Anqi Guo, Yunze Lian, Shineun Yoon

Richard Brower

Parallel Computing

May 1st, 2021

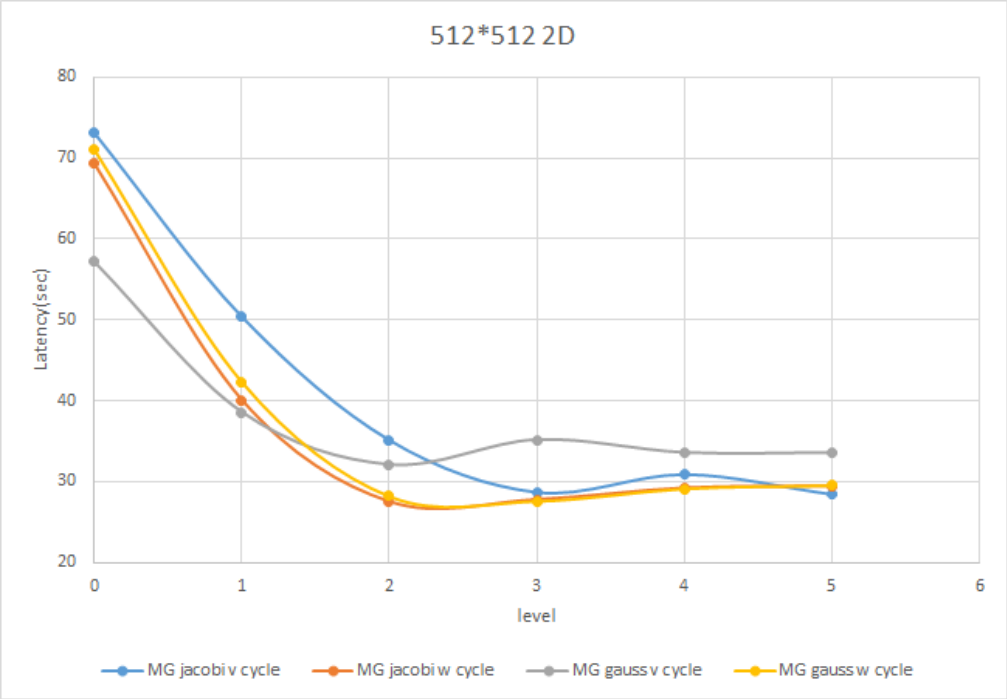
Solver Performance and Acceleration

As an acceleration of the iterative method, multigrid method (MG) and conjugate gradient (CG) has been discussed.

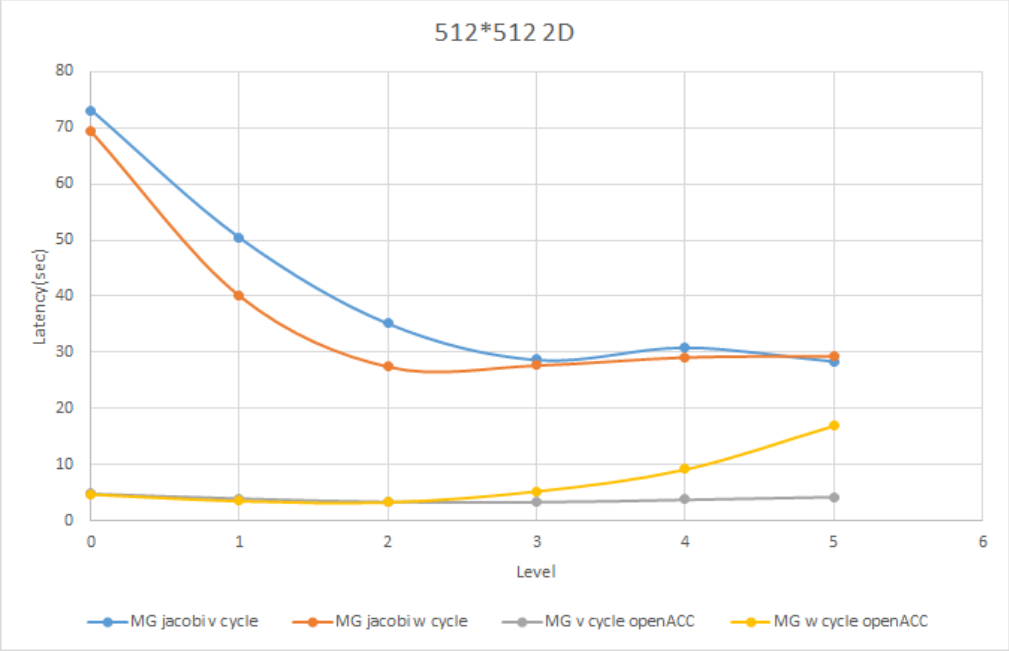
Multigrid Method (MG)

Multigrid method (MG method) is an algorithm for solving differential equations using a hierarchy of discretizations. The problem we tried to solve in multigrid is a 2-dimensional time-dependent heat conduction equation. A non-zero initial heat field that varies with time is implemented in the middle of a 2D lattice grid with boundaries fixed at zero. The heat source changes based on sin function and the program simulates 100 units of time. In each time unit, the program simulates the heat flow until the whole 2D lattice grid settles.

V-cycle and w-cycle multigrid method with two iterative solvers (Jacobi and Gauss-Seidel) performance is evaluated and accelerated with openACC using GPU.



The first figure shows the performance difference between v cycle and w cycle multigrid using jacobi and gauss seidel iterative solvers with different levels. The result shows the more level the better performance we will get. As level goes more than 3, each method will not get better latency. W-cycle with gauss seidel solvers gets the best performance when we have more than 2 levels.



The second figure is the performance comparison between MG serial code and openACC. We can see that openACC significantly accelerate the performance of MG. OpenACC gives us 6 times better performance

when level is larger than 2. However, when level increases, the latency of MG w cycle with openACC increases which seems a little weird which might be bottlenecked by memory.

Conjugate Gradient (CG)

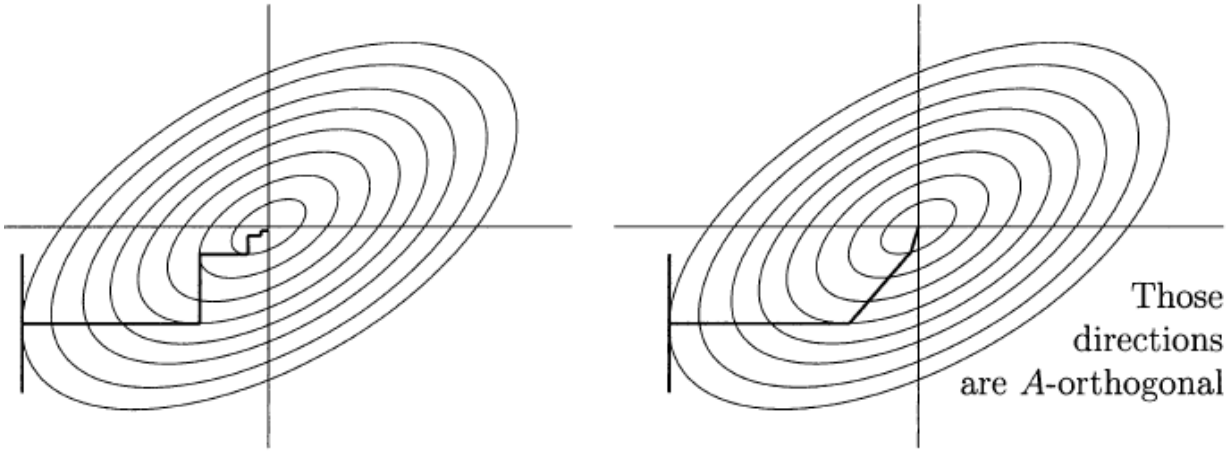
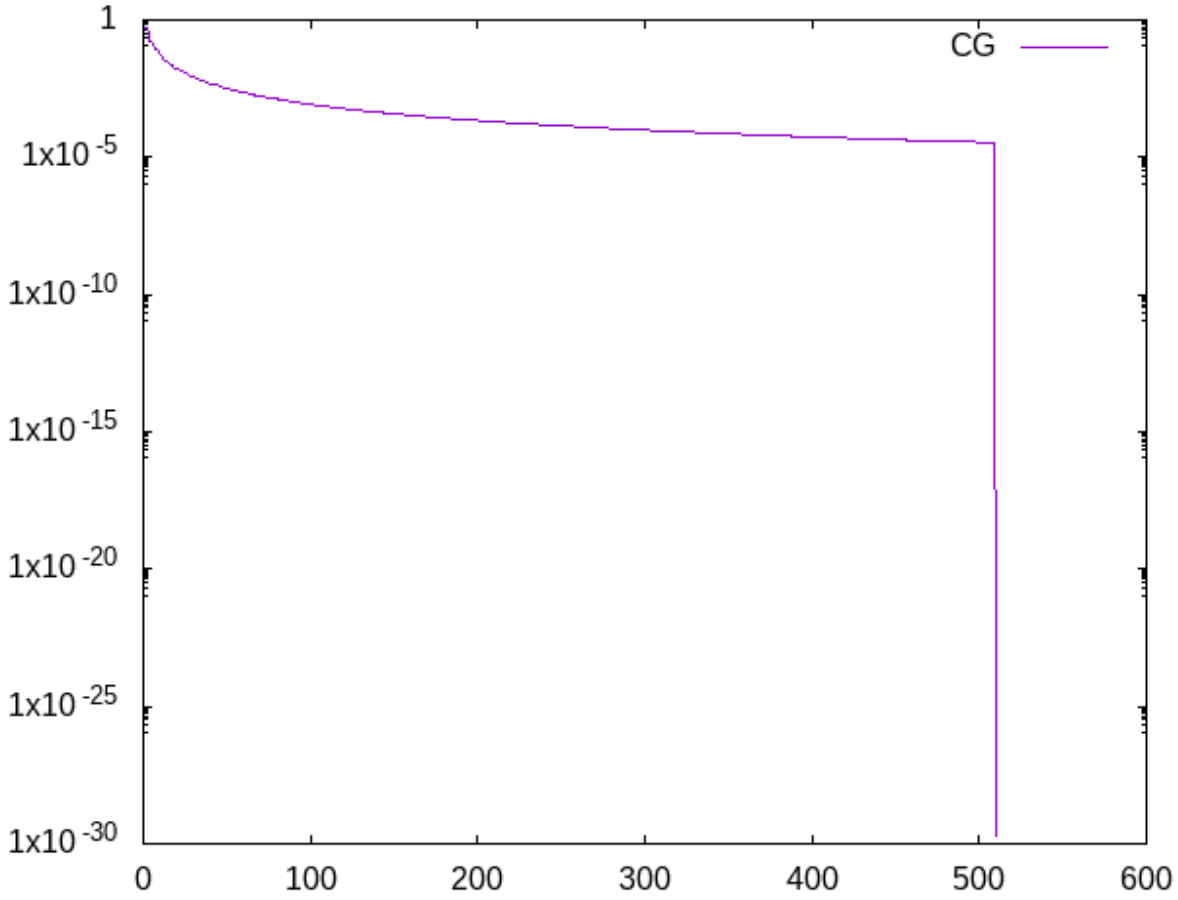


Figure 7.13: Steepest descent (many small steps) vs. conjugate gradients.

Solving the equation Ax=b is the same as getting the minimum point of the function recursively. Conjugate gradient method is just like an improved edition of gradient descent. When we use gradient descent, the X goes like the left figure. Go up and right, up and right for a lot of time. It could be faster if we just go right once and go up once, so we have conjugate gradient method.



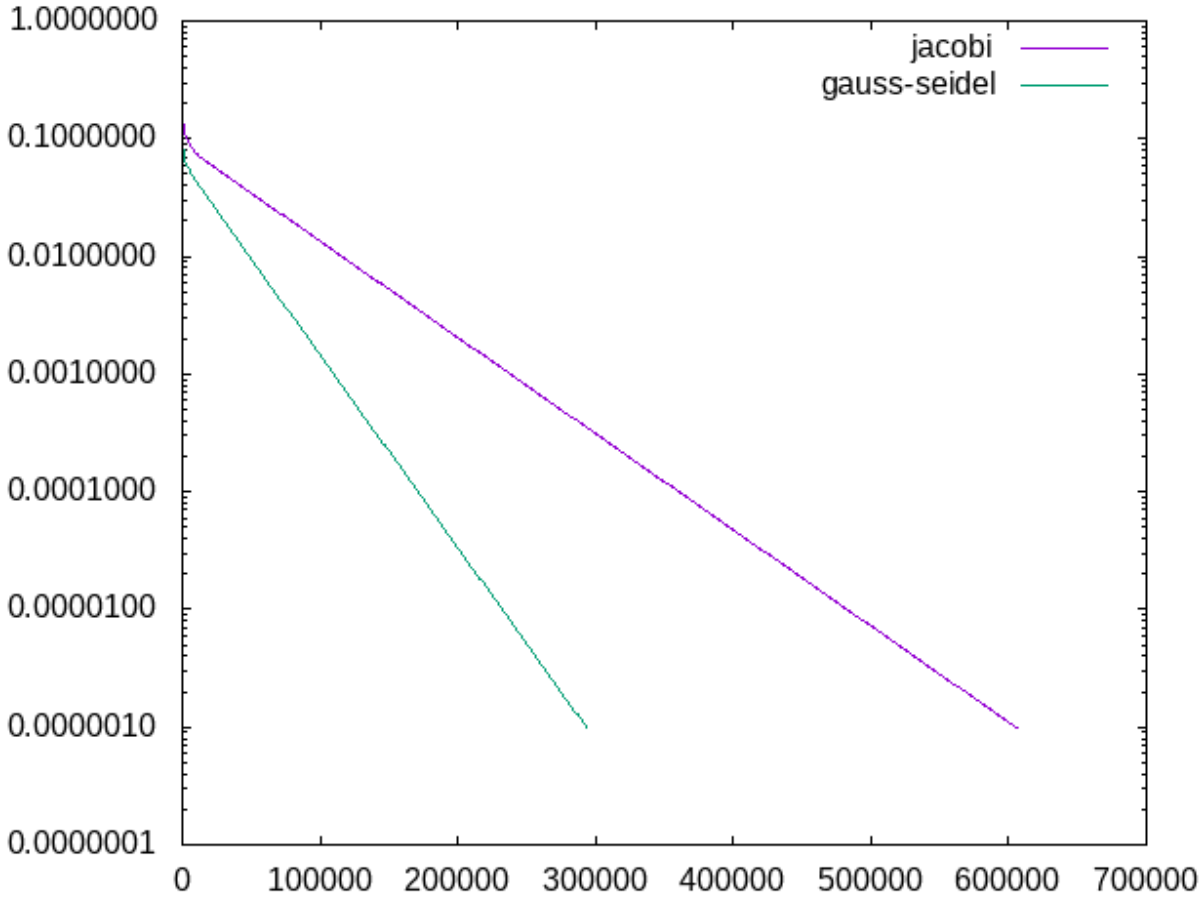
In CG method we try to let the direction d be orthogonal to error to make the error as small as possible. But in fact we can't find such a d. (If we can find it, we could get the solution X directly). So instead we try to get a d which is conjugate orthogonal to the error. So we let the direction be conjugate orthogonal with all the errors. For a N dimension problem, we need to work at most N times.

Other Iteration Methods

Rewrite $Ax = b$ $\quad Sx = Tx + b.$

Pure iteration $Sx_{k+1} = Tx_k + b.$

$Se_{k+1} = Te_k$ which means $e_{k+1} = S^{-1}Te_k.$



The basic logic of jacobi and gauss-seidel are the same. It uses a matrix ST to do linear transformation to error. So the speed of the jacobi and gauss-seidel depend on the eigenvalue of the matrix ST. The gauss-seidel is faster because it has smaller eigenvalue.

References

On Solvers: Multigrid Methods. COMSOL Multiphysics. (n.d.). https://www.comsol.com/blogs/on-solvers-multigrid-methods/.