Multigrid Poisson Solver

2020/06/30

Outlines

1. Model Problems
   1. 2D Poisson equation
2. Motivation
   1. Limitation of exact solvers
3. Algorithms of Multigrid methods
   1. V cycle
   2. W cycle
   3. F cycle
   4. Full multigrid(FMG)
   5. Method of GPU parallelism
4. Results
5. Conclusion

Model Problems

Poisson equation

We choose the following two 2D equations for the test of multigrid method.

Both of them have homogeneous BC and are confined within [0, 1].

Typically, multigrid method is used for solving elliptic PDEs.

Equation A

$$L\,\phi \;=\; -\left(k_x^2 + k_y^2\right)\sin k_x x \;*\; \sin k_y y$$
$$\phi \;=\; \sin k_x x \;*\; \sin k_y y$$

2D Fourier mode:

Test the performence of different Fourier modes independently

Equation B

$$L\,\phi \;=\; 2x(y-1)(y-2x+xy+2)\exp(x-y)$$
$$\phi \;=\; x(1-x)y(1-y)\exp(x-y)$$

Test the performence of multigrid method of an arbitrary problem

Motivation

Exact solvers

Here we introduce 2 kinds of exact solver:

1. Inverse Matrix
2. Successive Over-Relaxation (SOR)

There are other exact solvers, such as FFT, tridiagonal matrix algorithm, and etc., for solving discrete Poisson equation.

Limitation of Inverse matrix

1) Inverse matrix

For sin test with $k = \pi/L$,

N = 5 takes 2.269 sec to solve,

N = 9 takes >30 min to solve,

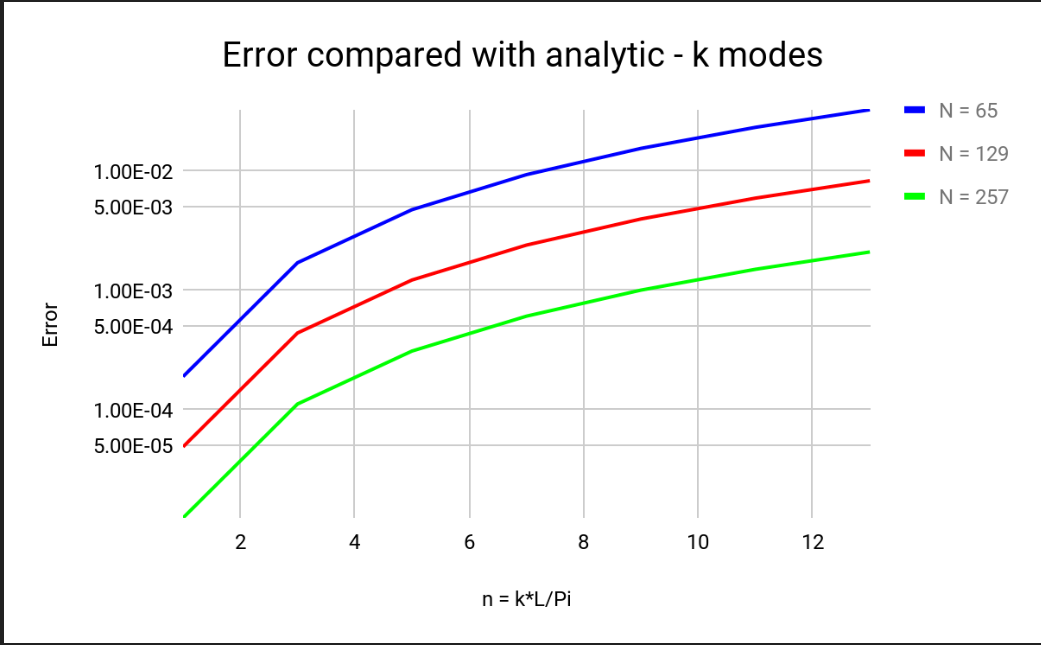
It is not realistic to solve the problem by inverse matrix generally

Limitation of SOR

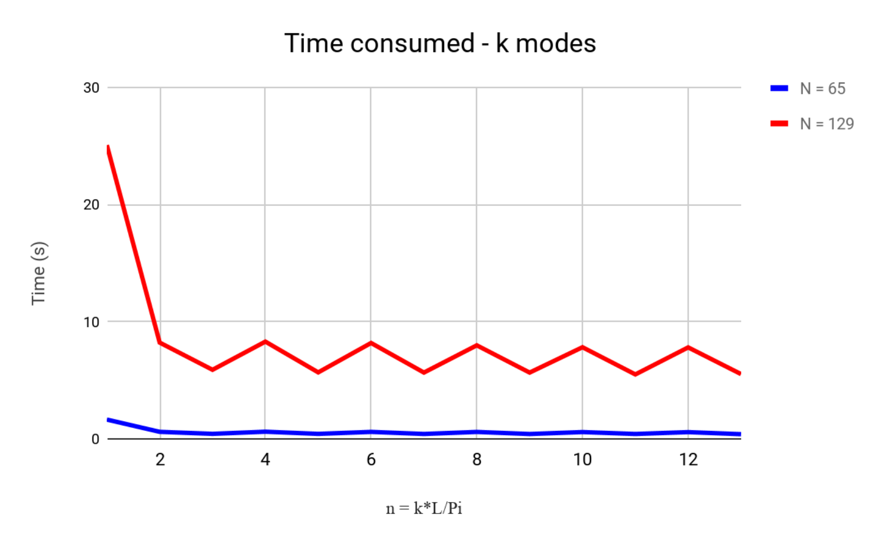
2) SOR

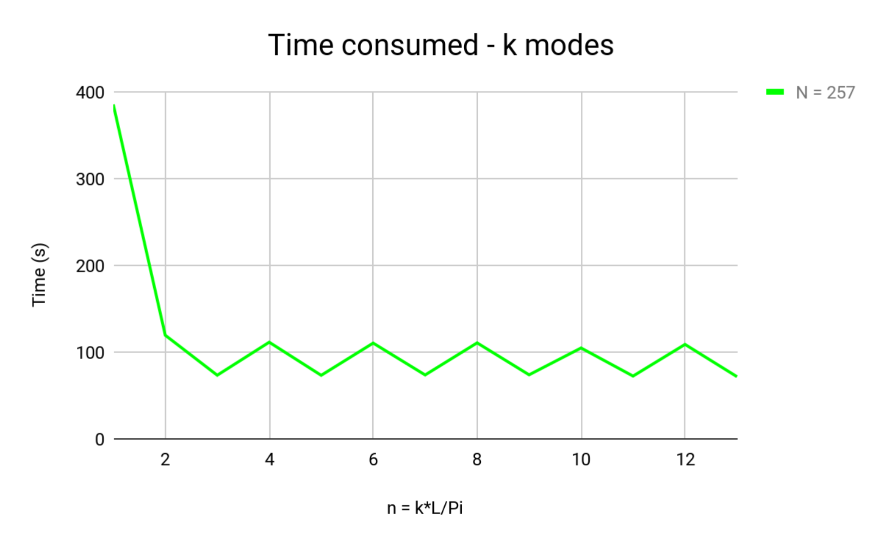
Convergance rate is defined as 1e-15. ω = 1.

For sin test with different k modes in defferent resolutions,

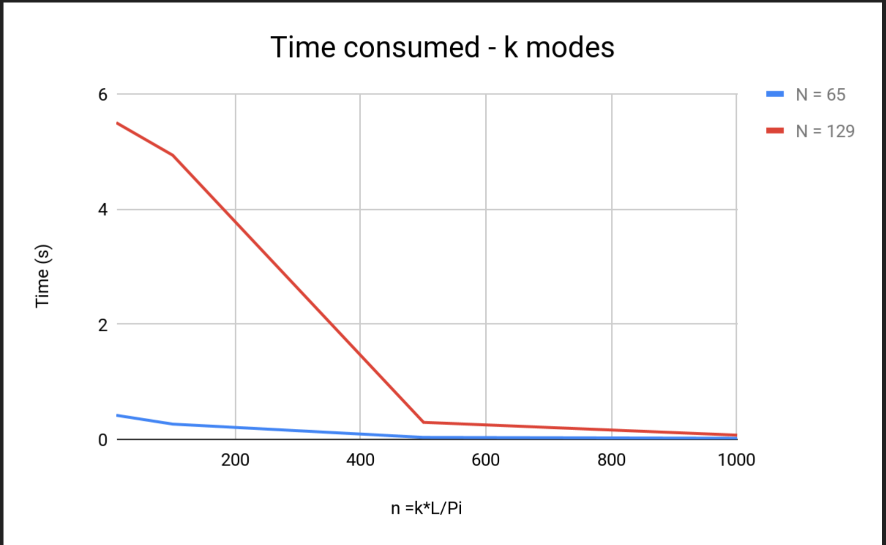


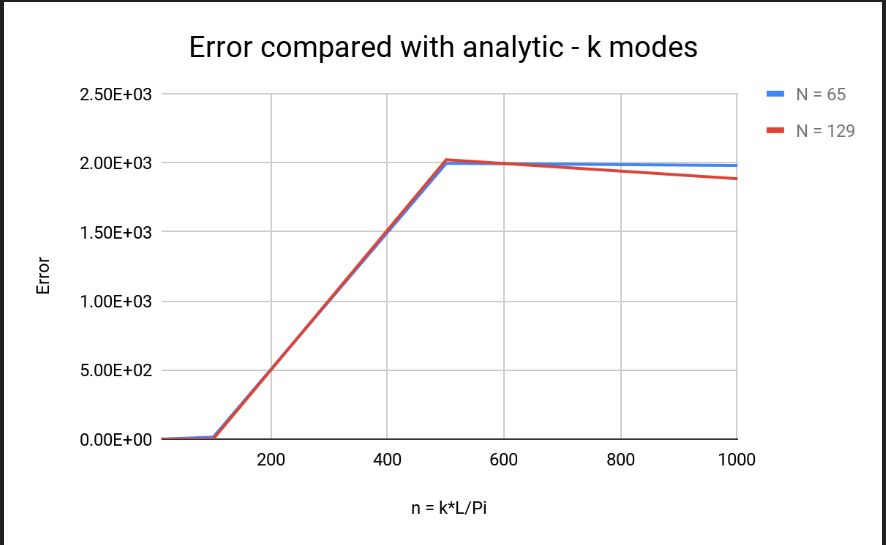
Limitation of SOR





Limitation of SOR





Limitation of SOR

1. Higher resolution solves the problem with lower error.
2. Error increases as k increases.
3. Higher resolution takes more time to solve.
4. Higher k mode spends less time to solve.

High resolution gives better result but more time consuming.

High k modes have greater error but less time consuming.

Limitation of SOR

1. For a general problem which has low k modes, it could be very expensive for solving.
2. N>257, it takes more than 1 min in general. Low resolution can solve the problem more quickly.

→ Using low resolution to solve the low k modes could be a better way of solving general problem.

→ Multigrid method

Algorithms of Multigrid Methods

Residual & Correction

I. Target equation $L\,\phi = \rho$

II. Define residual $\xi \equiv \rho - L\,\bar{\phi}$

III. Define correction $\xi = L\left(\phi - \bar{\phi}\right),\ \phi \equiv \bar{\phi} + \phi^{corr}$

IV. Rewrite equation $L\,\phi^{corr} = \xi$

Instead of solving target equation, we solve the equivalent residual equation, which is easier to be solved if we have a good enough guess.

(If the exact solver is inverse matrix, it does not matter which equation you are solving.)

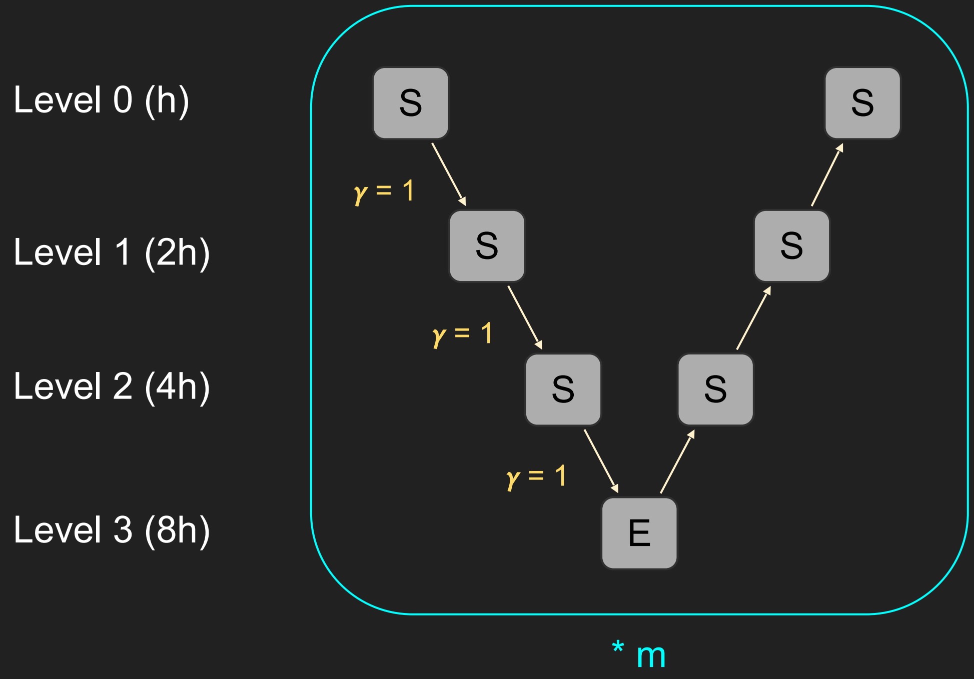
Residual & Correction

For two-grid scheme with exact solver SOR, it is better to restrict residual to coarser level.

Because the solution of the residual equation at coarser level is expected to be fluctuated around zero, we can simply choose the initial guess as zero, which is a good enough guess as we expected.

By introducing even coarser level, we have the multigrid method.

V cycle ($\gamma = 1$)



V cycle ($\boldsymbol{\gamma} = 1$)

Level 0

1. Initialization $\bar{\phi}_h^{old} = 0$
2. Smoothing $\bar{\phi}_h^{old,s} = S\,\bar{\phi}_h^{old}$
3. Residual $\xi_h = \rho_h - L_h\,\bar{\phi}_h^{old,s}$

Restriction $\rho_{2h} = R\,\xi_h$

Level 1

1. Initialization $\bar{\phi}_{2h}^{old} = 0$
2. Smoothing $\bar{\phi}_{2h}^{old,s} = S\,\bar{\phi}_{2h}^{old}$
3. Residual $\xi_{2h} = \rho_{2h} - L_{2h}\,\bar{\phi}_{2h}^{old,s}$

Restriction $\rho_{4h} = R\,\xi_{2h}$

V cycle ($\boldsymbol{\gamma} = 1$)

Restriction $\rho_{2^n h} = R\,\xi_{2^{n-1}h}$

Prolongation $\bar{\phi}^{corr}_{2^{n-1}h} = P\,\phi_{2^n h}$

Level n

1. Exact solver $\rho_{2^n h} = L_{2^n h}\,\phi_{2^n h}$

V cycle ($\boldsymbol{\gamma} = 1$)

Level 0

1. Updating $\bar{\phi}_h^{new} = \bar{\phi}_h^{old,s} + \bar{\phi}_h^{corr}$
2. Smoothing $\bar{\phi}_h^{new,s} = S\,\bar{\phi}_h^{new}$

Prolongation $\bar{\phi}_h^{corr} = P\,\bar{\phi}_{2h}^{new,s}$

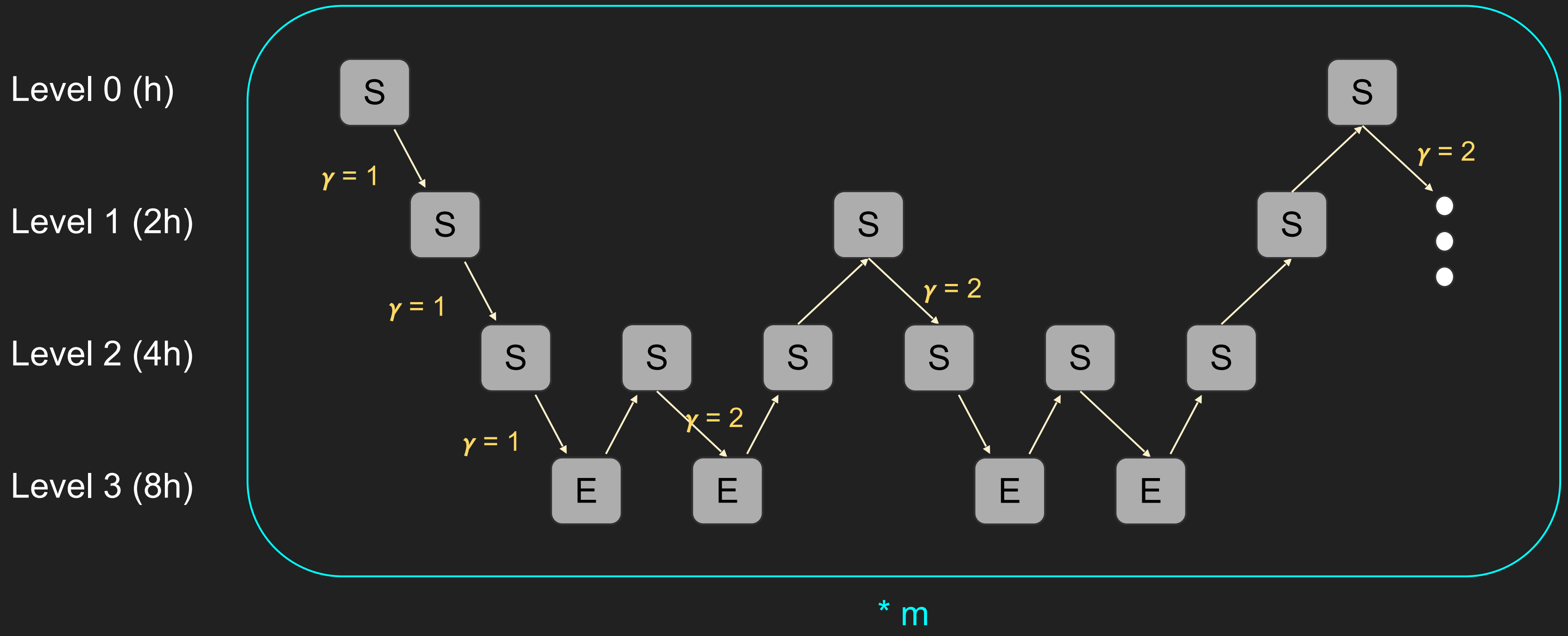
Level 1

1. Updating $\bar{\phi}_{2h}^{new} = \bar{\phi}_{2h}^{old,s} + \bar{\phi}_{2h}^{corr}$
2. Smoothing $\bar{\phi}_{2h}^{new,s} = S\,\bar{\phi}_{2h}^{new}$

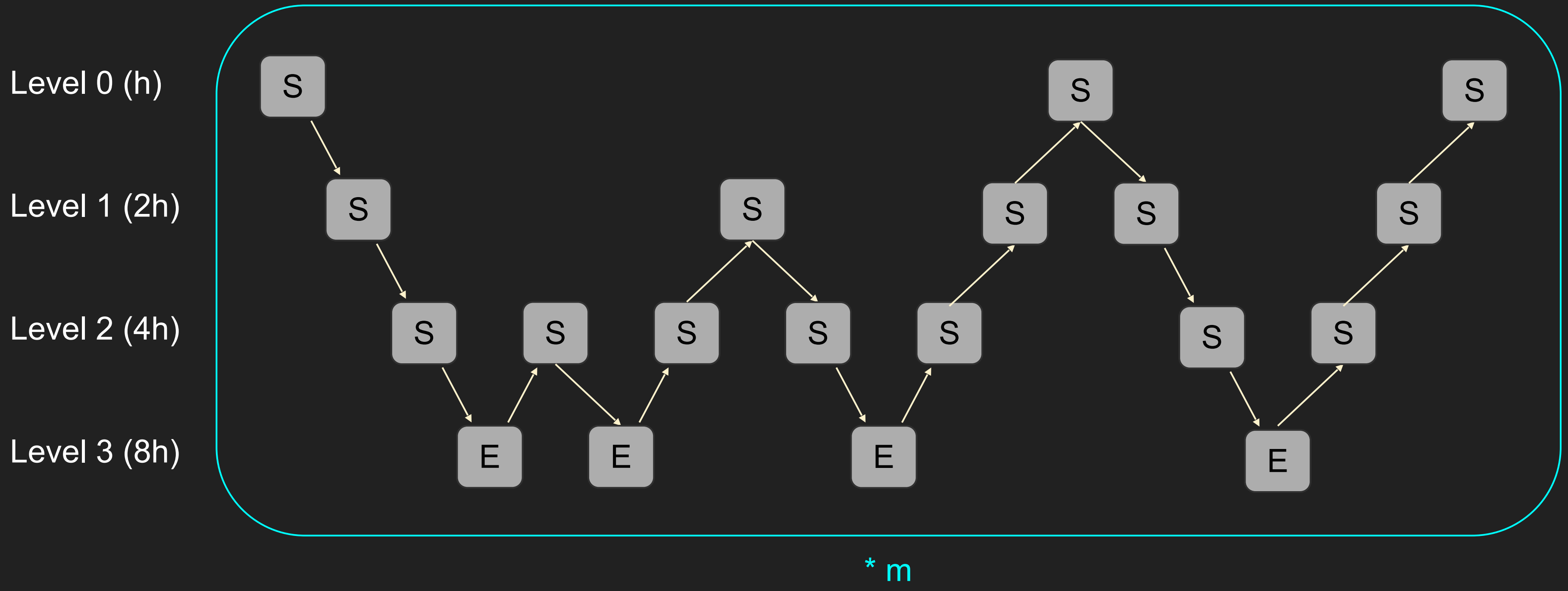
Prolongation $\bar{\phi}_{2h}^{corr} = P\,\bar{\phi}_{4h}^{new,s}$

⋮

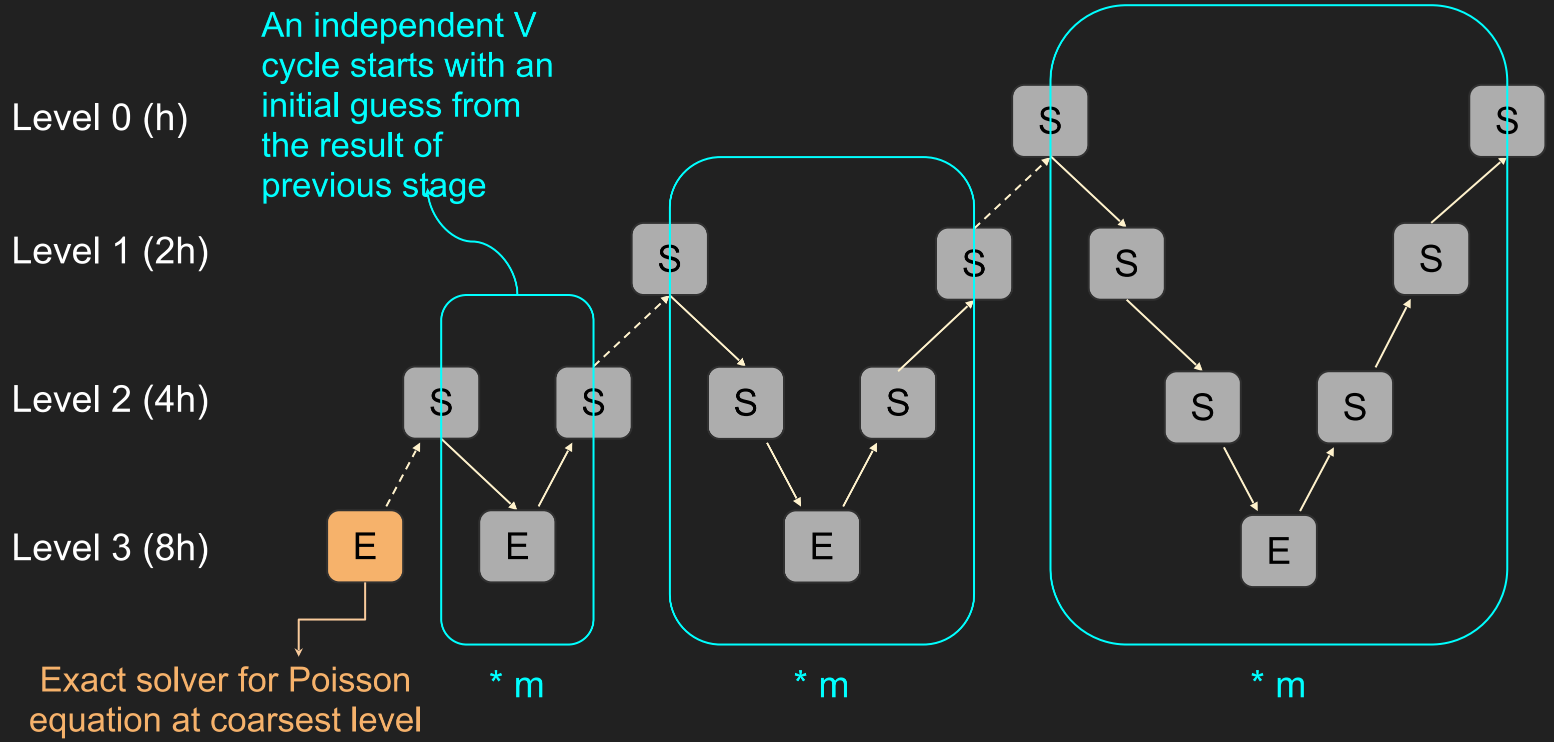
W cycle ($\gamma = 2$)



F cycle


Level 0 (h)
Level 1 (2h)
Level 2 (4h)
Level 3 (8h)
S
S
S
S
S
S
S
S
S
S
S
S
S
S
S
E
E
E
E
* m

FMG



Level 0 (h)
Level 1 (2h)
Level 2 (4h)
Level 3 (8h)
An independent V cycle starts with an initial guess from the result of previous stage
S
S
S
S
S
S
S
S
S
S
S
S
S
E
E
E
E
Exact solver for Poisson equation at coarsest level
* m
* m
* m

FMG

=

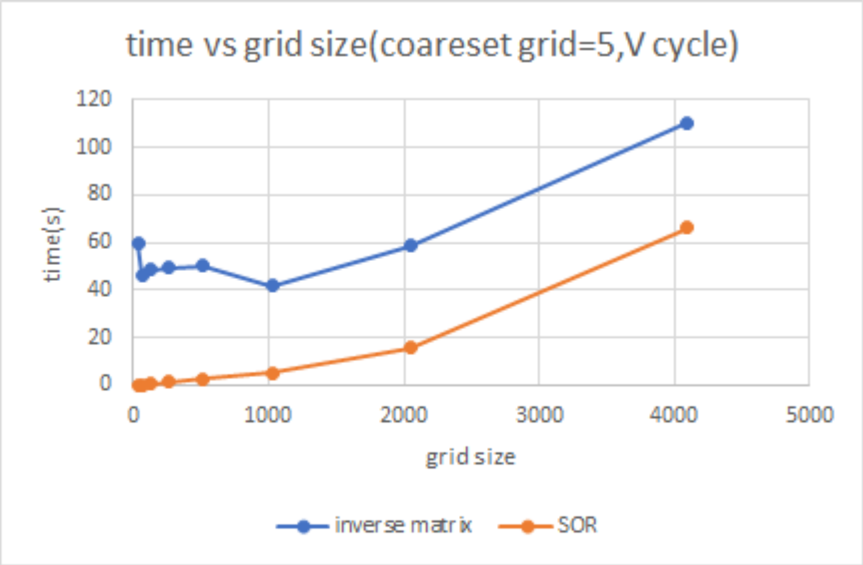
Prolongation $\bar{\phi}^{old}_{2^{n-1}h} = P\,\phi_{2^n h}$

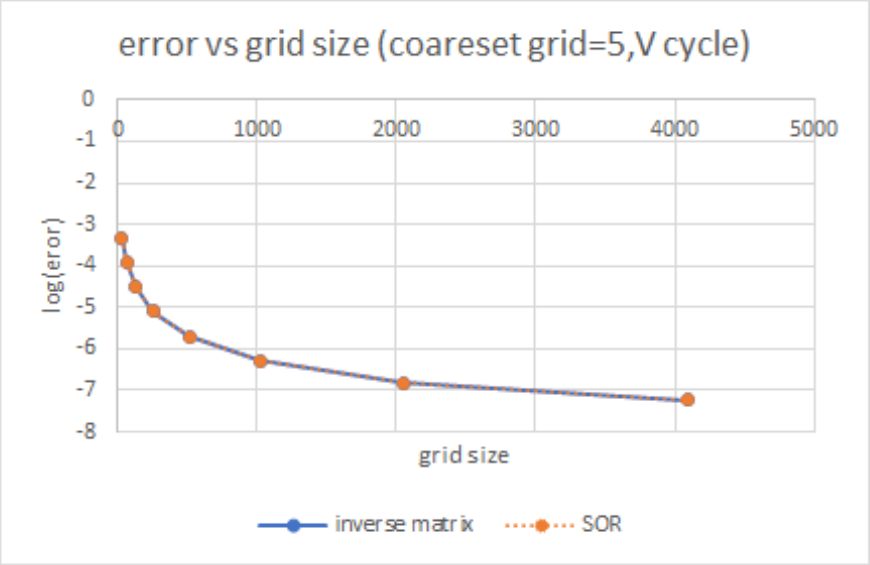
Prolongation $\bar{\phi}^{old}_{2^{i-1}h} = P\,\bar{\phi}^{new,s}_{2^i h}$ for i ≠ n

Primary difference between Multigrid method & FMG:
“FMG starts from exact solution at coarsest level rather than an arbitrary initial guess”

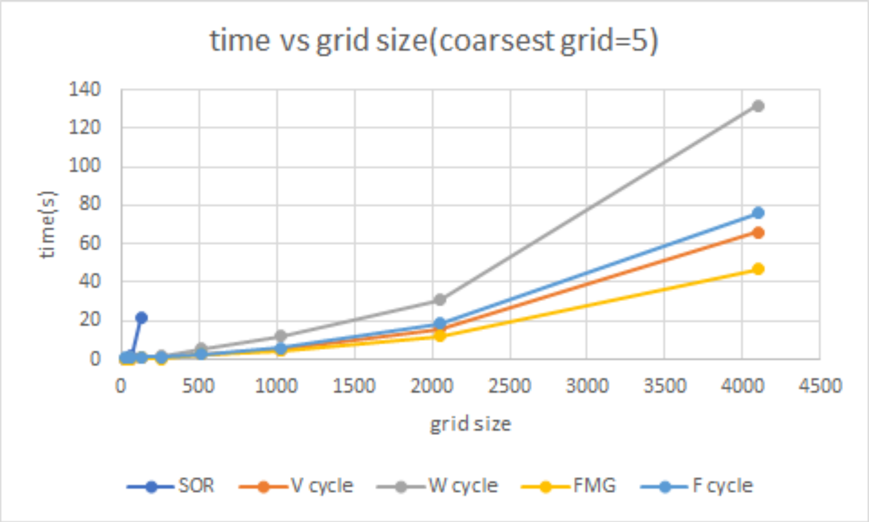
Results

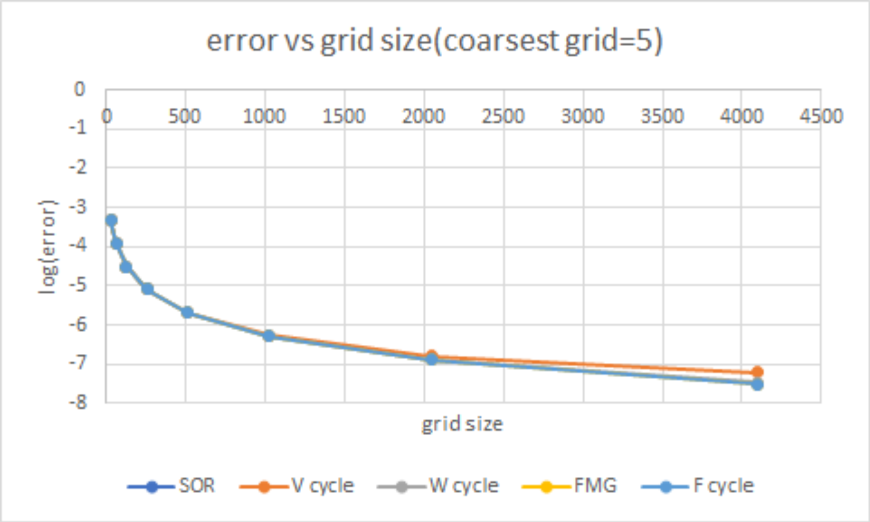
Why using inverse matrix as exact solver is stupid





Compare different types of cycle (with certain conv_rate)

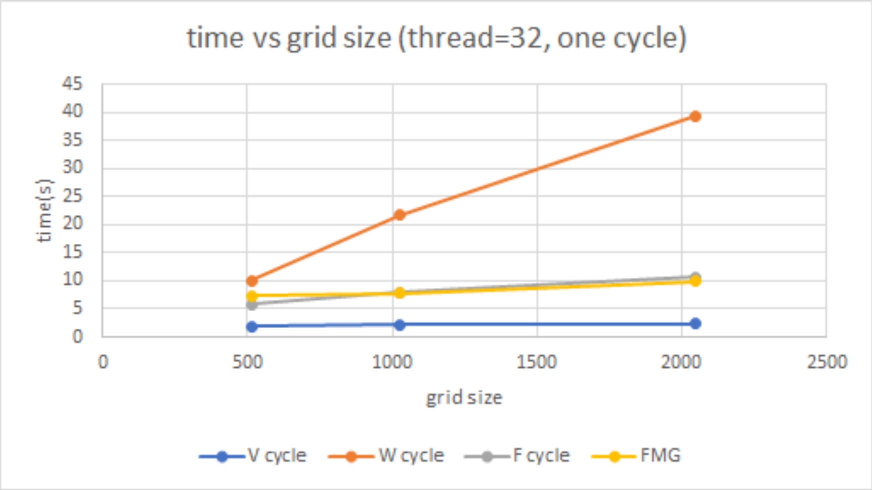


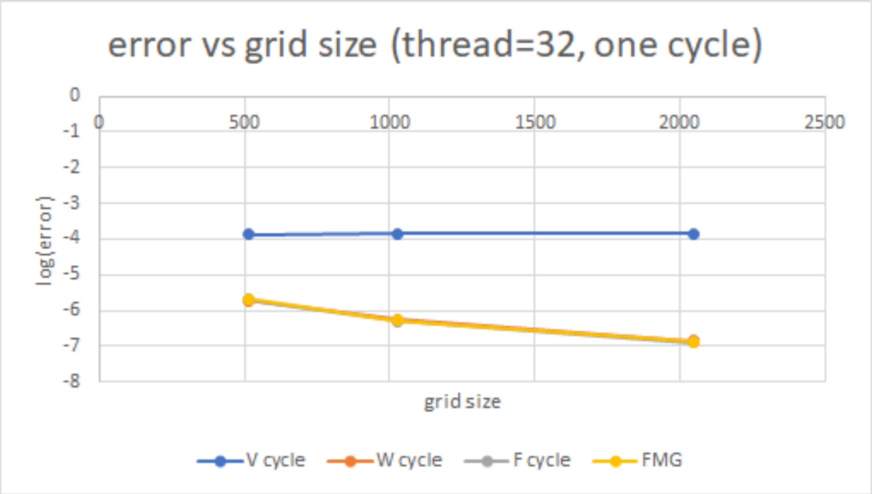


Since we fix the conv_rate, the errors of different methods are roughly the same at certain grid size.

W cycle is the slowest one among these method.

Compare different types of cycle (with one cycle)

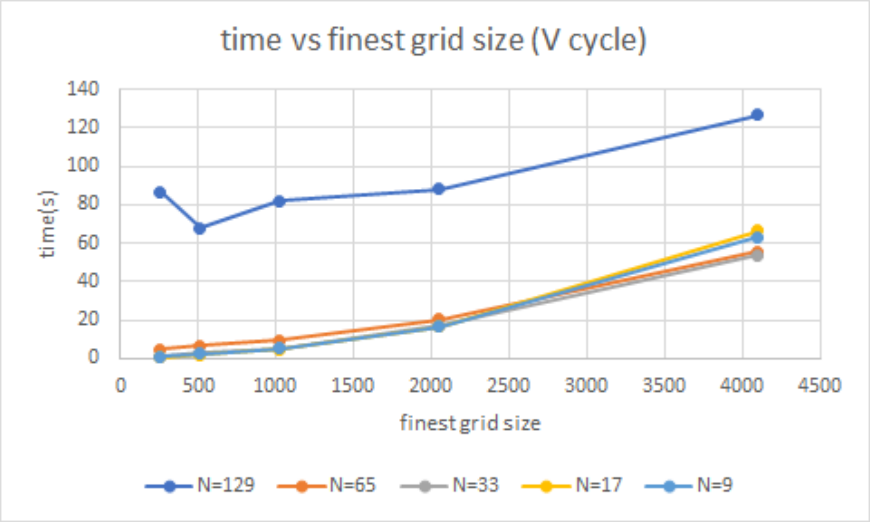


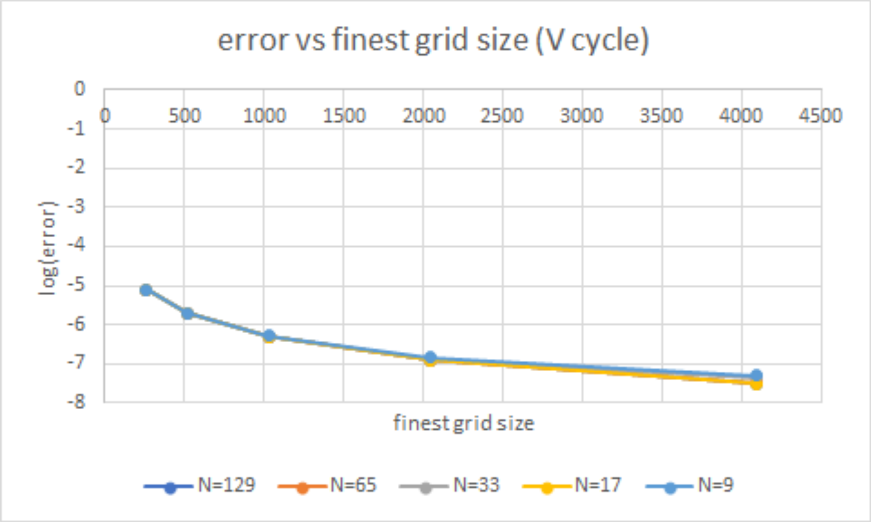


For a single cycle, W cycle is also the most time consuming one.

V cycle has the greatest error.

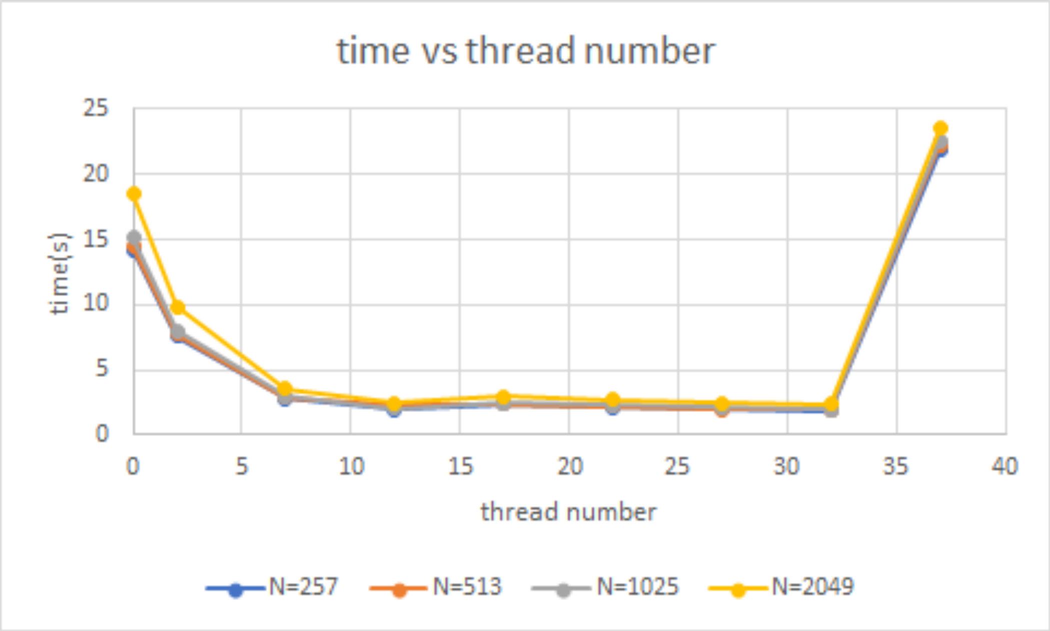
Down sample to different coarsest level





For the same finest level, the error does not depend on the coarsest level it reached heavily.

OMP parallel efficiency



Efficiency depends on how many physical threads the computer has.

Method of GPU parallelism

1. Parallel all the for-loop you see.
2. Reduce time of passing data.
3. Notice the maximum number of thread in hardware.

Conclusion

Conclusion

Error wrt analytic solution is only affected by test problem and total number of cells.

We seek for a way with the least consumed time for high efficiency. There are two ways:

1. choose the right cycle
2. go to the coarestest grid possible
3. minimize smoothing times at fine grids, since it doesn’t make much difference to error wrt analytic solution.

Thanks for your attention