Abstract: Current CPU (processor) and GPU (graphics) architectures heavily use data and instruction parallelism at different levels. Finite Difference algorithms on these systems tend to be memory bandwidth limited, like many other numerical schemes. In order to tune several Finite Difference kernels, we will discuss different cache aware algorithms, different vectorization strategies, different memory layouts, the use of larger numbers of registers, and automatic parameter tuning. An optimal choice depends on the size and shape of the Difference Stencils and on the CPU or GPU type. The results include data of recent AMD, Intel, and Nvidia architectures.