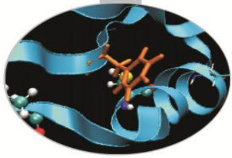
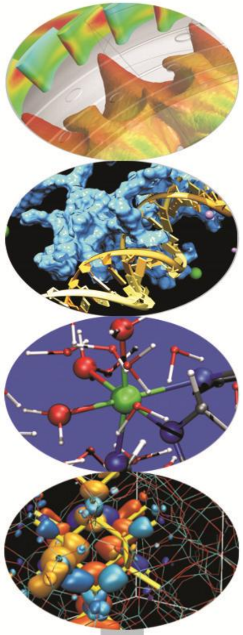
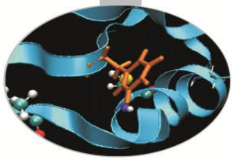


Parallel Fast Fourier Transform

Training session



Exercise 1



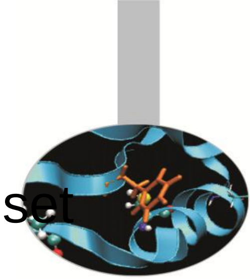
Make a program that computes the Fourier Transform of a 1D set of data, using the following specifications:

- Use Fortran or C.
- Input array: Complex array of 1024 points.
- Input function:

$$f(i)=\begin{cases}0 & 0\le i<255\\ 1 & 256\le i\le 767\\ 0 & 768<i\le 1023\end{cases}$$

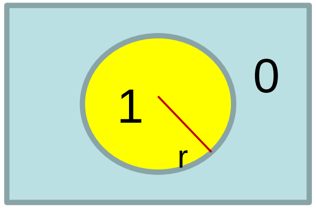
- Output array: Complex array of 1024 points.
- Using the results, compute the anti-transform, and compare the results.
- For FFTW commands you can use the FFTW on line manual at http://www.fftw.org/doc/
- Compare the time needed to execute the DFT, and the time used to create the plan, using both fftw_estimate and fftw_measure flags

Exercise 2



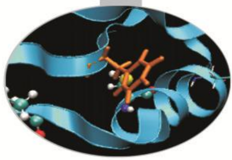
Make a parallel program that computes the Fourier Transform of a 2D set of data, using the following specifications:

- Use Fortran or C.
- Use MPI and MPI version of FFTW.
- Input array: Complex) array of 1024x1024 points.
- Input function: 2D-*Top Hat* function centered in (511,511) (or 512 for Fortran users) and radius 256 points:



- Output array: Complex array of 1024x1024 points.
- Using the results, compute the anti-transform, and compare the results.
- For FFTW commands you can use the FFTW on line manual at http://www.fftw.org/doc/
- Compare the time needed to execute the DFT, and the time used to create the plan, using both fftw_estimate and ftw_measure flags
- Only for the fftw_estimate case compare the times of execution obtained using 2,4, 8, 12, 16, 32, 48 cores.

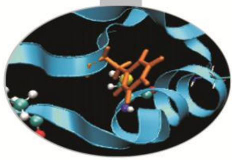
Exercise 3



Make a parallel program that computes the Fourier Transform of a 3D set of data, using the following specifications:

- Use Fortran or C.
- Use MPI and MPI version of FFTW.
- Input array: Complex array of 128x128x128 points.
- Input function: 3D-*Top Hat* function centered in (63,63) (or (64,64) for Fortran users) and radius 32 points:
- Output array: Complex array of 128x128x128 points.
- Using the results, compute the anti-transform, and compare the results.
- For FFTW commands you can use the FFTW on line manual at http://www.fftw.org/doc/
- Use MPI_WTIME function: MPI_Wtime.html
- Compare the times of execution obtained using 64,128, 256, 512 cores.

Exercise 4



Repeat Exercise 3 using 2Decomp&FFT library

•Please use Fortran language (currently this library is not compatible with C language).

•Compare the times of execution obtained in exercise 3 with the ones obtained now.