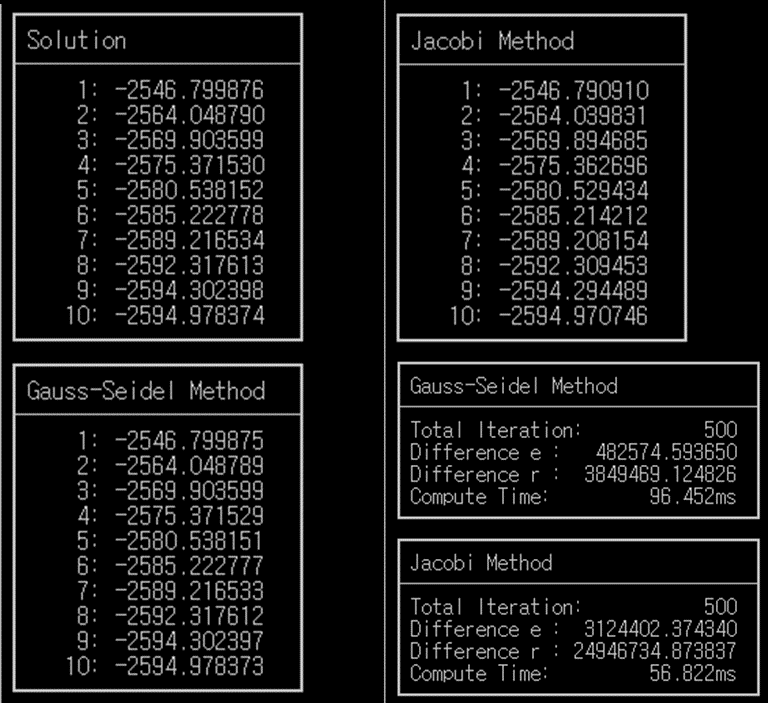
***수치 컴퓨팅 및 GPU 프로그래밍 – 숙제 4***

***Jacobi Method and Gauss-Seidel Method***

120170420 신광수

* 과제에서 요구한 문제 중 문제 1 및 문제 2만 작성하였음.
* 사용자 인터페이스 설명
  + 
  + 프로그램 시작 시, 사용자로부터 입력 받을 데이터와 총 Iteration 수를 받는다.
  + 파일로부터 데이터를 읽고, 알고 있는 해를 출력하며, 순차적으로 Gauss-Seidel 및 Jacobi의 결과로부터 얻은 해를 출력한다. 그 이후 각 방법의 Iteration 및 오차, 계산에 소요된 시간을 출력한다.
  + 
  + 이 때 difference E 및 R은 다음과 같이 계산하였다.
    - 를 계산한 후, 절대값을 취한 후에
    - 그 값을 모든 행에 대해 더한 값을 취하였다.
  + 사용된 모든 메모리를 해제하고 프로그램을 종료한다.
* 문제 2의 경우 2는 주어진 데이터 셋에 대해 일어나지 않았으므로 따로 구현하지 않았다.
* 식 별로 Iteration 과 Difference 및 Compute Time 의 변화는 다음과 같았다.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 1번 | Gauss-Seidel Method | | | Jacobi Method | | |
| Iteration | e | r | Compute Time | e | r | Compute Time |
| 500 | 0.010527 | 0.122455 | 92.929 | 2.859599 | 34.105152 | 52.384 |
| 1000 | 0.002105 | 0.019061 | 181.595 | 0.008428 | 0.097881 | 104.958 |
| 5000 | 0.002104 | 0.01906 | 951.738 | 0.002104 | 0.01906 | 533.510 |
| 10000 | 0.002104 | 0.01906 | 1908.259 | 0.002104 | 0.01906 | 1073.557 |
| 2번 | Gauss-Seidel Method | | | Jacobi Method | | |
| Iteration | e | r | Compute Time | e | r | Compute Time |
| 500 | 482574.5937 | 3849469.125 | 96.597 | 3124402.374 | 24946734.87 | 53.737 |
| 1000 | 10247.04923 | 81740.0287 | 194.475 | 457316.7663 | 3651381.375 | 109.208 |
| 5000 | 0.004952 | 0.036366 | 1055.918 | 0.088054 | 0.706128 | 630.276 |
| 10000 | 0.004952 | 0.036366 | 1958.755 | 0.004952 | 0.036366 | 1101.330 |
| 3번 | Gauss-Seidel Method | | | Jacobi Method | | |
| Iteration | e | r | Compute Time | e | r | Compute Time |
| 500 | 1210462.872 | 9660271.493 | 86.198 | 5527204.974 | 44148994.97 | 48.782 |
| 1000 | 47402.52912 | 378295.69 | 170.465 | 1135867.008 | 9072010.759 | 100.085 |
| 5000 | 0.008634 | 0.064652 | 838.986 | 2.630024 | 21.001088 | 500.521 |
| 10000 | 0.008634 | 0.064651 | 1724.632 | 0.008634 | 0.064652 | 973.054 |
| 4번 | Gauss-Seidel Method | | | Jacobi Method | | |
| Iteration | e | r | Compute Time | e | r | Compute Time |
| 500 | 15957.82596 | 127036.3965 | 83.743 | 377946.0302 | 3013813.409 | 47.483 |
| 1000 | 10247.04923 | 81740.0287 | 162.575 | 457316.7663 | 3651381.375 | 94.977 |
| 5000 | 0.003077 | 0.23208 | 883.910 | 0.003077 | 0.023208 | 482.604 |
| 10000 | 0.003077 | 0.023208 | 1717.172 | 0.003077 | 0.023208 | 949.287 |

* + 결과에 대해 흥미로운 점은 같은 수행 횟수를 취하였을 때 Jacobi의 경우가 그 속도가 1.7배 가량 빠름을 알 수 있었다.
  + 다만 같은 수준의 difference를 구하기 위해서는 Jacobi 가 조금 더 수행 시간이 길어짐을 알 수 있었다. 즉, 해를 구하기 위한 수행 횟수가 더 증가함에 따라 하나의 수행에 대해서는 속도가 빠르더라도 전체 수행 시간에 대해서는 Jacobi 가 크게 빠름을 볼 수 없었다.
  + 작성을 하진 않았지만, 문제 3에서의 GPU 상의 Jacobi를 진행한다면 Gauss-Seidel에 비해 큰 속도로 빨라질 수 있다. Gauss-Seidel 방식의 경우 이전 행의 X의 값을 구해야지만 다음 행의 X를 구할 수 있으므로 병렬화가 어렵다. 하지만 Jacobi는 그런 부분에서 자유로우므로, 그 속도 또한 급격하게 증가할 수 있을 것이다.