PARALLEL CODE : DOMAIN DECOMPOSITION 1

- If the code were fully finite-difference:
  - all operatiions are local
  - 1D,2D or 3D domain decomposition possible
  - pass ghost points
- However, this code is HYBRID PSEUDO-SPECTRAL/FINITE-DIFFERENCE
  - Spectral representation is a global not local operation

OPTIONS:

- Parallel FFT (consult Dubey, Cattaneo, University of Chicago)
  - Communication what is needed between processors
- Transpose-based FFT
  - do all FFTs in processor
  - transpose data when necessary

PROS AND CONS:

- Parallel FFT
  - Pro: Flexible - 1D, 2D or 3D domain decomposition
  - Pro: Efficient - minimum communication
  - Con: Very tough to program (serious programmer, ~ 2 years)
- Transpose-based FFT
  - Con: Domain decomposition limited: must include at least one full dimension. **Max no of processors can use is NZ - severe restriction.**
  - Con: Theoretically less efficient
  - Pro: In practice, about the same efficiency!
  - Con: Have to work hard on transpose to get that efficiency
  - Pro: Much easier to program – use standard/vendor FFTs; only need to write transpose (idiot like me ~ few weeks)