Computational Lower Bounds for the Fourier Transform

Abstract:

The Fourier Transform is one of the most important linear transformation in engineering and science with applications in a wide variety of domains spanning signal processing, pattern recognition, cryptography, polynomial multiplication, complexity theory, learning theory and more. In 1964, Cooley and Tukey discovered the Fast Fourier Transform (FFT) algorithm, which computes the transformation in time $O(n \log n)$ for a signal in n-dimensions. In spite of its importance, not much was known until recently about lower bounds. I will show that if you speed up FFT in a machine of finite precision (say, 32 or 64 bits) using linear operations only (multiplications and additions), then there exist Omega (n) orthogonal directions in input space that either overflow (exceed the machine's numerical upper limit) or underflow (exceed the machine's precision) at some point during the computation. A quantitative tradeoff between the speedup factor and the resulting numerical abuse will be presented. The talk is self-contained. Some open problems will be presented at the end.