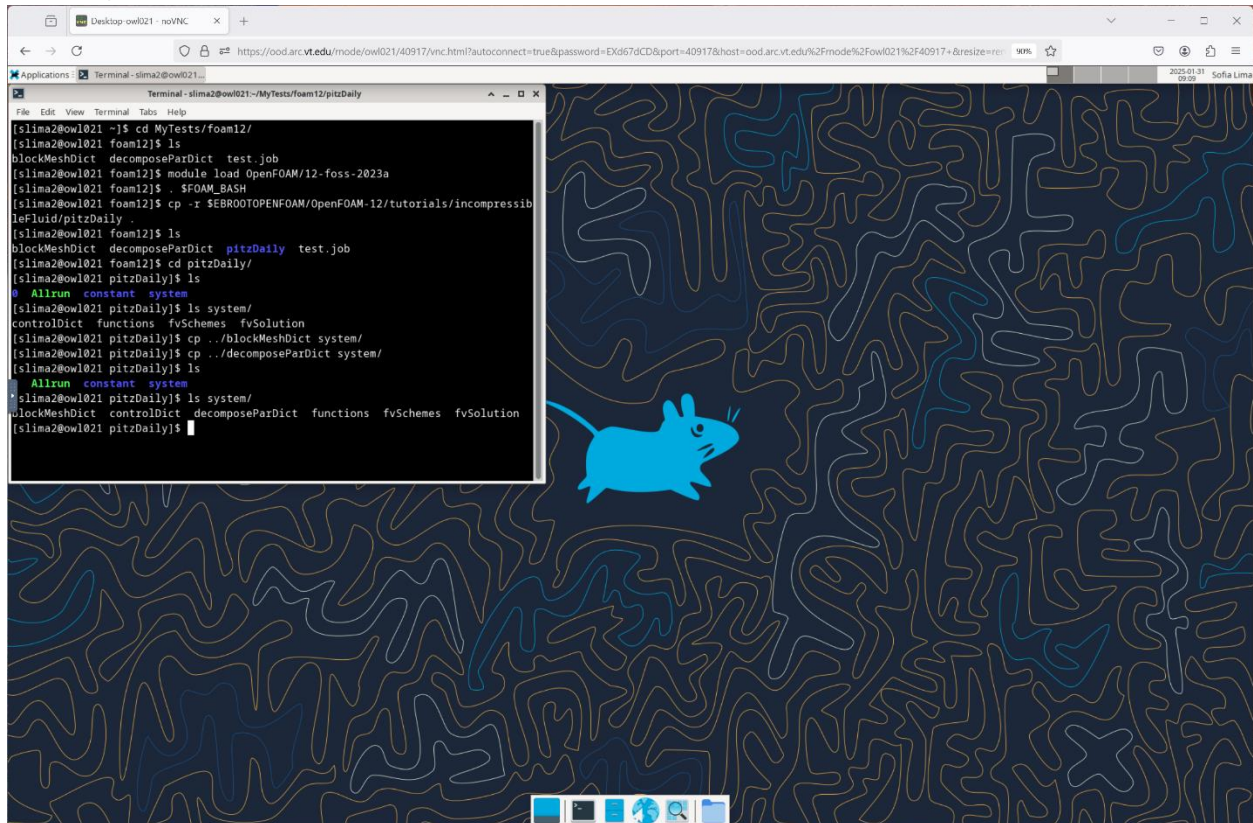


This terminal gives us access to a shell on the compute node with X-forwarding to allow for graphical rendering. This is different from the Terminal Emulator, which is a shell inside the container with “Apptainer>” on the command prompt.



Setup your working directory

First we will load the OpenFOAM software. Loading this module will define some environment variables, including \$FOAM_BASH and \$EBROOTOPENFOAM. OpenFOAM is unusual because we must source (.) the bash script \$FOAM_BASH in order to call the openfoam executables like blockMesh, decomposePar, and foamRun. Then, we will copy the example data into our working directory.

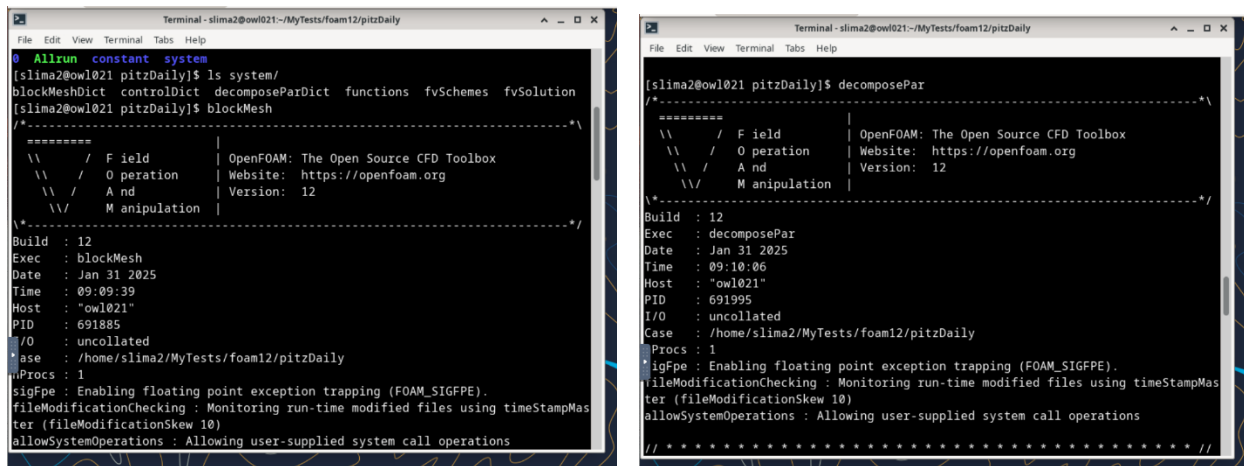


```
Terminal - slima2@owl021 - /MyTests/foam12/pitzDaily
[slima2@owl021 ~]$ cd MyTests/foam12/
[slima2@owl021 foam12]$ ls
blockMeshDict  decomposeParDict  test.job
[slima2@owl021 foam12]$ module load OpenFOAM/12-foss-2023a
[slima2@owl021 foam12]$ . $FOAM_BASH
[slima2@owl021 foam12]$ cp -r $EBROOTOPENFOAM/OpenFOAM-12/tutorials/incompressibleFluid/pitzDaily .
[slima2@owl021 foam12]$ ls
blockMeshDict  decomposeParDict  pitzDaily  test.job
[slima2@owl021 foam12]$ cd pitzDaily/
[slima2@owl021 pitzDaily]$ ls
0  Allrun  constant  system
[slima2@owl021 pitzDaily]$ ls system/
controlDict  functions  fvSchemes  fvSolution
[slima2@owl021 pitzDaily]$ cp ../blockMeshDict system/
[slima2@owl021 pitzDaily]$ cp ../decomposeParDict system/
[slima2@owl021 pitzDaily]$ ls
0  Allrun  constant  system
[slima2@owl021 pitzDaily]$ ls system/
blockMeshDict  controlDict  decomposeParDict  functions  fvSchemes  fvSolution
[slima2@owl021 pitzDaily]$
```

Run OpenFOAM

We will use openfoam commands to achieve our computational goals. We will follow the process to accelerate runtime by executing our simulation across multiple cores in parallel. This experiment is designed to run on 4 blocks of the mesh with 1 processor for each.

First we will run openfoam to block the mesh and decompose for parallel execution.

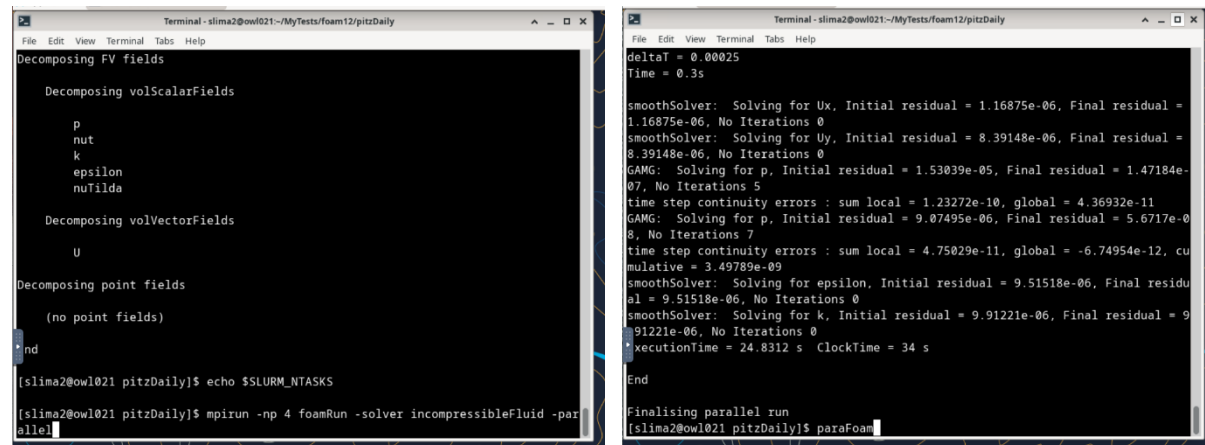


The first terminal window shows the execution of the `blockMesh` command. The output displays the OpenFOAM version (12), the build date (Jan 31 2025), and the host (owl021). It also shows the execution path and the number of processors (1). The second terminal window shows the execution of the `decomposePar` command. The output displays the OpenFOAM version (12), the build date (Jan 31 2025), and the host (owl021). It also shows the execution path and the number of processors (1).

```
[slima2@owl021 ~]$ ls system/
blockMeshDict controlDict decomposeParDict functions fvSchemes fvSolution
[slima2@owl021 ~]$ blockMesh
*****
\\ / F i e l d      | OpenFOAM: The Open Source CFD Toolbox
\\ / O p e r a t i o n | Website: https://openfoam.org
\\ / A n d           | Version: 12
\\ / M a n i p u l a t i o n |
*****
Build : 12
Exec  : blockMesh
Date  : Jan 31 2025
Time  : 09:09:39
Host  : "owl021"
PID   : 691885
/n0   : uncollated
ase   : /home/slima2/MyTests/foam12/pitzDaily
Procs : 1
sigFpe : Enabling floating point exception trapping (FOAM_SIGFPE).
fileModificationChecking : Monitoring run-time modified files using timeStampMaster (fileModificationSkew 10)
allowSystemOperations : Allowing user-supplied system call operations

[slima2@owl021 ~]$ decomposePar
*****
\\ / F i e l d      | OpenFOAM: The Open Source CFD Toolbox
\\ / O p e r a t i o n | Website: https://openfoam.org
\\ / A n d           | Version: 12
\\ / M a n i p u l a t i o n |
*****
Build : 12
Exec  : decomposePar
Date  : Jan 31 2025
Time  : 09:10:06
Host  : "owl021"
PID   : 691995
/n0   : uncollated
Case  : /home/slima2/MyTests/foam12/pitzDaily
Procs : 1
sigFpe : Enabling floating point exception trapping (FOAM_SIGFPE).
fileModificationChecking : Monitoring run-time modified files using timeStampMaster (fileModificationSkew 10)
allowSystemOperations : Allowing user-supplied system call operations
*****
```

Now we will run our simulation in parallel with OpenMPI. We will see the log in standard output; we could have redirected the output to a file by adding `> run_output.log` to the end of the command.



The first terminal window shows the execution of the `paraFoam` command. The output displays the decomposition of the mesh into 4 blocks. The second terminal window shows the execution of the `paraFoam` command. The output displays the simulation results, including the time step, the number of iterations, and the residuals.

```
[slima2@owl021 ~]$ paraFoam
Decomposing FV fields
Decomposing volScalarFields
p
nut
k
epsilon
nuTilda
Decomposing volVectorFields
U
Decomposing point fields
(no point fields)
*nd
[slima2@owl021 ~]$ echo $SLURM_TASKS
[slima2@owl021 ~]$ mpirun -np 4 foamRun -solver incompressibleFluid -parallel
[slima2@owl021 ~]$ paraFoam
deltaT = 0.00025
Time = 0.3s
smoothSolver: Solving for Ux, Initial residual = 1.16875e-06, Final residual = 1.16875e-06, No Iterations 0
smoothSolver: Solving for Uy, Initial residual = 8.39148e-06, Final residual = 8.39148e-06, No Iterations 0
GAMG: Solving for p, Initial residual = 1.53039e-05, Final residual = 1.47184e-07, No Iterations 5
time step continuity errors : sum local = 1.23272e-10, global = 4.36932e-11
GAMG: Solving for p, Initial residual = 9.07495e-06, Final residual = 5.6717e-08, No Iterations 7
time step continuity errors : sum local = 4.75029e-11, global = -6.74954e-12, cumulative = 3.49789e-09
smoothSolver: Solving for epsilon, Initial residual = 9.51518e-06, Final residual = 9.51518e-06, No Iterations 0
smoothSolver: Solving for k, Initial residual = 9.91222e-06, Final residual = 9.91222e-06, No Iterations 0
*ecutionTime = 24.8312 s ClockTime = 34 s
End
Finalising parallel run
[slima2@owl021 ~]$ paraFoam
```

Because we are on the Desktop app which gives us the access to a shell with X-forwarding, we can use the `paraFoam` command to visualize the results.

