Start a new calculation:

Example scenario: Start a job for fractional vortex r0=24, the order parameters are stored in re_n400_frac_r0_24 and im_n400_frac_r0_24. Intervals are in interval0.txt to interval5.txt (separate the intervals into multiple parts due to 4H time limit of a node). The intervals are expected to have no more than 550 eigenpairs per interval.

1. Make a copy of ~/physics/paral in the ~/scratch folder.
   Ex: `cp -r ~/physics/paral ~/scratch/run-frac`

2. Upload interval and order parameter files to the working folder.
   Ex: (In sftp)
   ```
   put re_n400_frac_r0_24 ~/scratch/run-frac
   put im_n400_frac_r0_24 ~/scratch/run-frac
   put interval*.txt ~/scratch/run-frac
   ```

3. Edit the values of "RE_S", "IM_S", "N" in **launch_fac.sh**. RE_S is the name of the file storing the real part of the order parameters, IM_S is the same for the imaginary part. N is N. No need to change other lines of the script.
   Ex: (In file launch_fac.sh)
   ```
   ...
   N=400
   ...
   RE_S=" re_n400_frac_r0_24"
   IM_S="im_n400_frac_r0_24"
   ...
   ```

4. Edit the values of "N", "K", and "XTOL" in **launch_solve.sh**. N is N, K is the number of eigenpairs to solve for each interval, and XTOL is the error tolerance of the solver as the number of multiple of machine epsilon.
   For double precision (always used), XTOL=3000 seem to work well enough.
   Value of K is set depending on the intervals we use.
   No need to change other lines of the script.
   Ex: (In file launch_solve.sh)
   ```
   ...
   N=400
   ...
   K=550
   XTOL=3000
   ...
   ```

5. Queue jobs using the Queue_Jobs.sh script. Usage: ./Queue_Jobs.sh <# of interval files>
   Ex:
   ```
   cd ~/scratch/run-frac
   ./Queue_Jobs.sh 6
   ```

6. Wait for all the jobs to finish. (~6 hours if no one else is using the double V100 nodes)
7. Check if all jobs ran without crashing:
   a) Check if all the **fac-interval#** folders are present in the working folder.
   b) Check if all solver completed by checking the contents of **fac-interval#/solve.log**
      i. If the last line of solve.log is "[Timestamp]: All results saved to disk.", then the solve for that part of intervals is successful.
      ii. If the last line is not above, then maybe the solver took too long and exceeded the 4-hour time limit. Consider splitting the intervals into more parts. It could also be other problem tho.
      iii. If solve.log doesn't exist for a fac-interval# folder, then the factorial didn't run successfully. Check the fac.log file in the folder for more information. Sometimes a node would not finish the LU decomposition of a part of intervals for some unknow reason, and retrying for that interval file would typically resolve the issue.
   c) Skim through each **solve.log** files and check if any interval had more than K eigenpairs found. If so, then we potentially missed some eigenpairs. This may or may not be a problem depending on how many we missed. We can check it and adjust the intervals using a python script (descripted later)
8. Copy the results to a folder in physics folder using the **copy_results.sh** script.
   Usage: ./copy_results.sh <desired result directory name> <# of intervals> <prefix to copy>
   Ex:
   ```
   ./copy_results.sh results_n400_frac_r0_24 6 E_
   ./copy_results.sh results_n400_frac_r0_24 6 V_
   ```
9. Queue the LDOS and fft jobs.
   Usage: sbatch Job_LDOS.sbash <N> <result dir> <output name> <ee>
   sbatch Job_fft.sbash <N> <result dir> <output name> <ee>

   Ex:
   ```
   sbatch Job_LDOS.sbash 400 ~/physics/results_n400_frac_r0_24 LDOS_n400_frac_r0_24.npy 0.005
   sbatch Job_fft.sbash 400 ~/physics/results_n400_frac_r0_24 spec_n400_frac_r0_24.npy 0.005
   ```
   Then when the jobs finish, you will see 5 .npy files in the working folders. LDOS, spectral function p and h, and the dispersion pw and hw.

10. (In Python) Download the results and draw graphs. Convert to matlab files.

Solve for interval files manually when an individual interval file failed:

Assuming the working folder is already set up and using double precision.

1. Remove output of the failed run (if exist).
   Ex: interval2.txt failed, but a folder named fac-interval2 was generated. Remove the folder using `rm -r fac-interval2`

2. Submit a job for factorization:
   `sbatch Job_fac.sbatch <interval file name> double`

   Ex: interval2.txt failed. `sbatch Job_fac.sbatch interval2.txt double`

3. Submit a job for solve and set it to be dependent on the success of the factorization job.
   `sbatch -d afterok:<jobid> --gres=gpu:V100:2 Job_solve.sbatch <interval folder name> double`

   Ex: interval2.txt failed. The command in step 2 returned jobid 12345678.
   `sbatch -d afterok:12345678 --gres=gpu:V100:2 Job_solve.sbatch fac-interval2 double`

Python Scripts:

The scripts are written in Jupyter notebook for ease of visualization. They are hopefully self-explanatory through comments and function/variable names.

`Viz_Result.ipynb` can visualize the LDOS and spectual function using matplotlib and can also save the results in matlab file.

`Viz_interval.ipynb` helps to estimate the distribution of eigenvalues of a system, and helps generate a new set of intervals that more accurately track the eigenvalues.