Multigrid method - illustration on 1D problem

- basic idea: transition from denser grids to coarser ones and back and for each grid we suppress „higher freguencies" of the error of the solution

- Jacobi method with a weight (weighted Jacobi method)

is used for suppressing „higher freguencies"

- it is similar to SOR(ω) method but instead of starting from the Gauss-Seidel method we use the Jacobi method

$$x_{m+1,j} = (1-\omega)\, x_{m,j} + \frac{\omega}{a_{jj}}\left(b_j - \sum_{k\neq j} a_{jk}\, x_{m,k}\right)$$

or in the matrix form $x_{m+1} = R^{\omega}_{Jac}\, x_m + c^{\omega}_{Jac}$

where $R^{\omega}_{Jac} = (1-\omega)\mathbb{I} + \omega \tilde{D}^{-1}(\tilde{L}+\tilde{U})$, $c^{\omega}_{Jac} = \omega \tilde{D}^{-1} b$

- for the 1D Poisson equation $\frac{d^2u}{dx^2} = f(x)$, $u(a)=u_a$, $u(b)=u_b$

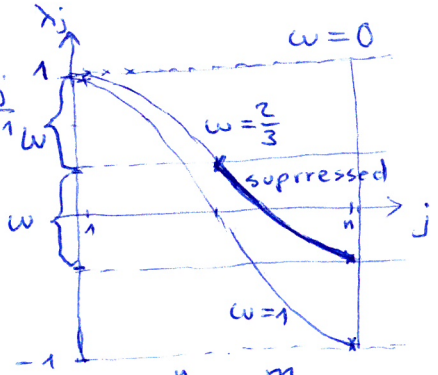
we can use approximation $\left.\frac{d^2u}{dx^2}\right|_{x_j} \simeq \frac{u(x_{j+1}) - 2u(x_j) + u(x_{j-1})}{h^2}$

to transform it into a system of linear equations $\sum_{k=1}^{n} T_{jk} v_k = -h^2 f(x_j)$ on the grid $x_1, \dots x_n$

where $T = \begin{pmatrix} 2 & -1 & & 0 \\ -1 & 2 & \ddots & \\ & \ddots & \ddots & -1 \\ 0 & & -1 & 2 \end{pmatrix} \Rightarrow \tilde{D}^{-1} = \begin{pmatrix} 1/2 & & 0 \\ & \ddots & \\ 0 & & 1/2 \end{pmatrix}$, $\tilde{L}+\tilde{U} = \begin{pmatrix} 0 & 1 & & 0 \\ 1 & \ddots & \ddots & \\ & \ddots & \ddots & 1 \\ 0 & & 1 & 0 \end{pmatrix}$

and thus we can write $R^{\omega}_{Jac} = (1-\omega)\mathbb{I} + \omega \begin{pmatrix} 0 & \frac{1}{2} & & 0 \\ \frac{1}{2} & \ddots & \ddots & \\ & \ddots & \ddots & \frac{1}{2} \\ 0 & & \frac{1}{2} & 0 \end{pmatrix} = 1 - \omega\frac{T}{2}$, $c^{\omega}_{Jac} = \frac{\omega b}{2}$

- eigenvalues are $\lambda_j\left(R^{\omega}_{Jac}\right) = 1 - \omega\left(1 - \cos\frac{\pi j}{n+1}\right) = 1 - \omega + \omega\cos\frac{\pi j}{n+1}$



to get convergence, we must use $0 < \omega \leq 1$

and for $\omega = \frac{2}{3}$, all eigenvalues for $j > \frac{n}{2}$ lie in the interval $(-\frac{1}{3}, \frac{1}{3})$

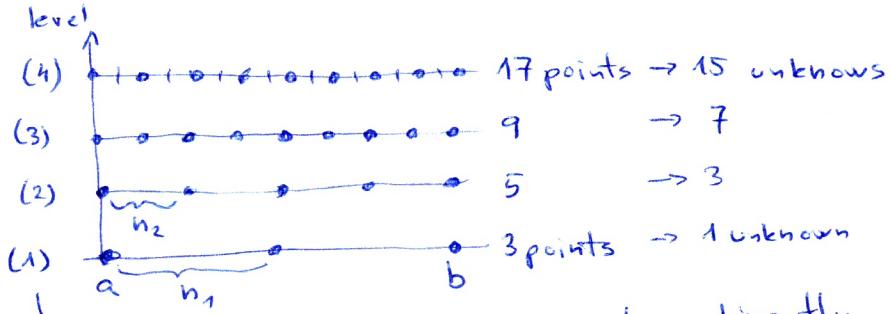
and if we expand the error as $e_0 = \sum_{j=1}^{n} c_j \xi_j$ (eigenvectors), we set $\left(R^{\omega}_{Jac}\right)^m e_0 = \sum_{j=1}^{n} c_j \lambda_j^m \xi_j$

and we see that contributions from higher frequencies ($j > \frac{n}{2}$) will be suppressed by a factor $\frac{1}{3}$ or smaller in each iteration, but low frequencies will be suppressed only slightly (note that higher frequencies in the solution will still be there thanks to the term c^w_{jnc})

- in the multigrid method, we suppress by the weighted Jacobi method higher frequencies for a certain grid and then by transition to a coarser grid we make some of lower frequencies higher one and we suppress then in the next step

- usually it is enough to make 2-5 Jacobi iterations and then to move to a coarser grid

- **transitions to a coarser grid and back**



on the (i)-th level we have $n_i + 2 = 2^i + 1$ points and $h_i = \frac{b-a}{n_i+1} = \frac{b-a}{2^i}$

→ on this level we can solve directly

$$\frac{u(a) - 2u(x_1) + u(b)}{h_1^2} = f(x_1) \Rightarrow u(x_1) = \frac{u_a + u_b}{2} - \frac{h_1^2 f(x_1)}{2}$$

- only at the highest level, we solve $T^{(i)} u^{(i)} = b^{(i)}$ where $b^{(i)}$ includes, if necessary, boundary conditions

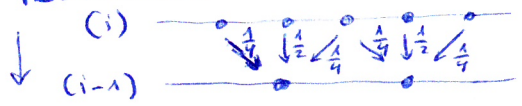
- at lower levels, we solve $T^{(i)} d^{(i)} = r^{(i)}$ where $d^{(i)}$ is a correction to $u^{(i)}$ and $r^{(i)}$ is a residuum given by $r^{(i)} = T^{(i)} u^{(i)} - b^{(i)} \Rightarrow T^{(i)} (\underbrace{u^{(i)} - d^{(i)}}_{\text{this is then a new approximation to the solution}}) = b^{(i)}$

- advantage is that $d^{(i)}$ should be ideally zero, thus we can use as an initial gues zero vector and it is not necessary to deal with non-zero boundary conditions

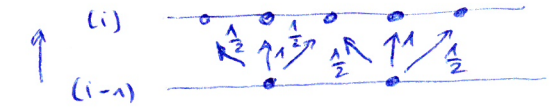
- transition to a lower level is done by averaging and to a higher level by linear interpolation or averaging

- in the 1D model problem we use

restriction



interpolation



or in the matrix form $r^{(i-1)} = P_{(i)}^{(i-1)} r^{(i)}$

where $P_{(i)}^{(i-1)} = \begin{pmatrix} \frac{1}{4} & \frac{1}{2} & \frac{1}{4} & 0 & \cdots & \cdots & 0 \\ 0 & 0 & \frac{1}{4} & \frac{1}{2} & \frac{1}{4} & 0 & \cdots \\ & & & \ddots & & & \\ 0 & \cdots & & & 0 & \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \end{pmatrix}$ $\left.\begin{matrix} \\ \\ \\ \end{matrix}\right\} 2^{i-1}-1$

$\underbrace{\qquad\qquad\qquad}_{2^i-1}$

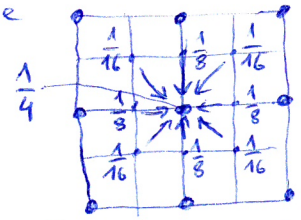
and $d^{(i)} = P_{(i-1)}^{(i)} d^{(i-1)}$ where

$P_{(i-1)}^{(i)} = \begin{pmatrix} \frac{1}{2} & 0 & & \\ 1 & 0 & & \\ \frac{1}{2} & \frac{1}{2} & & \\ & 1 & & \\ & \frac{1}{2} & \ddots & \\ & & & \frac{1}{2} \\ & 0 & & 1 \\ & & & \frac{1}{2} \end{pmatrix}$ $\left.\begin{matrix} \\ \\ \\ \\ \\ \\ \\ \end{matrix}\right\} 2^i-1$

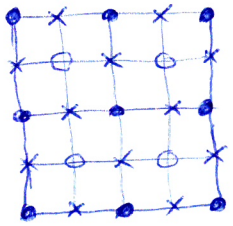
$\underbrace{\qquad\qquad}_{2^{i-1}-1}$

- in 2D one can use

for example when going from (i) to (i-1)

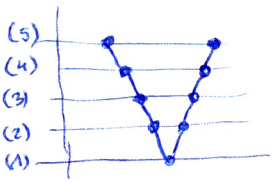


and for transition from (i-1) to (i):



- the same value
× average from two closest •
o average from four closest •

- the basic part of the multigrid method is the so-called V-cycle implemented using recursive function returning a new $v^{(i)}$ providing some initial $v^{(i)}$ and the right-hand site $b^{(i)}$



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

if $i=1$

solve exactly $v^{(1)} = \frac{b^{(1)}}{2}$

otherwise

$v^{(i)} = S(b^{(i)}, v^{(i)})$ - smoothing using a few iterations of weighted Jacobi

$r^{(i)} = T^{(i)} v^{(i)} - b^{(i)}$ - residue

$d^{(i)} = \text{Int}(\text{MGV}(4\,\text{Rest}(r^{(i)}), 0))$ - here we solve $T^{(i)} d^{(i)} = r^{(i)}$

↑ interpolation using $P_{(i-1)}^{(i)}$

↑ restriction using $P_{(i)}^{(i-1)}$

↑ estimate for right-hand side is multiplied by 4 because we have $(2h)^2$ on the coarse grid

$v^{(i)} = v^{(i)} - d^{(i)}$

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

return $v^{(i)}$

complexity for 1D problem:

$\sigma(n)$ →

(tridiagonal matrix T) $\sigma(n)$ →

$\sigma(n)$ →

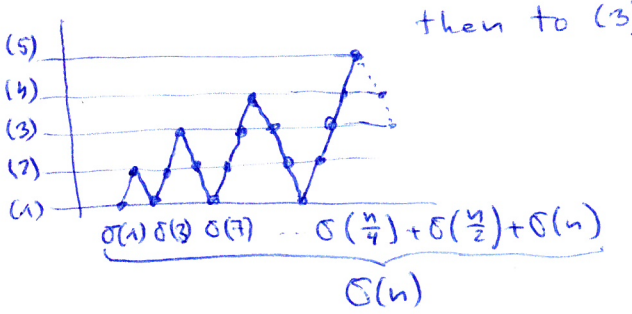
here Int and Rest are clearly $\sigma(n)$ and at lower levels we got $\sigma(\frac{n}{2}) + \sigma(\frac{n}{4}) + \ldots \sigma(1) \approx \sigma(n)$

$\sigma(n)$ →

$\sigma(n)$ →

- thus if we start with n points then the total cost of one V-cycle is $\sigma(n)$

- **full miltigrid method**
  - V-cycle is ~~its~~ basic part
  - we start at level (1), then move to (2) and do V-cycle, then to (3) etc.



function FMG $(b^{(k)}, u^{(k)})$ { ← returns solution on the (k)-th level

solve the problem at level (1) exactly

for $i=2, \dots, k$

$u^{(i)} = MGV\left(b^{(i)}, Int\left(u^{(i-1)}\right)\right)$

}

- if we know the smallest $h^{(k)}$ which should provide a sufficiently accurate solution (with respect to discretization error)

then (k) should be the highest level

and we just do several V-cycles from this level to obtain converged solution

- **note on convergence**
  - it can be shown (see Demmel's book, chap. 6.9 for an example of such analysis)

that in each V-cycle, the error is decreased by a certain factor independent of n (or level) which is smaller than 1 and usually relatively small fraction (for 1 iteration of Jacobi for 1D model problem it is $\sim \frac{1}{2}$)

and usually only a few V-cycles is sufficient to achieve requiered accuracy,

thus overall complexity is $\mathcal{O}(n)$!

- details can be found also in the paper

Yserentant: Acta Numerica 1993, p. 285