Homework Assignment 1

1. Modify the trapezoidal rule so that it will correctly estimate the integral even if comm_sz doesn't evenly divide n. (You can still assume that n ⩾ comm_sz.)

2. Suppose comm_sz = 4 and suppose that x is a vector with n = 14 components.
   a. How would the components of x be distributed among the processes in a program that used a block distribution?
   b. How would the components of x be distributed among the processes in a program that used a cyclic distribution?
   c. How would the components of x be distributed among the processes in a program that used a block-cyclic distribution with block size b = 2?

   You should try to make your distributions general so that they could be used regardless of what comm_sz and n are. You should also try to make your distributions "fair" so that if q and r are any two processes, the difference between the number of components assigned to q and the number of components assigned to r is as small as possible.

3. What do the various MPI collective functions do if the communicator contains a single process?

4. If comm_sz = p, we mentioned that the "ideal" speedup is p. Is it possible to do better?
   a. Consider a parallel program that computes a vector sum. If we only time the vector sum—that is, we ignore input and output of the vectors—how might this program achieve speedup greater than p?
   b. A program that achieves speedup greater than p is said to have superlinear speedup. Our vector sum example only achieved superlinear speedup by overcoming certain "resource limitations." What were these resource limitations? Is it possible for a program to obtain superlinear speedup without overcoming resource limitations?