Marco Ares

11526185

/\*\* C4.5.c: Gauss Elimination with Partial Pivoting \*\*/

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

#include <pthread.h>

#define N 8

double A[N][N + 1];

pthread\_barrier\_t barrier;

int NTHREADS;

int print\_matrix()

{

int i, j;

printf("------------------------------------\n");

for (i = 0; i < N; i++) {

for (j = 0; j < N + 1; j++)

printf("%6.2f ", A[i][j]);

printf("\n");

}

}

void\* ge(void\* arg) // threads function: Gauss elimination

{

int i, j, k, prow;

int myid = (int)arg;

double temp, factor;

for (i = 0; i < NTHREADS - 1; i++) {

if (i == myid) {

printf("partial pivoting by thread %d on row %d: ", myid, i);

temp = 0.0; prow = i;

for (j = i; j <= NTHREADS; j++) {

if (fabs(A[j][i]) > temp) {

temp = fabs(A[j][i]);

prow = j;

}

}

printf("pivot\_row=%d pivot=%6.2f\n", prow, A[prow][i]);

if (prow != i) { // swap rows

for (j = i; j < NTHREADS + 1; j++) {

temp = A[i][j];

A[i][j] = A[prow][j];

A[prow][j] = temp;

}

}

}

// wait for partial pivoting done

pthread\_barrier\_wait(&barrier);

for (j = i + 1; j < NTHREADS; j++) {

if (j == myid) {

printf("thread %d do row %d\n", myid, j);

factor = A[j][i] / A[i][i];

for (k = i + 1; k <= NTHREADS; k++)

A[j][k] -= A[i][k] \* factor;

A[j][i] = 0.0;

}

}

// wait for current row reductions to finish

pthread\_barrier\_wait(&barrier);

if (i == myid)

print\_matrix();

}

}

int main(int argc, char\* argv[])

{

int i, j;

double sum;

pthread\_t threads[N];

printf("Enter how many threads wher NTHREADS < 8\n");

scanf("%d", &NTHREADS);

printf("main: initialize matrix A[N][N+1] as [A|B]\n");

for (i = 0; i < N; i++)

for (j = 0; j < N; j++)

A[i][j] = 1.0;

for (i = 0; i < N; i++)

A[i][NTHREADS - i - 1] = 1.0 \* NTHREADS;

for (i = 0; i < NTHREADS; i++) {

A[i][NTHREADS] = (NTHREADS \* (NTHREADS + 1)) / 2 + (NTHREADS - i) \* (NTHREADS - 1);

}

print\_matrix(); // show initial matrix [A|B]

pthread\_barrier\_init(&barrier, NULL, NTHREADS); // set up barrier

printf("main: create N=%d working threads\n", NTHREADS);

for (i = 0; i < NTHREADS; i++) {

pthread\_create(&threads[i], NULL, ge, (void\*)i);

}

printf("main: wait for all %d working threads to join\n", NTHREADS);

for (i = 0; i < NTHREADS; i++) {

pthread\_join(threads[i], NULL);

}

printf("main: back substitution : ");

for (i = NTHREADS - 1; i >= 0; i--) {

sum = 0.0;

for (j = i + 1; j < NTHREADS; j++)

sum += A[i][j] \* A[j][NTHREADS];

A[i][NTHREADS] = (A[i][NTHREADS] - sum) / A[i][i];

}

// print solution

printf("The solution is :\n");

for (i = 0; i < NTHREADS; i++) {

printf("%6.2f ", A[i][NTHREADS]);

}

printf("\n");

}

Enter how many threads wher NTHREADS < 8

4

main: initialize matrix A[N][N+1] as [A|B]

------------------------------------

1.00 1.00 1.00 4.00 22.00 1.00 1.00 1.00 0.00

1.00 1.00 4.00 1.00 19.00 1.00 1.00 1.00 0.00

1.00 4.00 1.00 1.00 16.00 1.00 1.00 1.00 0.00

4.00 1.00 1.00 1.00 13.00 1.00 1.00 1.00 4.00

1.00 1.00 1.00 1.00 1.00 1.00 1.00 4.00 0.00

1.00 1.00 1.00 1.00 1.00 1.00 4.00 1.00 0.00

1.00 1.00 1.00 1.00 1.00 4.00 1.00 1.00 0.00

1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 0.00

main: create N=4 working threads

partial pivoting by thread 0 on row 0: pivot\_row=3 pivot= 4.00

thread 1 do row 1

thread 2 do row 2

main: wait for all 4 working threads to join

thread 3 do row 3

partial pivoting by thread 1 on row 1: pivot\_row=2 pivot= 3.75

------------------------------------

4.00 1.00 1.00 1.00 13.00 1.00 1.00 1.00 0.00

0.00 3.75 0.75 0.75 12.75 1.00 1.00 1.00 0.00

0.00 0.75 3.75 0.75 15.75 1.00 1.00 1.00 0.00

0.00 0.75 0.75 3.75 18.75 1.00 1.00 1.00 4.00

1.00 1.00 1.00 1.00 1.00 1.00 1.00 4.00 0.00

1.00 1.00 1.00 1.00 1.00 1.00 4.00 1.00 0.00

1.00 1.00 1.00 1.00 1.00 4.00 1.00 1.00 0.00

1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 0.00

thread 3 do row 3

thread 2 do row 2

partial pivoting by thread 2 on row 2: pivot\_row=2 pivot= 3.60

------------------------------------

4.00 1.00 1.00 1.00 13.00 1.00 1.00 1.00 0.00

0.00 3.75 0.75 0.75 12.75 1.00 1.00 1.00 0.00

0.00 0.00 3.60 0.60 13.20 1.00 1.00 1.00 0.00

0.00 0.00 0.60 3.60 16.20 1.00 1.00 1.00 4.00

1.00 1.00 1.00 1.00 1.00 1.00 1.00 4.00 0.00

1.00 1.00 1.00 1.00 1.00 1.00 4.00 1.00 0.00

1.00 1.00 1.00 1.00 1.00 4.00 1.00 1.00 0.00

1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 0.00

thread 3 do row 3

------------------------------------

4.00 1.00 1.00 1.00 13.00 1.00 1.00 1.00 0.00

0.00 3.75 0.75 0.75 12.75 1.00 1.00 1.00 0.00

0.00 0.00 3.60 0.60 13.20 1.00 1.00 1.00 0.00

0.00 0.00 0.00 3.50 14.00 1.00 1.00 1.00 4.00

1.00 1.00 1.00 1.00 1.00 1.00 1.00 4.00 0.00

1.00 1.00 1.00 1.00 1.00 1.00 4.00 1.00 0.00

1.00 1.00 1.00 1.00 1.00 4.00 1.00 1.00 0.00

1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 0.00

main: back substitution : The solution is :

1.00 2.00 3.00 4.00

/\*\* plu.c: rewrote lu.c as parallel plu.c program \*\*/

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

#include <pthread.h>

#define N 8

double A[N][N], L[N][N], U[N][N];

double B[N], b[N];

int P[N];

double Y[N], X[N];

pthread\_barrier\_t barrier;

int print(char c, double x[N][N])

{

int i, j;

printf("------------- %c -----------------------\n", c);

for (i = 0; i < N; i++) {

for (j = 0; j < N; j++)

printf("%6.2f ", x[i][j]);

printf("\n");

}

}

int printV(char c, double x[N])

{

int i;

printf("--------- %c vector-----------\n", c);

for (i = 0; i < N; i++)

printf("%6.2f ", x[i]);

printf("\n");

}

int printP()

{

int i;

printf("--------- P vector-----------\n");

for (i = 0; i < N; i++)

printf("%d ", P[i]);

printf("\n");

}

// LU decomposition function

int \*plu(void\* arg)

{

int i, j, k, m, itemp, prow;

int myid = (int)arg;

double max;

double temp;

double si;

// LU decomposition loop

for (k = 0; k < N; k++) {

if (k == myid) {

max = 0;

printf("partial pivoting by thread %d on row %d: ", myid, k);

for (i = k; i < N; i++) {

if (max < fabs(A[i][k])) { // partial pivoting

max = fabs(A[i][k]);

j = i;

}

prow = i;

}

printf("pivot\_row= %d pivot=%6.2f\n", prow, A[prow][k]);

if (max == 0) {

printf("zero pivot: singular A matrix\n");

exit(1);

}

print('A', A);

// swap P[k] and P[j];

itemp = P[k]; P[k] = P[j]; P[j] = itemp;

// swap row A[k] and row A[j]

for (m = 0; m < N; m++) {

temp = A[k][m]; A[k][m] = A[j][m]; A[j][m] = temp;

}

//swap L[k][0,k-2] and L[j][0,k-2]

for (m = 0; m < k - 2; m++) {

temp = L[k][m]; L[k][m] = L[j][m]; L[j][m] = temp;

}

}

pthread\_barrier\_wait(&barrier);

// compute L U entries

U[k][k] = A[k][k];

for (i = k + 1; i < N; i++) {

L[i][k] = A[i][k] / U[k][k];

U[k][i] = A[k][i];

}

// row reductions on

for (i = k + 1; i < N; i++) {

if (i == myid) {

printf("thread %d do row %d\n", myid, i);

for (m = k + 1; m < N; m++) {

A[i][m] -= L[i][k] \* U[k][m];

}

}

}

pthread\_barrier\_wait(&barrier);

if (k == myid) {

print('A', A); print('L', L); print('U', U); printP();

getchar();

}

}

}

int main(int argc, char\* argv[])

{

int i, j, k;

double si;

pthread\_t threads[N];

printf("main: initialize matrix A[N][N], B[N], L, U and P\n");

for (i = 0; i < N; i++)

for (j = 0; j < N; j++)

A[i][j] = 1.0;

for (i = 0; i < N; i++)

A[i][N - 1 - i] = 1.0 \* N;

for (i = 0; i < N; i++) {

B[i] = (N) \* (N + 1) / 2 + (N - i) \* (N - 1);

}

for (i = 0; i < N; i++) {

for (j = 0; j < N; j++) {

U[i][j] = 0.0;

L[i][j] = 0.0;

if (i == j)

L[i][j] = 1.0;

}

}

for (i = 0; i < N; i++) {

P[i] = i;

}

print('A', A); print('L', L); print('U', U);

printV('B', B); printP();

pthread\_barrier\_init(&barrier, NULL, N); //set up barrier

printf("main: create N=%d working threads\n", N);

for (i = 0; i < N; i++) {

pthread\_create(&threads[i], NULL, plu, (void\*)i);

}

printf("main: wait for all %d working threads to join\n", N);

for (i = 0; i < N; i++) {

pthread\_join(threads[i], NULL);

}

// P L U are all computed; solve P\*U\*L\*X = P\*B

printf("main: back substitution : ");

printf("The solution is :\n");

// apply P to B to get b[ ]

printV('B', B);

for (i = 0; i < N; i++) {

b[i] = B[P[i]];

}

printV('b', b);

// solve L\*Y = PB = b

for (i = 0; i < N; i++) { // forwar substitution

Y[i] = b[i];

for (j = 0; j < i; j++) {

Y[i] -= L[i][j] \* Y[j];

}

}

printV('Y', Y);

// solve U\*X=Y

for (i = N - 1; i >= 0; i--) { // backward substitution

si = 0.0;

for (j = i + 1; j < N; j++)

si += U[i][j] \* X[j];

X[i] = (Y[i] - si) / U[i][i];

}

printV('X', X);

}

main: initialize matrix A[N][N], B[N], L, U and P

------------- A -----------------------

1.00 1.00 1.00 1.00 1.00 1.00 1.00 8.00

1.00 1.00 1.00 1.00 1.00 1.00 8.00 1.00

1.00 1.00 1.00 1.00 1.00 8.00 1.00 1.00

1.00 1.00 1.00 1.00 8.00 1.00 1.00 1.00

1.00 1.00 1.00 8.00 1.00 1.00 1.00 1.00

1.00 1.00 8.00 1.00 1.00 1.00 1.00 1.00

1.00 8.00 1.00 1.00 1.00 1.00 1.00 1.00

8.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00

------------- L -----------------------

1.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

0.00 1.00 0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 1.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 1.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 1.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 1.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00 1.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00 0.00 1.00

------------- U -----------------------

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

--------- B vector-----------

92.00 85.00 78.00 71.00 64.00 57.00 50.00 43.00

--------- P vector-----------

0 1 2 3 4 5 6 7

main: create N=8 working threads

partial pivoting by thread 0 on row 0: pivot\_row= 7 pivot= 8.00

thread 5 do row 5

thread 6 do row 6

thread 4 do row 4

thread 7 do row 7

main: wait for all 8 working threads to join

thread 3 do row 3

thread 1 do row 1

thread 2 do row 2

partial pivoting by thread 1 on row 1: pivot\_row= 7 pivot= 0.88

------------- A -----------------------

8.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00

0.00 7.88 0.88 0.88 0.88 0.88 0.88 0.88

0.00 0.88 0.88 0.88 0.88 7.88 0.88 0.88

0.00 0.88 0.88 0.88 7.88 0.88 0.88 0.88

0.00 0.88 0.88 7.88 0.88 0.88 0.88 0.88

0.00 0.88 7.88 0.88 0.88 0.88 0.88 0.88

0.00 0.88 0.88 0.88 0.88 0.88 7.88 0.88

0.00 0.88 0.88 0.88 0.88 0.88 0.88 7.88

------------- L -----------------------

1.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

0.12 1.00 0.00 0.00 0.00 0.00 0.00 0.00

0.12 0.00 1.00 0.00 0.00 0.00 0.00 0.00

0.12 0.00 0.00 1.00 0.00 0.00 0.00 0.00

0.12 0.00 0.00 0.00 1.00 0.00 0.00 0.00

0.12 0.00 0.00 0.00 0.00 1.00 0.00 0.00

0.12 0.00 0.00 0.00 0.00 0.00 1.00 0.00

0.12 0.00 0.00 0.00 0.00 0.00 0.00 1.00

------------- U -----------------------

8.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

--------- P vector-----------

7 6 2 3 4 5 1 0

thread 3 do row 3

thread 7 do row 7

thread 2 do row 2

thread 5 do row 5

thread 4 do row 4

thread 6 do row 6

partial pivoting by thread 2 on row 2: pivot\_row= 7 pivot= 0.78

------------- A -----------------------

8.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00

0.00 7.88 0.88 0.88 0.88 0.88 0.88 0.88

0.00 0.00 7.78 0.78 0.78 0.78 0.78 0.78

0.00 0.00 0.78 0.78 7.78 0.78 0.78 0.78

0.00 0.00 0.78 7.78 0.78 0.78 0.78 0.78

0.00 0.00 0.78 0.78 0.78 7.78 0.78 0.78

0.00 0.00 0.78 0.78 0.78 0.78 7.78 0.78

0.00 0.00 0.78 0.78 0.78 0.78 0.78 7.78

------------- L -----------------------

1.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

0.12 1.00 0.00 0.00 0.00 0.00 0.00 0.00

0.12 0.11 1.00 0.00 0.00 0.00 0.00 0.00

0.12 0.11 0.00 1.00 0.00 0.00 0.00 0.00

0.12 0.11 0.00 0.00 1.00 0.00 0.00 0.00

0.12 0.11 0.00 0.00 0.00 1.00 0.00 0.00

0.12 0.11 0.00 0.00 0.00 0.00 1.00 0.00

0.12 0.11 0.00 0.00 0.00 0.00 0.00 1.00

------------- U -----------------------

8.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00

0.00 7.88 0.88 0.88 0.88 0.88 0.88 0.88

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

--------- P vector-----------

7 6 5 3 4 2 1 0

thread 4 do row 4

thread 5 do row 5

thread 7 do row 7

thread 3 do row 3

thread 6 do row 6

------------- A -----------------------

8.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00

0.00 7.88 0.88 0.88 0.88 0.88 0.88 0.88

0.00 0.00 7.78 0.78 0.78 0.78 0.78 0.78

0.00 0.00 0.00 0.70 7.70 0.70 0.70 0.70

0.00 0.00 0.00 7.70 0.70 0.70 0.70 0.70

0.00 0.00 0.00 0.70 0.70 7.70 0.70 0.70

0.00 0.00 0.00 0.70 0.70 0.70 7.70 0.70

0.00 0.00 0.00 0.70 0.70 0.70 0.70 7.70

------------- L -----------------------

1.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

0.12 1.00 0.00 0.00 0.00 0.00 0.00 0.00

0.12 0.11 1.00 0.00 0.00 0.00 0.00 0.00

0.12 0.11 0.10 1.00 0.00 0.00 0.00 0.00

0.12 0.11 0.10 0.00 1.00 0.00 0.00 0.00

0.12 0.11 0.10 0.00 0.00 1.00 0.00 0.00

0.12 0.11 0.10 0.00 0.00 0.00 1.00 0.00

0.12 0.11 0.10 0.00 0.00 0.00 0.00 1.00

------------- U -----------------------

8.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00

0.00 7.88 0.88 0.88 0.88 0.88 0.88 0.88

0.00 0.00 7.78 0.78 0.78 0.78 0.78 0.78

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

--------- P vector-----------

7 6 5 3 4 2 1 0

partial pivoting by thread 3 on row 3: pivot\_row= 7 pivot= 0.70

thread 7 do row 7

thread 6 do row 6

thread 5 do row 5

thread 4 do row 4

partial pivoting by thread 4 on row 4: pivot\_row= 7 pivot= 0.64

------------- A -----------------------

8.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00

0.00 7.88 0.88 0.88 0.88 0.88 0.88 0.88

0.00 0.00 7.78 0.78 0.78 0.78 0.78 0.78

0.00 0.00 0.00 7.70 0.70 0.70 0.70 0.70

0.00 0.00 0.00 0.00 7.64 0.64 0.64 0.64

0.00 0.00 0.00 0.00 0.64 7.64 0.64 0.64

0.00 0.00 0.00 0.00 0.64 0.64 7.64 0.64

0.00 0.00 0.00 0.00 0.64 0.64 0.64 7.64

------------- L -----------------------

1.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

0.12 1.00 0.00 0.00 0.00 0.00 0.00 0.00

0.12 0.11 1.00 0.00 0.00 0.00 0.00 0.00

0.12 0.11 0.10 1.00 0.00 0.00 0.00 0.00

0.12 0.11 0.10 0.09 1.00 0.00 0.00 0.00

0.12 0.11 0.10 0.09 0.00 1.00 0.00 0.00

0.12 0.11 0.10 0.09 0.00 0.00 1.00 0.00

0.12 0.11 0.10 0.09 0.00 0.00 0.00 1.00

------------- U -----------------------

8.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00

0.00 7.88 0.88 0.88 0.88 0.88 0.88 0.88

0.00 0.00 7.78 0.78 0.78 0.78 0.78 0.78

0.00 0.00 0.00 7.70 0.70 0.70 0.70 0.70

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

--------- P vector-----------

7 6 5 4 3 2 1 0

thread 6 do row 6

thread 7 do row 7

thread 5 do row 5

partial pivoting by thread 5 on row 5: pivot\_row= 7 pivot= 0.58

------------- A -----------------------

8.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00

0.00 7.88 0.88 0.88 0.88 0.88 0.88 0.88

0.00 0.00 7.78 0.78 0.78 0.78 0.78 0.78

0.00 0.00 0.00 7.70 0.70 0.70 0.70 0.70

0.00 0.00 0.00 0.00 7.64 0.64 0.64 0.64

0.00 0.00 0.00 0.00 0.00 7.58 0.58 0.58

0.00 0.00 0.00 0.00 0.00 0.58 7.58 0.58

0.00 0.00 0.00 0.00 0.00 0.58 0.58 7.58

------------- L -----------------------

1.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

0.12 1.00 0.00 0.00 0.00 0.00 0.00 0.00

0.12 0.11 1.00 0.00 0.00 0.00 0.00 0.00

0.12 0.11 0.10 1.00 0.00 0.00 0.00 0.00

0.12 0.11 0.10 0.09 1.00 0.00 0.00 0.00

0.12 0.11 0.10 0.09 0.08 1.00 0.00 0.00

0.12 0.11 0.10 0.09 0.08 0.00 1.00 0.00

0.12 0.11 0.10 0.09 0.08 0.00 0.00 1.00

------------- U -----------------------

8.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00

0.00 7.88 0.88 0.88 0.88 0.88 0.88 0.88

0.00 0.00 7.78 0.78 0.78 0.78 0.78 0.78

0.00 0.00 0.00 7.70 0.70 0.70 0.70 0.70

0.00 0.00 0.00 0.00 7.64 0.64 0.64 0.64

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

--------- P vector-----------

7 6 5 4 3 2 1 0

thread 7 do row 7

thread 6 do row 6

partial pivoting by thread 6 on row 6: pivot\_row= 7 pivot= 0.54

------------- A -----------------------

8.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00

0.00 7.88 0.88 0.88 0.88 0.88 0.88 0.88

0.00 0.00 7.78 0.78 0.78 0.78 0.78 0.78

0.00 0.00 0.00 7.70 0.70 0.70 0.70 0.70

0.00 0.00 0.00 0.00 7.64 0.64 0.64 0.64

0.00 0.00 0.00 0.00 0.00 7.58 0.58 0.58

0.00 0.00 0.00 0.00 0.00 0.00 7.54 0.54

0.00 0.00 0.00 0.00 0.00 0.00 0.54 7.54

------------- L -----------------------

1.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

0.12 1.00 0.00 0.00 0.00 0.00 0.00 0.00

0.12 0.11 1.00 0.00 0.00 0.00 0.00 0.00

0.12 0.11 0.10 1.00 0.00 0.00 0.00 0.00

0.12 0.11 0.10 0.09 1.00 0.00 0.00 0.00

0.12 0.11 0.10 0.09 0.08 1.00 0.00 0.00

0.12 0.11 0.10 0.09 0.08 0.08 1.00 0.00

0.12 0.11 0.10 0.09 0.08 0.08 0.00 1.00

------------- U -----------------------

8.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00

0.00 7.88 0.88 0.88 0.88 0.88 0.88 0.88

0.00 0.00 7.78 0.78 0.78 0.78 0.78 0.78

0.00 0.00 0.00 7.70 0.70 0.70 0.70 0.70

0.00 0.00 0.00 0.00 7.64 0.64 0.64 0.64

0.00 0.00 0.00 0.00 0.00 7.58 0.58 0.58

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

--------- P vector-----------

7 6 5 4 3 2 1 0

thread 7 do row 7

partial pivoting by thread 7 on row 7: pivot\_row= 7 pivot= 7.50

------------- A -----------------------

8.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00

0.00 7.88 0.88 0.88 0.88 0.88 0.88 0.88

0.00 0.00 7.78 0.78 0.78 0.78 0.78 0.78

0.00 0.00 0.00 7.70 0.70 0.70 0.70 0.70

0.00 0.00 0.00 0.00 7.64 0.64 0.64 0.64

0.00 0.00 0.00 0.00 0.00 7.58 0.58 0.58

0.00 0.00 0.00 0.00 0.00 0.00 7.54 0.54

0.00 0.00 0.00 0.00 0.00 0.00 0.00 7.50

------------- L -----------------------

1.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

0.12 1.00 0.00 0.00 0.00 0.00 0.00 0.00

0.12 0.11 1.00 0.00 0.00 0.00 0.00 0.00

0.12 0.11 0.10 1.00 0.00 0.00 0.00 0.00

0.12 0.11 0.10 0.09 1.00 0.00 0.00 0.00

0.12 0.11 0.10 0.09 0.08 1.00 0.00 0.00

0.12 0.11 0.10 0.09 0.08 0.08 1.00 0.00

0.12 0.11 0.10 0.09 0.08 0.08 0.07 1.00

------------- U -----------------------

8.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00

0.00 7.88 0.88 0.88 0.88 0.88 0.88 0.88

0.00 0.00 7.78 0.78 0.78 0.78 0.78 0.78

0.00 0.00 0.00 7.70 0.70 0.70 0.70 0.70

0.00 0.00 0.00 0.00 7.64 0.64 0.64 0.64

0.00 0.00 0.00 0.00 0.00 7.58 0.58 0.58

0.00 0.00 0.00 0.00 0.00 0.00 7.54 0.54

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

--------- P vector-----------

7 6 5 4 3 2 1 0

------------- A -----------------------

8.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00

0.00 7.88 0.88 0.88 0.88 0.88 0.88 0.88

0.00 0.00 7.78 0.78 0.78 0.78 0.78 0.78

0.00 0.00 0.00 7.70 0.70 0.70 0.70 0.70

0.00 0.00 0.00 0.00 7.64 0.64 0.64 0.64

0.00 0.00 0.00 0.00 0.00 7.58 0.58 0.58

0.00 0.00 0.00 0.00 0.00 0.00 7.54 0.54

0.00 0.00 0.00 0.00 0.00 0.00 0.00 7.50

------------- L -----------------------

1.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

0.12 1.00 0.00 0.00 0.00 0.00 0.00 0.00

0.12 0.11 1.00 0.00 0.00 0.00 0.00 0.00

0.12 0.11 0.10 1.00 0.00 0.00 0.00 0.00

0.12 0.11 0.10 0.09 1.00 0.00 0.00 0.00

0.12 0.11 0.10 0.09 0.08 1.00 0.00 0.00

0.12 0.11 0.10 0.09 0.08 0.08 1.00 0.00

0.12 0.11 0.10 0.09 0.08 0.08 0.07 1.00

------------- U -----------------------

8.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00

0.00 7.88 0.88 0.88 0.88 0.88 0.88 0.88

0.00 0.00 7.78 0.78 0.78 0.78 0.78 0.78

0.00 0.00 0.00 7.70 0.70 0.70 0.70 0.70

0.00 0.00 0.00 0.00 7.64 0.64 0.64 0.64

0.00 0.00 0.00 0.00 0.00 7.58 0.58 0.58

0.00 0.00 0.00 0.00 0.00 0.00 7.54 0.54

0.00 0.00 0.00 0.00 0.00 0.00 0.00 7.50

--------- P vector-----------

7 6 5 4 3 2 1 0

main: back substitution : The solution is :

--------- B vector-----------

92.00 85.00 78.00 71.00 64.00 57.00 50.00 43.00

--------- b vector-----------

43.00 50.00 57.00 64.00 71.00 78.00 85.00 92.00

--------- Y vector-----------

43.00 44.62 46.67 49.00 51.55 54.25 57.08 60.00

--------- X vector-----------

1.00 2.00 3.00 4.00 5.00 6.00 7.00 8.00