AM213B Numerical Methods for the Solution of Differential Equations

Lecture 19

List of topics in this lecture

- Restriction operator, interpolation operator
- The two-grid V-cycle (TGV), the multigrid V-cycle (MGV), the full multigrid V-cycle (FMGV), the cost of FMGV
- Multigrid method based on FMGV

Review:

We study iterative methods for solving $T_1 u = b$ that use only evaluations of $T_1 w$.

Operator T_1 is in the difference form

$$T_1 : u \rightarrow T_1 u$$

$$(T_1 u)_i = -\left(\frac{u_{i-1} - 2u_i + u_{i+1}}{h^2}\right), \qquad 1 \le i \le (N-1)$$

with zero BCs $u_0 = u_N = 0$

The difference form is easily extended to 2D and 3D problems.

Weighted Jacobi iteration (damped Jacobi iteration) for solving $T_1 u = b$

$$u^{k+1} = u^k + \underbrace{\omega D^{-1} r^k}_{\text{increment}}, \qquad 0 < \omega \le 1$$

$$\underbrace{r^k}_{\text{residual}} = -T_1 u^k + b$$

For $\omega = 2/3$, all high wave number modes in $\varepsilon^k \equiv u - u^k$ decay very rapidly.

Key observation:

Low wave number modes decay faster on a coarser grid.

Multigrid method

Goal: to speed up the decay of low wave number modes in the error.

Strategy

- mapping the residual down to a coarse grid,
- doing iterations on the coarse grid to calculate increment, and
- mapping the increment back to the fine grid.

Advantage of the residual-increment formulation:

If residual is small, increment is small and the mapping error is small.

End of review

We define the two operators mapping down and up between G_i and G_{i-1}.

Restriction operator (mapping down)

$$\text{R}: \ r^{(i)} \longrightarrow r^{(i-1)}$$

$$r_j^{(i-1)} = \frac{r_{2j-1}^{(i)} + 2r_{2j}^{(i)} + r_{2j+1}^{(i)}}{4}, \qquad j = 1, 2, \ldots, N_{i-1} - 1$$

(This is the weighted average of 3 points).

Here

$r^{(i)} = \left(r_1^{(i)}, r_2^{(i)}, \ldots, r_{N_i-1}^{(i)}\right)^T$ is a vector of $(N_i - 1)$ components on G_i and

$r^{(i-1)} = \left(r_1^{(i-1)}, r_2^{(i-1)}, \ldots, r_{N_i/2-1}^{(i-1)}\right)^T$ is a vector of $(N_i/2 - 1)$ components on G_{i-1}.

Interpolation operator (mapping up)

$$\text{In}: \ v^{(i-1)} \longrightarrow v^{(i)}$$

$$v_{2j}^{(i)} = v_j^{(i-1)}, \qquad j = 1, 2, \ldots, N_i/2 - 1$$

$$v_{2j+1}^{(i)} = \frac{1}{2}\left(v_j^{(i-1)} + v_{j+1}^{(i-1)}\right), \qquad j = 0, 1, \ldots, N_i/2 - 1$$

with zero BCs $v_0^{(i-1)} = v_{N_i/2}^{(i-1)} = 0$

(This is the linear interpolation between 2 points).

where

$v^{(i-1)} = \left(v_1^{(i-1)}, v_2^{(i-1)}, \ldots, v_{N_i/2-1}^{(i-1)}\right)^T$ is a vector of $(N_i/2 - 1)$ components on G_{i-1} and

$v^{(i)} = \left(v_1^{(i)}, v_2^{(i)}, \ldots, v_{N_i-1}^{(i)}\right)^T$ is a vector of $(N_i - 1)$ components on G_i.

Next we write the two-grid V-cycle in the form of an algorithm. Note that in the algorithm form, we usually store the output of an operation over the input if the input is no longer needed. Consequently, we don't need to store all intermediate results and we won't have so many versions of v and V.

Steps in the two-grid V-cycle:

Input: $b^{(i)}$, the RHS of linear system $T_1^{(i)} u^{(i)} = b^{(i)}$.

$v^{(i)}$, an approximate solution of $T_1^{(i)} u^{(i)} = b^{(i)}$.

Output: $v^{(i)}$, an improved approximate solution of $T_1^{(i)} u^{(i)} = b^{(i)}$.

Superscript $^{(i)}$ refers to the grid level.

$v^{(i)} = S(b^{(i)}, v^{(i)})$ Apply the solution operator on G_i

$r^{(i)} = -T_1^{(i)} v^{(i)} + b^{(i)}$ Calculate the residual of $v^{(i)}$

$r^{(i-1)} = R(r^{(i)})$ Map $r^{(i)}$ from G_i down to G_{i-1}

$\underline{v}^{(i-1)} = S(r^{(i-1)}, 0)$ Apply the solution operator on G_{i-1}

$\underline{v}^{(i)} = In(\underline{v}^{(i-1)})$ Map $\underline{v}^{(i-1)}$ from G_{i-1} up to G_i

$v^{(i)} = v^{(i)} + \underline{v}^{(i)}$ Update the approximate solution

$v^{(i)} = S(b^{(i)}, v^{(i)})$ Apply the solution operator on G_i

Two-grid V-cycle (in the form of an algorithm):

function TGV($b^{(i)}$, $v^{(i)}$)

$v^{(i)} = S(b^{(i)}, v^{(i)})$

$r^{(i)} = -T_1^{(i)} v^{(i)} + b^{(i)}$

$v^{(i)} = v^{(i)} + In(S(R(r^{(i)}), 0))$

$v^{(i)} = S(b^{(i)}, v^{(i)})$

return $v^{(i)}$

The benefit of the two-grid V-cycle is limited

In the two-grid V-cycle, the modes in the error with wave numbers $n \geq N_i/2$ are reduced rapidly by doing iterations on G_i. The modes with wave numbers $N_i/4 \leq n < N_i/2$ are reduced rapidly by doing iterations on G_{i-1}. The modes with wave numbers $n < N_i/4$ are still not effectively reduced.

To reduce rapidly the modes with wave numbers $n < N_i/4$, we need to do iterations on G_{i-1}, then on G_{i-2}, then on G_{i-3}, all the way to G_1. This is best implemented in a recursive algorithm.

Multigrid V-cycle (MGV) in the form of a recursive algorithm

Inputs:

$b^{(i)}$, the right hand side of linear system $T_1^{(i)}u^{(i)} = b^{(i)}$

$v^{(i)}$, an approximate solution of $T_1^{(i)}u^{(i)} = b^{(i)}$

Output:

$v^{(i)}$, an improved approximate solution of $T_1^{(i)}u^{(i)} = b^{(i)}$

function MGV($b^{(i)}$, $v^{(i)}$)

if $i = 1$

solve $T_1^{(1)}u^{(1)} = b^{(1)}$ directly (exactly)

else

$v^{(i)} = \mathrm{S}(b^{(i)}, v^{(i)})$

$r^{(i)} = -T_1^{(i)}v^{(i)} + b^{(i)}$

$v^{(i)} = v^{(i)} + \mathrm{In}(\mathrm{MGV}(\mathrm{R}(r^{(i)}), 0))$ calling itself recursively

$v^{(i)} = \mathrm{S}(b^{(i)}, v^{(i)})$

endif

return $v^{(i)}$

Remarks:

1) The number of MGV-cycles required for converges is O(1).

In each MGV-cycle, all modes in the error are reduced by a factor that is independent of N_i. Therefore, the number of MGV-cycles required for convergence is O(1), independent of N_i. In contrast, if we simply do iterations on G_i, the number of iterations required for convergence is $O(N_i^2)$. This is a *huge* difference for large N_i.

2) The cost of one MGV-cycle starting on G_i is $O(N_i)$.

The cost on grid level G_i is $O(N_i)$:

The cost of $v^{(k)} = \mathrm{S}(b^{(k)}, v^{(k)})$ on G_k is $O(N_k)$.

The cost of $r^{(k)} = -T_1^{(k)}v^{(k)} + b^{(k)}$ on G_k is $O(N_k)$.

The cost of $\mathrm{R}(r^{(k)})$ from G_k to G_{k-1} is $O(N_k)$. …

The total cost on all grid levels is the sum of $O(N_i)$.

$$\sum_{k=1}^{i} N_k = \sum_{k=1}^{i} 2^k = 2^{i+1} - 1 = 2N_i - 1 = O(N_i)$$

$$\text{Cost of MGV}(b^{(i)}, v^{(i)}) = \sum_{k=1}^{i} O(N_k) = O\left(\sum_{k=1}^{i} N_k\right) = O(N_i)$$

3) To get started, the MGV-cycle needs an input of $v^{(i)}$.

$v^{(i)}$ is an approximate solution of $T_1^{(i)}u^{(i)} = b^{(i)}$ on grid G_i.

We can start with the default approximation (0) on grid G_i.

A more sensible approach is to start from the coarsest grid (G_1).

- find an approximate solution on G_1;
- map the approximate solution from G_1 to G_2, and improve the solution using MGV starting on G_2 to find an approximate solution on G_2;
- map the approximate solution from G_2 to G_3, and improve the solution using MGV starting on G_3 to find an approximate solution on G_3;
- repeat the process of mapping up to a finer grid, and improving …
- until we finish grid G_i.

This leads to the full multigrid V-cycle

Full multigrid V-cycle (FMGV):

Input:

$b^{(i)}$: the right hand side of the linear system on grid G_i.

Output

$v^{(i)}$: an approximation to the solution of $T_1^{(i)}u^{(i)} = b^{(i)}$.

function FMGV($b^{(i)}$)

map $b^{(i)}$ to $b^{(i-1)}$, $b^{(i-2)}$, …, $b^{(2)}$, $b^{(1)}$

solve $T_1^{(1)}u^{(1)} = b^{(1)}$ exactly

for k = 2 to i

$v^{(k)}$ = MGV($b^{(k)}$, In($v^{(k-1)}$))

endfor

return $v^{(i)}$

Remarks:

*) The FMGV cycle does not need an input of an approximate solution.

*) $T_1^{(1)}u^{(1)} = b^{(1)}$ is a 1×1 system. We solve it exactly.

*) The cost of one FMGV-cycle up to grid G_i is $O(N_i)$.

$$\text{Cost of FMGV}(b^{(i)}) = \sum_{k=1}^{i} \text{Cost of MGV}(b^{(k)}, v^{(k)}) = \sum_{k=1}^{i} C_{\text{MGV}} N_k$$

$$= C_{\text{MGV}} \left(\sum_{k=1}^{i} 2^k \right) \le 2 C_{\text{MGV}} 2^i = 2 C_{\text{MGV}} N_i$$

$$= 2 \times \text{Cost of MGV}(b^{(i)}, v^{(i)})$$

*) For a 2-D problem, the number of grid points = N_k^2.

Cost of MGV starting on grid $G_k = O(N_k{}^2)$

$$\text{Cost of FMGV}(b^{(i)}) = \sum_{k=1}^{i} \text{Cost of MGV}(b^{(k)}, v^{(k)}) = \sum_{k=1}^{i} C_{\text{MGV}} N_k{}^2$$

$$= C_{\text{MGV}} \left(\sum_{k=1}^{i} 4^k \right) \le \frac{4}{3} C_{\text{MGV}} 4^i = \frac{4}{3} C_{\text{MGV}} N_i{}^2$$

$$= \frac{4}{3} \times \text{Cost of MGV}(b^{(i)}, v^{(i)})$$

*) For a 3-D problem, the number of grid points = N_k^3.

Cost of MGV starting on grid $G_k = O(N_k{}^3)$

$$\text{Cost of FMGV}(b^{(i)}) = \sum_{k=1}^{i} \text{Cost of MGV}(b^{(k)}, v^{(k)}) = \sum_{k=1}^{i} C_{\text{MGV}} N_k{}^3$$

$$= C_{\text{MGV}} \left(\sum_{k=1}^{i} 8^k \right) \le \frac{8}{7} C_{\text{MGV}} 8^i = \frac{8}{7} C_{\text{MGV}} N_i{}^3$$

$$= \frac{8}{7} \times \text{Cost of MGV}(b^{(i)}, v^{(i)})$$

Finally we write out the multigrid method based on the FMGV-cycle.

Multigrid method based on FMGV:

$v^{(i)}$ = FMGV($b^{(i)}$)

$r^{(i)} = -T_1{}^{(i)} v^{(i)} + b^{(i)}$

while || $r^{(i)}$ || > tol

  $v^{(i)} = v^{(i)}$ + FMGV($r^{(i)}$)

  $r^{(i)} = -T_1{}^{(i)} v^{(i)} + b^{(i)}$

endwhile

Remark:

The number of full multigrid V-cycles required for convergence is O(1). The cost of each FMGV-cycle is $O(N_i)$. Therefore, the overall cost of the multigrid method is $O(N_i)$.

For a 2-D problem, the overall cost of the multigrid method is $O(N_i^2)$ where N_i is the number of grid points in each direction. In contrast, the overall cost of Jacobi (or weighted Jacobi) iteration is $O(N_i^2)O(N_i^2) = O(N_i^4)$.

For a 2-D problem, the overall cost of the multigrid method is $O(N_i^3)$ where N_i is the number of grid points in each direction. In contrast, the overall cost of Jacobi (or weighted Jacobi) iteration is $O(N_i^3)O(N_i^2) = O(N_i^5)$.