1. Matrix-vector multiplication

```
[mg5610@cuda2 homework4]$ ./matvecmult 2000
Time taken for serial portion: 0.007217
Time taken for CUDA portion: 0.179786
Error: 1.705303e-13
[mg5610@cuda2 homework4]$ ./matvecmult 20000
Time taken for serial portion: 0.440130
Time taken for CUDA portion: 0.485707
Error: 3.637979e-12
[mg5610@cuda2 homework4]$ ./matvecmult 80000
Time taken for serial portion: 7.093167
Time taken for CUDA portion: 0.158712
Error: 2.023309e+04
[mg5610@cuda2 homework4]$ ./matvecmult 30000
Time taken for serial portion: 0.999018
Time taken for CUDA portion: 0.915332
Error: 4.547474e-12
```

2.

Jacobi Serial version

```
Iteration: 790. Residual: 0.937298
Iteration: 800. Residual: 0.936894
Iteration: 810. Residual: 0.936493
Iteration: 820. Residual: 0.936094
Iteration: 830. Residual: 0.935697
Iteration: 840. Residual: 0.935303
Iteration: 850. Residual: 0.934911
Iteration: 860. Residual: 0.934522
Iteration: 870. Residual: 0.934134
Iteration: 880. Residual: 0.933749
Iteration: 890. Residual: 0.933367
Iteration: 900. Residual: 0.932986
Iteration: 910. Residual: 0.932607
Iteration: 920. Residual: 0.932231
Iteration: 930. Residual: 0.931856
Iteration: 940. Residual: 0.931484
Iteration: 950. Residual: 0.931113
Iteration: 960. Residual: 0.930745
Iteration: 970. Residual: 0.930378
Iteration: 980. Residual: 0.930013
Iteration: 990. Residual: 0.929650
Time elapsed: 1.888641
```

Jacobi CUDA version

```
Iteration: 780. Residual: 0.937704
Iteration: 790. Residual: 0.937298
Iteration: 800. Residual: 0.936894
Iteration: 810. Residual: 0.936493
Iteration: 820. Residual: 0.936094
Iteration: 830. Residual: 0.935697
Iteration: 840. Residual: 0.935303
Iteration: 850. Residual: 0.934911
Iteration: 860. Residual: 0.934522
Iteration: 870. Residual: 0.934134
Iteration: 880. Residual: 0.933749
Iteration: 890. Residual: 0.933367
Iteration: 900. Residual: 0.932986
Iteration: 910. Residual: 0.932607
Iteration: 920. Residual: 0.932231
Iteration: 930. Residual: 0.931856
Iteration: 940. Residual: 0.931484
Iteration: 950. Residual: 0.931113
Iteration: 960. Residual: 0.930745
Iteration: 970. Residual: 0.930378
Iteration: 980. Residual: 0.930013
Iteration: 990. Residual: 0.929650
Time elapsed: 0.949479
```

3. We have generated some training data on which we will run our Hogwild SGD on. Our next course of action is to write the CUDA code which will parallelize the SGD.