**Lab 7 Write-Up**

**Part 2**

The SOR implemented on the GPU using CUDA executed in **42933.718750ms,** whereas for CPU the implementation took **0.069ms.**

The following are the comparisons for the results of the **first 50** values in the array:

**CPU GPU**

0: 1557181440.00000000 1264305664.00000000

1: 993237312.00000000 547869760.00000000

2: 2064222976.00000000 1878329472.00000000

3: 165421824.00000000 440974688.00000000

4: 1497211520.00000000 1394771328.00000000

5: 906312512.00000000 339387712.00000000

6: 196663968.00000000 1203565568.00000000

7: 1661110784.00000000 359470144.00000000

8: 279437248.00000000 757453696.00000000

9: 399124672.00000000 1455555072.00000000

10: 772164480.00000000 421628512.00000000

11: 1031763008.00000000 269884608.00000000

12: 416787904.00000000 851543296.00000000

13: 1986810624.00000000 1034040768.00000000

14: 1427428608.00000000 878260480.00000000

15: 1636480768.00000000 2110470528.00000000

16: 864814912.00000000 1047292544.00000000

17: 1424373888.00000000 559716544.00000000

18: 1055845440.00000000 161955760.00000000

19: 1991412736.00000000 93015304.00000000

20: 297150752.00000000 52114584.00000000

21: 1937330176.00000000 2126837504.00000000

22: 1209663872.00000000 1299515648.00000000

23: 1968894336.00000000 830320640.00000000

24: 995229696.00000000 14245510.00000000

25: 1512426112.00000000 846971008.00000000

26: 440140448.00000000 183719520.00000000

27: 2111515648.00000000 959326656.00000000

28: 693171328.00000000 376500608.00000000

29: 795416768.00000000 1871880192.00000000

30: 1960820608.00000000 91905616.00000000

31: 102869040.00000000 1640806272.00000000

32: 1788654080.00000000 272266272.00000000

33: 1877559936.00000000 1970235136.00000000

34: 268290864.00000000 2081780992.00000000

35: 1138381952.00000000 1667037696.00000000

36: 636388864.00000000 162139152.00000000

37: 464954848.00000000 1137862912.00000000

38: 652009088.00000000 2026507776.00000000

39: 915826112.00000000 919592896.00000000

40: 864079488.00000000 445934368.00000000

41: 1424173568.00000000 300652640.00000000

42: 1947589120.00000000 1189477504.00000000

43: 1280867456.00000000 1297477632.00000000

44: 1263500416.00000000 1334693376.00000000

45: 1227534080.00000000 2067737984.00000000

46: 769864512.00000000 1260464512.00000000

47: 2128315392.00000000 234502272.00000000

48: 504424320.00000000 479970848.00000000

49: 1825709952.00000000 1422420224.00000000

As it can be seen, although the results for SOR for both CPU & GPU versions both have the same orders of magnitude, the values for the elements in the arrays are not entirely the same. This is expected owing to the fact that floating point values on the GPU are a little non-deterministic hence resulting in the different values. This leads to particularly divergent values on SOR, since this is a convergence function with many iterations (2000 in this case), and all data elements depend on other data elements in the array.

**Part 3**

After blocking, the GPU CUDA version of the code executed in **37.27ms**, whereas for CPU the SOR took **0.069ms.** As it is expected the execution time for the blocked GPU execution took significantly lesser time than the non-blocked version.

The following are the comparisons for the results of the **first 50** values in the array:

**CPU GPU**

0: 1557181440.00000000 1264305664.00000000

1: 993237312.00000000 547869760.00000000

2: 2064222976.00000000 1878329472.00000000

3: 165421824.00000000 440974688.00000000

4: 1497211520.00000000 1394771328.00000000

5: 906312512.00000000 339387712.00000000

6: 196663968.00000000 1203565568.00000000

7: 1661110784.00000000 359470144.00000000

8: 279437248.00000000 757453696.00000000

9: 399124672.00000000 1455555072.00000000

10: 772164480.00000000 421628512.00000000

11: 1031763008.00000000 269884608.00000000

12: 416787904.00000000 851543296.00000000

13: 1986810624.00000000 1034040768.00000000

14: 1427428608.00000000 878260480.00000000

15: 1636480768.00000000 2110470528.00000000

16: 864814912.00000000 1047292544.00000000

17: 1424373888.00000000 559716544.00000000

18: 1055845440.00000000 161955760.00000000

19: 1991412736.00000000 93015304.00000000

20: 297150752.00000000 52114584.00000000

21: 1937330176.00000000 2126837504.00000000

22: 1209663872.00000000 1299515648.00000000

23: 1968894336.00000000 830320640.00000000

24: 995229696.00000000 14245510.00000000

25: 1512426112.00000000 846971008.00000000

26: 440140448.00000000 183719520.00000000

27: 2111515648.00000000 959326656.00000000

28: 693171328.00000000 376500608.00000000

29: 795416768.00000000 1871880192.00000000

30: 1960820608.00000000 91905616.00000000

31: 102869040.00000000 1640806272.00000000

32: 1788654080.00000000 272266272.00000000

33: 1877559936.00000000 1970235136.00000000

34: 268290864.00000000 2081780992.00000000

35: 1138381952.00000000 1667037696.00000000

36: 636388864.00000000 162139152.00000000

37: 464954848.00000000 1137862912.00000000

38: 652009088.00000000 2026507776.00000000

39: 915826112.00000000 919592896.00000000

40: 864079488.00000000 445934368.00000000

41: 1424173568.00000000 300652640.00000000

42: 1947589120.00000000 1189477504.00000000

43: 1280867456.00000000 1297477632.00000000

44: 1263500416.00000000 1334693376.00000000

45: 1227534080.00000000 2067737984.00000000

46: 769864512.00000000 1260464512.00000000

47: 2128315392.00000000 234502272.00000000

48: 504424320.00000000 479970848.00000000

49: 1825709952.00000000 1422420224.00000000

Again, although the results for both CPU & GPU versions have the same orders of magnitude, the values for the elements in the arrays are not entirely the same. The reasoning is the same- this is expected owing to the fact that floating point values on the GPU are a little non-deterministic hence resulting in the different values. This leads to particularly divergent values on SOR, since this is a convergence function with many iterations (2000 in this case), and all data elements depend on other data elements in the array.