MA6252 Homework 3

Liu Zhaoqiang A0109983

Question 1: see Hw3_1.cpp

Question 2: see Hw3_2.cpp

For messages sent in different processes, the MPI implementation is fair. And for messages sent in the same process, the messages sent earlier will be received by process 0 earlier.

The partial result for is shown below:

First running:

```
[a0109983@hawk1 ~]$ mpirun -np 4 ./Hw3_2
Received: message 0 of process 1
Received: message 1 of process 1
Received: message 0 of process 3
Received: message 2 of process 1
Received: message 1 of process 3
Received: message 3 of process 1
Received: message 2 of process 3
Received: message 4 of process 1
Received: message 3 of process 3
Received: message 5 of process 1
Received: message 4 of process 3
Received: message 6 of process 1
Received: message 5 of process 3
Received: message 7 of process 1
Received: message 6 of process 3
Received: message 8 of process 1
Received: message 7 of process 3
Received: message 9 of process 1
Received: message 8 of process 3
Received: message 10 of process 1
Received: message 9 of process 3
Received: message 11 of process 1
Received: message 10 of process 3
Received: message 12 of process 1
Received: message 11 of process 3
Received: message 13 of process 1
Received: message 12 of process 3
Received: message 14 of process 1
Received: message 13 of process 3
Received: message 15 of process 1
Received: message 14 of process 3
Received: message 16 of process 1
Received: message 15 of process 3
Received: message 17 of process 1
Received: message 16 of process 3
Received: message 18 of process 1
Received: message 17 of process 3
Received: message 19 of process 1
Received: message 18 of process 3
Received: message 20 of process 1
Received: message 19 of process 3
```

Second running:

```
[a0109983@hawk1 ~]$ mpirun -np 4 ./Hw3_2
Received: message 0 of process 2
Received: message 1 of process 2
Received: message 0 of process 3
Received: message 2 of process 2
Received: message 1 of process 3
Received: message 3 of process 2
Received: message 2 of process 3
Received: message 4 of process 2
Received: message 3 of process 3
Received: message 5 of process 2
Received: message 4 of process 3
Received: message 6 of process 2
Received: message 5 of process 3
Received: message 7 of process 2
Received: message 6 of process 3
Received: message 8 of process 2
Received: message 7 of process 3
Received: message 9 of process 2
Received: message 8 of process 3
Received: message 10 of process 2
Received: message 9 of process 3
Received: message 11 of process 2
Received: message 10 of process 3
Received: message 12 of process 2
Received: message 11 of process 3
Received: message 13 of process 2
Received: message 12 of process 3
Received: message 14 of process 2
Received: message 13 of process 3
Received: message 15 of process 2
Received: message 14 of process 3
Received: message 16 of process 2
Received: message 15 of process 3
Received: message 17 of process 2
Received: message 16 of process 3
Received: message 18 of process 2
Received: message 17 of process 3
Received: message 19 of process 2
Received: message 18 of process 3
Received: message 20 of process 2
Received: message 19 of process 3
```

Question 3: see Hw3_3.cpp.

In this Question, we consider the computation domain as [0,1], and the boundary value is u(0)=a, u(1)=b. Therefore, the exact solution should be u(x)=(b-a)x+a. The main part for the Jacobi algorithm in this question is:

$$x_i^{k+1} = 0.5(x_{i-1}^k + x_{i+1}^k)$$

Where $x_0^k = a, x_n^k = b$(n is the total number of small intervals).