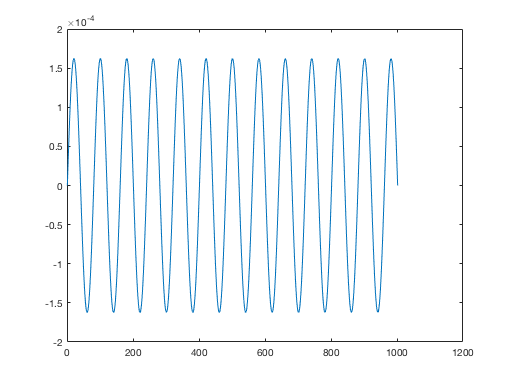
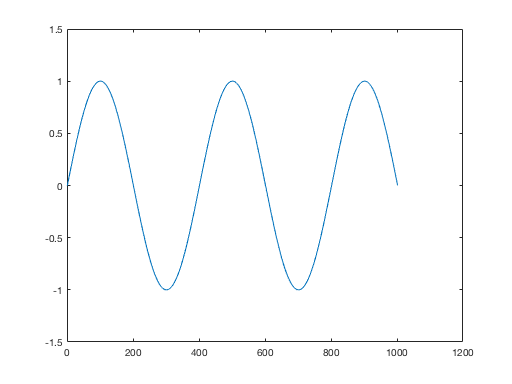
Problem 3

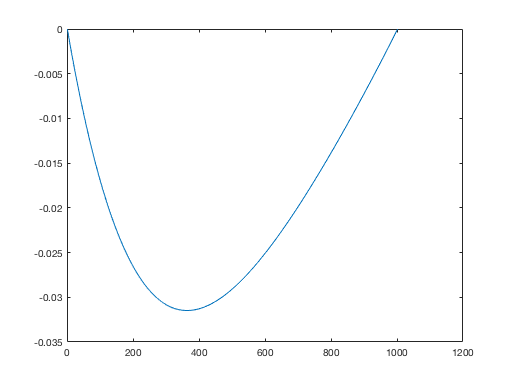
Let r(x)=-2x+3 and f(x) = -sin(25\*pi\*x) and let the program run on 25 nodes. MATLAB plot:



Let r(x)=-x and f(x) = -(25.0\*(pi^2)I+x)\*sin(5.0\*pi\*x); and let the program run on 25 nodes. MATLAB plot:



Let r(x)=-3x and f(x) = exp^(-3x); and let the program run on 25 nodes. MATLAB plot:



Source code:

#include <iostream>

#include <fstream>

#include <cmath>

#include <vector>

#include "mpi.h"

using namespace std;

// coefficient functions, analytical answer is u = sin(5pi\*x)

double r(const double x) {

return -3\*x;

}

double f(const double x) {

return exp(-3\*x);

}

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

const int n = 1000, iter\_max = 1000000;

const double h = 1.0/(n+1);

int rank, size, tag = 100;

MPI\_Status status;

MPI\_Init(&argc, &argv);

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

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

int L = n/size, R = n%size;

int I = (n+size-rank-1)/size;

int first = rank\*L+min(rank, R)+1;

typedef vector<double> Vec;

Vec rr(I+2), ff(I+2);

double x\_i;

for(int i = 0; i<ff.size(); ++i) {

x\_i = (first-1+i)\*h;

rr[i] = r(x\_i);

ff[i] = f(x\_i);

}

Vec u(I+2, 0), u\_new(I+2, 0);

int right = rank+1, left = rank-1;

if(rank == size-1) {

right = MPI\_PROC\_NULL;

} else if(rank == 0) {

left = MPI\_PROC\_NULL;

}

for(int step = 0; step < iter\_max; ++step) {

if(rank%2 == 0) { // red

MPI\_Send(&u[I], 1, MPI\_DOUBLE, right, tag, MPI\_COMM\_WORLD);

MPI\_Recv(&u.back(), 1, MPI\_DOUBLE, right, tag, MPI\_COMM\_WORLD, &status);

MPI\_Send(&u[1], 1, MPI\_DOUBLE, left, tag, MPI\_COMM\_WORLD);

MPI\_Recv(&u.front(), 1, MPI\_DOUBLE, left, tag, MPI\_COMM\_WORLD, &status);

} else { // black

MPI\_Recv(&u.front(), 1, MPI\_DOUBLE, left, tag, MPI\_COMM\_WORLD, &status);

MPI\_Send(&u[1], 1, MPI\_DOUBLE, left, tag, MPI\_COMM\_WORLD);

MPI\_Recv(&u.back(), 1,MPI\_DOUBLE, right, tag, MPI\_COMM\_WORLD, &status);

MPI\_Send(&u[I], 1,MPI\_DOUBLE, right, tag, MPI\_COMM\_WORLD);

}

// Jacobi iteration over inner points

for(int i = 1; i < u.size() - 1; ++i) {

u\_new[i] = (u[i-1]+u[i+1]-h\*h\*ff[i])/(2.0-h\*h\*rr[i]);

}

u.swap(u\_new); // runs in O(1)

}

const char file[] = "out";

ofstream fout;

MPI\_Recv(0, 0, MPI\_CHAR, left, tag, MPI\_COMM\_WORLD, &status);

if(rank == 0) {

fout.open(file); fout << 0 << ' ';

} // left boundary

else fout.open(file, ios::app);

for(int i = 1; i < u.size()-1; ++i)

fout << u[i] << ' ';

if(rank == size-1) fout << 0;

fout.close();

MPI\_Send(0, 0, MPI\_CHAR, right, tag, MPI\_COMM\_WORLD);

MPI\_Finalize();

return 0;

}