Project 4, due on 05/02.

Problem. Implementing parallel Gaussian elimination method for solving Ax = b using OpenMP.

Use the algorithm described below to implement Gaussian elimination without pivoting.

```
    do pivot = 1, (n-1)
!$omp parallel do private(xmult) schedule(runtime)
        do i = (pivot+1), n
            xmult = a(i,pivot) / a(pivot,pivot)
            do j = (pivot+1), n
                a(i,j) = a(i,j) - (xmult * a(pivot,j))
            end do
            b(i) = b(i) - (xmult * b(pivot))
        end do
!$omp end parallel do
    end do
```

The backward substitution step does not have to be in parallel, but you are encouraged to think about possible ways to parallelize this step using OpenMP.

Hand-In.

1. The hardcopy of your source code (Also send the source code to me by email. Please use the email title: Project 4: your name).

2. A report which contains results of performance test on the Gaussian elimination part and validation of the complete algorithm for solving Ax = b.

For performance test, use sizes of matrix A 10×10, 20×20, 40×40, 80×80 and 160×160 respectively to see how the algorithm scales with respect to a fixed number of threads. You can use 4, 8, or 12 threads depending on the node chosen to run the code.

3 To validate the complete algorithm for solving Ax=b, generate a strictly diagonally dominant square matrix as the coefficient matrix A. The vector b can be anything.

Reference:

S.F. MicGinn, R.E. Shaw and S. John, Parallel Gaussian Elimination Using OpenMP and MPI, Proceedings of the 16th Annual International Symposium on High Performance Computing Systems and Applications (HPCS'02) 0-7695-1626-2/02.