Fast Solvers 1. Lecture

Our overall goal is to solve the linear system of equations:

Find $x \in \mathbb{R}^n$ such that $Ax = f$ (1)

Here A is (real) matrix of size $n \times n$, $A \in \mathbb{R}^{n,n}$.

For now we only assume that A is nonsingular.

We know from elementary courses that Gaussian Elimination will solve this problem (1). In fact

$N_{LU} := \frac{2}{3}n^3 - \frac{1}{2}n^2 - \frac{1}{6}n$ is the number of arithmetic float point operation $(+ - \times \div)$ needed to compute the LU-factorisation of A with (Naive) G.E. Note that these operation counts are the best estimate for "work".

We define $G_n := \frac{2}{3}n^3$ so in big O notation $\mathcal{O}(n^3)$

2010 Fastest PC CPU peaked at $109 \cdot 10^9$ flops

So "theoretically" the work of $n \approx 5000$ can be done in 1 second and $n \approx 200000$ in 24 hours. Disclaimer: This does not account for memory or communication. Or that the peak can be reach for these kinds of problems.

" / $n = 5000$ A (and A^{-1}) requrie 0.2gb each

$n = 200\,000$ — " — requrie 320gb

Note if you ran out RAM, you are in truble.

However we are not solving general non singular matrices A.
We will focus on matrices arising from discretizations of PDE or other things ~~like~~ related to PDE.

For these discretization we assume we are basis functions with local support, that is, we assume A is sparse (Most entries are zeroes)

This allows us to reduce memory and computational requirements for solving (1).

Note: If A is sparse, then A^{-1} is typically dense.

We start with a very simple Model problem.

1D Poisson Problem

Given $f(x)$ solve $-u''(x) = f(x)$ on $(0,1)$
with $u(0) = u(1) = 0$

For simplicity we use a FDM

Lecture 4: Linear iterative methods

The Richardson iteration

$$u^k = u^{k-1} - \tau (Au^{k-1} - b) \qquad \text{(R.I.)}$$

damping / relaxation parameter (acceleration) → Cost: $\mathcal{O}(n)$ if A is sparse.

This is an important method, but by itself, it is somewhat useless.

Let $e^k := u^k - u$ be the error

$$\text{(R.I.)} - u \Rightarrow \quad e^k = e^{k-1} - \tau (Au^{k-1} - \overset{Au}{\overset{\shortparallel}{b}}) = e^{k-1} - \tau A e^{k-1}$$

ℓ^2-norm $\quad \|e^k\| = \|e^{k-1} - \tau A e^{k-1}\| \leq \|I - \tau A\| \, \|e^{k-1}\|$

If $\|I - \tau A\| \leq 1$ we have convergence

$$\boxed{\text{Matrix / Operator norm} \quad \|A\| = \max_{x \neq 0} \frac{\|Ax\|}{\|x\|} = \lambda_{max} \text{ if } A \text{ is SPD}}$$

Assume for simplicity that A is SPD

$$\|I - \tau A\| = \max_{0 \neq x} \frac{\|(I - \tau A)x\|}{\|x\|} = \begin{cases} 1 - \tau \lambda_{min} \\ -(1 - \tau \lambda_{max}) \end{cases}$$

Let $\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$ be the eigenvalue in order

$\rho := \|I - \tau A\|$ Convergence Factor / contraction number

Optimal τ: $1 - \tau\lambda_1 = -(1 - \tau\lambda_n)$

$$\tau_{opt} = \frac{2}{\lambda_1 + \lambda_n}$$

Optimal $\rho = \max_{\lambda} |1 - \tau\lambda| \underset{\text{let us assume}}{=} 1 - \tau\lambda_1 = 1 - \frac{2\lambda_1}{\lambda_1 + \lambda_n}$

$$= \frac{\lambda_n - \lambda_1}{\lambda_n + \lambda_1} = \frac{\kappa - 1}{\kappa + 1} \qquad \kappa := \frac{\lambda_n}{\lambda_1} \text{ Condition Number}$$

The same result if we use λ_n.

$$\|e^k\| = \|(I - \tau A)e^{k-1}\| \leq \rho \|e^{k-1}\| \leq \left(\frac{\kappa - 1}{\kappa + 1}\right)^n \|e^0\|$$

So this is optimal "convergence". Note we will later derive a similar result for Krylov-methods.

Stopping criterias

When do we stop the iteration?

If we have u, then we can stop at $\frac{\|e^k\|_{\ell^2}}{\sqrt{n}} \leq \varepsilon \approx 10^{-16}$ or some other norm (infinity norm)

Normally we do not have u...

One option $\|u^{k+1} - u^k\|_Q \leq \varepsilon$ (+ scaling)

Can be bad!

4

$$\lambda_n = \frac{4}{h^2}\sin^2\left(\frac{n\pi h}{2}\right) = \frac{4}{h^2}\underbrace{\sin^2\left(\frac{n\pi}{2}\overbrace{\frac{1}{n+1}}^{=h}\right)}_{\approx 1} \approx \frac{4}{h^2}$$

$$K = \frac{\lambda_n}{\lambda_1} \approx \frac{4\pi^2}{h^2} \qquad K \sim \frac{1}{h^2} \sim n^2$$

$$\rho = \frac{K-1}{K+1}$$

n	10	100	1000
K	48	4134	406095
ρ	0,9595	0,99952	0,999995
it est	446	38073	3740273

$\|e^k\| = \rho^k \|e^0\|$

$\frac{\|e^k\|}{\|e^0\|} = \epsilon = 10^{-8}$

$k = \frac{\log \epsilon}{\log \rho}$ ← estimated number of iterations.

More general iterative solvers:

$$(*) \qquad u^k = u^{k-1} - L^{-1}(Au^{k-1} - b)$$

(we will later call L^{-1} the smoother

We use this formulation for analyzing purposes. The implementation will probably differ.

Note for Richardson $L^{-1} = \tau I$

Let $A = D - A_L - A_R$ (Left/right strictly triangular)

Jacobi iteration: $L^{-1} = D^{-1}$ (5)

$$u^k = u^{k-1} - D^{-1}(Au^{k-1} - b)$$
$$= u^{k-1} - D^{-1}((D - A_L - A_R)u^{k-1} - b)$$
$$= D^{-1}(A_L + A_R)u^{k-1} + D^{-1}b$$

Can be damped

Gauss-Seidel (forward): $L^{-1} = (D - A_L)^{-1}$

$$u^k = u^{k-1} - (D - A_L)^{-1}((D - A_L - A_R)u^{k-1} - b)$$
$$= (D - A_L)^{-1}(A_R u^{k-1} + b)$$

Gauss-Seidel (backward) $L^{-1} = (D - A_L)^{-1}$

$$u^k = (D - A_R)^{-1}(A_R u^{k-1} + b)$$

Symmetric Gauss-Seidel: Forward + Backward

the cost is 1.5 of a normal GS (allegedly)

Lemma Symmetric GS has the form (*) with $L^{-1} = (D - A_R)^{-1} D (D - A_L)^{-1}$. Note if A is symmetrix then L is symmetrix.

Proof Symmetrix GS is

$$u^{k-\frac{1}{2}} = (D - A_L)^{-1}(A_R u^{k-1} + b)$$
$$u^k = (D - A_R)^{-1}(A_L u^{k-\frac{1}{2}} + b)$$

which gives

$$u^k = (D - A_R)^{-1}\left[A_L(D - A_L)^{-1}(A_R u^{k-1} + b) + b\right]$$

We want $u^k = u^k - L^{-1}(Au^{k-1} - b)$

Multig grid methods

In lecture 4 we consider linear iterative methods like:
Richardson, Jacobi, Gauss-Seidel, SOR.

$$(*) \quad u^k = u^{k-1} - L^{-1}(Au^{k-1} - b)$$

They are slow if $\mathcal{K}$ is large, like for Poisson problem.

Recall the 1D Poisson: $\lambda_k = \frac{4}{h^2} \sin^2\left(\frac{k\pi h}{2}\right)$ and

$u_k = \sin(k\pi x) \qquad k = 1, 2, \ldots, n$

The error can be represented as a linear compination of u_k. If analysed we can see that the high frequency components (k is large) are reduced fast, while low frequency components (k is small) reduced very slowly.

Idea of Multigrid:

Use smoothers to reduse high frequency error. Then transfer the problem to a coarser mesh/grid. Use smoother on the coarser problem and continue this procedure.

Simple two-grid method.

$A_H u_H = b_H$ coarse

$A_h u_h = b_h$ fine

Smooth r-times $u_h^{k,m} = u_h^{k,m-1} - L_h^{-1}(A_h u_h^{k,m-1} - b_h)$

Restrict to coarse level $\cancel{R u_h^{k,r}}$ $m = 1,2,\dots,r$

$r_H := R(b_h - A_h u_h^{k,r})$

Solve for p_H $A_H p_H = r_H$

Prolongate and update solution $u_h^{k+1,0} = u_h^{k,r} + P p_H$

Geometric MG

Assume $A_H u_H = b_H$ and $A_h u_h = b_h$ results from FEM/IGA discretizations. I will use u_h for both functions and vectors (Note: Be careful)

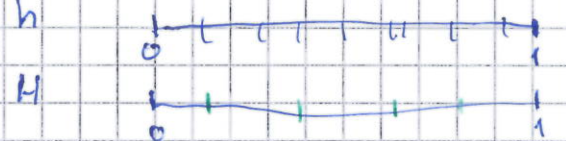
Let $u_h \in V_h$ and $u_H \in V_H$.

If $V_H \subset V_h$ (nested spaces) we can use the canonical embedding I_H^h as P and its transpose $I_h^H = (I_H^h)^T$ as R.

Example (IGA)

Assume we only have $A_h u_h = b_h$

We can find a coarse space by looking at the underlying grid (knot vector)



For tensor product B-splines $V_H \subset V_h$

We can also play around with multiplicity and spline degree haven choosen a coarser grid we can construct I_H^h (Knot insertion algorithm's)

we now need A_H. B_H

IF I_H^h and A_H are matrices we have

$$A_H = I_h^H A_h I_H^h$$ However this is expencive to evaluate.

Instead we assemble A_H on V_H.

Similar approach for FEM

Note normally the final grid/mesh is obained via refinement. In these cases we get the coarse grid/mesh for "Free".

Practical MG algorithm

we describe a 3-level method $h, 2h, 4h$

$\tilde{M} u_h^0 = \tilde{u}_h^r$ let u_h^0 be the initial guess.

$$u_h^0 \xrightarrow{S} u_h^r : \quad u_h^{k+1} = u_h^k - L_h^{-1}(A_h u_h^k - b_h) \quad k = 0, 1, \dots, r$$

3 $u_h^0 \xrightarrow{S} u_h^r$ $r_h = b_h - A_h u_h^r$ $\tilde{u}_h^0 = u_h^r + e_h$ $\tilde{u}_h^0 \xrightarrow{S} \tilde{u}_h^r$

$u_{2h}^0 = 0$
$b_{2h} = I_h^{2h} r_h$

$e_h = I_{2h}^h \tilde{u}_{2h}^r$

2 $u_{2h}^0 \xrightarrow{S} u_{2h}^r$ $r_{2h} = b_{2h} - A_{2h} u_{2h}^r$

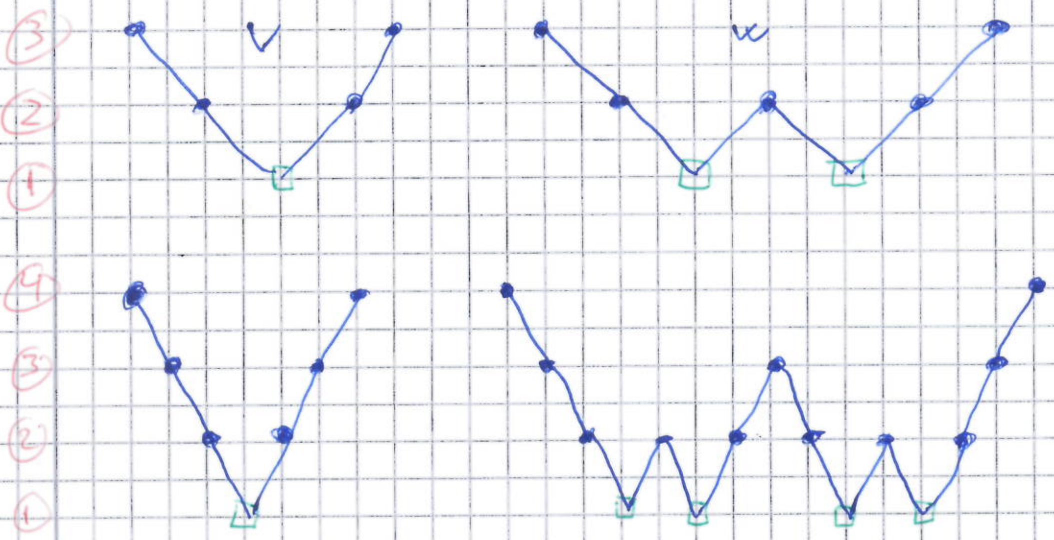
$\tilde{u}_{2h}^0 = u_{2h}^r + e_{2h}$, $\tilde{u}_{2h}^0 \xrightarrow{S} \tilde{u}_{2h}^r$

$b_{4h} = I_{2h}^{4h} r_{2h}$

$e_{2h} = I_{4h}^{2h} e_{4h}$

1 $e_{4h} = A_{4h}^{-1} b_{4h}$

What we saw was a V-cycle



Error of the two-grid method (Hackbusch)

$$e^1 = u - u^1 = (I - T_H)\, S_h^{\nu} \underbrace{(u - u^0)}_{e^0}$$

where $S_h = I - \tau L_h^{-1} A_h$ $\qquad$ $T_H = I_H^h A_H^{-1} A_h^H A_h$

Smoother $\qquad\qquad$ coarse-grid correction

Assume A and L are SPD

We show convergence in the L-norm i.e.

$$q := \|(I - T_H) S_h^{\nu}\|_{L_h} < 1$$

$$\|e^1\|_{L_h} \le q \|e^0\|_{L_h}$$

First we prove a usefull equality

M is a square matrix and L is SPD

$$\|M\|_L = \|L^{\frac{1}{2}} M L^{-\frac{1}{2}}\|$$

proof

$$\|M\|_L^2 = \left(\sup_{x \in \mathbb{R}^n} \frac{\|Mx\|_L}{\|x\|_L} \right)^2 = \sup_{x \in \mathbb{R}^n} \frac{\langle LMx, Mx \rangle}{\langle Lx, x \rangle} \quad \text{define } y := L^{\frac{1}{2}} x$$

$$= \sup_{y \in \mathbb{R}^n} \frac{\langle LML^{-\frac{1}{2}} y, ML^{-\frac{1}{2}} y \rangle}{\langle y, y \rangle} = \sup_{y \in \mathbb{R}^n} \frac{\|L^{\frac{1}{2}} M L^{-\frac{1}{2}} y\|^2}{\|y\|} = \|L^{\frac{1}{2}} M L^{-\frac{1}{2}}\|^2$$

Using this we get

$$\|(I - T_H) S_h^{\nu}\|_{L_h} = \|L_h^{\frac{1}{2}} (I - T_H) S_h^{\nu} L_h^{-\frac{1}{2}}\|$$

$$= \|L_h^{\frac{1}{2}} (I - T_H) \underbrace{A_h^{-1} L_h^{\frac{1}{2}} L_h^{-\frac{1}{2}} A_h}_{= I} S_h^{\nu} L_h^{-\frac{1}{2}}\|$$

$$\leq \underbrace{\|L_h^{\frac{1}{2}} (I - T_H) A_h^{-1} L_h^{\frac{1}{2}}\|}_{\leq C_A} \; \underbrace{\|L_h^{-\frac{1}{2}} A_h S_h^{\nu} L_h^{-\frac{1}{2}}\|}_{\leq C_S \nu^{-1}} \leq C_A C_S \nu^{-1}$$

Approximation property Smoothing property.

The idea is to prove these two inequalitys such that $C_A C_S$ is independent of n (h).

Lemma 1

$$\|(I-T_H)u_h\|^2_{L_h} \le C_A \|u_h\|_A^2 \quad \forall\, u_h \in V_h \qquad *$$

is equivalent to the approximation property.

Proof

Note $T_H = I_H^h A_H^{-1} I_h^H A_h$ is the A-ortogonal projector from V_h to V_H $\Rightarrow$

$$\left((I-T_H)A_h^{-1}\right)^T = (I-T_H)A_h^{-1} \qquad **$$

$$* \quad \sup_{u_h \in V_h} \frac{\|(I-T_H)u_h\|^2_{L_h}}{\|u_h\|_A^2} \overset{w_h := A^{\frac{1}{2}} u_h}{=} \sup_{w_h} \frac{\langle L_h (I-T_H) A_h^{-\frac{1}{2}} w_h, (I-T_H) A^{-\frac{1}{2}} w_h \rangle}{\|w_h\|^2}$$

$$= \sup_{w_h} \frac{\|\overbrace{L_h^{\frac{1}{2}}(I-T_H)A_h^{-\frac{1}{2}}}^{X_h :=} w_h\|^2}{\|w_h\|^2} \le C_A$$

$$\|X_h\| \le C_A^{\frac{1}{2}} \quad \Rightarrow \quad \|X_h X_h^T\| \le C_A$$

$$X_h X_h^T = L_h^{\frac{1}{2}}(I-T_H)A_h^{-1}(I-T_H)^T L_h^{\frac{1}{2}} \overset{**}{=} L_h^{\frac{1}{2}}(I-T_H)^{\cancel{2}} A_h^{-1} L_h^{-\frac{1}{2}} \quad \leftarrow \text{Projector}$$

$\square$