The overall goal of this homework is to obtain some hands-on experience on implementing Krylov methods on NVIDIA GPUs, and to invoke awareness of the communication bottleneck.

1. Implement a GPU-accelerated Conjugate Gradient (CG) solver based on routines taken from NVIDIA’s cuBLAS and cuSPARSE library. Use the matrix I/O functions developed in the last homework to read in a matrix in CSR format. Use a right-hand-side $b$ with all entries equal to one and solve the linear system of the form $Ax=b$ for the test matrix Trefethen_2000.mtx available in the University of Florida Matrix Collection. Report the iteration count and the runtime needed.

2. Estimate the data transfers between GPU multiprocessors and GPU main memory needed in every iteration of your CG implementation for linear system of size $n$ and a matrix containing $nnz$ nonzeros. You may distinguish between reads and writes.

3. Optimize the communication pattern by designing algorithm-specific routines that reduce the overall transfer balance.

4. Implement the algorithm-specific routines using the CUDA language.

5. Run the CG solver using the algorithm-specific routines and report iteration count and runtime.

Resources:

University of Florida Matrix Collection: http://www.cise.ufl.edu/research/sparse/matrices/

cuBLAS reference: http://docs.nvidia.com/cuda/cublas/

cuSPARSE reference: http://docs.nvidia.com/cuda/cusparse/

Project:

Implement the IDR method using the cuBLAS and cuSPARSE routines, design and implement algorithm-specific routines for reduced data transfers, compare runtime and iteration count of the two implementations.