Additively Damped AFAC Variants:
Asynchronous Stencil Construction

Charles Murray, Tobias Weinzierl
c.d.murray@durham.ac.uk, tobias.weinzierl@durham.ac.uk

Durham University
Department of Computer Science

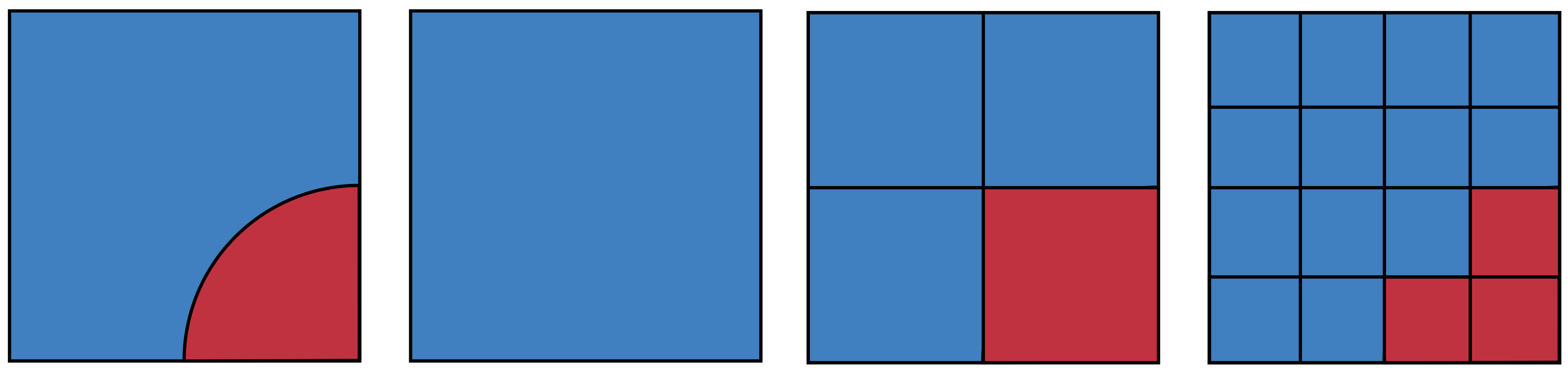
adaFAC-PI

- Simpler alternative damping parameter construction

$$-PI\omega_\ell M_\ell^{-1}(b_\ell - A_\ell u_\ell)$$

- Approximate coarse grid equation impact
- Avoid expensive coarse grid construction

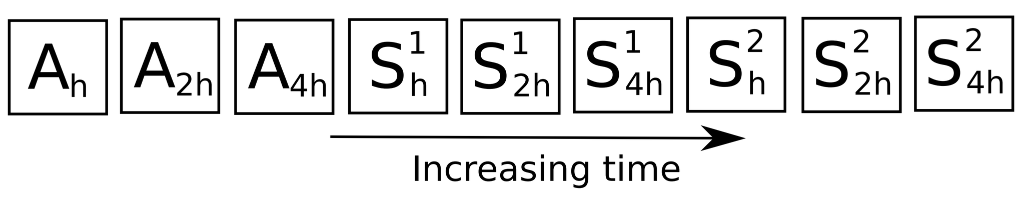
Iterative Quadrature



Accurate material data (left); Three quadrature iterates to capture material data (right)

Stencil Integration

- Construction of accurate matrix equation expensive
- Worsened by Multigrid coarse grid equations

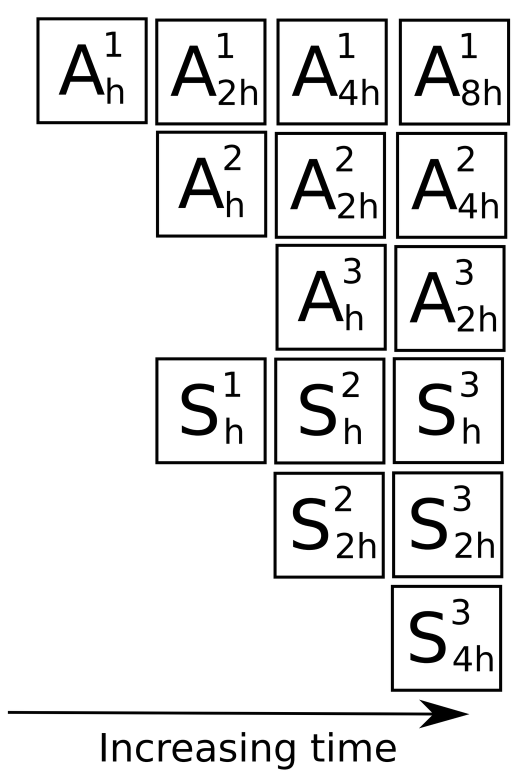


Breaking Multigrid operations into tasks. $A_{k^d h}$ corresponds to matrix construction, $S^i_{k^d h}$ corresponds to smoothing

- Multigrid smoothing reduces error
- Exact fine grid equation not required—just need to be "good enough"
- Matrix equation is a finite element integration—write as quadrature
- Compute quadrature as iterative process (start with low number of samples, p, and increase to improve accuracy of quadrature)

Multitasking

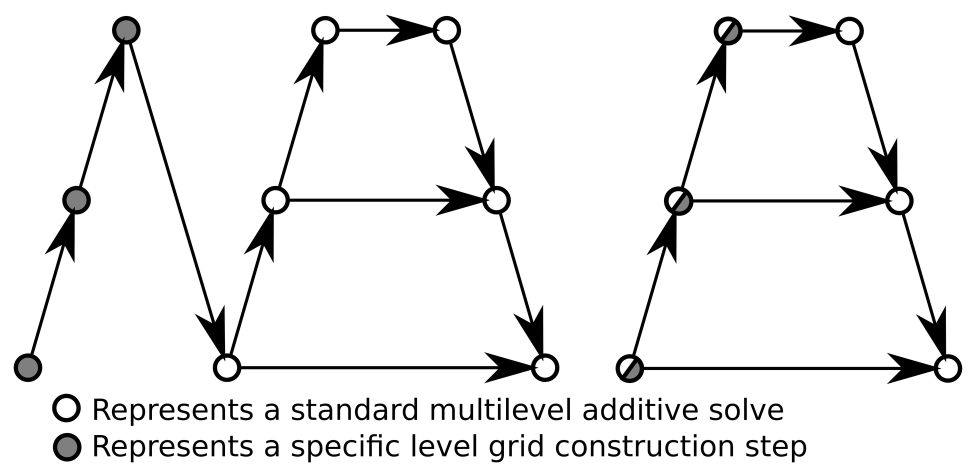
- Break down operations into series of tasks with known dependencies



- Three classes of task:
  1. Construction of the fine grid matrix equation (A_h)
  2. Subsequent construction of coarse grid matrix equations ($A_{k^d h}$)
  3. Multigrid smoothing (S)
- Their dependencies :
  1. Fine grid approximation depends on earlier fine grid approximations
  2. Coarse grid approximation depends a fine grid equation
  3. Smoothing on a level depends on equation existing at that level

Delayed Evaluation

- Enforcing syncronicity between multigrid algorithm and equation construction increases time to solution—multigrid must wait on integration (the assembly)
- Use a local equation approximate instead
- Asynchronously improve local estimate parallel to solve



The two grid construction methods as v-cycles

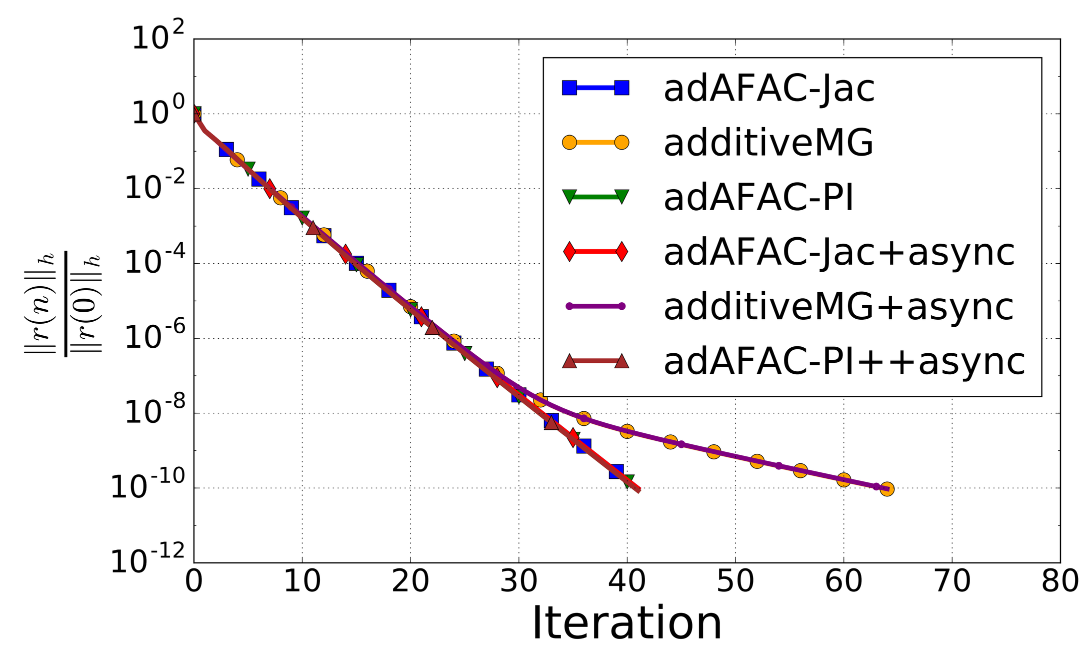
- Start smoothing on coarse grids once approximation of the equation exists on those grids
- Ritz-Galerkin formulation requires a matrix triple product that is dependent upon the fine grid equation
- Compute as traverse grid—updates ripple up single grid level at a time

Greedy Algorithm

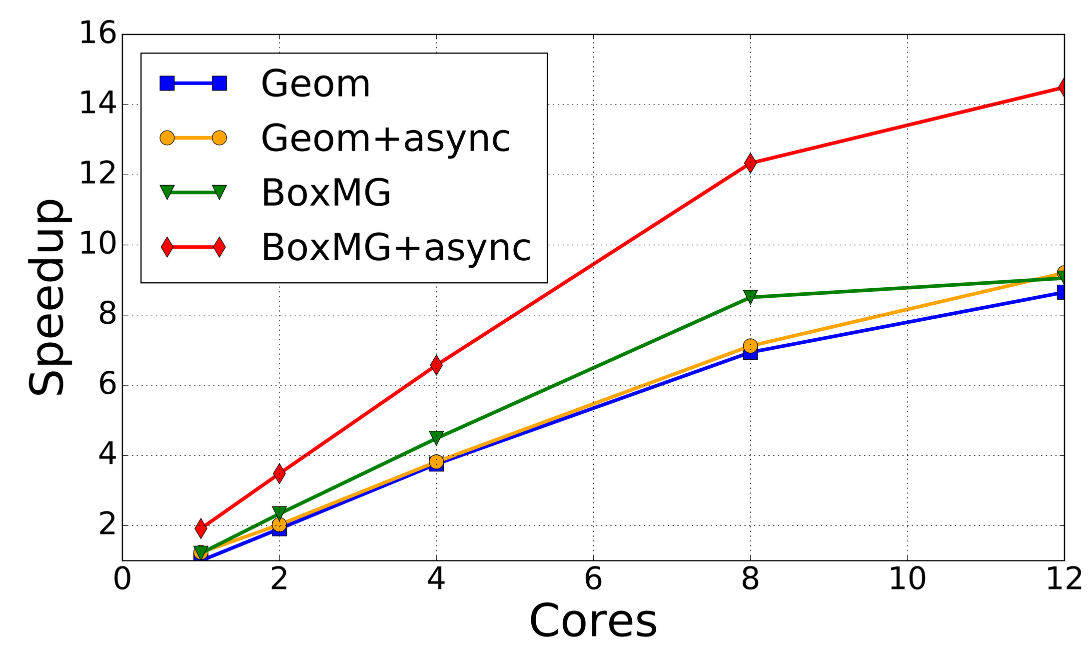
Algorithm 2 This function is called when a fine grid cell is encountered for the first time during a traversal of the grid.

```
function GREEDY-INTEGRATION
    if p = ⊥ then
        p ← 1
        INTEGRATESTENCIL(p)
        Wait for initial integration
        LocalStencil ← NewStencil
        p ← 2
        INTEGRATESTENCIL(p)
    else if p = ⊤ then
        Take no action
    else
        if IntegrationTerminated then
            if NewStencil ≈ LocalStencil then
                p ← ⊤
            else
                p ← p + 1
                INTEGRATESTENCIL(p)
            end if
            LocalStencil ← NewStencil
            IntegrationTerminated ← False
        end if
    end if
end function
```

Consistency validation



adaFAC-Jac Scalability Study



Takeaway

- adaFAC-Jac improves the rate of convergence of Additive Multigrid
- The stability improvements are similar to those from BoxMG
- Asynchronous construction produces consistent convergence rates for the same setup
- Time to solution is reduced by a third due to delayed evaluation of the matrix equations through background tasks

References

L. Hart and S. F. McCormick. Asynchronous multilevel adaptive methods for solving partial differential equations on multiprocessors: Basic ideas. *Parallel Computing*, 12:131–144, 1989.

C. D. Murray and T. Weinzierl. Dynamically adaptive FAS for an additively damped AFAC variant. *arXiv preprint arXiv:1903.10367*, 2019.

C. D. Murray and T. Weinzierl. Lazy stencil integration in multigrid algorithms. In *Parallel Processing and Applied Mathematics: 13th International Conference, PPAM 2020*, LNCS. Springer, (in press).