A Proposed Parallel Architecture for Matrix Triangularization with Diagonal Loading

C. M. Rader
MIT Lincoln Laboratory *
Lexington, MA

April 22, 2004

Abstract

We are given M training samples of N-element column vectors in a matrix X and a predefined constant λ. We want to compute the lower-triangular matrix which is the Cholesky factor of $R = XX^t + \lambda I$ using highly parallel hardware, either using FPGAs or ASICs. Adding λ is called diagonal loading. In most adaptive processing applications, diagonal loading is used to reduce the sensitivity of the adaptation to errors due to insufficient sample support and to slight errors in the target model.

Mathematically, we first prefix $\sqrt{\lambda}I$ to X and then we use N size $M+1$ Householder postmultiplication, each carried out in a *virtual* superprocessor. We format the computation so that each Householder operation affects the same number of columns, but with fewer and fewer rows. *Actual* superprocessors each share the work of two virtual superprocessors. This allows each superprocessor to be physically identical with each other, while all are used with 100% efficiency. Data is moved from one superprocessor to another a row at a time.

*This work is sponsored by Defense Advanced Research Projects Agency, under Air Force Contract F19628-00-C-0002. Opinions, interpretations, conclusions, and recommendations are those of the author and are not necessarily endorsed by the United States Government.

Most of the arithmetic operations in a Householder transformation are simple in two important ways. They can be prescheduled, and they cannot give unbounded results. Hence they are easily parallelized. These simple operations are segregated into two groups. One group is dot-products and the other group uses operations that multiply one row by a scalar and add it to another row. Both these operations allow us to flow data with each column's elements moving only vertically and maintaining their order, a perfect recipe for systolic computation. Each row is used as soon as it is transferred.

The small number of more complicated operations we need, a square root and a few multiplications and divisions, are carried out in a physically separate part of the superprocessor. All the floating point operations are confined to this part of the processor. They don't need to be fast, because we can keep the multipliers and adders 100% occupied by working on several different triangularizations at the same time.