Numerical software of the future will be different from what it looks nowadays. Nevertheless, reliability, portability across the rapidly evolving platforms, and satisfactory run time efficiency will remain to be standard requirements requested from high quality software. However, continually increasing problem sizes, computing environments which are exceedingly difficult to utilize, and new ways of software usage via the Internet and computational grids make it harder and harder to achieve these traditional goals. The only way of dealing with all these problems is to automatically generate and to automatically optimize code. This is not only economically more viable, but also has shown to lead to code that is better than what humans can achieve.

The talk will present methods to handle instruction set extensions like fused multiply-add (FMA) instructions and short vector SIMD instructions available on most of the current microprocessors. This topic will be discussed within the context of automatic performance tuning for digital signal processing (DSP) algorithms like discrete Fourier transforms, sine and cosine transforms, and wavelet transforms.

The following results are presented:

- Optimization of FFT kernels with respect to arithmetic cost and memory traffic by utilization of FMA instructions. These kernels are replacements for standard FFTW kernels and provide superior performance characteristics.

- FFTW-GEL, a two-way short vector SIMD version of FFTW where straight-line code is vectorized and a special compiler backend enables new levels of performance on AMD and Intel machines.

- An n-way vectorization of FFTW, that is portable across different short vector SIMD extensions, even for different vector lengths.

- SPIRAL-SIMD, an n-way vectorization of SPIRAL where a vectorizing formula compiler replaces SPIRAL’s SPL compiler. The new formula compiler provides vectorization of DSP algorithms on the algorithm level rather than on the code level. It is portable across short vector extensions and can handle arbitrary vector lengths.

The combination of all the results summarized above leads to the worldwide fastest FFT implementations currently available for general purpose microprocessors featuring short vector SIMD extensions (Intel Pentium III, Pentium 4, and AMD processors). In addition, very fast implementations for other DSP transforms are provided.

Performance of current single-precision FFT programs on an Intel Pentium 4 running at 3.05 GHz. SPIRAL SIMD outperforms all other high-performance FFT implementations including Intel’s hand-coded library MKL.