Problem 6

1. The code is shown below.

/\* sample hello world program \*

\* C Michael Hanke 2006-10-19 \*/

#include <stdio.h>

#include <string.h>

#include <assert.h>

#include <mpi.h>

#define M 2048

#define N 2048

typedef struct complex {

double real;

double imag;

} Complex;

int mb(Complex c) {

int count, num\_iter;

Complex z;

double foo;

count = 0;

num\_iter = 256;

z.real = 0;

z.imag = 0;

while ((z.real\*z.real+z.imag\*z.imag<4.0) && (count<num\_iter)) { // we know if |z|<2 then z is considered converge

foo = z.real\*z.real-z.imag\*z.imag+c.real;

z.imag = 2\*z.real\*z.imag+c.imag;

z.real = foo;

count++;

}

return count;

}

main(int argc, char \*\*argv) {

int rank, size, tag, rc, i, j, k, num\_rows, start, end;

double r, rx, ry, foo;

int \*data, \*data\_foo;

Complex c;

MPI\_Status status;

FILE \*fp;

rc = MPI\_Init(&argc, &argv);

rc = MPI\_Comm\_size(MPI\_COMM\_WORLD, &size);

rc = MPI\_Comm\_rank(MPI\_COMM\_WORLD, &rank);

// test if divided

assert(N%size == 0);

num\_rows = N/size;

// allocate memories for each calculations

data = (int\*)malloc(num\_rows\*M\*sizeof(int));

data\_foo = data; // a pointer for further use.

start = rank\*num\_rows;

end = start+num\_rows-1;

r = 0.05; // display range and radius.

rx = 0.25;

ry = -0.025;

for (i = start; i < end; i++) {

c.imag = i/((double)M)\*(2\*r)+ry; // map array position to 2d-position

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

c.real = j/((double)N)\*(2\*r)+rx;

foo = mb(c);

\*data++ = foo;

}

}

data = data\_foo;

tag = 100;

if (rank == 0) {

// write to file.

fp = fopen("color.txt", "w");

printf("num\_rows %d\n", num\_rows);

for (i = 0; i < num\_rows; i++) {

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

fprintf(fp, "%hhu ", (unsigned char)data[j+i\*M]);

}

fprintf(fp, "\n");

}

fclose(fp);

// then others' results.

for (k = 1; k < size; k++) {

MPI\_Recv(data, num\_rows\*N, MPI\_INTEGER, k, tag, MPI\_COMM\_WORLD, &status);

printf("received message from process %d\n", k);

fp = fopen("color.txt", "a");

for (i = 0; i < num\_rows; i++) {

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

fprintf(fp, "%hhu ", (unsigned char)data[j+i\*M]);

}

fprintf(fp, "\n");

}

fclose(fp);

}

} else {

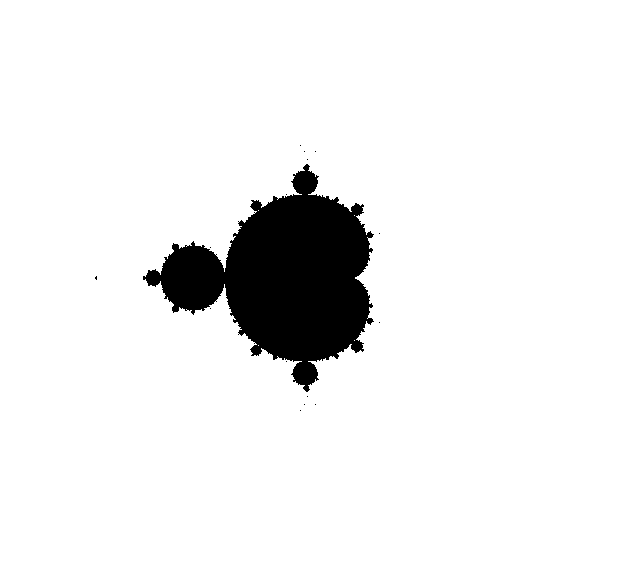
rc = MPI\_Send(data, num\_rows\*N, MPI\_INTEGER, 0, tag, MPI\_COMM\_WORLD);

}

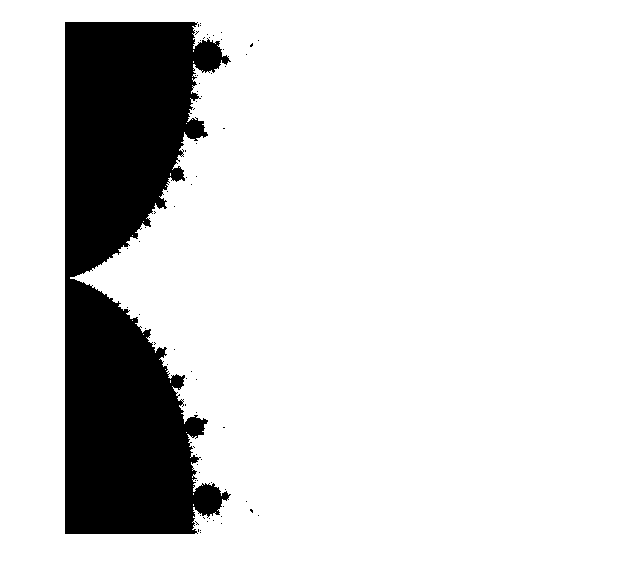
rc = MPI\_Finalize();

}

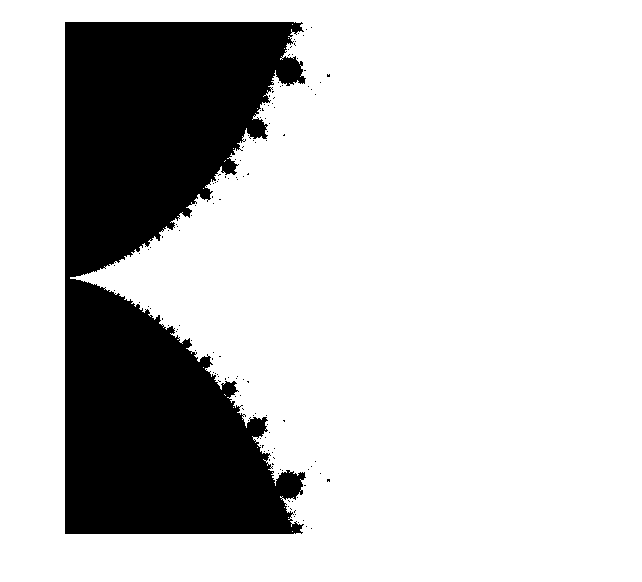
b) The figure from the lecture note is shown below.



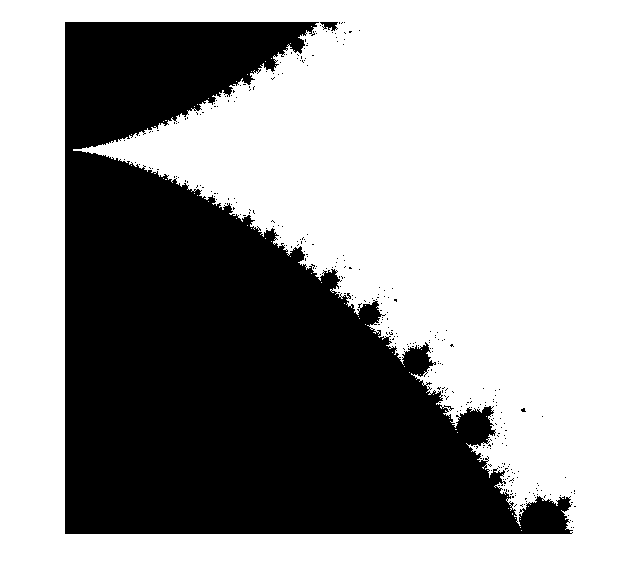
c) We tested figures surrounding point (0.25,0) since we know f will reach a boundary between convergence and divergence at z=0.25+0i. Below are figures when the radius converge from 0.25 to 0.125, 0.05, 0.01 respectively.



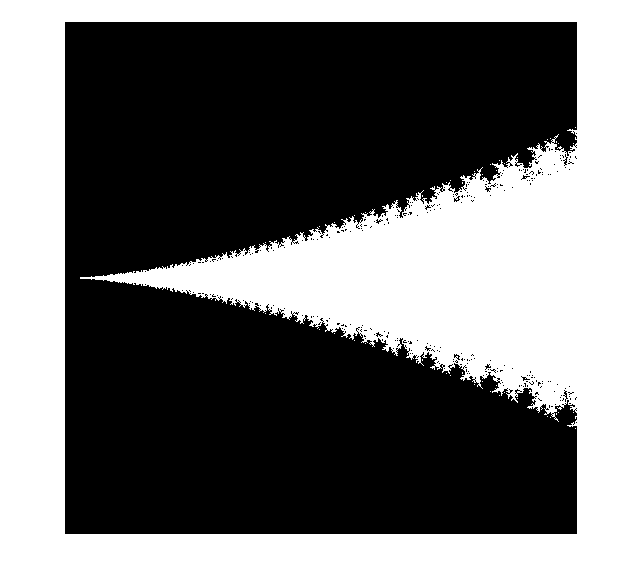
r=0.25



r=0.125



r=0.05



r=0.01. We can clearly see the fractal copies at every scale.