

PHYS 511: Computational Modeling and Simulation - Fall 2016
Assignment #6, due Thursday Dec 1, by 7:00 pm
(no late submissions will be accepted)

Introduction to parallel computing with MPI

1. Write a Fortran (or C) program that evaluates the expectation value of the kinetic energy operator for an atomic orbital defined on a 3D numerical grid. The program should use MPI and should be capable of running on N_p processors (this is not a fixed number, and can be changed by the user when the program is executed). You will need to figure out how to split data and work between MPI processes more or less equally.

Suppose you have an atomic orbital $\varphi(x, y, z) = e^{-ax^2 - by^2 - cz^2}$ (atomic units are used throughout), with $a = 0.7$, $b = 0.6$, and $c = 0.5$. This orbital is defined on a $n_x \times n_y \times n_z$ numerical grid with grid spacing h . In your simulations you can use $n_x = 101$, $n_y = 101$, $n_z = 101$, and $h = 0.1$ – these parameters essentially define a box of size $[-5\dots 5]$ atomic units of length in all three directions. The expectation value of the kinetic energy operator \hat{T} is defined as follows (φ are assumed to be real):

$$\langle \hat{T} \rangle = \frac{\iiint_{-\infty}^{+\infty} \varphi \left(-\frac{1}{2} \nabla^2 \varphi \right) dx dy dz}{\iiint_{-\infty}^{+\infty} \varphi^2 dx dy dz}$$

The analytic boundary conditions for atomic orbitals are such that they vanish when $x, y, z \rightarrow \pm\infty$. If you look at $\varphi(x, y, z)$ you can see that it is essentially zero at the boundaries of your $[-5\dots 5]$ box, so you can assume that it is zero anywhere beyond those boundaries. To compute the integrals use the rectangular rule:

$$\iiint f(x, y, z) dx dy dz \approx \sum_{i,j,k} f(x_i, y_j, z_k) \Delta V.$$

Despite being simple, it is actually very accurate for this particular case of an integral when the integrand vanishes at the boundaries.

To discretize the Laplace operator ($\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}$) use the five-point approximation for the second derivatives (see formula (8) in lecture #2). Notice that the five-point approximation requires two points in each (positive and negative) direction. Therefore, you will need at least two extra layers of points in each direction for the 3D array(s) where you store φ . Please use memory efficiently. Each MPI process should allocate only as much memory for data storage as it is actually necessary. There is no need for an MPI process to allocate the entire $n_x \times n_y \times n_z$ array when you run calculations with two processors. In this case you can organize the calculations in such a way that each MPI process does calculations in roughly 1/2 of the box. Keep in mind that your program should be general and work on any number of processors.

Please do not follow the practice of hardcoding any parameter values (e.g. n_x, n_y, n_z, a, b, c) in the expressions that appear in the body of your program. Instead, all those parameters should be defined in the beginning of the program (for example as constants) and then referred to in all relevant expressions or loops by their name. This way one can easily modify these parameters and rerun calculations without having the hassle of looking and changing hardcoded values (e.g. 0.5) in the body of the program. In other words do not write things such as `t=exp(-0.5*x*x)`. Instead write it as `t=exp(-a*x*x)`.

2. Upload all relevant files (e.g. `as06.f90` and `report.txt`) in your google-drive directory that is shared with the instructor. Do not forget to include the result, $\langle T \rangle$, in file `report.txt`.