CSE551 Assignment #3
Short Review of Domain Decomposition Method

Mustafa Kutay Yabaş

December 3, 2016

Domain Decomposition is a divide and conquer method used to solve numerical problems. Most importantly it allows for parallel computing in large problems by dividing the problem into subdomains.[2] Then each subproblem is solved in its subdomain where the solutions are merged together at the end. It is flexible and efficient in its nature and applicable to a large variety of problems. It allows numerical solutions for irregular shapes or usage of different computational approaches in different subdomains.

While there are different methods of Domain Decomposition (DD), overlapping DD is easy to implement and has an intuitive setup. As an example application we can explain the solution of Laplacian equation with finite difference method. In the finite difference method we create a hypothetical stencil or a grid, then calculate the value of the each cell of the grid depending on the differences of the neighbouring cells. The numbers of the required neighbours increase by increasing orders of derivatives and accuracy. So if we slice the whole grid into sub-grids we should also consider the required neighbours.

Laplacian is a 2^{nd} order derivative and if we want to solve the equation with finite difference method with 1^{st} order accuracy then we need at least one neighbouring cell. So if we create a 10×10 grid and want to solve the problem in 4, 5×5 sub-grids, in practice the grids should be 6×6. The solution of the 5×5 matrix will require the information of 6^{th} row and column or other corresponding columns for other sub-grids. So the overlapping parts are these excess columns and rows in each sub-grid. They are not calculated and can be omitted when the sub-grids are merged at the end. So a large

$N \times N$ matrix can be divided into smaller sub-grids, each sub-grid can be computed in parallel to be merged later. The only disadvantage is the excess rows and columns present in every sub-grid causing increased memory usage but it is an efficient trade of in return of decreased computing time.

A very efficient case will be a problem with emerging patterns or cyclic functions. Instead of recalculation of repeated patterns one can slice the grid wisely then just solve the one of a kind grids and duplicate the solutions to fill the pattern.

My area of research includes extensive usage of Monte Carlo methods for simulations, sampling and inference. A similar approach like in domain decomposition model may be beneficial for parallelisation of the simulations processes to obtain more samples in the same unit of time. However the overlaps are not predefined in a Monte Carlo solution as the process is based on randomness. In the course of computation, events that pass the boundaries of the subdomains may arise and they should be sent between threads asynchronously.[1] This results in threads waiting for each other and messages to be transmitted which decreases the true benefits of parallel computing. So the trade off may be bigger in the Monte Carlo case compared to deterministic problems.

Another problem is to attain independent randomness between threads. Order of execution or other dependencies between threads will probably introduce autocorrelation and introduce a randomisation bias in the problem which degrades the whole benefits of using Monte Carlo methods.

There surely are methods to eliminate biases in the above-mentioned, however in practice it is a more common approach to run separate instances of the entire Monte Carlo problem in independent threads then combine the results. After all the solution is a random sample of the population and different random samples merged together continue to be random. On the other hand if the single Monte Carlo thread converges very slowly this approach is not desirable. So domain decomposition methods are efficient to overcome slow, non-uniform convergence of variables and high memory requirements in high dimensional simulations.[3]

In conclusion domain decomposition methods are beneficial in some cases of Monte Carlo methods and can be used to increase the speed of convergence of the variables to their expected values.

References

[1] Thomas A Brunner and Patrick S Brantley. An efficient, robust, domain-decomposition algorithm for particle monte carlo. *Journal of Computational Physics*, 228(10):3882–3890, 2009.

[2] X Cai. Overlapping domain decomposition methods. In *Advanced Topics in Computational Partial Differential Equations*, pages 57–95. Springer, 2003.

[3] John C Wagner, Scott W Mosher, Thomas M Evans, Douglas E Peplow, and John A Turner. Hybrid and parallel domain-decomposition methods development to enable monte carlo for reactor analyses. *Progress in nuclear science and technology*, 2:815–820, 2011.