GPU-Acceleration of Hejhal's Algorithm

Alex Karlovitz

A *graphics processing unit* (GPU) is a specialized electronic circuit originally designed for efficient computer graphics and image processing. GPUs support parallel computation on large blocks of data, making them useful for large-scale linear algebra routines. In this document, we go through the details for running Hejhal's algorithm on a GPU. The software is implemented in PyTorch, a python library for machine learning which supports GPU operations.

Appendix: Series Expansion of the K-Bessel Function

The K-Bessel function is a solution to the modified Bessel's equation

$$x^2 y'' + xy' - (x^2 + \nu^2)y = 0 \tag{1}$$

where ν is a (complex) parameter. Specifically, one defines a pair of standard solutions I_ν and K_ν to (1) as follows.

$$I_\nu(x) = \left(\frac{x}{2}\right)^\nu \sum_{k=0}^{\infty} \frac{1}{k!(\nu+1)_k} \left(\frac{x}{2}\right)^{2k}$$

where $(a)_n$ is the rising Pochhammer symbol. K_ν is the solution to (1) with the asymptotic

$$K_\nu(x) \sim \sqrt{\pi/(2x)}e^{-x}$$

as $x \to \infty$. To write down a series expansion, we can use the formula

$$K_\nu(z) = \frac{1}{2}\pi \frac{I_{-\nu}(z) - I_\nu(z)}{\sin(\nu\pi)}$$

Note: this formula is valid only for non-integer ν; however, the limit exists as ν approaches integers, so we can extend the formula in this manner.

In any case, ν will not be an integer in the applications to Hejhal's algorithm, so we can directly use the relation to I_ν to derive a series expansion for K_ν. Combining the series expansions for $I_{-\nu}$ and I_ν, this gives

$$K_\nu(z) = \frac{\pi}{2\sin(\nu\pi)} \sum_{k=0}^{\infty} \frac{(z/2)^{2k}}{k!} \left(\frac{(z/2)^{-\nu}}{(1-\nu)_k} - \frac{(z/2)^\nu}{(1+\nu)_k} \right)$$