PSPFFT: Multi-threaded Parallel FFT-Based 3D Poisson Solver
Reuben D. Budiardja
National Institute for Computational Sciences, University of Tennessee - Knoxville

**Introduction**
- Many physics simulations require the solution of Poisson’s equation
- Common example: Newtonian gravitational potential, potential of electric charge, spectral method
- We implement a method that employ Fourier transform to solve the discretized Poisson’s equation on 3D system
- The solver, named PSPFFT (‘Poisson Solver Parallel FFT’) solves the equation globally on mesh block distributed across multiple processes on parallel computer
- It is suitable for large-scale parallel simulations

**Poisson’s Equation**
- We need to solve \( \nabla^2 \Phi(x) = S(x) \) with boundary condition \( \Phi(x) \to 0 \) as \( |x| \to \infty \)
- Formal solution: \( \Phi(x) = \int dx' G(x-x')S(x') \)
- By convolution theorem, evaluate the integral as: \( \tilde{\Phi}(k) = \tilde{G}(k)\tilde{S}(k) \)

**Mesh Decomposition**
- Discrete Fourier transform is done with FFT
- \( n \log(n) \) operation
- Optimized implementation provided by FFTW (can use other library)
- Not convenient for typical brick mesh decomposition

**Code Description**
- PSPFFT is written in Fortran 2003 standard
- Follow object-oriented principle with abstraction, encapsulation, and polymorphism
- Currently uses FFTW, but usage is abstracted in one Fortran module such that other FFT libraries could be used without widespread code change
- Uses the latest FFTW Fortran interface and provides façade pattern for its advanced API
- Release will be available at [http://eagle.phys.utk.edu/pspfft](http://eagle.phys.utk.edu/pspfft)

**Parallel Three-Dimensional FFT**
- Transform ‘bricks’ to ‘pillar’ decomposition:
  - Each MPI process performs multiple (x-pillar width times) 1D Fourier transform in parallel
  - Multi-dimensional FFTs: sets of one-dimensional transform in each dimension
  - Pillars decomposition has to be transposed
  - Multiple MPI sub-communicators are created to transpose data in parallel

**Results and Test Problems**
- \( \Phi(x) \) of homogeneous spheroid (and its relative error distribution)
- Error convergence & weak scaling:
- MPI with OpenMP comparison:

---

**Multi-Threading using OpenMP**
- Multiple numbers of 1D transform in each MPI process can be done in parallel using a team of threads
- Each thread is completely independent transform \( \rightarrow \) linear scaling within an MPI process
- Thread-safe FFTW plan is required

Visit Our Webpage for More Information