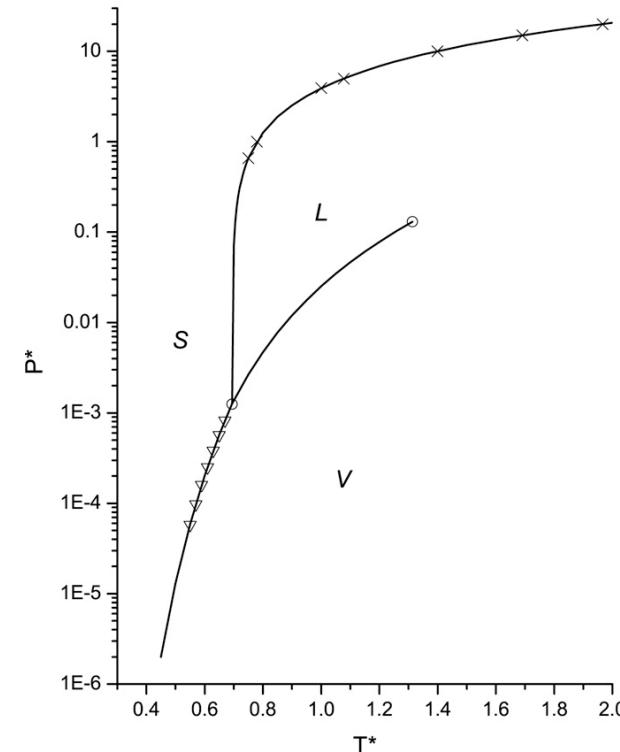


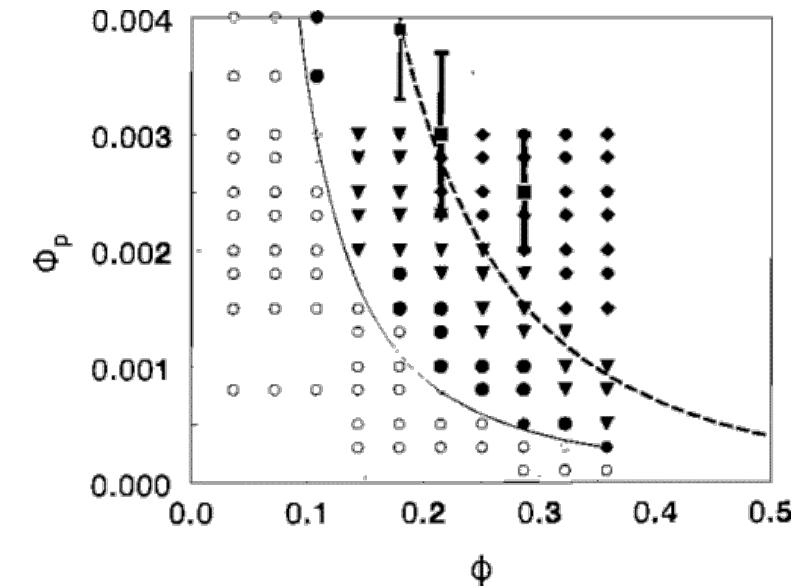
# Simple Fluids 2: Batch Jobs & Analyzing Phases

# The Problem: Determining a Phase Diagram

- Phase diagram of a Lennard-Jones system
- Hard, correct way: *free energy*
- Simpler, fraught way: check a grid
- That's a lot of points – how to run *many* simulations?
- This session:
  - Choose a Pressure-Temperature space
  - set pressure, run a *melting study* and record melting point
- We have 1 hour – statistics are bad!



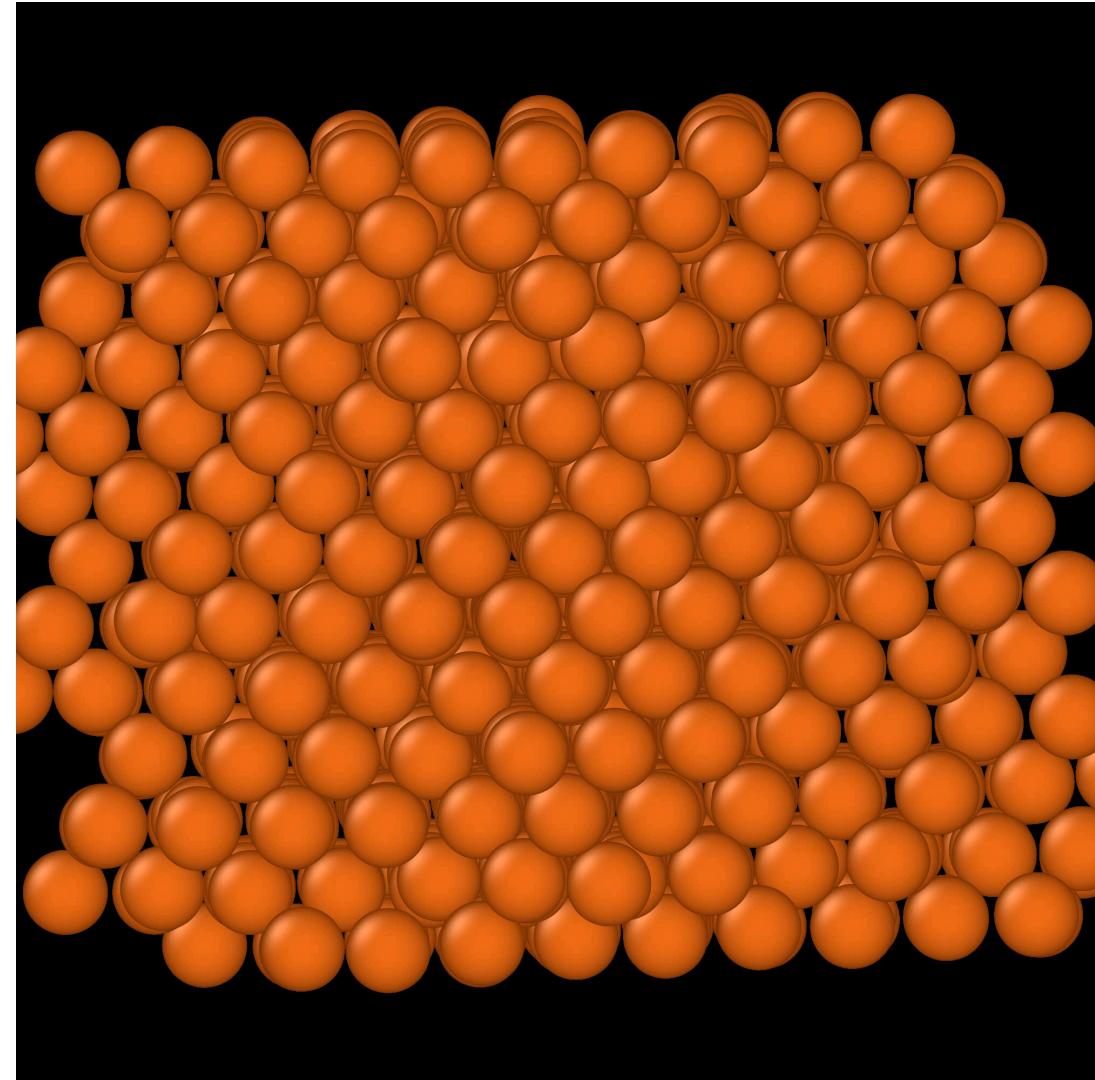
LJ Phase Diagram



Protein phase diagram  
Determined by a grid search

# The Problem: Determining a Phase Diagram

- Phase diagram of a Lennard-Jones system
- Hard, correct way: *free energy*
- Simpler, fraught way: check a grid
- That's a lot of points – how to run *many* simulations?
- This session:
  - Choose a Pressure-Temperature space
  - set pressure, run a *melting study* and record melting point
- We have 1 hour – statistics are bad!



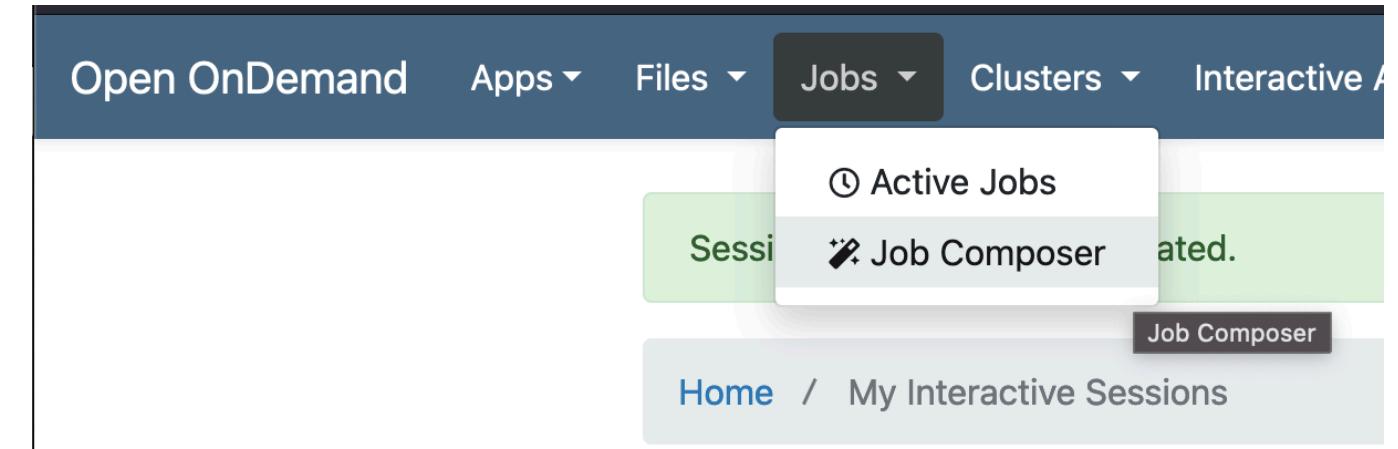
# SLURM – Managing Compute Resources

- Many jobs are too big to be run in Jupyter Notebooks
  - Or don't need interactivity
- Slurm: software on **nearly all** HPCs which manages running “jobs”
- Uses SBATCH commands to set HPC parameters
  - Output locations
  - Project that has the compute time (see220002p for this workshop)
  - Estimated length of the job
  - Number of CPUs, GPUs, etc.
  - And more!
- Run in the terminal with “sbatch” command
- Or use OnDemand Job Composer

```
#!/bin/bash
#SBATCH --job-name=phase-test
#SBATCH --output=%j.o
#SBATCH --error=%j.e
#SBATCH --nodes=1
#SBATCH --ntasks-per-node=1
#SBATCH --partition=RM-shared
#SBATCH --time=02:00:00
#SBATCH --account=see220002p

singularity exec /ocean/projects/see220002p/shared/icomse_latest.sif python phase_diagram.py
```

Rachael@Home ~ \$ sbatch cpu.slurm



# Array Jobs

- Give a list of *job indices* to submit many jobs at once
- Access the index through `$SLURM_ARRAY_TASK_ID` to assign each job different parameters
  - Index into a list
  - Use directly as a numerical value

```
#!/bin/bash
#SBATCH --job-name=phase-test
#SBATCH --output=%j-%a.o
#SBATCH --error=%j-%a.e
#SBATCH --nodes=1
#SBATCH --ntasks-per-node=1
#SBATCH --partition=RM-shared
#SBATCH --time=00:30:00
#SBATCH --account=see220002p

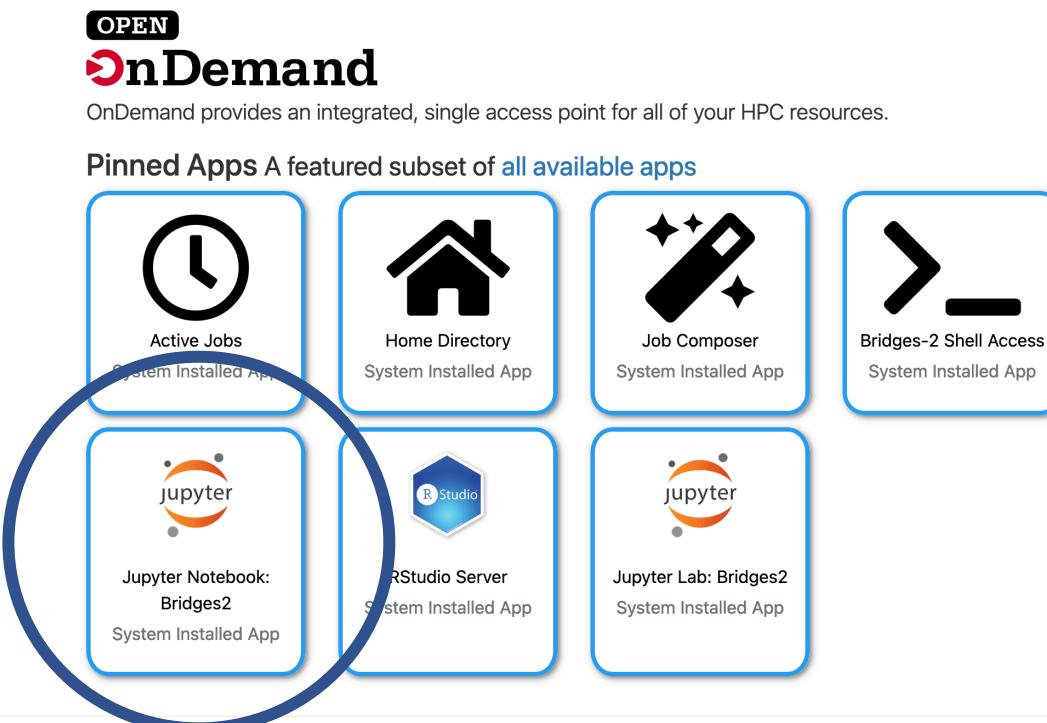
singularity exec /ocean/projects/see220002p/shared/icomse_latest.sif python3 array_phase_diagram.py $SLURM_ARRAY_TASK_ID

[rss364@bridges2-login014 session2]$ sbatch --array=1-10 array-cpu.slurm
```

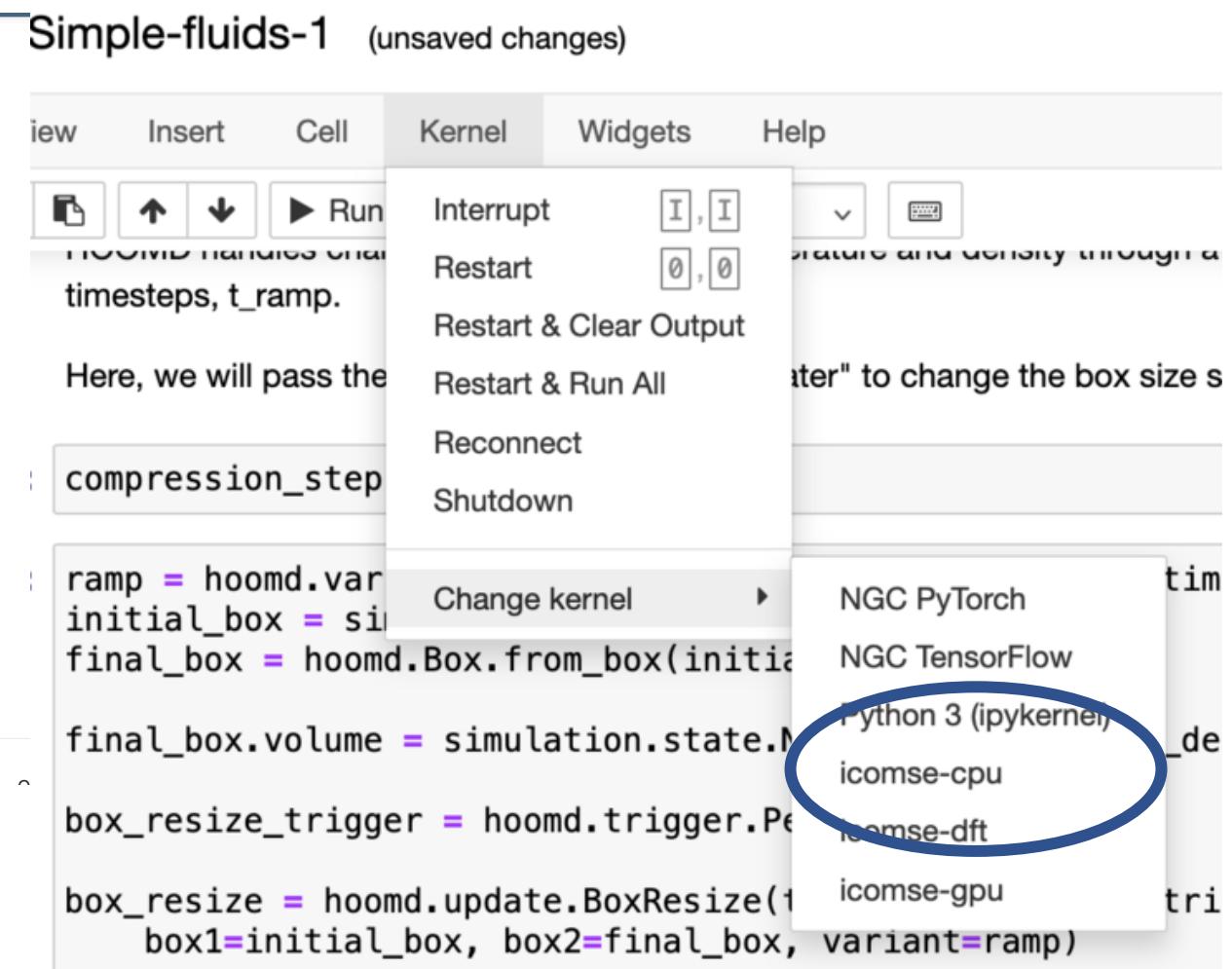
```
7 import sys
8 pressure = float(sys.argv[1])
9
10 print('pressure is ', pressure)
11 print(type(pressure))
12
```

# Accessing OnDemand

## Jupyter Notebook



## Choose the correct kernel



# Job Request

**Interactive Apps**

|                               |
|-------------------------------|
| Servers                       |
| Jupyter Lab:<br>Bridges2      |
| Jupyter Notebook:<br>Bridges2 |
| RStudio Server                |

**Jupyter Notebook: Bridges2**

This app will launch a Jupyter Notebook server on one or more nodes.

Number of hours

Number of nodes

Account

Partition

For help please review the available [Bridges Partitions](#).

Extra Slurm Args

Extra Jupyter Args

I would like to receive an email when the session starts

**Launch**

# Uploading/Downloading Data

Through the command line:

```
(base) Rachael@Home ~/example_folder $ scp -r username@data.bridges2.psc.edu:path/to/file path/to/destination  
username@data.bridges2.psc.edu's password:  
Rachael@Home:~$
```

Through OnDemand:

The screenshot shows the OnDemand web interface. At the top, there is a navigation bar with links for Open OnDemand, Apps, Files, Jobs, Clusters, Interactive Apps, and a user icon. Below the navigation bar is a toolbar with buttons for Open in Terminal, Refresh, New File, New Directory, Upload, Download, Copy/Move, and Delete. On the left, there is a sidebar with Home Directory, Ocean, and Examples sections. The main area shows a file list in a terminal window. The terminal path is /jet/home/rss364/. The file list includes:

| Type | Name                                  | Size    | Modified at          |
|------|---------------------------------------|---------|----------------------|
| □    | ondemand                              | -       | 7/3/2024 11:08:19 AM |
| □    | tmp_ondemand_ocean_see220002p_symlink | -       | 7/3/2024 11:47:12 AM |
| ✓    | Simple-fluids-1.ipynb                 | 38.6 KB | 7/3/2024 6:19:12 PM  |
| □    | Untitled.ipynb                        | 1.34 KB | 7/3/2024 11:48:23 AM |

A red arrow points to the 'Simple-fluids-1.ipynb' file in the list.

# Simulation Analysis: Radial Distribution Function

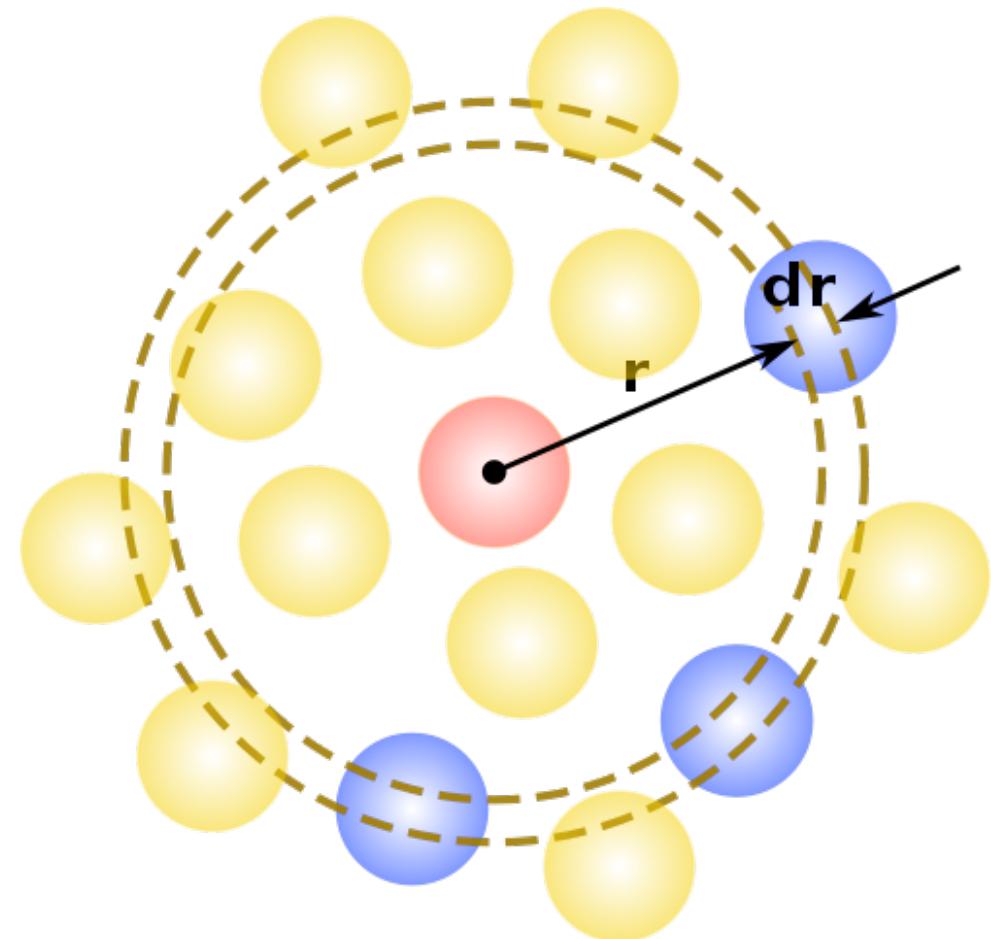
# Radial Distribution Function (RDF)

- In words: the ratio between the *actual* density and the total system density at a distance  $r$  away from a particle
- In equations:

$$g(r) = \frac{\langle \rho(r) \rangle}{\rho}$$

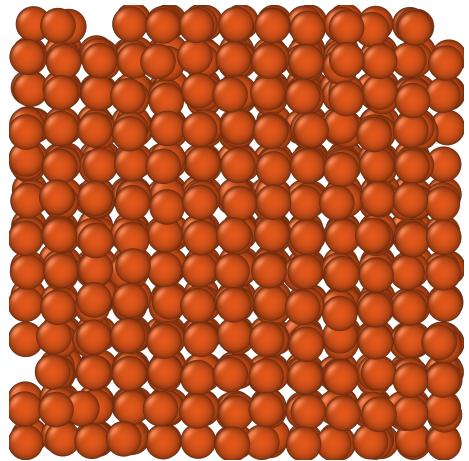
$$g(r) = V \frac{N_{\text{reference particles}}}{N_{\text{particles}}} \langle \delta(r) \rangle$$

- Algorithm:
  - Choose a “shell width”  $dr$
  - Choose a particle
    - Count the number of particles inside  $dr$  and convert to density
    - Move out by  $dr$  and repeat
  - Repeat for all particles and take the average

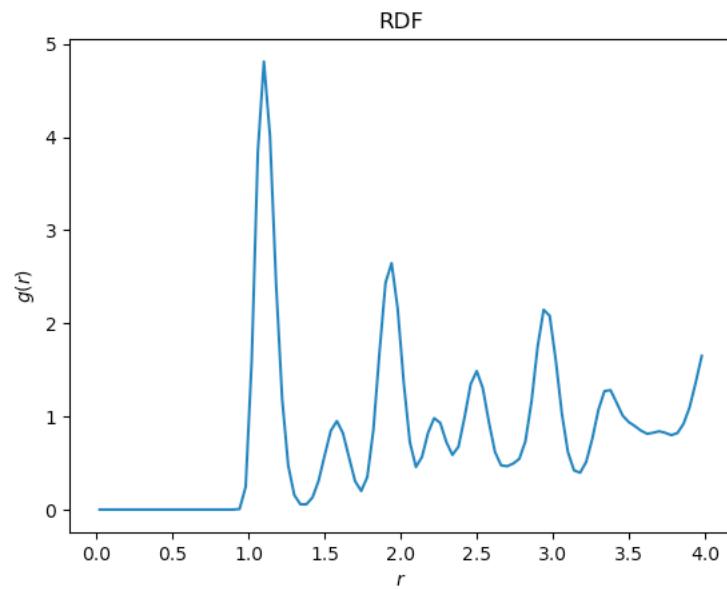


*Graphical depiction of calculating the RDF, from Wikipedia*

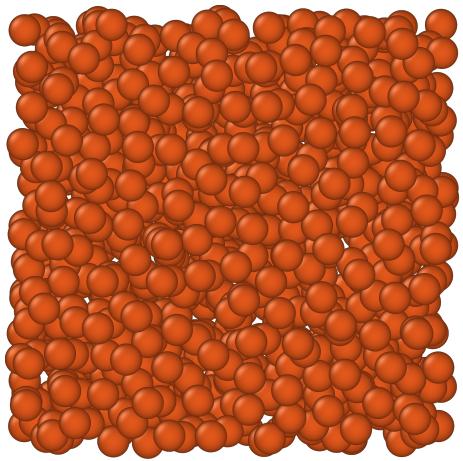
# Sample RDFs



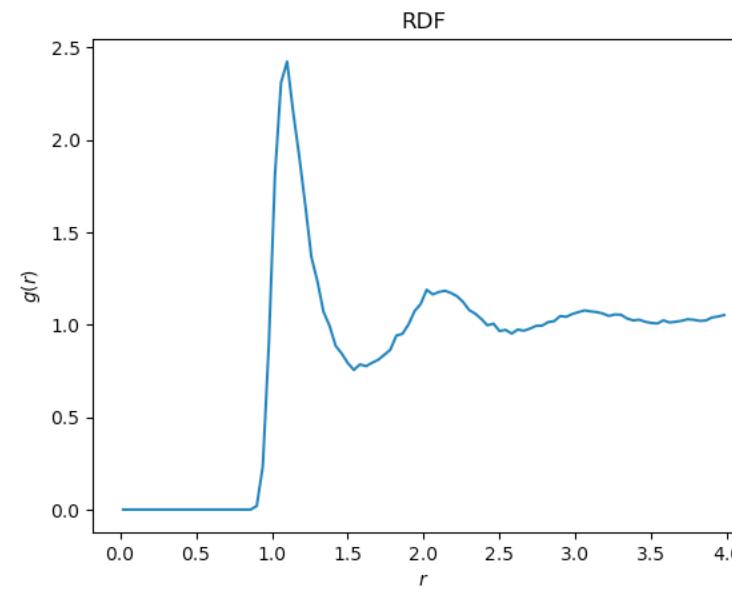
RDF



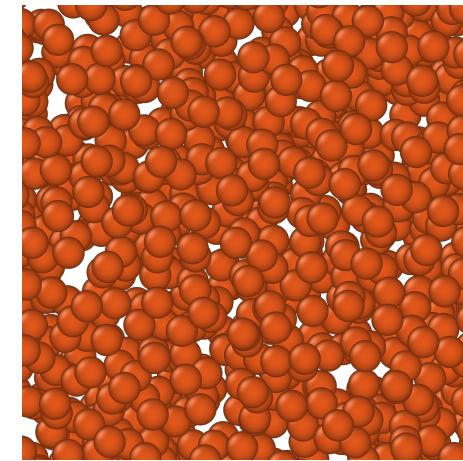
Ordered Solid/Crystal



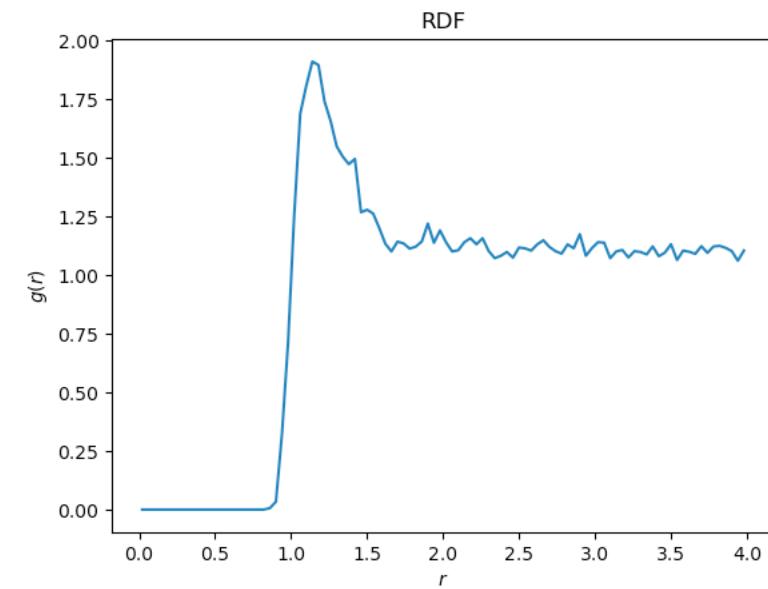
RDF



Liquid



RDF



Gas

# Viewing RDF in Ovito

