MCS 572: Homework 2 (Revised)
University of Illinois at Chicago (Professor Nicholls)
Fall 2009

Due Monday, October 12 by 2pm.

1. (a) Modify the matrix–vector multiplication code (`matvec.f`) to perform the Jacobi iteration for the solution of linear systems of equations $A\vec{x} = \vec{b}$.

   (b) Consider the matrix $A \in \mathbf{R}^{n \times n}$ which is $(n+1)$ on the diagonal and 1 everywhere else. Pick the "exact solution" $\vec{x}_{exact}$ which is all 1's. What is the corresponding $\vec{b}_{exact} = A\vec{x}_{exact}$ which gives this solution?
   For $n = 10$ run the code (perform the Jacobi iteration with this $(\vec{x}_{exact}, \vec{b}_{exact})$ pair until a relative error of 10^{-10} is achieved) with $N = 2, 4, 8$ processes. How many iterations are required? What speed–up do you notice?

   (c) Repeat part (b) with $n = 100$.

   (d) Repeat part (b) with $n = 1000$.

2. Exercises (Heath): 2.7, 2.21, 2.26, 2.31

3. Computer Problems (Heath): 2.2, 2.4, 2.6, 2.17

4. Exercises (Heath): 11.6, 11.7

5. Computer Problems (Heath): 11.12

```
      program main
      include 'mpif.h'
      integer MAX_ROWS, MAX_COLS, rows, cols
      parameter (MAX_ROWS = 1000, MAX_COLS = 1000)
      double precision a(MAX_ROWS,MAX_COLS), b(MAX_COLS), c(MAX_ROWS)
      double precision buffer(MAX_COLS), ans

      integer myid, master, numprocs, ierr, status(MPI_STATUS_SIZE)
      integer i, j, numsent, sender
      integer anstype, row

      call MPI_INIT( ierr )
      call MPI_COMM_RANK( MPI_COMM_WORLD, myid, ierr )
      call MPI_COMM_SIZE( MPI_COMM_WORLD, numprocs, ierr )
      master = 0
      rows   = 100
      cols   = 100

      if ( myid .eq. master ) then
c    master initializes and then dispatches
c    initialize a and b  (arbitrary)
      do 20 j = 1,cols
        b(j) = 1
        do 10 i = 1,rows
           a(i,j) = i
 10     continue
 20   continue
      numsent = 0
c     send b to each slave process
      call MPI_BCAST(b, cols, MPI_DOUBLE_PRECISION, master,
     &               MPI_COMM_WORLD, ierr)
c     send a row to each slave process; tag with row number
      do 40 i = 1,min(numprocs-1,rows)
        do 30 j = 1,cols
           buffer(j) = a(i,j)
 30     continue
        call MPI_SEND(buffer, cols, MPI_DOUBLE_PRECISION, i,
     &                i, MPI_COMM_WORLD, ierr)
        numsent = numsent+1
 40   continue
      do 70 i = 1,rows
        call MPI_RECV(ans, 1, MPI_DOUBLE_PRECISION,
     &                MPI_ANY_SOURCE, MPI_ANY_TAG,
     &                MPI_COMM_WORLD, status, ierr)
        sender     = status(MPI_SOURCE)
```

```
        anstype    = status(MPI_TAG)       ! row is tag value
        c(anstype) = ans
        if (numsent .lt. rows) then        ! send another row
           do 50 j = 1,cols
              buffer(j) = a(numsent+1,j)
 50        continue
           call MPI_SEND(buffer, cols, MPI_DOUBLE_PRECISION,
     &                   sender, numsent+1, MPI_COMM_WORLD, ierr)
           numsent = numsent+1
        else      ! Tell sender that there is no more work
           call MPI_SEND(MPI_BOTTOM, 0, MPI_DOUBLE_PRECISION,
     &                   sender, 0, MPI_COMM_WORLD, ierr)
        endif
 70    continue
       else
c         slaves receive b, then compute dot products until
c         done message received
          call MPI_BCAST(b, cols, MPI_DOUBLE_PRECISION, master,
     &                   MPI_COMM_WORLD, ierr)
c         skip if more processes than work
          if (rank .gt. rows)
     &       goto 200
 90       call MPI_RECV(buffer, cols, MPI_DOUBLE_PRECISION, master,
     &                  MPI_ANY_TAG, MPI_COMM_WORLD, status, ierr)
          if (status(MPI_TAG) .eq. 0) then
             go to 200
          else
             row = status(MPI_TAG)
             ans = 0.0
             do 100 i = 1,cols
                ans = ans+buffer(i)*b(i)
 100         continue
             call MPI_SEND(ans, 1, MPI_DOUBLE_PRECISION, master,
     &                     row, MPI_COMM_WORLD, ierr)
             go to 90
          endif
 200      continue
       endif

       call MPI_FINALIZE(ierr)
       stop
       end
```