**HEAT\_MPI\_COMM**

**Communication Time vs Number of Processes**

**HEAT\_MPI\_COMM**

**Computation Time vs Number of Processes**

****

**CHARTS FROM ASSIGNMENT 2 (PI MONTE CARLO)**

**PI\_MPI\_COMM: Communication Time**

****

**PI\_MPI\_COMM: Computation Time**

**Answer to Question for Assignment 3:**

Compared to the charts from Assignment 2, the communication chart for Assignment 3 exhibit a more predictable and linear trend. In assignment, the more processes we have, the more ghost rows we need to exchange between processes. . For every 1 process we add, we need to exchange 1 more set of ghost rows. Therefore the communication time would increase at a roughly linear rate. For assignment 2, the communication rate increased drastically between 16 and 49 processes. The increase has a higher slope than any trend exhibited in Assignment 3. This may be due to the fact that during the monte-carlo pi simulation, the root process must wait for inputs from all the other processes. In the heat\_mpi example, each process only exchanges a set or two of ghost rows with neighboring processes. Therefore, the trade happens in parallel. Therefore, the increase in communication time is not as steep as the monte-carlo simulation.

The computation time for Assignment 2 initially decreased greatly with additional processes then leveled off. This may be due to the fact that there are only a billion points to cycle through no matter how many processes we invoke. Therefore, we should see a big jump between having 1 process handling a billion point to 4 processes handling 250 million points each. But as we keep increasing the processes, less and less points are available to distribute, therefore we will see the computation time level off. The same trend should be exhibited for Assignment 3. Initially we can split our 1000 row grid between 4 processes calculating 250 rows each. Therefore we see a big initial drop in computation time. But as we keep increasing the number of processes, there are only 1000 rows to be split. So we see the drop in computation time level off.