Assignment #08

Dec, 11, 2017

JinYeong Wang
Parallel Computing Lab.
Mechanical Engineering
Hanyang University

Problem Definition

- Compute

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = \cos[\pi x]\sin[y] + \pi^2 \cos[\pi x]\sin[y]$$

- While

$$0 \le x \le 2 \; and \; 0 \le y \le 2$$

- Exact solution

$$u_{exact}(x, y) = \cos[\pi x]\sin[\pi + y]$$

- Using Jacobi method, compute till

$$L_1 \; Error = \sum \left| u_m^{k+1} - u_m^k \right| < 10^{-4}$$

- Number of grid nodes are at least 500, more than 5 cases
  - 500, 525, 550, 575, 600

- Plot L2 error vs. Grid node

Pseudo Code

If size == 2

rank	U	displs	counts	U_p
0	0	displs [rank] = 0	counts [rank] = 3	0
	1			1
	2			2
1	3	displs [rank] = 3		0
	4			1
	5			2

Parallelize Memory Indexing

```
1.  N_fix = N * N
2.  While( N_fix % size )
3.    N_fix++
4.  rows_p = N_fix / size
5.  If( rank == size-1 )
6.    rows_p = N * N - N_fix / size  * (size - 1)
7.  U = memory allocation( N * N )
8.  U_p = memory allocation( rows_p )
9.  counts = memory allocation( size )
10. displs = memory allocation( size )
11. Sum = 0
12. for( i, 0 to size - 1 )
13.   counts[ i ] = N_fix / size
14.   displs[ i ] = sum
15.   sum += counts[ i ]
16. counts[ size – 1 ] = N * N – sum
17. displs[ size – 1 ] = sum
```

Pseudo Code

Boundary Condition

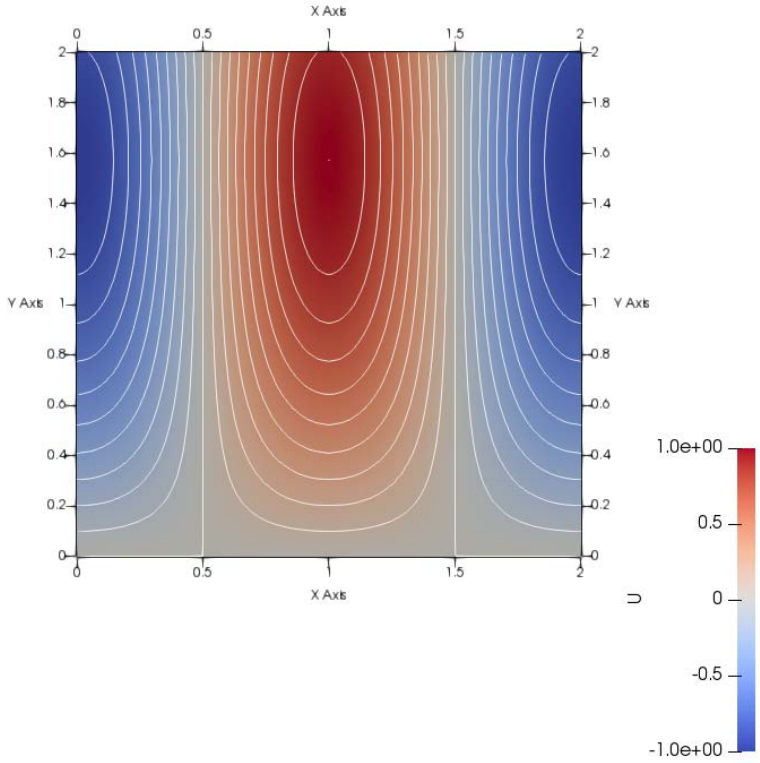
```
1.  for ( index, 0 to counts[rank] )
2.    pos = index + displs[rank]
3.    i = pos % N
4.    j = pos / N
5.    if( pos is not boundary )
6.      U_p[pos] = 0.0
7.    else if ( 0,y )
8.      U_p[pos] = sin(pi*delta*j)
9.    else if ( 2,y )
10.     U_p[pos] = sin(pi*delta*j)
11.   else if ( x,0 )
12.     U_p[pos] = 0.0
13.   else if ( x,2 )
14.     U_p[pos] = cos(pi*delta*i)*sin(pi+2.0)
```

Jacobi Iteration Method

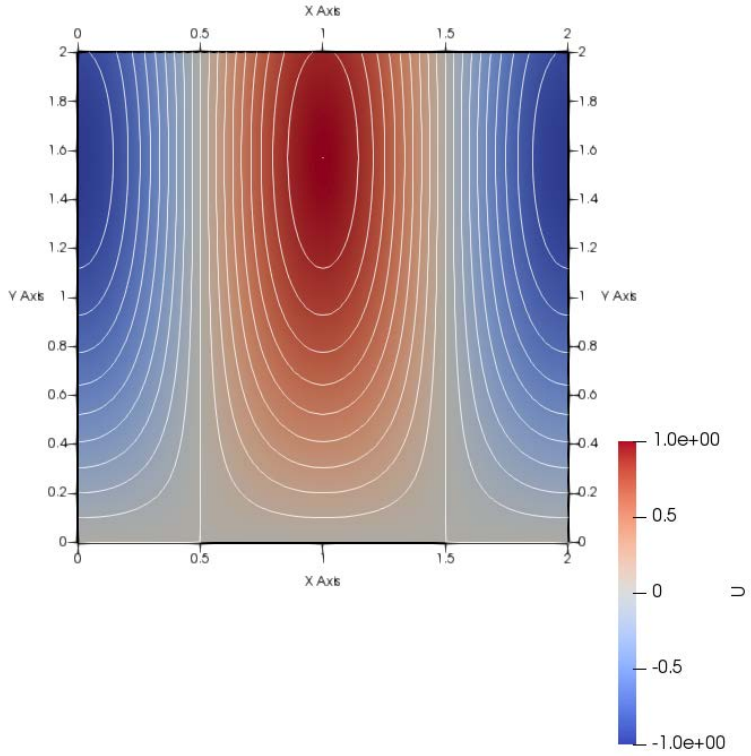
```
1.  Tol = 0.00001
2.  error = 1.0
3.  repeat
4.    MPI_Allgatherv( U_p to U )
5.    for ( index, 0 to counts[rank] )
6.      pos = index + displs[rank]
7.      i = pos % N
8.      j = pos / N
9.      x = delta * i
10.     y = delta * j
11.     f = cos(pi*x)sin(y)+pi*pi*cos(pi*x)sin(y)
12.     if( pos is not boundary )
13.       U_p[pos] = (U[pos+1]+U[pos-
    1]+U[pos+N]+U[pos-N]-delta*delta*f)*0.25
14.   error_p = 0.0
15.   for ( index, 0 to counts[rank] )
16.     pos = index + displs[rank]
17.     error_p = fabs(U_p[index] – U[pos])
18.   MPI_Allreduce(error_p to error, sum)
19. until error < Tol
```

N=500

Solve by Jacobi method

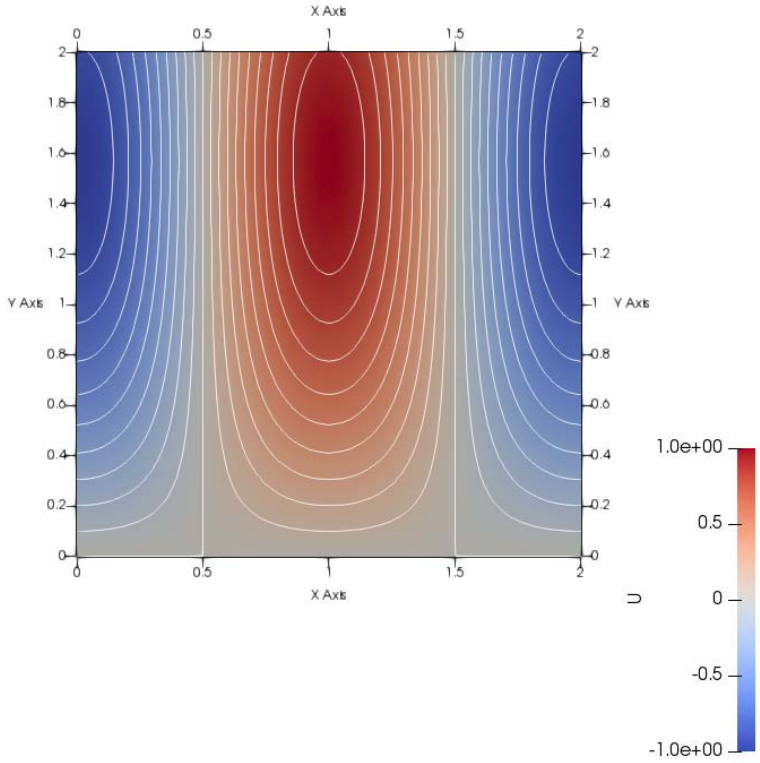


Exact solution

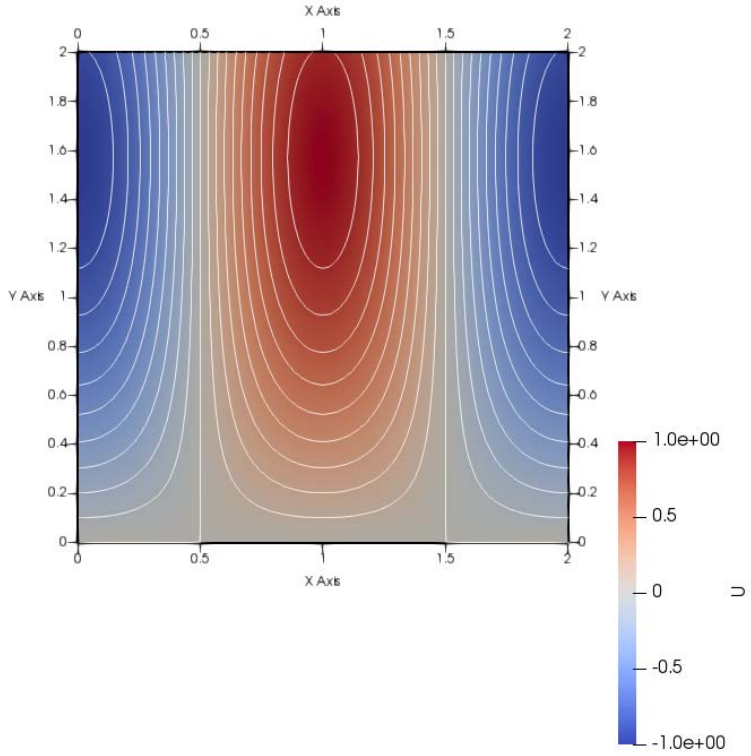


N=525

Solve by Jacobi method

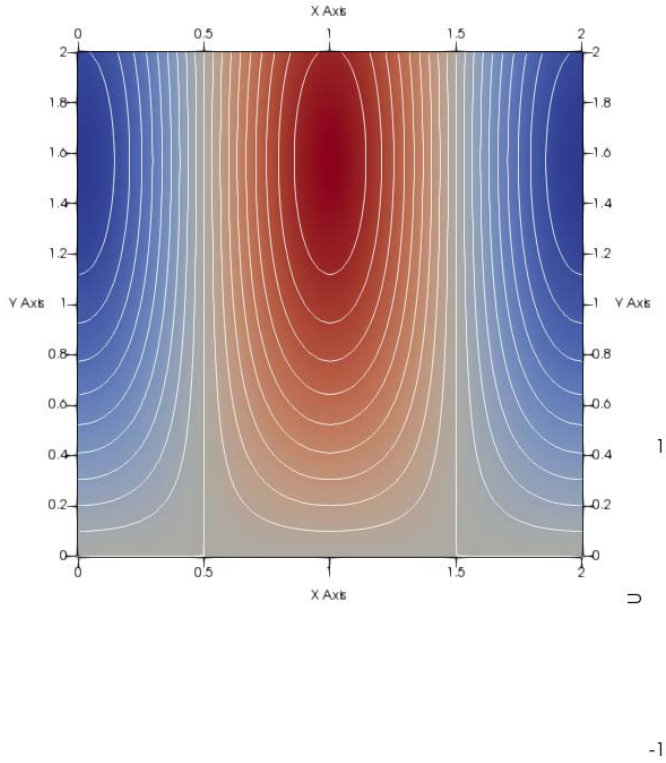


Exact solution

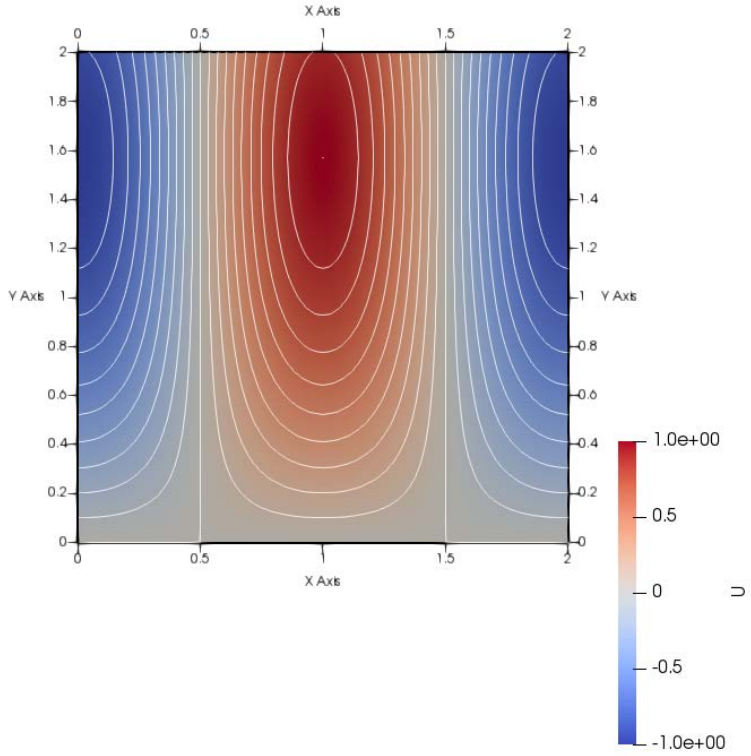


N=550

Solve by Jacobi method

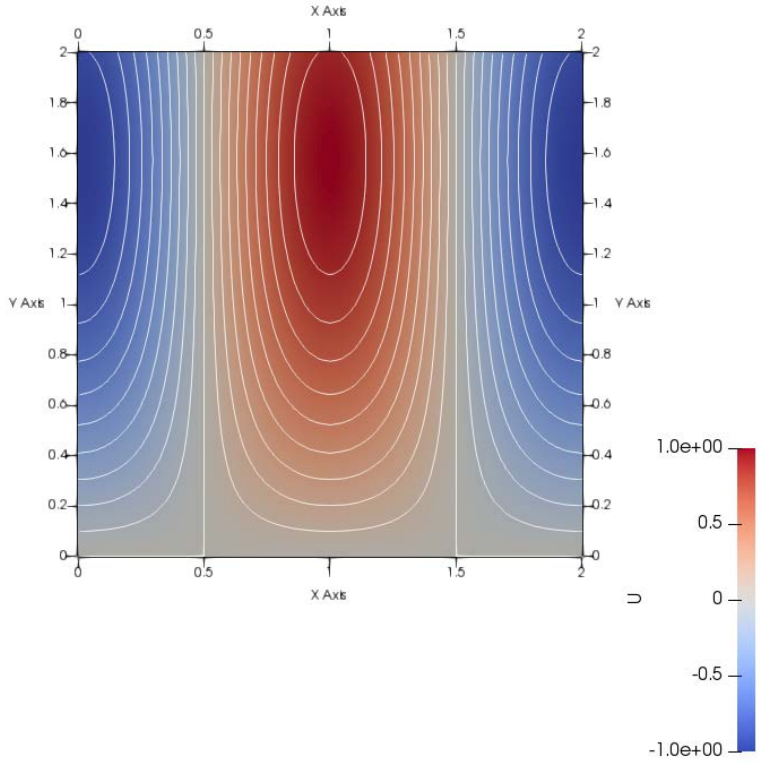


Exact solution

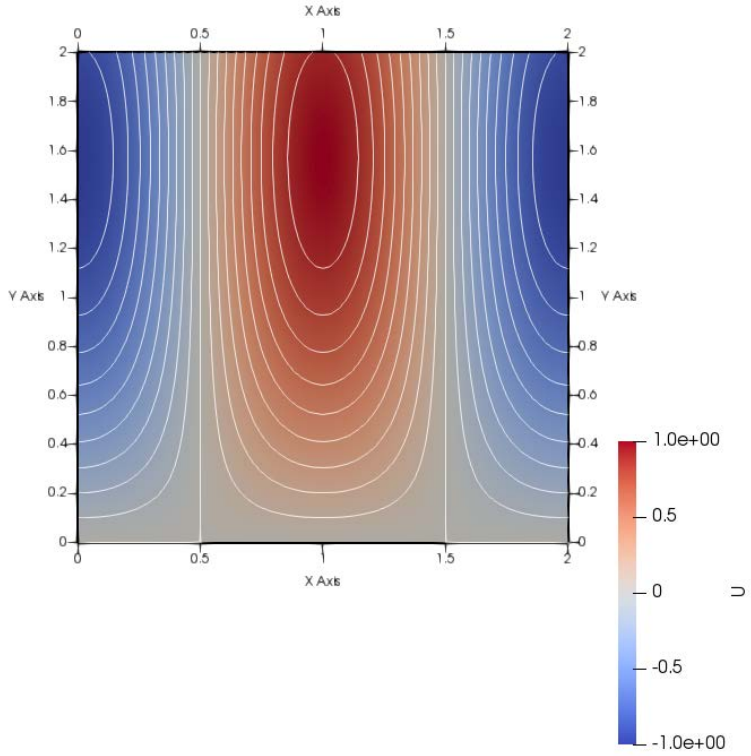


N=575

Solve by Jacobi method

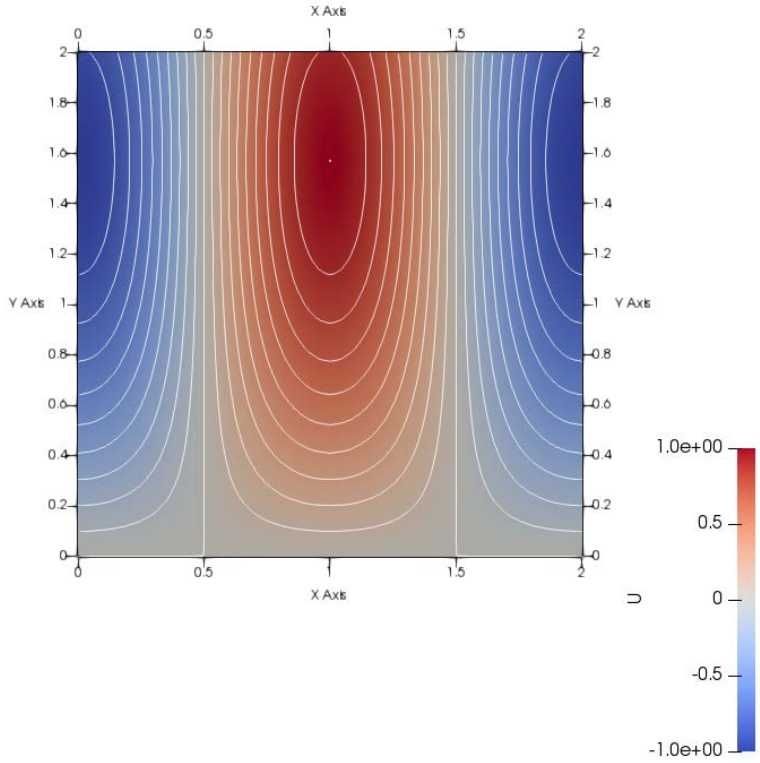


Exact solution

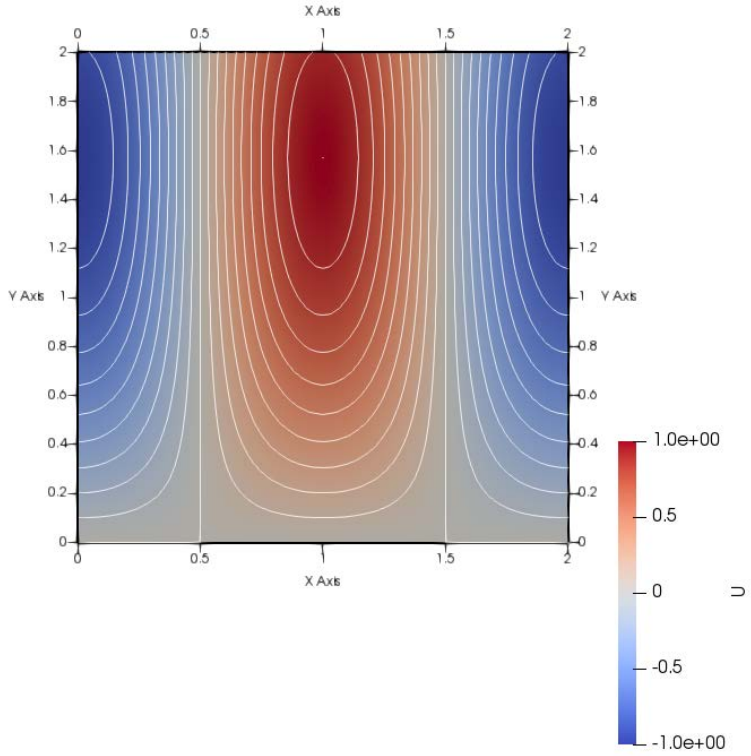


N=600

Solve by Jacobi method



Exact solution



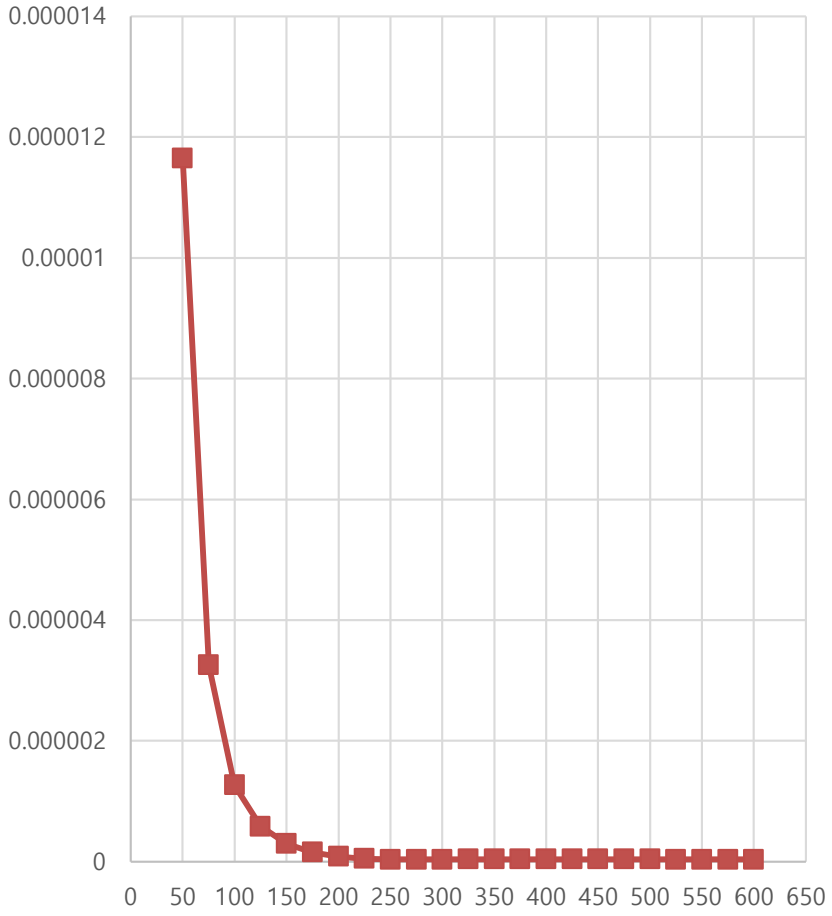
Result

Elliptic, N=500, Exact
Elliptic, N=500, L2=0.0000000391478, np=1, t=4555.590961, Jacobi
Elliptic, N=500, L2=0.0000000391478, np=2, t=2383.236201, Jacobi
Elliptic, N=500, L2=0.0000000391478, np=3, t=1703.346945, Jacobi
Elliptic, N=500, L2=0.0000000391478, np=4, t=1290.001672, Jacobi
Elliptic, N=500, L2=0.0000000391478, np=5, t=1086.956939, Jacobi
Elliptic, N=500, L2=0.0000000391478, np=6, t=954.962086, Jacobi
Elliptic, N=525, Exact
Elliptic, N=525, L2=0.0000000382156, np=1, t=5476.488216, Jacobi
Elliptic, N=525, L2=0.0000000382156, np=2, t=2950.183918, Jacobi
Elliptic, N=525, L2=0.0000000382156, np=3, t=2062.157977, Jacobi
Elliptic, N=525, L2=0.0000000382156, np=4, t=1570.662798, Jacobi
Elliptic, N=525, L2=0.0000000382156, np=5, t=1317.836718, Jacobi
Elliptic, N=525, L2=0.0000000382156, np=6, t=1169.407257, Jacobi

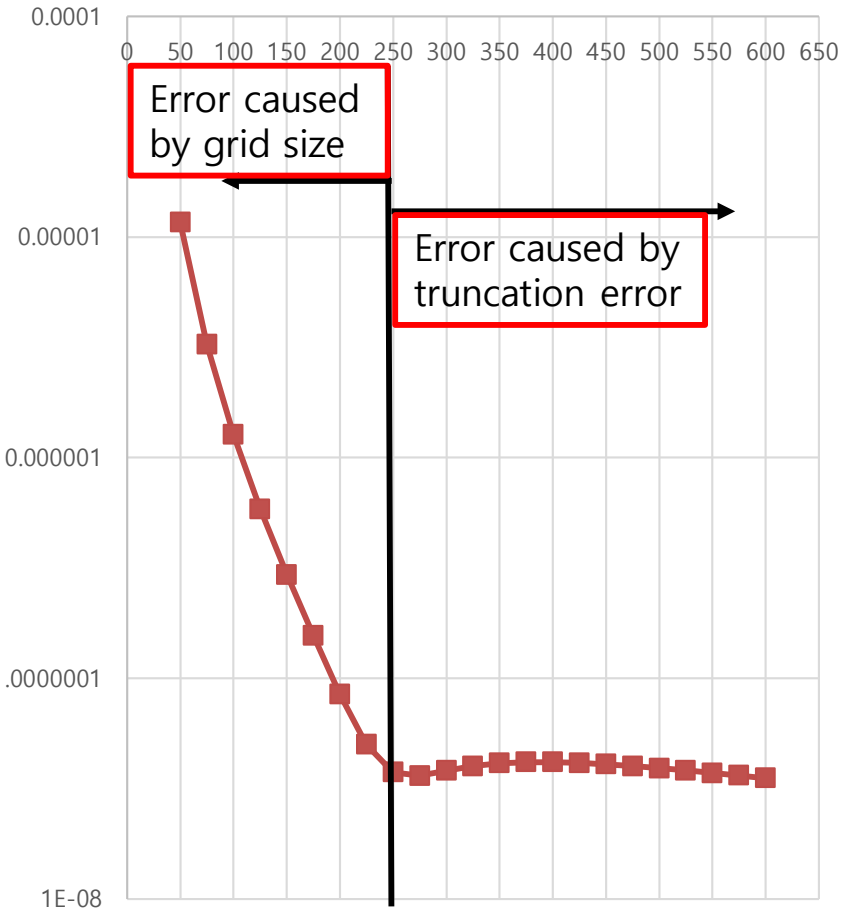
Elliptic, N=550, Exact
Elliptic, N=550, L2=0.0000000372543, np=1, t=6655.835934, Jacobi
Elliptic, N=550, L2=0.0000000372543, np=2, t=3568.105374, Jacobi
Elliptic, N=550, L2=0.0000000372543, np=3, t=2486.185437, Jacobi
Elliptic, N=550, L2=0.0000000372543, np=4, t=1902.978690, Jacobi
Elliptic, N=550, L2=0.0000000372543, np=5, t=1596.473933, Jacobi
Elliptic, N=550, L2=0.0000000372543, np=6, t=1434.189380, Jacobi
Elliptic, N=575, Exact
Elliptic, N=575, L2=0.0000000362858, np=1, t=7971.567872, Jacobi
Elliptic, N=575, L2=0.0000000362858, np=2, t=4261.728387, Jacobi
Elliptic, N=575, L2=0.0000000362858, np=3, t=2976.539548, Jacobi
Elliptic, N=575, L2=0.0000000362858, np=4, t=2282.359113, Jacobi
Elliptic, N=575, L2=0.0000000362858, np=5, t=1923.139598, Jacobi
Elliptic, N=575, L2=0.0000000362858, np=6, t=1741.285618, Jacobi
Elliptic, N=600, Exact
Elliptic, N=600, L2=0.0000000353238, np=1, t=9247.522723, Jacobi
Elliptic, N=600, L2=0.0000000353238, np=2, t=5080.173374, Jacobi
Elliptic, N=600, L2=0.0000000353238, np=3, t=3598.920527, Jacobi
Elliptic, N=600, L2=0.0000000353238, np=4, t=2773.763258, Jacobi
Elliptic, N=600, L2=0.0000000353238, np=5, t=2351.110034, Jacobi
Elliptic, N=600, L2=0.0000000353238, np=6, t=2152.320612, Jacobi

Result

L2 error vs # of grid nodes

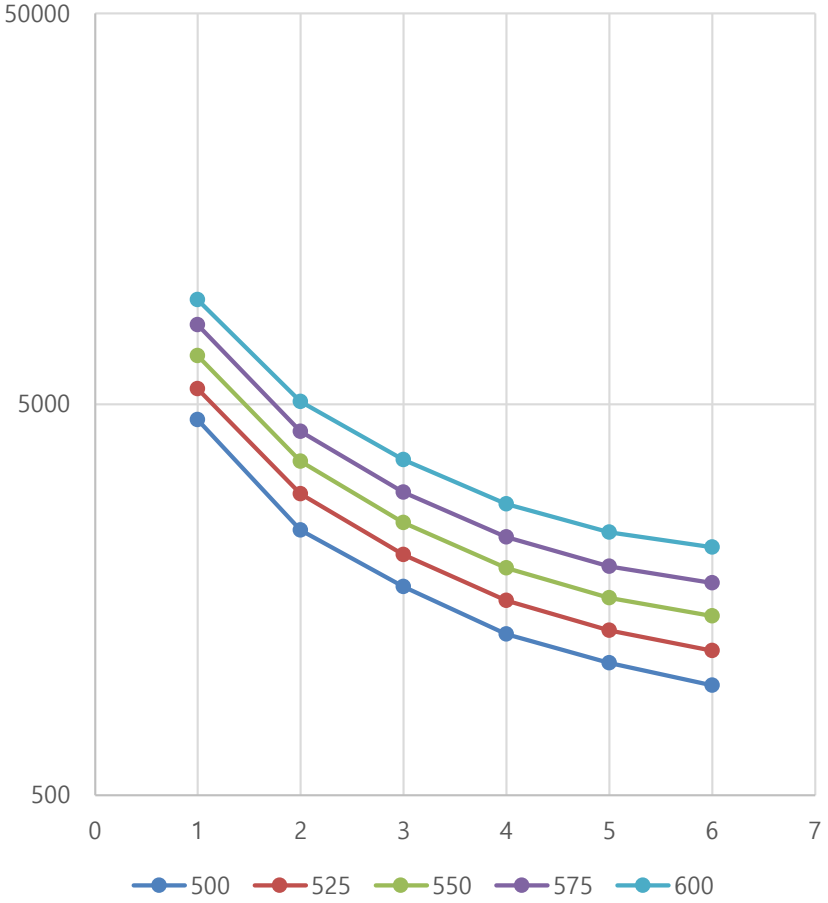


Log(L2 error) vs # of grid nodes



Result

Log(Time) vs NP



Speedup