A high arithmetic intensity multigrid preconditioner based on matrix polynomials

Bram Reps

Department of Mathematics and Computer Science,
University of Antwerp,
Belgium
bram.reps@uantwerpen.be
joint work with Wim Vanroose

One of the current tendencies in the development of new computer hardware is the increasing number of processors per chip. This naturally brings along the challenge of larger communication costs, relative to the speed of computation. The number of useful floating point operations per data-read from slow memory, i.e. the arithmetic intensity, plays a determining role in the efficiency of a numerical algorithm. Indeed, communication is avoided if more computations can be done with the data that is already in cache. In addition, algorithms require a minimal arithmetic intensity to benefit from vectorization inside a processor. In this presentation we discuss the efficiency of a multigrid method to solve the sparse linear system

$$Ax = b,$$

given by a stencil for matrix A, taking into account the arithmetic intensity.

A single Chebyshev smoothing step, $x_{i+1} = x_i + p_m(A)r_i$, relies on m successive matrix vector multiplications $w = Av$. The arithmetic intensity of the smoother can therefore be raised with a stencil compiler that rearranges the nested loops over the vector elements for optimized temporal data-locality and vectorization. As a consequence, the average time of one multigrid cycle drops with an increasing degree m of the Chebyshev polynomial. And thus, although the effect on the error reduction per multigrid cycle might be minor, the time to solution on multi- or many-core hardware can reduce if m increases.

Chebyshev polynomials are also used to improve the convergence rate in preconditioned Krylov subspace methods. The polynomial q_m of degree $m-1$ is chosen such that the condition number of the preconditioned system,

$$q_{m-1}(A)Ax = q_{m-1}(A)b \Leftrightarrow p_m(A)x = \hat{b}, \tag{1}$$

is optimally reduced. The method then requires less iterations to converge, yet in each iteration the matrix vector multiplication $w = Av$ is now replaced by $w = p_m(A)v$ at a higher computational cost. However, the latter operation can be computed with a higher arithmetic intensity, based on a stencil compiler.

Finally we choose a Krylov subspace method with a suboptimal polynomial p_m that only reduces the high frequency eigenmodes of A. Our multigrid method is then used as a complementary preconditioner for system (1), built upon the same computational kernel $w = p_m(A)v$ with high arithmetic intensity in the smoother.