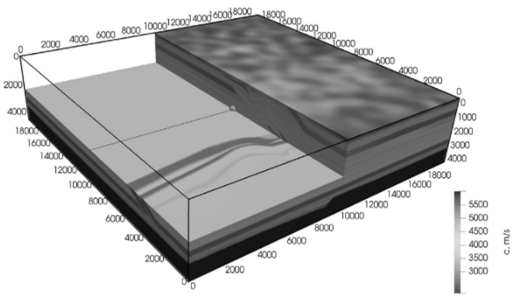
Advanced Solvers for Numerical Partial Differential Equations (PDEs)

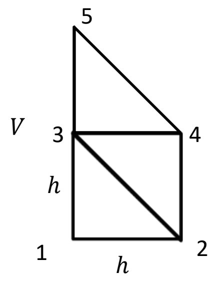


Lecture 2

Nikolay Yavich

n.yavich@skoltech.ru

Exercise 1



$$-\Delta u = 1 \quad \text{в } V,$$
$$\frac{\partial u}{\partial n} = 0 \quad \text{на } \Gamma$$

Derive the P_1 finite-element system for this domain/problem

2

Part 1
Introduction to modern iterative solvers

FD Equation System

$$-\Delta u = f \;\; in \;\; \Omega$$
$$u = \varphi \quad on \;\; \Gamma$$

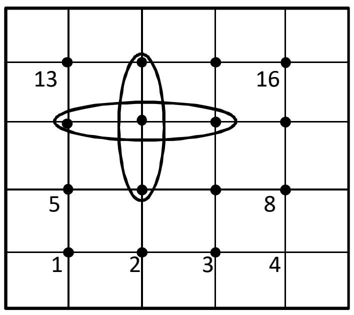
We receive a large equation system

$$A\, u_h = f_h$$

Its size

$$N = (n_x - 1)(n_y - 1)$$

$$\left[\begin{array}{cccc|cccc|cccc|cccc}
4 & -1 & & & -1 & & & & & & & & & & & \\
-1 & 4 & -1 & & & -1 & & & & & & & & & & \\
& -1 & 4 & -1 & & & -1 & & & & & & & & & \\
& & -1 & 4 & & & & -1 & & & & & & & & \\
\hline
-1 & & & & 4 & -1 & & & -1 & & & & & & & \\
& -1 & & & -1 & 4 & -1 & & & -1 & & & & & & \\
& & -1 & & & -1 & 4 & -1 & & & -1 & & & & & \\
& & & -1 & & & -1 & 4 & & & & -1 & & & & \\
\hline
& & & & -1 & & & & 4 & -1 & & & -1 & & & \\
& & & & & -1 & & & -1 & 4 & -1 & & & -1 & & \\
& & & & & & -1 & & & -1 & 4 & -1 & & & -1 & \\
& & & & & & & -1 & & & -1 & 4 & & & & -1 \\
\hline
& & & & & & & & -1 & & & & 4 & -1 & & \\
& & & & & & & & & -1 & & & -1 & 4 & -1 & \\
& & & & & & & & & & -1 & & & -1 & 4 & -1 \\
& & & & & & & & & & & -1 & & & -1 & 4
\end{array}\right]$$



$$n_x = n_y = 5$$
$$N = 16$$

$$A = \begin{pmatrix} C & -I & \\ -I & C & \ddots \\ & \ddots & \ddots \end{pmatrix}$$

Overall properties

$$A\,u_h = g$$

1. **Gauss elimination / LU factorization**
   Computational complexity $Q = O(N^3)$.

2. **Block tridiagonal solver**

$$D\,u_{i-1} + Cu_i + Du_{i+1} = f_i$$
$$u_i = M_i\,u_{i+1} + t_i$$

Complexity $Q = O(N^{2.5})$.

$$\begin{pmatrix} C & D & \\ D & C & \ddots \\ & \ddots & \ddots \end{pmatrix}\begin{pmatrix} \vdots \\ u_i \\ \vdots \end{pmatrix} = \begin{pmatrix} \vdots \\ f_i \\ \vdots \end{pmatrix}$$

These methods ignore sparsity of the blocks.
Thus iterative solver would be a better fit for this problem.

Single multiplication complexity $O(N)$.
Our goal is thus to design a solver with complexity of $O(N^2)$, $O(N\,log(N))$, or even $O(N)$.

5

Iterative solvers

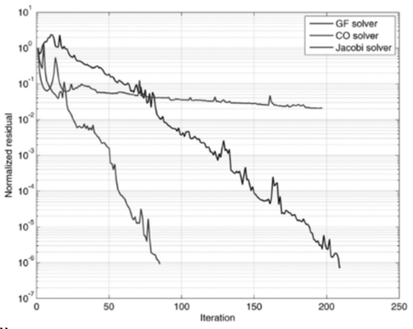
$Au = f$
$\varepsilon > 0$ tolerance

u^0 Initial guess
$u^1 \cdots u^s \cdots$

$Au^s - f$ residual
$u^s - u$ error

Convergence speed ρ

$$\|u^s - u\| \le \rho\|u^{s-1} - u\|$$
$$\le \rho^s\|u^0 - u\| \le \varepsilon\,\|u^0 - u\|$$



Iteration count to reduce the error in $1/\varepsilon$ times

$$s = \left\lceil \frac{\ln\varepsilon}{\ln\rho} \right\rceil$$

Fast solver = ρ close to 0 Slow solver = ρ close to 1

6

Richardson Iteration

$Au = f$ $\quad A > 0$ $\quad$ Richardson, 1910

$u_{k+1} = u_k + \tau(f - Au_k)$ $\quad \tau > 0$

$$\tau_{opt} = \frac{2}{\lambda_{min} + \lambda_{max}} \qquad \rho_{opt} \approx 1 - 2\frac{\lambda_{min}}{\lambda_{max}}$$

Steepest Descent

$u_{k+1} = u_k + \tau_k(f - Au_k)$

$\phi(u) = \frac{1}{2}(Au, u) - (f, u)$ $\qquad \tau_k$ s.th. $\phi(u_{k+1}) \to \min$

$$\tau_k = \frac{(r_k, r_k)}{(r_k, Ar_k)} \qquad \rho \approx 1 - 2\frac{\lambda_{min}}{\lambda_{max}}$$

Equivalent to $r_{k+1} \perp r_k$

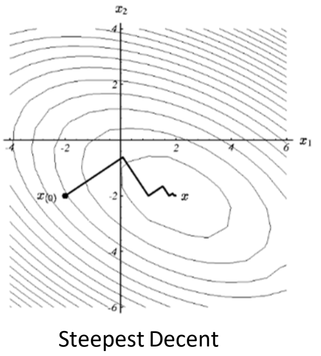
Conjugate Gradient Method (CG)

$Au = f$ $\qquad A = A^{\mathrm{T}} > 0$

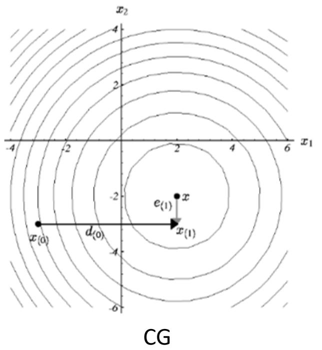
$\phi(u) = \frac{1}{2}(Au, u) - (f, u)$

u_k minimizes $\phi(u)$ over $K_m = span(f, Af, \cdots) + u_0$

Krylov subspace



Steepest Decent



CG

Conjugate Gradient Method (CG)

* Combines SD with Gram-Schmidt orthonormalization
* Avoids repeating directions

$$r_k = Au_k - f$$

$$u_{k+1} = u_k - \beta_k p_k$$

$$p_k = r_k - \alpha_k p_{k-1}$$

$$\alpha_k = \frac{(r_{k-1}, Ap_{k-1})}{(p_{k-1}, Ap_{k-1})} \qquad \beta_k = \frac{(r_{k-1}, r_{k-1})}{(p_{k-1}, Ap_{k-1})}$$

$$\rho \approx 1 - 2\sqrt{\frac{\lambda_{min}}{\lambda_{max}}}$$

Hestenes & Stiefel, 1952

* Most commonly used algorithm for problems with SPD matrices

9

Complexity of different solvers

Gauss elimination	$O(N^3)$
Block tridiagonal solver	$O(N^{2.5})$
Jacobi and Seidel	$O(N^2 \log\frac{1}{\varepsilon})$
Richardson and steepest descent	$O(N^2 \log\frac{1}{\varepsilon})$
Conjugate gradient	$O(N^{\frac{3}{2}} \log\frac{1}{\varepsilon})$
???	
Optimal estimate	$O(N)$

10

Part 2
Introduction to Preconditioning

Preconditioned Iterative Solvers

For solving $A\,u = g$ we already discussed

$$u^{k+1} = u^k + \tau\left(g\ - Au^k\right).$$

What if we substitute scalar τ with a matrix B?

$$u^{k+1} = u^k + B^{-1}\left(g\ - Au^k\right)$$

Two pathalogical cases
- $B^{-1} = \tau I$, nothing really changes
- $B = A$, convergence in 1 iteration, but complexity is of the original problem.

B is a preconditioner. The goal is to design B, ***close*** to A,
yet easier to invert.

Jacobi Preconditioner

$$u^{k+1} = u^k + B^{-1}(g - Au^k)$$

Take

$$B = \text{diag}(A).$$

How good will such a solver for the model problem?

For that we have to analyze $S = I - B^{-1}A$

Note $B_{ii} = \frac{4}{h^2}$.

Studying the eigenvalue problem

$$(I - B^{-1}A)v = \lambda v$$

Leads to

$$\lambda_{km} = 1 - \frac{h^2}{4}\frac{4}{h^2}\left[\sin\left(\frac{\pi kh}{2}\right)^2 + \sin\left(\frac{\pi mh}{2}\right)^2\right].$$

Thus

$$\rho = 1 - 2\sin\left(\frac{\pi h}{2}\right)^2 \approx 1 - \frac{\pi^2 h^2}{2},$$

- Arithmetical complexity $Q = cN^2 \ln\frac{1}{\varepsilon}$

13

Part 3
FFT solver as a direct method

14

Motivation

$$-\Delta u = f \quad \text{в} \quad \mathbb{R}^2$$
$$u = 0 \ \text{при} \ |x| + |y| \to \infty$$

Apply double Fourier transform

$$U(k_x, k_y) = \iint_{-\infty}^{\infty} e^{-2\pi i(k_x x + k_y y)} u(x, y) dx dy$$

$$F(k_x, k_y) = \iint_{-\infty}^{\infty} e^{-2\pi i(k_x x + k_y y)} f(x, y) dx dy$$

$$(2\pi)^2 (k_x^2 + k_y^2) U(k_x, k_y) = F(k_x, k_y)$$

Apply inverse double Fourier transform $U(k_x, k_y)$

$$u(x, y) = \iint_{-\infty}^{\infty} e^{2\pi i(k_x x + k_y y)} U(k_x, k_y) dx dy$$

15

Discrete Fourier Transform

$$v_k = \sum_{n=0}^{N-1} e^{-\frac{2\pi i\, k\, n}{N}} p_n \qquad k = 0..n-1$$

This summation has a complexity $O(N^2)$

Cooley Tukey, 1965 г.
Fast Fourier Tranform
$O(N\ log(N))$

16

Separation-by-variable Solver

$$-\Delta u = f \quad \text{в} \ (0,1)^2, \qquad u = 0 \quad \text{на} \quad \Gamma \qquad\qquad A\,u = g$$

Spectral decomposition

$$A = W\,D\,W^{-1}$$

Solution could be written

$$u = W\,D^{-1}\,W^{-1}\,g.$$

This would be practical if we could multiply by W and W^{-1} efficiently.

Unfortunately, W is dense thus single multiplication takes $O(N^2)$.

17

Separation-by-variable Solver

We know entries of W

$$W_{ij} = C \sin(\pi i_x j_x h) \sin(\pi i_y j_y h),$$

i_x and i_y are indices of node i

$$i = (i_y - 1)(n-1) + i_x$$

Multiplication $v = Wp$ is equivalent to

$$v_i = \sum_{j=1}^{N} W_{ij} p_j = C \sum_{j_y=1}^{n-1} \sin(\pi i_y j_y h) \sum_{j_x=1}^{n-1} \sin(\pi i_x j_x h)\, p_{(j_y-1)(n-1)+j_x}$$

This is discrete double sine transform!

Both forward and inverse could be performed in $O(N\ log\ N)$ using FFT

18

Separation-by-variable Solver

Thus solution

$$u = W\, D^{-1}\, W^{-1}\, g$$

computed as

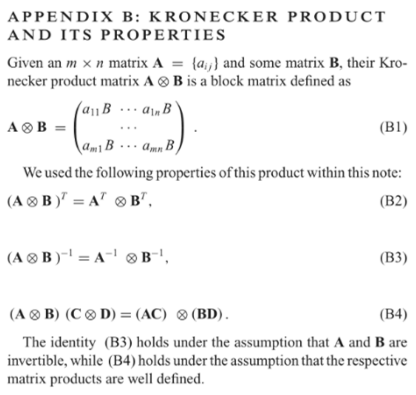
1) $v = W^{-1}g$ (inverse 2D DST);
2) $p = D^{-1}v$ (diagonal matrix scaling)
3) $u = Wp$ (forward 2D DST).

The overall complexity is $O(N\, log\, N)$,
For comparison, with the CG we had $O\left(N^{\frac{3}{2}} \log\frac{1}{\varepsilon}\right)$

- This is a **direct** method
- We looked at uniform grids, constant coefficients.
- Let as check few generalizations

Kronecker Product Representation

$$-\Delta u = f \quad \text{в} \;\; (0,1)^2, \qquad A\, u = g$$
$$u = 0 \quad \text{на} \;\; \Gamma$$



APPENDIX B: KRONECKER PRODUCT AND ITS PROPERTIES

Given an $m \times n$ matrix $\mathbf{A} = \{a_{ij}\}$ and some matrix $\mathbf{B}$, their Kronecker product matrix $\mathbf{A} \otimes \mathbf{B}$ is a block matrix defined as

$$\mathbf{A} \otimes \mathbf{B} = \begin{pmatrix} a_{11}B & \cdots & a_{1n}B \\ & \cdots & \\ a_{m1}B & \cdots & a_{mn}B \end{pmatrix}. \qquad \text{(B1)}$$

We used the following properties of this product within this note:

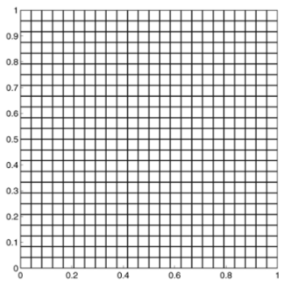
$$(\mathbf{A} \otimes \mathbf{B})^T = \mathbf{A}^T \otimes \mathbf{B}^T, \qquad \text{(B2)}$$

$$(\mathbf{A} \otimes \mathbf{B})^{-1} = \mathbf{A}^{-1} \otimes \mathbf{B}^{-1}, \qquad \text{(B3)}$$

$$(\mathbf{A} \otimes \mathbf{B})\,(\mathbf{C} \otimes \mathbf{D}) = (\mathbf{AC}) \otimes (\mathbf{BD}). \qquad \text{(B4)}$$

The identity (B3) holds under the assumption that $\mathbf{A}$ and $\mathbf{B}$ are invertible, while (B4) holds under the assumption that the respective matrix products are well defined.

Yavich & Zhdanov, 2020



$$A = I_y \otimes \mathrm{L}_{xx} + \mathrm{L}_{yy} \otimes I_x$$

Eigenvector matrix $W = W_y \otimes \mathrm{W}_x$

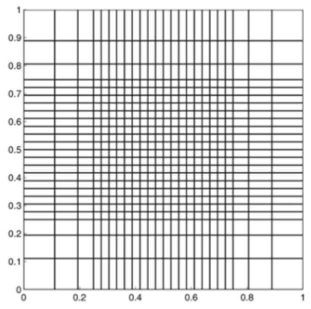
L_{xx} L_{yy} tridiaganol

$$WA\, W^T\, Wu = Wg$$

$$T\;\; \tilde{u} = \tilde{g}$$

T diagonal !

Non-uniform Grids



The system and eigenvector matrices still admit this representations

$$A = I_y \otimes \mathrm{L}_{xx} + \mathrm{L}_{yy} \otimes I_x$$

$$W = W_y \otimes \mathrm{W}_x$$

Althoght W_x and W_y are not sines any more

Yet we can perform spectral decomposition of L_{xx} and L_{yy} and find W_x and W_y

21

1D coefficients

$$-\frac{\partial}{\partial x}\left(a(x)\frac{\partial u}{\partial x}\right) - \frac{\partial}{\partial y}\left(b(y)\frac{\partial u}{\partial y}\right) = g \text{ в } V = (0,1)^2$$
$$u = 0 \text{ на } \Gamma$$

The system and eigenvector matrices still admit this representations

$$A = I_y \otimes \mathrm{L}_{xx} + \mathrm{L}_{yy} \otimes I_x$$

$$W = W_y \otimes \mathrm{W}_x$$

with W_x and W_y easy to compute

22

Part 4
FFT solver as a preconditioner

Preconditioned Richardson Method

$$A\,u = g$$

u^0 initial guess

B some preconditioner

ε tolerance

residual

$$u^{k+1} = u^k + \tau B^{-1}\left(g - Au^k\right) \qquad \left\|g - Au^k\right\| < \varepsilon\ ?$$

How fast the convergence will be / how many iterations will be needed?

u^* exact solution

$z^k = u^* - u^k$ error

$$z^{k+1} = z^k - \tau B^{-1} A z^k$$

$$= (I - \tau B^{-1} A) z^k$$

$$\left\|z^{k+1}\right\| \leq \|S\|\left\|z^k\right\|$$

Convergence depends on the norm/largest eigenvalue of

$$S = I - \tau B^{-1} A$$

The smaller the faster convergence!

Spectrally-equivalent preconditioner

$$A\,u = g$$

a preconditioner B e.g. in

$$u^{k+1} = u^k + \tau B^{-1}\left(g - Au^k\right)$$

spectrally-equivalent if

$$C_1\, Bv \cdot v \le Av \cdot v \le C_2\, Bv \cdot v \qquad \forall v$$

with C_1 and C_2 independent of problem size

$$S = I - \tau B^{-1} A$$

$\rho(S)$ independent of h!

With such a preconditioner iteration count depends only C_1, C_2 и ε!

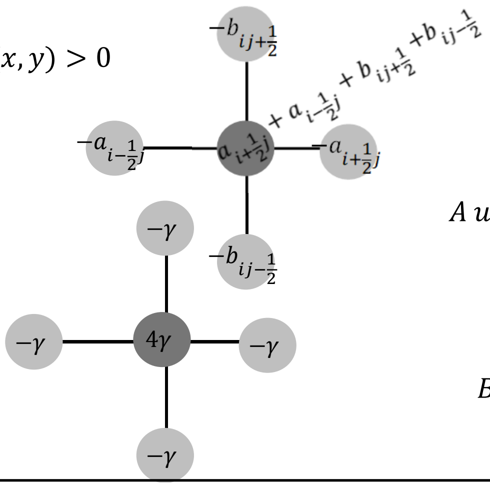
Anisotropic Diffusion Equation

$$-\frac{\partial}{\partial x}\left(a(x,y)\frac{\partial u}{\partial x}\right) - \frac{\partial}{\partial y}\left(b(x,y)\frac{\partial u}{\partial y}\right) = g \;\text{ в }\; V = (0,1)^2$$

$$u = 0 \;\text{ на }\; \Gamma$$

$a(x,y), b(x,y) > 0$

uniform grid



$$A\,u = g$$

$$-\gamma\,\Delta u = f \;\text{ в }\; V$$

$$u = 0 \;\text{ на }\; \Gamma$$

$\gamma > 0$ - const

$$B\,u = f$$

FFT as preconditioner

- Matrices A and B are of the same size
- Vector $B^{-1}v$ could be computed with FFT

$$u^{k+1} = u^k + \tau B^{-1}\left(g - Au^k\right)$$

Claim. Assume

$$c_1\,\gamma \le a(x,y) \le c_2\,\gamma \quad \text{в } V$$
$$c_1\,\gamma \le b(x,y) \le c_2\,\gamma$$

then B is spectrally-equivalent A with constants c_1 and c_2

i.e. solver convergence speed will depend on closeness of $a(x,y)$ and $b(x,y)$ to γ.

27

FFT as preconditioner

□ $$Av\cdot v =$$

$$\frac{1}{h^2}\left(\ldots a_{i-\frac{1}{2}j}\left(v_{i-1j} - v_{i\,j}\right)\ldots a_{i-\frac{1}{2}j}\left(v_{ij} - v_{i-1\,j}\right)\ldots\right)\cdot \left(\ldots v_{i-1j}\; v_{ij}\ldots\right) =$$

$$\ldots\frac{a_{i-\frac{1}{2}j}}{h^2}\left(v_{ij} - v_{i-1\,j}\right)^2\ldots \le$$
$$\ldots\frac{c_2\gamma}{h^2}\left(v_{ij} - v_{i-1\,j}\right)^2\ldots =$$

$$= c_2 Bv\cdot v$$

Similarly,

$$Av\cdot v \ge c_1 Bv\cdot v$$

■

We thus designed a solver with complexity $O\left(N\,log\,N\log\frac{1}{\varepsilon}\right)$

28

Part 2

Multigrid Method

R. Fedorenko, 1961
A. Brand, 1973
W. Hackbusch, 1977

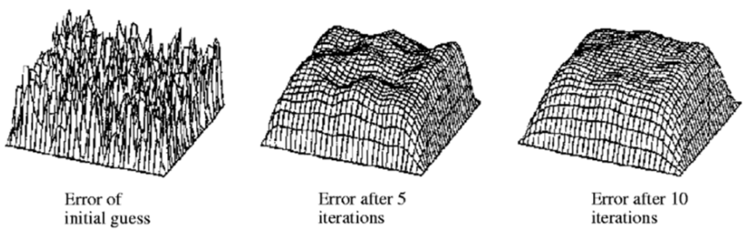
Motivation

Model problem

$$-\Delta u = g \quad \text{в } V$$
$$u = 0 \quad \text{на } \Gamma$$

$$A_h\, u_h = g_h$$

Why the Jacobi solver is so slow?

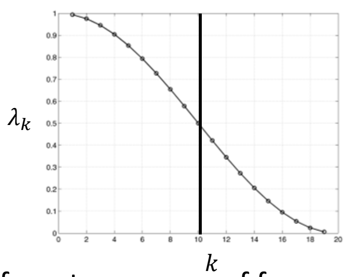


U. Trottenberg (2001)

Motivation

Why Jacobi solver smooths our the error?

$$S = E - D^{-1}A$$
$$D = diag(A)$$
$$\lambda_k = 1 - \sin\left(\frac{\pi kh}{2}\right)^2 \qquad u^h_{k,j} = \sin(\pi jkh)$$



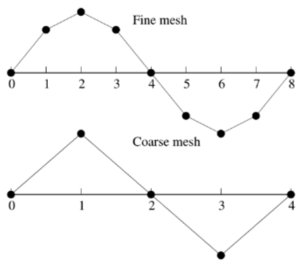
$n = 20$

We can form two groups of frequencies:

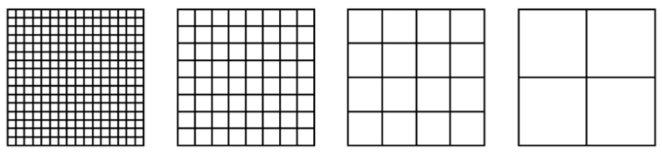
- Low frequencies $k < n/2\,, \frac{1}{2} < \lambda < 1$
- High frequencies $\frac{n}{2} \leq k,\ \lambda \leq 1/2$

Motivation

- High-frequency error components smoothed out well on fine grids
- Low-frequency error components represented well on coarse grids



- Coarser problems are easer to solve

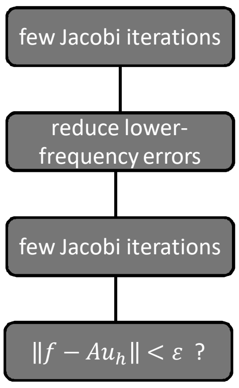


The Big Idea of Multigrid

presmoothing — reduces high-frequency errors

coarse grid correction

postsmoothing — reduces high-frequency errors



Coarse grid correction

$A_h \, u_h = g_h$

u_h^* exact sol-n

u_h current guess

$r_h \; = g_h - A_h u_h$

$e_h \; = u_h^* - u_h$

Note: $e_h \; = A_h^{-1} r_h$

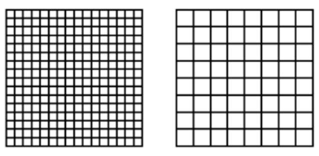
If we were able to compute the sum

$$u_h := u_h \; + e_h \; = u_h^*$$

then the problem would be solved

$A_h e_h \; = r_h \Longrightarrow$

- $r_H \; = R \, r_h$
- $A_H d_H \; = r_H$
- $d_h \; = P \, d_H$
- $u_h := u_h \; + d_h$

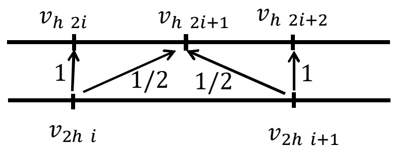


h $\quad$ $H = 2h$

We need to define

- restriction operator R
- coarse grid operator A_H
- prolongation operator P

Prolongation Operator



h

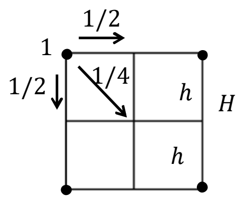
$H = 2h$

$$v_{h\ 2i} = v_{2h\ i}$$

$$v_{h\ 2i+1} = \frac{1}{2}(v_{2h\ i} + v_{2h\ i+1})$$

$$v_h = P\, v_{2h}$$

$$P = \frac{1}{2}\begin{bmatrix} 1 & & & \\ 2 & & & \\ 1 & 1 & & \\ & 2 & & \\ & 1 & 1 & \\ & & \vdots & \\ & & \vdots & \\ & & 1 & 1 \\ & & & 2 \\ & & & 1 \end{bmatrix}$$



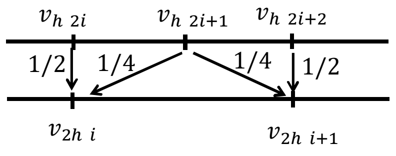
Stencil $P = \frac{1}{2}[1\ \ 2\ \ 1]$

$$P = \frac{1}{4}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

35

Restriction Operator

$$(u_H, Rv_h)_H = (Pu_H, v_h)_h$$



h

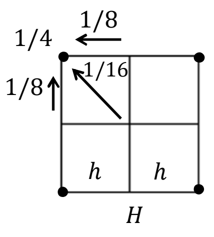
$H = 2h$

$$v_{2h\ i} = \frac{1}{4}(v_{h\ 2i-1} + 2v_{h\ 2i} + v_{h\ 2i+1})$$

$$v_{2h} = R\, v_h$$

$$R = \frac{1}{4}\begin{bmatrix} 1 & 2 & 1 & & & & & \\ & & 1 & 2 & 1 & & & \\ & & & & 1 & 2 & 1 & \\ & & & & \cdots & \cdots & \cdots & \\ & & & & & & 1 & 2 & 1 \end{bmatrix}$$

$R = \frac{1}{4}[1\ \ 2\ \ 1]$



$$R = \frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

$P = 2\ R^T$ in 1D
$P = 4\ R^T$ in 2D

36

Coarse Grid Operator

- Direct method
  A_H – approximation of the differential operator on grid H
- Galerkin projection method
  $$A_H = R\ A_h\ P$$
  Yet the product should be computed implicitly, e.g.
  $$A_H v = (R\ (A_h\ (P\ v)))$$

Two-grid Algorithm

$A_h\, u_h\ = g_h$

$$\begin{tikzcd} n_{pre} \arrow[dr, "R"'] & & n_{post} & h \\ & A_H^{-1} r_H \arrow[ur, "P"'] & & H \end{tikzcd}$$

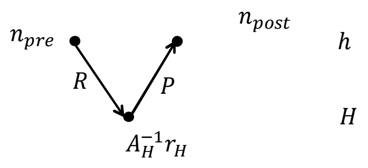
Algorithm $u_h^0,\ A_h,\ g_h$

```
$u_h :=$ Jacobi_Iter$(n_{pre}, u_h^0, A_h, g_h)$      *presmoothing*
$r_h = g_h - A_h u_h$
$r_H = R\ r_h$                                        *Restriction of the residual*
Solve $A_H d_H = r_H$                                 *Coarse grid problem*
$d_h = P\ d_H$                                        *Prolongation*
$u_h := u_h + d_h$
$u_h :=$ Jacobi_Iter$(n_{post}, u_h, A_h, g_h)$       *postsmoothing*
$\|g_h - A_h u_h\| < \varepsilon$ **?**
```

Two-grid Algorithm

$A_h\, u_h \;= g_h$

$D = diag(A_h)$



Iteration matrix

$$S = (I - D^{-1}A_h)^{n_{post}}\left(I - PA_H^{-1}RA_h\right)(I - D^{-1}A_h)^{n_{pre}}$$

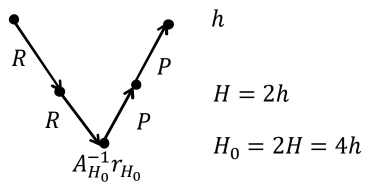
Theorem

$\rho(S) < \frac{1}{2}$ is independent of h, but depends on n_{post}, n_{post}

$n_{post} + n_{pre} > 0$

We designed a spectrally-equivalent preconditioner, although we have solve $A_H e_H = r_H$ …

Multigrid algorithm

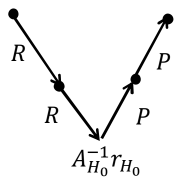


$A_h\, u_h \;= g_h$

Algorithm u_h =V-cycle($u_h^0,\ A_h,\ g_h$)

```
$u_h :=$ Jacobi_Iter($n_{pre}, u_h^0, A_h, g_h$)
$r_h = g_h - A_h u_h$
$r_H = R\, r_h$
if  $H \neq H_0$  $e_H$=V-cycle(0, $A_H, r_H$)          *Recursion*
else   solve $A_H e_H = r_H$          *We are on the coarsest grid*
$e_h = P\, e_H$
$u_h := u_h + e_h$
$u_h :=$ Jacobi_Iter($n_{post}, u_h, A_h, g_h$)
Return $u_h$
```

Computational Complexity



Assume the coarsest problem has size of $O(1)$

Then complexity of a single iteration is

$$\text{1D: } N + \frac{1}{2}N + \frac{1}{4}N + \cdots = O(N)$$
$$\text{2D: } N + \frac{1}{4}N + \frac{1}{16}N + \cdots = O(N)$$

i.e. multigrid had linear computational complexity

What will be the convergence speed?

41

Multigrid Iteration Matrix

Assume we L grids from $k = L$ (fine) to $k = 1$ (coarsest)

A_k	system matrix	$R_{k\to k-1}$
S_k	iteration matrix	$P_{k-1\to k}$

$$S_1 = 0$$
$$S_k = \left(I - D_k^{-1}A_k\right)^{n_{post}} \left(I - P_{k-1\to k}\,(I - S_{k-1})A_{k-1}{}^{-1}\; R_{k\to k-1}A_k\right)\left(I - D_k^{-1}A_k\right)^{n_{pre}}$$

Full iteration matrix $S = S_L$

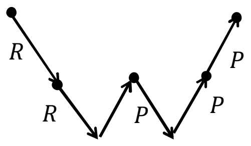
$\rho(S)$ = 0.05 .. 0.5 for model problems

Spectrally-equivalent preconditioner

Solver with optimal complexity $\quad \mathrm{O}(N \log \frac{1}{\varepsilon})$

42

W-cycles



Algorithm u_h =W-cycle($u_h^0,\ A_h,\ g_h$)

```
$u_h := \text{Jacobi\_Iter}(n_{pre}, u_h^0, A_h, g_h)$
$r_h = g_h - A_h u_h$
$r_H = R\, r_h$
if  $H \neq H_0$  $e_H$=W-cycle(0, $A_H, r_H$)          Two recursion calls
              $e_H$=W–cycle($e_H,\ A_H,\ r_H$)
 else    solve $A_H e_H = r_H$                    We are on the coarsest grid
$e_h = P\, e_H$
$u_h := u_h + e_h$
$u_h := \text{Jacobi\_Iter}(n_{post}, u_h, A_h, g_h)$
Return $u_h$
```

- More robust than V-cycle
- Also there is also FMG.

43

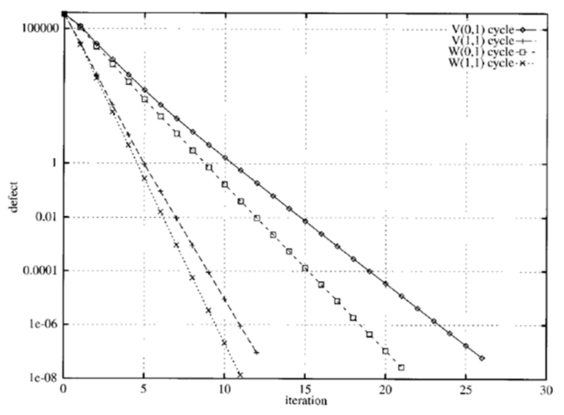
Complexity of different solvers

Gauss elimination	$O(N^3)$
Jacobi and Seidel	$O(N^2 \log\frac{1}{\varepsilon})$
Richardson and steepest descent	$O(N^2 \log\frac{1}{\varepsilon})$
Conjugate gradient	$O(N^{\frac{3}{2}} \log\frac{1}{\varepsilon})$
FFT-based	$O(N \log N)$
Multigrid V-cycle	$O(N \log\frac{1}{\varepsilon})$
Optimal estimate	$O(N)$

44

Example

$$-\Delta u = g \;\text{ в }\; V$$
$$u = 0 \;\text{ на }\; \Gamma \qquad\qquad h = \frac{1}{256}$$



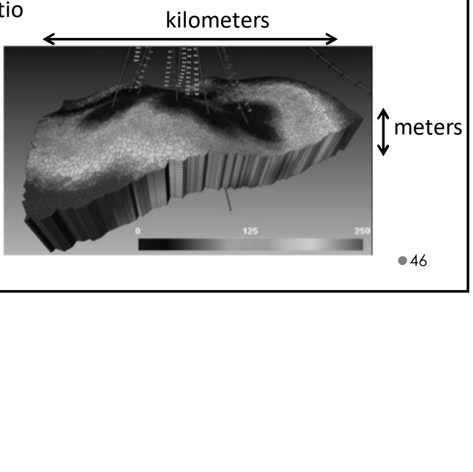
Trottenberg et al, 2001

Resumé on Multigrid

- Solver with optimal complexity
- Has many generalization (nonlinear equations etc)

The concept we discusses is only robust for
- Square grids
- Heterogeneous equations but isotropic coefficients

These cases require special treatment
- Stretched grids / high aspect ratio
- Anisotropic coefficients


kilometers
meters
0
125
250