**Notes on performance modeling for stencil evaluation**

The discussion of 4 March 2013 prompts the following thoughts on generalizing the analysis beyond the particulars of memory-bandwidth limited performance bounds on 8th-order-in-space, 2nd-order-in-time constant-coefficient star-like stencils on Cartesian lattices.

There are generalizations in many directions that are of interest and that should be part of a comprehensive treatment of this subject in context, e.g., a PhD thesis or a field-defining review article.

*Number of dimensions for the independent variables*. So far, we have 3 space and 1 time. This is probably the most common requirement in practice. However, there are higher dimensional stencils, e.g., in Boltzmann problems (where the extra dimensions are in velocity phase space) or in radiation transport problems (where the extra dimensions are in angular coordinates for the radiation emanating from a spatial point, or in the energy of particles in different bins). Lower dimensional problems are also sometimes of interest.

*Number of dimensions for the dependent variables*. So far, we have 1 dependent variable. This is the least advantageous for performance, since there is little computation to do at one point. In practice, there are often many degrees of freedom defined at each spatial point, e.g., in combustion there may be a hundred interacting species or more. In compressible Navier-Stokes alone, there are five. The coupling of degrees of freedom at a point is not always dense, but the dense case is an interesting one. It leverages the group’s complementary work on dense linear algebra, and it increases arithmetic intensity.

*Spatial order*. Eighth order is an interesting case, since spatial order is increasing throughout numerical analysis, partly in response to improved understanding of the modeling advantages and partly due to the superior arithmetic intensity of high order. But we should make the semi-stencil bandwidth a parameter.

*Temporal order*. Second order is an interesting case, since it is the minimum that should be used in wave problems. But we should make the temporal order a parameter. It mainly enters through the number of copies of the spatial domain to be kept and has less reuse than the spatially connected points.

*Stencil shape*. So far, we have considered star-like stencils. Box-like stencils enter with mixed derivatives or various discretizations of convective derivatives.

*Variable coefficients*. Constant coefficient operators are perhaps rare in real problems – certainly they are rare in seismic problems where the entire point of doing imaging is to determine the variation in the parameters that affect the wave speed. We cannot get very far without considering the memory pressure from variable coefficients, since these coefficients typically have no reuse. An intermediate case is the piecewise constant case. This is relevant to seismic modeling. It allows significant reduction in storage at the expense of complexity of keeping track of where the layers change.

*Extent per dimension*. As noted in the Erlangen presentation in the form of the layer condition, the extent of the arrays per dimension is an important parameter in reuse. It motivates the local reuse of the diamond and trapezoidal structure of the KAUST research. It is relevant to study the extremes.

*Arithmetic precision*. As is well known, attached processors may have very different performance with precision, and the Bytes per word also directly (or inversely) affects storage and bandwidth when measured in words, so multiple precision analysis is important.

*Stencil structure*. All considerations so far are for structured Cartesian lattices. Many of the same considerations apply to unstructured, where there are analogous policies for maximizing memory locality. The difference is the synchronization and latency of the integer operations to trace neighbors.

*Limiting resources*. The analyses yesterday assumed that the only limiting resource was memory bandwidth and cache size in the form of the layer condition. A good way to study performance when there are many *N* potentially limiting resources is to assume that *N-1* of them are not limiting (operating at ideal) and to separately examine the bound for each resource in turn. Relevant hardware features include: storage, bandwidth, replacement policies at each level of memory, load and store units for integers and floats, prefetching hardware, and granularity of SIMDization,