Mitigating the Curse of Dimensionality: A Highly Parallel Fast Sweeping Method

UC SANTA BARBARA engineering

Miles Detrixhe, Frederic Gibou, Chohong Min
Department of Mechanical Engineering, University of California Santa Barbara

COMPUTATIONAL APPLIED SCIENCE LABORATORY

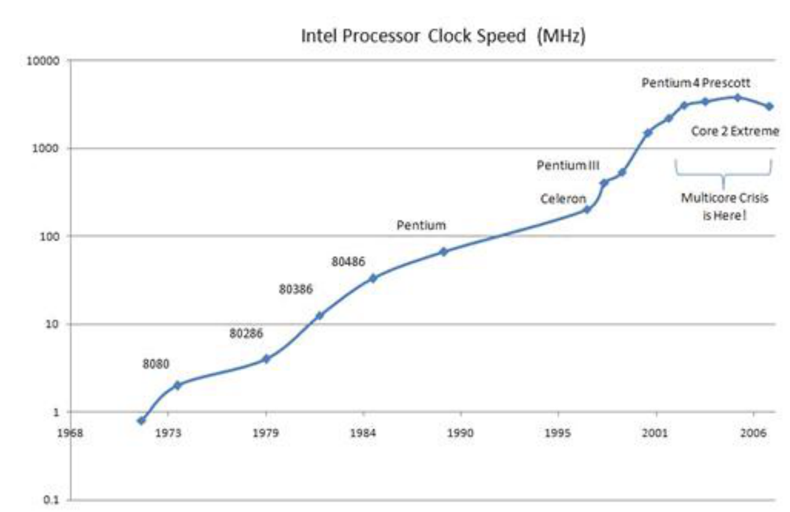
Introduction

Often, the types of problems computational scientists can study are limited by computational resources. Parallel computing offers a tool for approaching previously out-of-reach problems. In order to take full advantage of parallel computer architecture, however, software must follow suit.

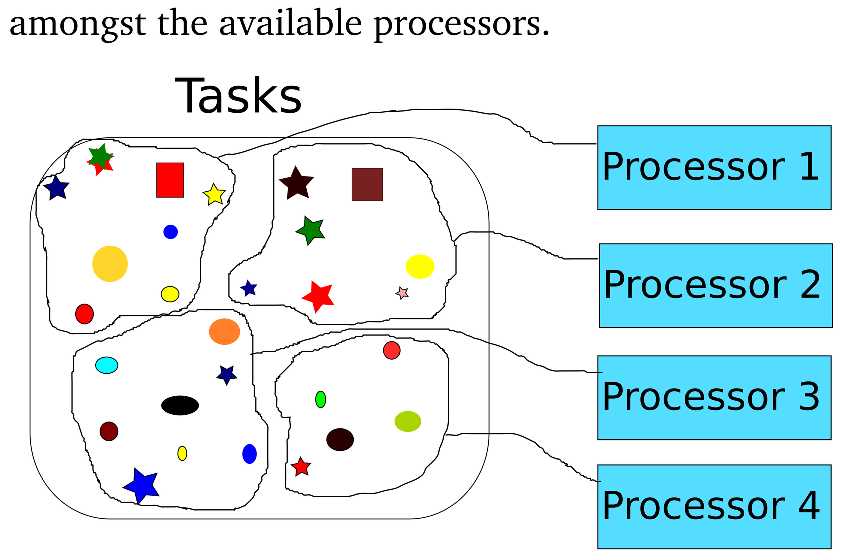
The Eikonal equation has applications in which the state space can be large. Large dimensional problems lead to prohibitively huge computations. We present a parallel method that does not plateau at a large number of threads. This can decrease computational time by orders of magnitude.

The Future of Computing

- In the recent history of computing, clock speed has been increasing at an incredible rate. This steady increase in computation power (Moore's Law) has held true until the last few years.



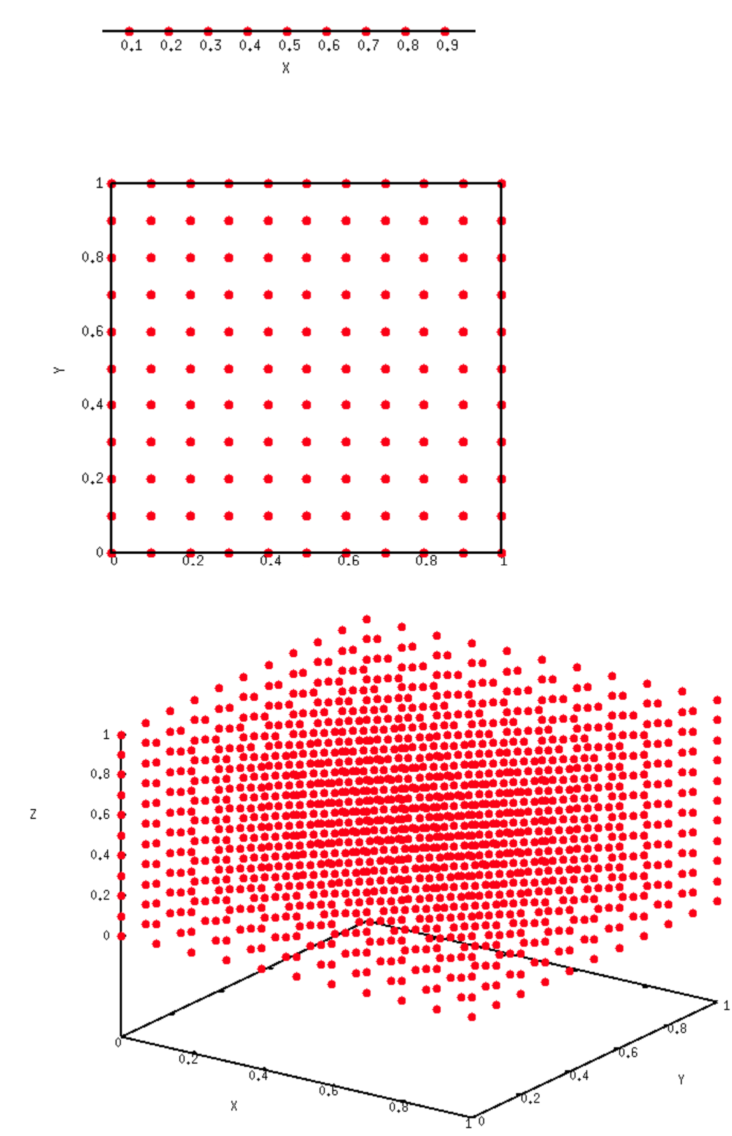
- Since physical limitations are prohibiting clock speed from increasing much further, new technology has arisen. Parallel computing is the next logical step in increasing computational power.
- In order to utilize parallel computation, software must be designed accordingly. The basis of parallel computing is to divide the work into tasks (the shapes below) and distribute them amongst the available processors.



- Some problems are simple to parallelize; the tasks can simply be divided and solved simultaneously by the available processors (as above). Others require more clever thinking. Imagine approaching the problem above if all squares had to be completed before the star tasks could be started. The best algorithm may not be obvious. The method we present falls into the latter category.

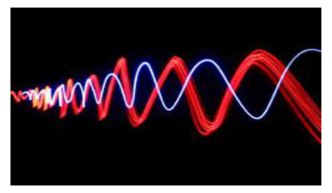
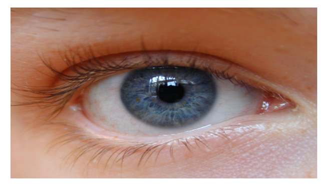
Curse of Dimensionality

- In order to solve a problem numerically, the domain must first be *discretized*. Then a numerical solution is constructed on the discrete grid nodes.
- The 'Curse of Dimensionality' refers to the fact that as the dimension of a problem increases, the number of grid nodes, and thus the number of computations, increases exponentially.
- The diagrams at right illustrate this. Shown is a one, two, and three dimensional domain [0,1] in each direction with a .1 spacing between grid nodes. The number of points required to describe the domain quickly balloons.



- Below is a chart estimating the amount of time to solve a modest problem (~100 operations per grid node, 200 nodes in each direction) in N-dimensional Euclidean space on a standard single core desktop computer.
- The problem becomes prohibitively large very quickly when the dimension of a problem is increased.

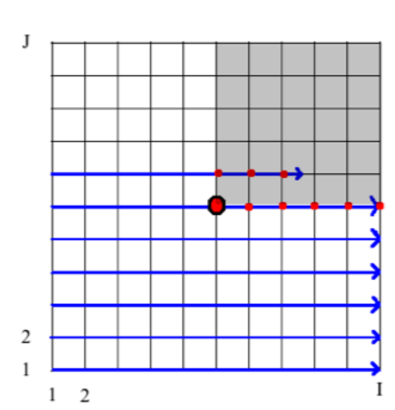
N	# of nodes	time (s)	about as much time as:
1	200	4e-6	Light transiting 1km
2	4e4	9e-4	Camera shutter
3	8e6	.2	A human eye blinking
4	16e8	30	Stoplight cycle
5	3.2e11	7000	The movie Jurassic Park
6	6.4e13	1e6	Hiking the JMT
8	2.6e18	6e10	Sequoia lifespan
11	2e25	4e17	Age of the Universe

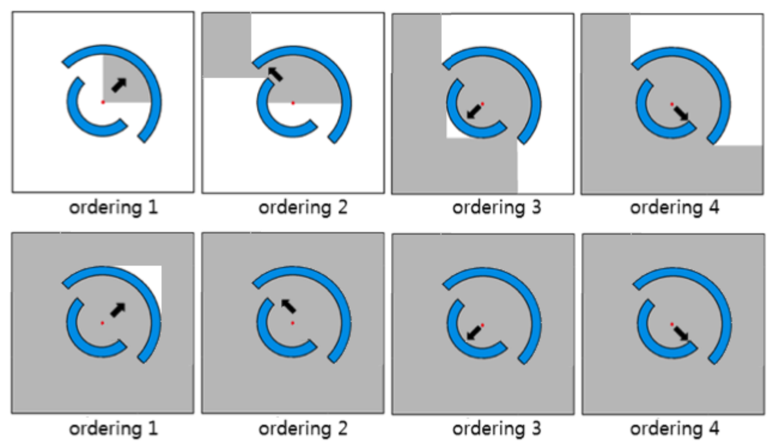

Eikonal Equation

$$\begin{aligned} |\nabla u(\mathbf{x})| &= f(\mathbf{x}) \quad \mathbf{x} \in \Omega \subset R^n \\ u(\mathbf{x}) &= \phi(\mathbf{x}) \quad \mathbf{x} \in \Gamma \subset \Omega \end{aligned}$$

- The Eikonal equation (above) has applications in optimal control, path planning, computer vision, and interface tracking.
- The Fast Marching Method is a computational method for solving the Eikonal equation that takes advantage of the fact that boundary data propagates along characteristics. By iteratively sweeping through the data in a specific order, information is propagated along all characteristic directions.

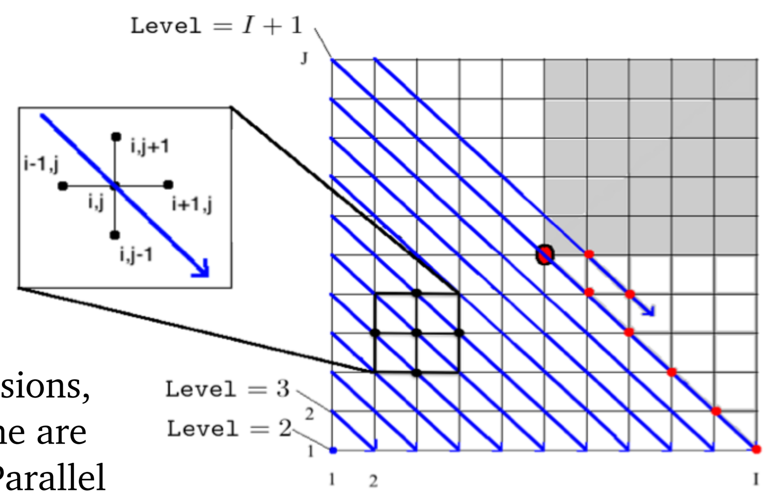


- In two dimensions, there are four characteristic directions, and therefore, four specific sweeping orderings. The example below shows how each ordering propagates information in a specific direction, and with iteration, will produce a solution across the whole domain.

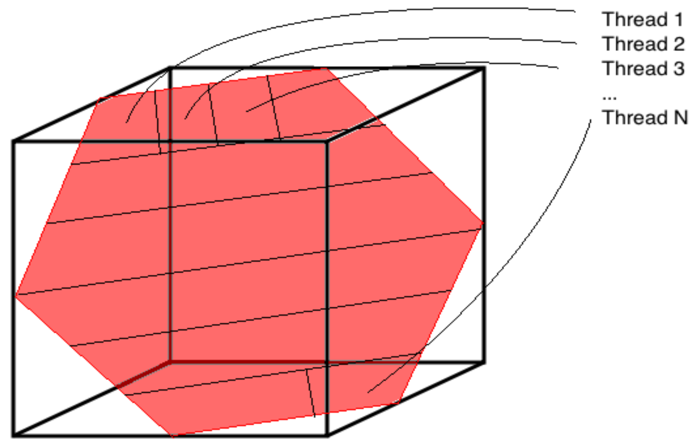


Parallel Method

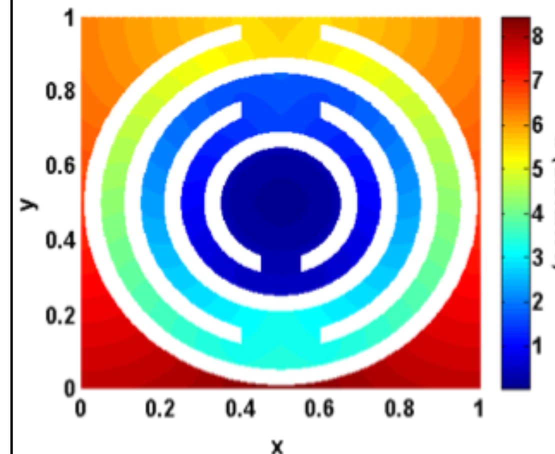
- Previously, the fast sweeping method was parallelized by computing each ordering in parallel. This limits the speedup drastically (only four threads can be used in 2D). Our method utilizes an alternate ordering that allows parallelization within each sweep.
- The computation of the solution uses the standard 5-point stencil. When we sweep along a diagonal, all of the points on the stencil are independent of the other nodes along the diagonal. We can compute all of these nodes in parallel. This allows for a large number of threads to be used simultaneously.



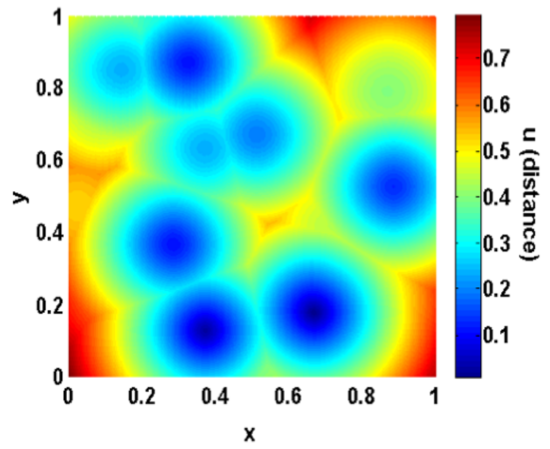
- In three dimensions, nodes in a plane are independent. Parallel load sharing is shown in the illustration (below right).



Results

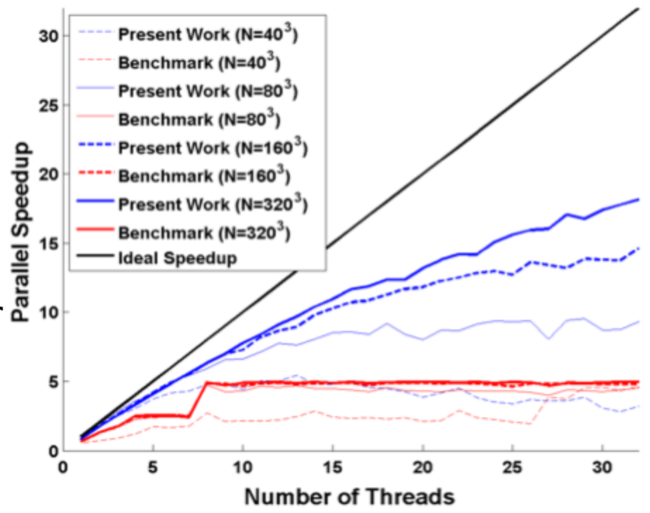


Example problem 1: distance from center with spherical obstacles..



Example problem 2: distance from a random set of points in 3D domain.

- Our results and a parallel benchmark (method by Zhao[2]) were tested on a large shared memory parallel machine.
- The results show superior speedup. As the problem size becomes large, the speedup approaches the ideal case. The benchmark method plateaus when the thread limit is reached.
- Even on a modest (32 core) shared memory machine, we can reduce computational time by over one order of magnitude.



Conclusion: We present an algorithm for solving the Eikonal equation that has ideal theoretical parallel speedup O(N/p). We validate the method and provide results that show our method approaches ideal parallel speedup in the limit of large N.

[1] Detrixhe, M., Min, C., & Gibou, F. (2011). A Parallel Fast Sweeping Method for the Eikonal Equation
[2] Zhao, H. (2007). Parallel Algorithms For the Fast Sweeping Method. Journal of Computational Mathematics, 25(4), 421-429.