**ME964**
**High Performance Computing for Engineering Applications**
**Default Midterm Project: Solving a Large, Dense Banded Linear System**
**Date Due: April 12, 2012 – 23:59 PM**
**Intermediate Report Due: March 29 – 23:59 PM**

You will have to write CUDA code that solves on a GTX480 a linear system $\mathbf{Ax} = \mathbf{b}$, where $\mathbf{A} \in \mathbb{R}^{n\times n}$, and $\mathbf{b} \in \mathbb{R}^{n\times m}$. The easy way out is to take $m = 1$. Kudos to you if your program handles the nontrivial case $m > 1$; however, you will not be penalized if your code only handles the $m = 1$ case. Note that a quick intro to the concept of banded matrix is available on Wikipedia [1].

This project is only vaguely specified in terms of the size and structure of the dense matrix $\mathbf{A}$. It will be up to you to push the limit on the value of $n$ and the value of the bandwidth $k$. The goal is to solve systems as large as possible, as fast as possible. Note that between solving a system with $n = 10^7$ and a small value of $k$ such as $k = 20$, I prefer the scenario where the matrix has a smaller dimension but a larger bandwidth. That is, I am more interested in cases where the values of $n$ and $k$ are relatively close, say $k \approx 0.5n$. However, it is ok if you prefer the former scenario. In terms of input, generate your own $\mathbf{A}$ and $\mathbf{b}$ inputs. To keep things simple, have $\mathbf{A}$ be diagonally dominant, set $\mathbf{x} = [1,1,\ldots,1]^T$, and choose $\mathbf{b} = \mathbf{A}\cdot\mathbf{x}$ (in other words, you know what the solution should be).

In your report, you will have to touch on the following:

- The mathematical algorithm embraced to solve this problem
- The format in which the code expects the inputs $\mathbf{A}$ and $\mathbf{b}$ to be provided.
- Your software design solution. Comment on
    a) your use of shared memory, if any
    b) the type of global memory access (coalesced vs. non-coalesced)
    c) use of synchronization barriers
    d) any other CUDA features relevant to your design
- Run a **`cuda-memcheck`** on the final version of your code from within **`cuda-gdb`** and provide a printout of the report produced by **`cuda-memcheck`**. Comment on any unusual output you notice in that report.
- Profile your code using **`nvvp`** and interpret/comment on the profiling results. Include pictures if helpful.
- Run a scaling analysis. To this end, consider a variety of dimensions $n$ and a variety of bandwidths $k$.
- Compare your linear solver against the CULA banded solver over a spectrum of dimensions $n$ and bandwidths $b$. The CULA banded solver is available on Euler.

REMARKS:

a) If you write code that systematically beats the CULA banded solver over a reasonable spectrum of dimensions $n$ and bandwidths $k$ you will earn an automatic A grade in ME964.
b) I would be very happy to meet with you and discuss algorithm design ideas. This can happen during or outside office hours.
c) You can work alone or team up with one ME964 colleague to work on this project.
d) An intermediate report that documents your progress towards finishing this project is due on March 29.

REFERENCES:

[1] Band Matrix: Wikipedia http://en.wikipedia.org/wiki/Band_matrix (accessed March 13, 2012)