Multigrid Solution of Sets of FDEs

See Achi Brandt, Math. of Computation, 31 (1977) pp 333-390

Trivial Multigrid:

We can find a solution very quickly on a coarse grid, but the solution has considerable truncation error. We can find a solution with very little truncation error on a fine grid, but it takes a long time.

So—Start on a coarse grid; Find a solution; Interpolate the solution to a finer grid as an initial guess and solve again. Continue to finer grids until you achieve sufficient accuracy.

This is especially helpful for nonlinear problems where we must iterate, and convergence depends on having a good initial guess.

Multigrid Solution of Sets of FDEs

Recall P^h projects a function onto a grid G^h with characteristic discretization scale h.

Let P^h_{2h} represent a "prolongation" operator that interpolates solutions from grid G^{2h} to G^h

$$u^h = P^h_{2h} u^{2h}$$

Note: if $u^{2h} = P^{2h} u$
then $u^h \neq P^h u$ in general

So, trivial Multigrid goes as follows:

- Start with initial guess $\tilde{u}$
- $\tilde{u}^h = P^h \tilde{u}$
- Solve $\mathscr{L}^h u^h = f^h$ using $\tilde{u}^h$ as the initial guess
- $\tilde{u}^{h/2} = P^{h/2}_h u^h$
- Solve $\mathscr{L}^{h/2} u^{h/2} = f^{h/2}$ using $\tilde{u}^{h/2}$ as the initial guess

⋮

Multigrid Solution of Sets of FDEs

Recall the definition of truncation error:

$$\mathcal{L}^h P^h u = f^h + \tau^h \quad ; \quad \tau^h \equiv \mathcal{L}^h P^h u - P^h \mathcal{L} u$$

If we knew τ^h (which we can't unless we know the analytic solution), then we could eliminate the error from our solution !

What if, instead, we could estimate τ^h well enough so that we could correct the differencing scheme on G^{2h} so that it has the same error $(P^{2h}u - u^{2h})$ as the differencing scheme on G^h?

If we could do this, then we could solve the problem on the coarser grid!

Realistically, we can never get as good a solution on G^{2h} as we can get on G^h. The solutions on G^{2h} cannot represent the "highest frequency" components of the solutions on G^h.

(ie, features with the shortest length scales can't be represented; Fourier perspective)

Multigrid Solution of Sets of FDEs

Let us define a “relative” local truncation error.

To begin, we define r_h^{2h} to be a “restriction” operator which interpolates a function from $G^h \rightarrow G^{2h}$

$$r_h^{2h}\mathscr{L}^h u^h = r_h^{2h} f^h \approx f^{2h} \quad \Rightarrow \quad r_h^{2h}\mathscr{L}^h u^h + \left(\mathscr{L}^{2h}u^{2h} - \mathscr{L}^{2h}u^{2h}\right) \approx f^{2h}$$

$$\therefore \quad \mathscr{L}^{2h}u^{2h} \approx f^{2h} + \mathscr{L}^{2h}u^{2h} - r_h^{2h}\left(\mathscr{L}^h u^h\right) = f^{2h} + \tau_h^{2h}$$

So:

$$\tau_h^{2h} \equiv \mathscr{L}^{2h}u^{2h} - r_h^{2h}\left(\mathscr{L}^h u^h\right)$$

is the relative local truncation error, which requires only u^h to compute.

If r_h^{2h} is chosen well, then the solution of

$$\mathscr{L}^{2h}u^{2h} = f^{2h} + \tau_h^{2h}$$

has roughly the same error as the solution on G^h.

Multigrid Solution of Sets of FDEs

2-level "Full Approximation Storage" (FAS) Scheme:

There are other 2-level schemes, but this is very flexible and handles non-linear equations.

Let $\acute{u}^h$ be the current guess on G^h

1) Perform 2-3 "smoothing sweeps" on G^h to reduce high-frequency errors in $\acute{u}^h$. [Gauss-Seidel relaxation]

2) Restrict $\acute{u}^h$ to G^{2h}: $r_h^{2h}\acute{u}^h$

This term is zero when h is the finest spacing. On coarser grids, $\tau_{h/2}^h$ must be considered as a correction to the operator $\mathscr{L}^h$

3) Compute $\tau_h^{2h} = \mathscr{L}^{2h}(r_h^{2h}\acute{u}^h) - r_h^{2h}(\mathscr{L}^h\acute{u}^h - \tau_{h/2}^h)$

4) Solve $\mathscr{L}^{2h}u^{2h} = f^{2h} + \tau_h^{2h}$ using $r_h^{2h}\acute{u}^h$ as an initial guess for $\acute{u}^{2h}$

$\rightarrow$ u^{2h} is a solution on G^{2h} for u^h on G^h, having nearly the same truncation error.

The error will be small, especially in what were low-frequency modes on G^h, but the highest frequency information will be missing.

Multigrid Solution of Sets of FDEs

2-level "Full Approximation Storage" (FAS) Scheme:

There are other 2-level schemes, but this is very flexible and handles non-linear equations.

5) Compute $v^{2h} = u^{2h} - r_h^{2h} \acute{u}^h$

   $\rightarrow$ v^{2h} is the amount by which $r_h^{2h} \acute{u}^h$ changed in order to be a good solution on G^{2h}. It is a smooth function which is the correction needed to eliminate the low-frequency error modes in $\acute{u}^h$; in fact it will correct all the error modes except the highest frequency modes which cannot be represented on G^{2h}.

6) $\acute{u}^h \mapsto \acute{u}^h + P_{2h}^h v^{2h}$

7) Perform 2-3 "smoothing sweeps" on G^h to reduce any high-frequency error introduced by prolonging v^{2h} to G^h.

Multigrid Solution of Sets of FDEs

2-level "Full Approximation Storage" (FAS) Scheme:

Solving $\mathscr{L}u = f$ on G^{2h} is always faster than solving it on G^{h}.

We could use SOR or a direct solver on G^{2h}

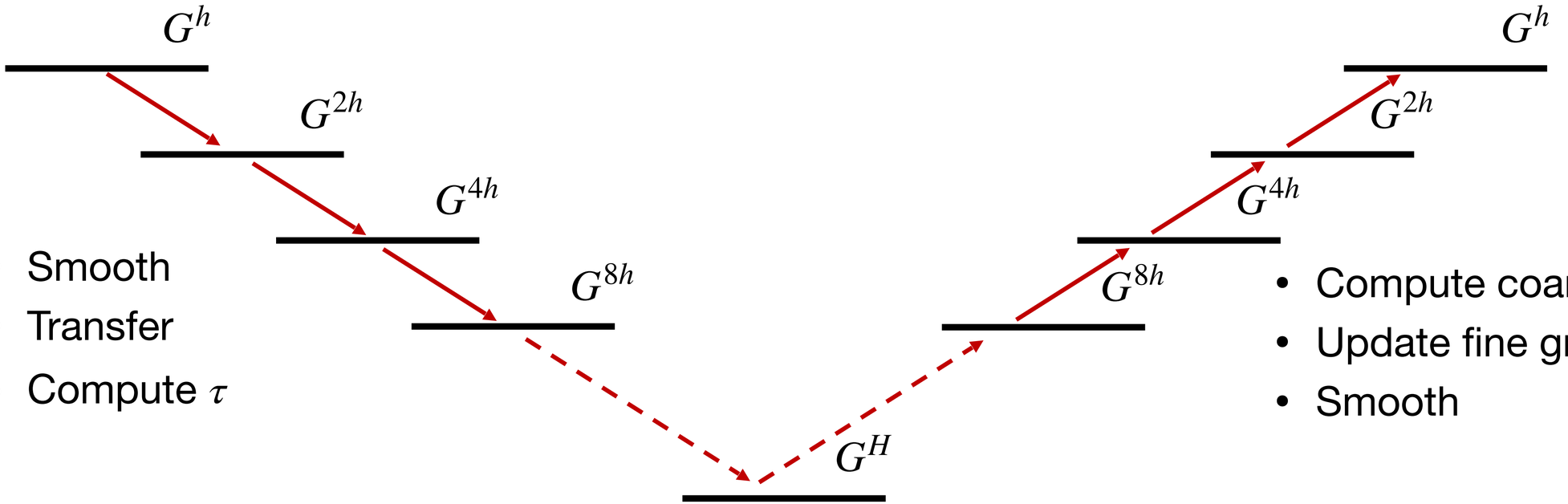
OR

Transfer the solution to $G^{4h.}$ $\Rightarrow$ Multigrid

Keep transferring the problem to coarser grids until solving the problem becomes trivial.

Multigrid Solution of Sets of FDEs

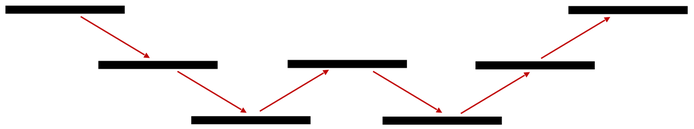
V-cycle (FAS) multigrid algorithm:



- Smooth
- Transfer
- Compute τ

- Compute coarse grid correction
- Update fine grid
- Smooth

Solve $\mathscr{L}^H u^H = f^H + \tau^H_{H/2}$

Can also perform W-cycles  etc…

Multigrid Solution of Sets of FDEs

Convergence Properties:

One full multigrid cycle $G^h \rightarrow G^H \rightarrow G^h$ will typically reduce the residual by about 1 order-of-magnitude. Also, the computational effort scales as $\mathcal{O}(N)$.

When do we stop iterating? $\tau_h^{2h} \sim \tau^{2h} - \tau^h$,

but $\tau^{2h} \sim 4\tau^h$ for 2nd-order FDEs.

$$\therefore \quad \tau_h^{2h} \sim 3\tau^h$$

So, we can stop iterating when $\|R^h\| \lesssim \frac{1}{3}\|\tau_h^{2h}\|$

(residual is about the same size as the truncation error)

Typically, we want $\|R^h\|$ an order of magnitude smaller to be safe.

Multigrid Solution of Sets of FDEs

Important Details of Multigrid

Most of the work (coding and computing) is in the smoothing and inter-level transfers:

1) Smoothing:
   Never use SOR! Sometimes you may need to use “line smoothing”.

2) Inter-level transfers:
   You must use high-enough order interpolation to capture the accuracy you need, but not too high to introduce high-frequency errors.
   The interpolation of functions and “densities” is different. Volume weighted residuals are “densities”!