

CSE 569M
Parallel Architectures and Algorithms
Assignment #2

Due: February 19, 2009.

1. An example MPI program that implements a version of the Monte Carlo calculation of π is on the class web page. In the example code, the pseudo random numbers are all computed in one place and distributed to the slave nodes for use. Empirically evaluate the performance of this code, plotting speedup vs. number of processors for 2 to 8 processors.
2. Alter the example program above to partition the workload as in the shared memory parallel Monte Carlo program of assignment #1. Here, the pseudo random numbers should be computed in parallel on the slave nodes. Empirically evaluate the performance of this code as above. Comment on the performance of the two implementations.
3. Gaussian elimination is a well-known technique for solving simultaneous linear systems of equations. Variables are eliminated one by one until there is only one left, and then the discovered values of variables are back-substituted to obtain the values of other variables. In practice, the coefficients of the unknowns in the equation system are represented as a matrix A , and the matrix is first converted to an upper-triangular matrix (a matrix in which all elements below the main diagonal are 0). Then back-substitution is used. Let us focus on the conversion to an upper-triangular matrix by successive variable elimination. Pseudocode for sequential Gaussian elimination is shown below. The diagonal element for a particular iteration of the loop k is called the *pivot element*, and its row is called the *pivot row*.
 - a. Assuming a decomposition into rows and an assignment into blocks of contiguous rows, implement a shared memory parallel Gaussian elimination code. Measure the performance of this code, showing speedup for 2, 3, and 4 processors.
 - b. Modify the code to use an interleaved (cyclic) assignment of rows to processes. Measure the performance of this code, showing speedup for 2, 3, and 4 processors.
 - c. Comment on the performance implications of the assignments used above.

```
procedure Eliminate (A)          /* triangularize the matrix A */
  for k <- 0 to n-1 do          /* loop over all diagonal (pivot) elements */
    for j <- k+1 to n-1 do      /* for all elements in the row of, and to the
                                right of, the pivot element */
      A[k,j] = A[k,j]/A[k,k];   /* divide by pivot element */
    endfor
    A[k,k] <- 1;
    for i <- k+1 to n-1 do      /* for all rows below the pivot row */
      for j <- k+1 to n-1 do    /* for all elements in the row */
        A[i,j] <- A[i,j] - A[i,k]*A[k,j];
      endfor
      A[i,k] <- 0;
    endfor
  endfor
end procedure
```