WS 2009/2010

Numerical Linear Algebra

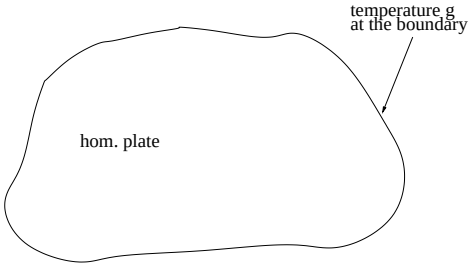
Professor Dr. Christoph Pflaum

Contents

1 Linear Equation Systems in the Numerical Solution of PDE's

1.1 Examples of PDE's

1. Heat Equation



Heat source f in the interior of the plate.

Question: What is the temperature inside of the plate?

Poisson Problem (P)

Let $\Omega \subset \mathbb{R}^n$ open, bounded, $f \in C(\overline{\Omega})$, $g \in C(\delta\Omega)$.

Find $u \in C^2(\overline{\Omega})$ such that

$$\begin{aligned} -\Delta u &= f \quad \text{on } \Omega \\ u\big|_{\delta\Omega} &= g \\ \text{where} \quad \Delta &= \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \end{aligned}$$

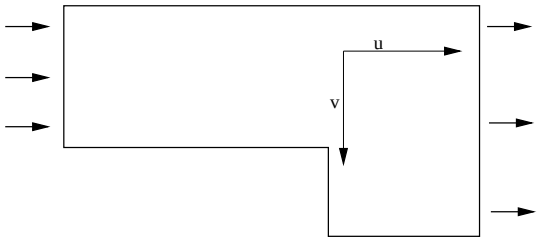
2. Convection-Diffusion-Problem

Find $u \in C^2(\overline{\Omega})$ such that

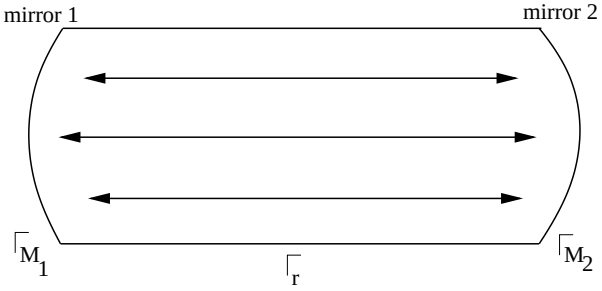
$$\begin{aligned} -\Delta u + \vec{b} \cdot \nabla u + cu &= f \quad \text{on } \Omega \\ u\big|_{\delta\Omega} &= 0 \\ \text{where} \quad \vec{b} &\in (C(\Omega))^2, \quad f, c \in C(\Omega) \end{aligned}$$

3. Navier-Stokes-Equation

$$\begin{aligned} \frac{\partial u}{\partial t} + \frac{\partial p}{\partial x} + \frac{\partial (u^2)}{\partial x} + \frac{\partial (uv)}{\partial y} &= \frac{1}{\mathrm{Re}} \Delta u \\ \frac{\partial u}{\partial t} + \frac{\partial p}{\partial y} + \frac{\partial (uv)}{\partial x} + \frac{\partial (v^2)}{\partial y} &= \frac{1}{\mathrm{Re}} \Delta v \\ \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} &= 0 \end{aligned}$$



4. Laser simulation



$$\Gamma_M = \Gamma_{M_1} \cup \Gamma_{M_2}$$

Find $u \in C^2_{\mathbb{C}}(\overline{\Omega}),\ \lambda \in \mathbb{C}$ such that

$$\begin{aligned} -\Delta u - k^2 u &= \lambda u \\ u\big|_{\Gamma_M} &= 0 \\ \frac{\partial u}{\partial \vec{n}}\Big|_{\Gamma_{\text{rest}}} &= 0 \qquad \text{(or boundary condition third kind)} \end{aligned}$$

We apply the ansatz

$$u = u_r e^{-i\tilde{k}z} + u_l e^{i\tilde{k}z}$$

where $\tilde{k}$ is an average value of k.
This leads to the equivalent eigenvalue problem:
Find u_r, u_l, λ such that

$$\begin{aligned} -\Delta u_r \ + \ 2i\tilde{k}\ \frac{\partial u_r}{\partial z} \ + \ (\tilde{k}^2 - k^2)u_r &= \lambda u_r \\ -\Delta u_l \ - \ 2i\tilde{k}\ \frac{\partial u_l}{\partial z} \ + \ (\tilde{k}^2 - k^2)u_l &= \lambda u_l \\ u_r + u_l\big|_{\Gamma_M} = 0, \qquad \frac{\partial u_r}{\partial z} - \frac{\partial u_l}{\partial z}\Big|_{\Gamma_M} &= 0 \\ \frac{\partial u_r}{\partial \vec{n}}\Big|_{\Gamma_{\text{rest}}} = \frac{\partial u_l}{\partial \vec{n}}\Big|_{\Gamma_{\text{rest}}} &= 0 \end{aligned}$$

1.2 Finite-Difference-Discretization of Poisson's Equation

Assume $\Omega =]0,1[^2$ and that an exact solution of (P) exists. We are looking for an approximate solution u_h of (P) on a grid Ω_h of meshsize h. Choose $h = \frac{1}{m}$ where $m \in \mathbb{N}$.

$$\begin{aligned} \Omega_h &= \big\{(ih, jh) \big| i, j = 1, \ldots, m-1\big\} \\ \overline{\Omega_h} &= \big\{(ih, jh) \big| i, j = 0, \ldots, m\big\} \end{aligned}$$

Discretization by Finite Differences:
Idea: Replace second derivative by difference quotient.
Let $e_x = (1,0)$ and $e_y = (1,0)$,

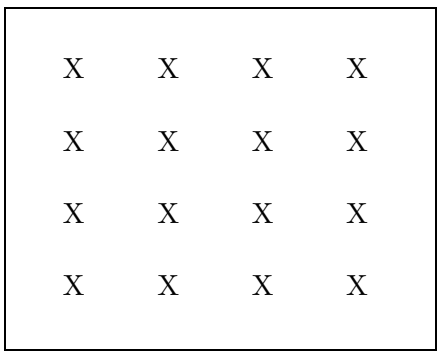
$$-\Delta u(z) = \left(-\frac{\partial^2 u}{\partial x^2} - \frac{\partial^2 u}{\partial y^2}\right)(z) = f(z) \quad \text{for } z \in \Omega_h$$

$$\begin{aligned} -\frac{u_h(z+he_x) - 2u_h(z) + u_h(z-he_x)}{h^2} & \\ -\frac{u_h(z+he_y) - 2u_h(z) + u_h(z-he_y)}{h^2} &= f(z) \end{aligned}$$

$$\begin{aligned} \text{and} \qquad u(z) &= g(z) & \\ \approx & = & \text{for } z \in \overline{\Omega_h} \backslash \Omega_h \\ u_h(z) &= g(z) & \end{aligned}$$

This leads to a linear equation system $L_h\, U_h = F_h$ where $U_h = (u_h(z))_{z \in \Omega_h}$, L_h is $|\Omega_h| \times |\Omega_h|$ matrix. The discretization can be described by the stencil

$$\begin{pmatrix} & -\frac{1}{h^2} & \\ -\frac{1}{h^2} & \frac{4}{h^2} & -\frac{1}{h^2} \\ & -\frac{1}{h^2} & \end{pmatrix} = \begin{pmatrix} m_{-1,1} & m_{0,1} & m_{1,1} \\ m_{-1,0} & m_{0,0} & m_{1,0} \\ m_{-1,-1} & m_{0,-1} & m_{1,-1} \end{pmatrix}$$



Let us abbreviate $U_{i,j} := u_h(ih, jh)$ and $f_{i,j} := f(ih, jh)$. Then, in case of $g = 0$, the matrix equation $L_h U_h = F_h$ is equivalent to:

$$\sum_{k,l=-1}^{1} m_{kl} U_{i+k,j+l} = f_{i,j}$$

1.3 FD Discretization for Convection-Diffusion

Let Ω, Ω_h as above.

$$-\Delta u + b\frac{\mathrm{d}u}{\mathrm{d}x} = f$$

Assume that b is constant.

1. Discretization by central difference:

$$\frac{\mathrm{d}u}{\mathrm{d}x}(z) \approx \frac{u_h(z + he_x) - u_h(z - he_x)}{2h}$$

   This leads to the stencil

$$\begin{pmatrix} & -\frac{1}{h^2} & \\ -\frac{1}{h^2} - \frac{b}{2h} & \frac{4}{h^2} & -\frac{1}{h^2} + \frac{b}{2h} \\ & -\frac{1}{h^2} & \end{pmatrix}$$

   $\rightarrow$ unstable for large b.

2. Upwind discretization:

$$\frac{\mathrm{d}u}{\mathrm{d}x}(z) \approx \frac{u_h(z) - u_h(z - he_x)}{h}$$

   This leads to the stencil

$$\begin{pmatrix} & -\frac{1}{h^2} & \\ -\frac{1}{h^2} - \frac{b}{h} & \frac{4}{h^2} + \frac{b}{h} & -\frac{1}{h^2} \\ & -\frac{1}{h^2} & \end{pmatrix}$$

1.4 Irreducible and Diagonal Dominant Matrices

Definition 1. *A $n \times n$ matrix A is called* strong diagonal dominant*, if*

$$|a_{ii}| > \sum_{i \neq j} |a_{ij}| \qquad 1 \leq i \leq n \tag{1}$$

A is called weak diagonal dominant, if there exists at least one i such that (1) holds and such that

$$|a_{ii}| \geq \sum_{i \neq j} |a_{ij}| \qquad 1 \leq i \leq n$$

Definition 2. *A is called reducible, if there exists a subset $J \overset{\neq}{\subset} \{1, 2, \ldots, n\}$, $J \neq \emptyset$. such that*

$$a_{ij} = 0 \qquad \text{for all } i \notin J,\ j \in J$$

A not reducible matrix is called irreducible.

Remark. An reducible matrix has the form

$$\begin{pmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{pmatrix}$$

→ The equation system separates in two parts.

Example:

1. Poisson FD:
   diagonal: $a_{ii} = \frac{4}{h^2}$
   non-diagonal: $a_{ij} = \begin{cases} -\frac{1}{h^2} & \text{if } i \text{ is N,S,W,O of } j \\ 0 & \text{else} \end{cases}$

   - A is not strong diagonal dominant, but weak diagonal dominant. To see this, consider a point i such that j is N of i. Then
   $$a_{ij} = \begin{cases} -\frac{1}{h^2} & \text{if } i \text{ is S,W,O of } j \\ 0 & \text{else} \end{cases}$$
   - A is irreducible.
   Proof: If A is reducible, then, $\{1, 2, ..., n\}$ is the union of two different sets of colored points, where one set is J. Then, there is a point $j \in J$ such that one of the points i=N,W,S,E is not contained in J, but i is contained in $\{1, 2, ..., n\}$. This implies $a_{j,i} \neq 0$. $\Rightarrow$ contradiction.

2. Convection-Diffusion-Equation

- centered difference

$$\begin{aligned} |a_{ii}| &= \frac{4}{h^2} \\ \sum_{i\neq j} |a_{ij}| &= \frac{4}{h^2} 2 \cdot \frac{1}{h^2} + \left(\frac{1}{h^2} + \frac{b}{2h}\right) + \left|\frac{1}{h^2} - \frac{b}{2h}\right| \\ &= 3\frac{1}{h^2} + \frac{b}{2h} + \left|\frac{1}{h^2} - \frac{b}{2h}\right| \end{aligned}$$

  Thus, $|a_{ii}| \geq \sum_{i\neq j} |a_{ij}|$, if and only if $\frac{1}{h^2} - \frac{b}{2h} \leq 0$.
  This shows $|a_{ii}| \geq \sum_{i\neq j} |a_{ij}|$, if and only if $h < \frac{2}{b}$

- upwind

$$\begin{aligned} |a_{ii}| &= \frac{4}{h^2} + \frac{b}{h} \\ &\geq \\ \frac{4}{h^2} + \frac{b}{h} &\geq \sum_{i\neq j} |a_{ij}| \quad \text{for all } h, b > 0 \end{aligned}$$

- Conclusion
  central: A is weak diagonal dominant if and only if $h < \frac{2}{b}$.
  upwind: A is weak diagonal dominant.
  A is irreducible in both cases.

Definition 3. *Let A be an $n \times n$ matrix. Consider n points $P_1, ..., P_n$. Draw an edge between $\overrightarrow{P_i, P_j}$ if $a_{i,j} \neq 0$. The directed graph of A is this set of points $P_1, ..., P_n$ with these edges $\overrightarrow{P_i, P_j}$.*

Definition 4. *A directed graph is called* strongly connected*, if for every pair of disjoint points P_i, P_j there exists a directed path in the graph. This means there exists a path $\overrightarrow{P_{i_0} P_{i_1}}, \overrightarrow{P_{i_2} P_{i_3}}, ..., \overrightarrow{P_{i_{r-1}} P_{i_r}}$ such that $P_{i_0} = P_i$ and$P_{i_r} = P_j$.*

Theorem 1. *A $n \times n$ matrix A is irreducible, if and only if its directed graph is connected.*

Proof. Let A be irreducible.
Let $1 \leq i_0 \leq n$ be an index. Let

$$J := \{j \mid \text{there is a directed path from } P_{i_0} \text{ to } P_j.\}.$$

J is not empty. Otherwise, $a_{i_0,j} = 0$ for every j and choosing $\tilde{J} = \{i_0\}$ would lead to a contradiction to A to be irreducible. Let us assume $J \neq \{1, 2, ..., n\}$.

Then, $a_{i,j} = 0$ for every $i \in J$ and $j \notin J$. Otherwise, there is a connected path from P_{i_0} to P_i and to P_j.

The above property of J is a contradiction to A irreducible.

Let the directed graph of A be connected.
Assume that A is reducible. Then, there are disjoint sets J, I such that $a_{i,j} = 0$ for every $i \in I, j \in J$. Let $P_{i_0}\overrightarrow{P_{i_1}}, P_{i_2}\overrightarrow{P_{i_3}}, ..., P_{i_{r-1}}\overrightarrow{P_{i_r}}$ be a directed path from $i_0 \in I$ to $i_r \in J$. Then, there must be a index s such that $i_{s-1} \in I$ and $i_s \in J$. This implies $a_{i_{s-1},i_s} \neq 0$. This is a contradiction to the properties of J and I. □

1.5 FE (Finite Element) Discretization

Definition 5. *$\mathcal{T} = \{T_1, \ldots, T_M\}$ is a conform triangulation of Ω if*

- $\overline{\Omega} = \bigcup_{i=1}^{M} T_i$, $\quad$ *T_i is triangle or square*
- *$T_i \cap T_j$ is either*
  - *empty or*
  - *one common corner or*
  - *one common edge.*

Remark.

- Let us write $\mathcal{T}_h$, if the diameter h_T of every element $T \in \mathcal{T}_h$ is less or equal h:

  $$h_T \leq h.$$

- A family of triangulations $\{\mathcal{T}_h\}$ is called quasi-uniform, if there exists a constant $\rho > 0$ such that the radius ρ_T of the largest inner ball of every triangle $T \in \mathcal{T}_h$ satisfies

  $$\rho_T > \rho h.$$

Definition.

- *Let $\mathcal{T}_h$ be a triangulation of Ω. Then, let V_h be the space of linear finite elements defined as follows:*

  $$\begin{aligned} V_h &= \left\{ v \in C^0(\overline{\Omega}) \;\middle|\; v\big|_T \text{ is linear for every } T \in \mathcal{T}_H \right\} \\ \overset{0}{V}_h &= V_h \cap H_0^1(\Omega) \end{aligned}$$

  $v\big|_T$ is linear means that $v\big|_T(x, y) = a + bx + cy$.

- *Let $\Omega =]0, 1[^2$, $h = \frac{1}{m}$ and*

  $$\mathcal{T}_h = \left\{ [ih, (i+1)h] \times [jh, (j+1)h] \;\middle|\; i, j = 0, \ldots, m-1 \right\}$$

The space of bilinear finite elements on Ω is defined as follows

$$V_h = \left\{ v \in C^0(\overline{\Omega}) \;\middle|\; v\big|_T \text{ is bilinear for every } T \in \mathcal{T}_H \right\}$$

$v\big|_T$ is bilinear means that $v\big|_T(x,y) = a + bx + cy + dxy$.

- *Let V_h be the space of linear or bilinear finite elements on $\mathcal{T}_h$ and $\mathcal{N}_h$ the set of corners of $\mathcal{T}_h$. Then, define the nodal basis function $v_p \in V_h$ at the point p by:*

$$v_p(x) = \begin{cases} 1 & \text{if } x = p \\ 0 & \text{if } x \neq p \end{cases} \qquad \text{for } x \in \mathcal{N}_h$$

Observe that

$$V_h = span \left\{ v_p \;\middle|\; p \in \mathcal{N}_h \right\}$$

This means that every function $u_h \in V_h$ can be represented as

$$u_h = \sum_{p \in \mathcal{N}_h} \lambda_p v_p$$

Finite Element Discretization of Poisson's equation:

$$\begin{aligned} -\Delta u &= f \\ u\big|_{\delta\Omega} &= 0 \end{aligned}$$

Thus, for every $v_h \in \overset{0}{V}_h$, we get:

$$\begin{aligned} -\Delta u\, v_h &= f\, v_h \\ &\Downarrow \\ \int_\Omega \nabla u\, \nabla v_h \, \mathrm{d}(x,y) + \int_\Gamma \frac{\partial u}{\partial \vec{n}}\, v_h \, \mathrm{d}(x,y) &= \int_\Omega f\, v_h \, \mathrm{d}(x,y) \\ &\Downarrow \\ \int_\Omega \nabla u\, \nabla v_h \, \mathrm{d}(x,y) &= \int_\Omega f\, v_h \, \mathrm{d}(x,y) \qquad \forall v_h \in \overset{0}{V}_h \end{aligned}$$

FE Discretization: Find $u_h \in \overset{0}{V}_h$ such that

$$\int_\Omega \nabla u\, \nabla v_h \, \mathrm{d}(x,y) = \int_\Omega f\, v_h \, \mathrm{d}(x,y) \qquad \forall v_h \in \overset{0}{V}_h \tag{2}$$

Stiffness matrix.

$$\begin{aligned} a_{p.q} &:= \int_\Omega \nabla v_p \, \nabla v_q \, \mathrm{d}(x,y), \qquad f_q := \int_\Omega f \, v_q \, \mathrm{d}(x,y) \\ A &:= (a_{p,q})_{p,q \in \overset{0}{\mathcal{N}}_h}, \qquad \overset{0}{\mathcal{N}}_h := \mathcal{N}_h \cap \Omega \\ u_h &= \sum_{p \in \overset{0}{\mathcal{N}}_h} \lambda_p \, v_p \end{aligned}$$

Then, (2) implies

$$\begin{aligned} \sum_{p \in \overset{0}{\mathcal{N}}_h} \lambda_p \int_\Omega \nabla v_p \, \nabla v_q \, \mathrm{d}(x,y) &= \int_\Omega f \, v_q \, \mathrm{d}(x,y) \qquad \text{for all } q \in \overset{0}{\mathcal{N}}_h \\ &\Downarrow \\ \sum_{p \in \overset{0}{\mathcal{N}}_h} \lambda_p \, a_{p,q} &= f_q \qquad \forall q \in \overset{0}{\mathcal{N}}_h \\ &\Downarrow \\ A \, U_h &= F_h \qquad \text{where} \quad \begin{array}{l} U_h = (\lambda_p)_{p \in \overset{0}{\mathcal{N}}_h} \\ F_h = (f_q)_{q \in \overset{0}{\mathcal{N}}_h} \end{array} \end{aligned}$$

The matrix A is called the stiffness matrix of the FE discretization.

1.6 Discretization Error and Algebraic Error

Let $||\cdot||$ be a suitable norm. Then, $||U_h - U||$ is called discretization error, with respect to this norm.

Example 1. *Poisson on a square*

- *FD,* $u \in C^4(\overline{\Omega})$*, then*

$$||U_h - U||_{L^\infty(\Omega_h)} = O(h^2)$$

- *FE,* $u \in H^2(\overline{\Omega})$*, then*

$$\begin{aligned} ||U_h - U||_{L^2(\Omega)} &= O(h^2) \\ ||U_h - U||_{H^1(\Omega)} &= O(h) \end{aligned}$$

Problem. The solution u_h cannot be calculated exactly, since L_h (or A) is a very large matrix and

$$A\, U_h = F_h.$$

Therefore, we need iterative solvers if $n > 10.000$ (or $n > 100.000$). By such an iterative solver, we get an approximation $\tilde{u}_h$ of u_h. $||\tilde{u}_h - u_h||$ is called algebraic error.

1.7 Basic Theory for LInear Iterative Solvers

Let A be a non singular $n \times n$ matrix and b a vector, $b \in \mathbb{R}^n$.

Problem:
Find $x \in \mathbb{R}^n$ such that $A\, x = b$.

A basic approach to construct an iterative solver is to use a decomposition

$$A = M - N$$

where M is a matrix, which is easy to invert (which can be inverted by a small number of operations). Then we get

$$\begin{aligned} M\, x &= N\, x + b \\ &\Downarrow \\ x &= M^{-1} N\, x + M^{-1} b \end{aligned}$$

By this formulas, we get the algorithm:

Algorithm:

Let x^0 be the start guess. Then
$x^{k+1} := M^{-1}N\, x^k + M^{-1}b$

Let us write the iteration formula as

$$x^{k+1} = C\, x^k + d,$$

where $C = M^{-1}N$ and $d = M^{-1}b$. This is the general form of a linear iterative solver.

Theorem 1. x^k converges to x for every start vector x^0 if and only if

$$\rho(C) < 1$$

Here $\rho(C)$ is the spectral radius of C,

$$\rho(C) = \max\left\{|\lambda| \middle| \lambda \text{ is eigenvalue of } C\right\}$$

(Observe the eigenvalues may be complex.)

Furthermore, the following convergence result holds:

$$||x^k - x|| \leq ||C^k||\; ||x^0 - x|| \tag{3}$$

If C is a normal matrix, then

$$||x^k - x||_2 \leq (\rho(C))^k\; ||x^0 - x||_2 \tag{4}$$

There exist start vectors x^0, such that the equal sign holds in the above inequality.

Proof. By $x^{k+1} = C\, x^k + d$ and $x = C\, x + d$, we get

$$x^{k+1} - x = C\left(x^k - x\right)$$

This implies

$$x^k - x = C^k(x^0 - x) \tag{5}$$

This implies (3).
Let us assume, that $x^0 - x = e$ is an eigenvector of C with eigenvalue λ such that

$$|\lambda| = \rho(C)$$

Then, we get

$$||x^k - x|| = \rho(C)^k ||x^0 - x||$$

This shows:

- if $\rho(C) \geq 1$, then x^k does not converge to x.
- the equal sign holds in equation (4).

Now, let us assume that x^0 is a general start vector. Let us assume $\rho(C) < 1$. We want to prove $\lim_{k\to\infty} x^k = x$. By (3), it is enough to prove

$$||C^k|| \to 0 \qquad \text{for } k \to \infty$$

Since all norms are equivalent in a finite dimensional vector space, it is enough to show this for the $\|\cdot\|_2$-norm.

1. C is normal. Then, there exists a unitary matrix T such that

   $$T^{-1} D\, T = C$$

   where $D = \text{diag}(\lambda_1, \ldots, \lambda_n)$ is the diagonal matrix of eigenvalues. Then, we get

   $$||C^k||_2 = ||T^{-1} D^k T||_2 \leq ||T^{-1}||_2\, ||T||_2\, ||D^k||_2 = ||D^k||_2 = \rho(C)^k.$$

   This shows (4).

2. C is a general matrix. Then, we have to apply the Jordan decomposition

   $$T^{-1} J\, T = C$$

   Then, we get

   $$||C^k|| \leq ||T^{-1}||\, ||T||\, ||J^k||$$

   Thus, it is enough to show

   $$\lim_{k\to\infty} ||J^k|| = 0$$

   It is enough to study an Jordan block

   $$\tilde{J} = \lambda\, E + N,$$

   where E is the unit matrix and

   $$N = \left.\begin{pmatrix} 0 & 1 & 0 & \\ & \ddots & \ddots & \\ & & \ddots & 1 \\ 0 & & & 0 \end{pmatrix}\right\} \text{ s rows}$$

Since $\rho(C) < 1$, it follows $|\lambda| < 1$. A short calculation shows

$$\begin{aligned} ||N^i|| &\leq 1 \qquad \text{for all } i \\ N^s &= 0 \end{aligned}$$

Since $N\ E = E\ N$ it follows:

$$\begin{aligned} ||\tilde{J}^k|| &= ||\sum_{i=0}^{k} \binom{k}{i} (\lambda E)^{k-i}\, N^i|| \leq \\ &\leq \sum_{i=0}^{s-1} \binom{k}{i} \lambda^{k-i} \leq \\ &\leq s\ k^s\ \lambda^{k-s} = \\ &= (s\ \lambda^{-s})\ k^s\ \lambda^k \to 0 \qquad \text{for } k \to \infty \end{aligned}$$

NR: $$\binom{k}{i} = \frac{k(k-1)\cdot \ldots \cdot (k-i+1)}{i!} \leq k^i \leq k^s$$

$$\frac{(k+1)^s \lambda^{k+1}}{k^s \lambda^k} = \left(1 + \frac{1}{k}\right)^s \lambda \leq \frac{\lambda + 1}{2} < 1 \qquad \text{for large } k$$

1.8 Effective Convergence Rate

In several applications one would like to know, how many iterations s are needed to reduce the algebraic error by a certain factor. Let us assume that this factor is $\frac{1}{2}$. Thus, we would like to know how many iterations s are needed to obtain

$$\|x^s - x\| \leq \frac{1}{2}\|x^0 - x\|.$$

To this end , let us assume that there is an estimation

$$\|x^s - x\| \leq \rho^s \|x^0 - x\|.$$

In case of a linear iteration method with symmetric iteration matrix C, we can choose $\rho = \rho(C)$.

Obviously,

$$s = \frac{\ln\left(\frac{1}{2}\right)}{\ln(\rho)}$$

since $\rho(C)^s = \frac{1}{2}$. s and $\rho(C)$ are not the effective convergence rate. To estimate the effective convergence rate , the computational amount has to

be included. Let Op the number of operations for one iteration. Then, the effective convergence rate is:

$$G_{eff} := s \cdot \frac{Op}{\text{number of unknowns}} = \frac{\ln \frac{1}{2}}{\ln(\rho)} \cdot \frac{Op}{n}$$

Example 2. *Gauss elimination*

$$G_{eff} = O(n^2)$$

1.9 Jacobi and Gauss-Seidel Iteration

The Jacobi-iteration is a „one-step“ method. The Gauss-Seidel-iteration is a successive relaxation method.

1.9.1 Ideas of Both Methods

Relaxation of the i-th unknown x_i:
Correct x_i^{old} by x_i^{new} such that the i-th equation of the equation system

$$A \cdot x = b$$

is correct.

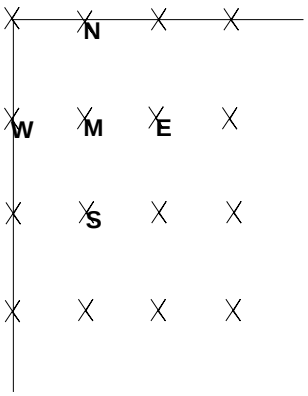
Jacobi-iteration:
„Calculate the relaxations simultaneously for all $i = 1, \ldots, n$“
This means: If $x^{old} = x^k$, then
let $x^{k+1} = x^{new}$

Gauss-Seidel-iteration:
„Calculate relaxation for $i = 1, \ldots, n$ and use the new values“
This means: $x^{old,1} = x^k$
Iterate for $i = 1, \ldots, n$:
Calculate $x^{new,i}$ by relaxation of the i-th component
Put $x^{old,i+1} = x^{new,i}$
$x^{k+1} = x^{new,n}$

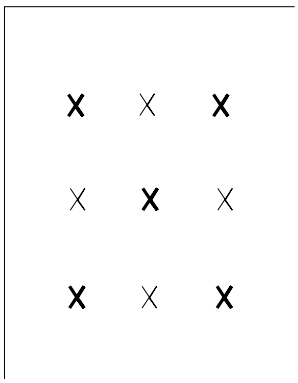
Remark.

- Jacobi-iteration is independent of the numbering of the grid points
- The convergence rate of the Gauss-Seidel iteration depends on the numbering of the grid points

Example 3. *Model problem, FD for Poisson*

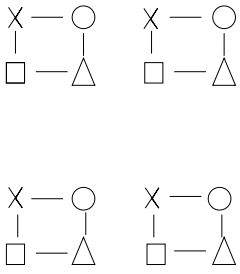


$$u_M^{new} = \frac{1}{4}\left(u_N^{old} + u_S^{old} + u_E^{old} + u_W^{old}\right) + f_M$$



red-black Gauss-Seidel

A four color Gauss-Seidel-relaxation is used for a 8-point stencil



$$\begin{matrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{matrix}$$

- better relaxation property
- after relaxation of one color all equations at those points are correct

Relaxation for the Convection-Diffusion:
A convection-diffusion problem is a so-called singular perturbed problem.
To see this write the convection-diffusion problem in the form:

$$-\epsilon\Delta u + \frac{\partial u}{\partial x} = \tilde{f} \quad , \quad \epsilon > 0$$

$\epsilon \to 0$ is the difficult case.

(Hackbusch's) rule for relaxing singular perturbed problems: Construct the iteration such that it is an exact solver for $\epsilon = 0$

For $\epsilon = 0$ we get the stencil (for upwind FD):

$$\begin{pmatrix} & 0 & \\ -\frac{1}{h} & \frac{1}{h} & 0 \\ & 0 & \end{pmatrix}$$

Thus a Gauss-Seidel relaxation with a numbering of the grid points from left to right leads to an exact solver

1	2	3
4	5	6
7	8	9

This can be done also for more complicated convection directions. Exception: Circles!

1.9.2 Description of Jacobi and Gauss-Seidel Iteration by Matrices

Let A be a $n \times n$ matrix. Decompose $A = D - L - R =$

$$\begin{pmatrix} * & 0 & & \dots & 0 \\ 0 & * & \ddots & & \vdots \\ \vdots & & \ddots & & \\ & & \ddots & * & 0 \\ 0 & \dots & & 0 & * \end{pmatrix} - \begin{pmatrix} 0 & 0 & & \dots & 0 \\ * & 0 & & & \vdots \\ \vdots & \ddots & \ddots & & \\ & & \ddots & 0 & 0 \\ * & \dots & & * & 0 \end{pmatrix} - \begin{pmatrix} 0 & * & & \dots & * \\ 0 & 0 & & & \vdots \\ \vdots & \ddots & \ddots & & \\ & & \ddots & 0 & * \\ 0 & \dots & & & 0 \end{pmatrix}$$

Let $x_0 \in \mathbb{R}^d$ be a start vector.

Jacobi-iteration

$$D\,x^{k+1} - (L+R)\,x^k = b \qquad \Rightarrow$$
$$x^{k+1} = D^{-1}\,(L+R)\,x^k + D^{-1}b$$

Decomposition: $A = D - (L+R) = M - N$
Thus, the iteration matrix is

$$C_J = D^{-1}\,(L+R)$$

Gauss-Seidel-iteration

$$(D-L)\,x^{k+1} - R\,x^k = b \qquad \Rightarrow$$
$$x^{k+1} = (D-L)^{-1}\,R\,x^k + (D-L)^{-1}\,b$$

Thus, the iteration matrix is

$$C_{GS} = (D-L)^{-1}\,R$$

1.10 Convergence Rate of Jacobi and Gauss-Seidel Iteration

1.10.1 General theory for weak dominant matrices

1.10.2 Special theory for upwind FD

1.10.3 FE analysis, variational approach

1.10.4 Eigenvector, eigenvalue analysis: „Fourier-analysis“

1.10.1 General Theory for Weak Dominant Matrices

Theorem 1.

1. *Assume that A is weak diagonal dominant and irreducible. Then, Jacobi and Gauss-Seidel iteration converge.*

2. *Assume that A is diagonal dominant. Then, the following estimate for the convergence rate holds:*

$$\rho \leq \max_{1 \leq i \leq n} \frac{1}{|a_{ii}|} \sum_{j \neq i} |a_{ij}| < 1$$

Proof. Let x be an eigenvector of C with eigenvalue λ, where $|\lambda| = \rho(C)$. Furthermore assume $||x||_\infty = 1$.
Assume that A is weak diagonal dominant.

- In case of the Jacobi iteration: $C = D^{-1}(L + R)$

$$|(Cx)_i| \leq \frac{1}{|a_{ii}|} \sum_{j \neq i} |a_{ij}| \, |x_j| \leq \left\{ \frac{1}{|a_{ii}|} \sum_{j \neq i} |a_{ij}| \right\} \, ||x||_\infty \leq 1 \qquad (6)$$

  This shows $||Cx||_\infty \leq 1$. Since x eigenvector with eigenvalue λ, it follows

$$1 \geq ||Cx||_\infty = ||\lambda x||_\infty = |\lambda| = \rho(C)$$

- In case of the Gauss-Seidel iteration:

$$C = (D - L)^{-1}R \Rightarrow (D - L)C = R \Rightarrow C = D^{-1}(LC + R)$$

Let us prove by induction $|(Cx)_i| \leq 1$ for $i = 1, \ldots, n$

$$\begin{aligned} |(Cx)_i| &\leq \frac{1}{|a_{ii}|} \left\{ \sum_{j<i} |a_{ij}| \, |(Cx)_j| + \sum_{j>i} |a_{ij}| \, |x_j| \right\} \qquad (7) \\ &\leq \frac{1}{|a_{ii}|} \left\{ \sum_{j \neq i} |a_{ij}| \right\} \leq 1 \end{aligned}$$

Analogously, we get $\rho(C) \leq 1$.

If A is diagonal dominant, then similar calculations show

$$|(Cx)_i| \leq \frac{1}{|a_{ii}|} \left\{ \sum_{j \neq i} |a_{ij}| \right\} < 1$$

This implies

$$||Cx||_\infty \leq \max_i \frac{1}{|a_{ii}|} \sum_{j \neq i} |a_{ij}| < 1$$

which shows

$$\rho(C) \leq \max_i \frac{1}{|a_{ii}|} \sum_{j \neq i} |a_{ij}| < 1$$

This completes the proof of 2.

Let us assume that A is irreducible and weak diagonal dominant. Let

$$J = \left\{ i \in \mathbb{N} \middle| 1 \leq i \leq n, |x_i| = 1 \right\}$$

Proof by contradiction. Assume $\rho = 1$. Then for all $i \in J$, the equal signs hold for all inequalities in (6),(7). This shows for every $i \in J$:

$$|a_{ij}| = 0 \qquad \text{if } j \notin J \qquad (8)$$

($j \notin J$ means $|x_j| < 1$)
By assumption, there is a i_0 such that the equal sign does not hold in (6),(7). This means

$$1 > |(Cx)_{i_0}| = |\rho x_{i_0}| = |x_{i_0}|$$

This shows that J is a real subset of $\{1, \ldots, n\}$. J is not the empty set, since $\|x\|_\infty = 1$.
This is a contradiction to A irreducible. □

Example. By the examples in 1.4, Gauss-Seidel iteration and Jacobi iteration converge for Poisson problem and convection-diffusion problem and FD upwind. But: no estimation of the convergence rate.

1.10.2 Special Theory for the FD-Upwind

Definition. *Let us assume, that $q > 0$. Then define the upwind norm*

$$||x||_{up,q} := \max_i |q^i x_i|$$

Theorem 2. *Assume that*

$$\sum_{j<i} \frac{|a_{ij}|}{|a_{ii}|} q^{i-j} < 1.$$

Then,

$$\rho(C_{GS}) \leq \max_{i=1}^{n} \frac{\sum_{j>i} \left|\frac{a_{ij}}{a_{ii}}\right| q^{i-j}}{1 - \sum_{j<i} \left|\frac{a_{ij}}{a_{ii}}\right| q^{i-j}}$$

Proof. Let us assume, that

$$||x||_{up,q} = 1 \qquad \Rightarrow |q^i x_i| \leq 1 \quad \forall i$$

Assume, that x is eigenvector of C_{GS} with eigenvalue λ such that $|\lambda| = \rho(C_{GS})$. Choose i such that $|q^i x_i| = 1$. Then $C_{GS} = (D-L)^{-1}R \Rightarrow C_{GS} = D^{-1}(LC_{GS} + R)$

$$\begin{aligned}
|\lambda| q^{-i} &= |\lambda|\,|x_i| = |(\lambda x)_i| = |(C_{GS}(x))_i| = \left|(D^{-1}(LC_{GS}x + Rx))_i\right| \leq \\
&\leq \frac{1}{|a_{ii}|}\left\{\sum_{j<i} |a_{ij}||(C_{GS}x))_j| + \sum_{j>i} |a_{ij}||x_j|\right\} \leq \\
&\leq \frac{1}{|a_{ii}|}\left\{\sum_{j<i} |a_{ij}||(\lambda x))_j| + \sum_{j>i} |a_{ij}||x_j|\right\} \leq \\
&\leq \frac{1}{|a_{ii}|}\left\{\sum_{j<i} |a_{ij}||\lambda| q^{-j} + \sum_{j>i} |a_{ij}| q^{-j}\right\} \\
&\Downarrow \\
&\quad |\lambda|\left(1 - \sum_{j<i} \frac{|a_{ij}|}{|a_{ii}|} q^{i-j}\right) \leq \sum_{j>i} \frac{|a_{ij}|}{|a_{ii}|} q^{i-j}
\end{aligned}$$

This completes the proof. □

Example. FD - 1D convection diffusion

$$\begin{aligned}
-u'' + bu' &= f \qquad \text{on } [0,1],\ b > 0 \\
u(0) = u(1) &= 0
\end{aligned}$$

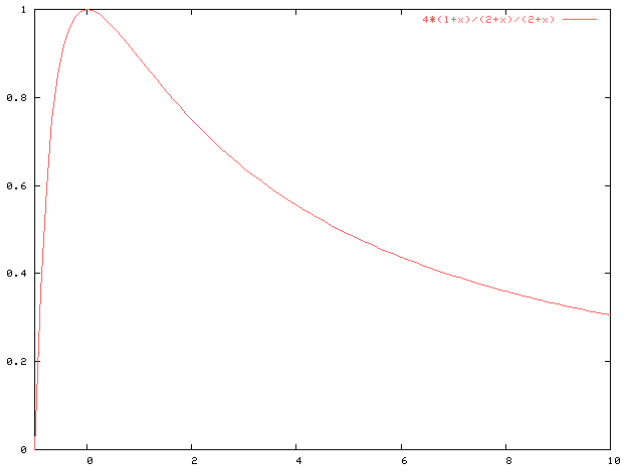


Figure 1: Estimation of the spectral radius of upwind Gauss-Seidel in 1D

Stencil of upwind discretization

$$\left(\; -\tfrac{1}{h^2} - b\tfrac{1}{h} \quad \tfrac{2}{h^2} + b\tfrac{1}{h} \quad -\tfrac{1}{h^2} \; \right) = \frac{1}{h^2} \left(\; -1-s \quad 2+s \quad -1 \; \right)$$

where $s = bh$. Normalized coefficients

$$\begin{aligned} a_{ii} &= 2+s \\ a_{ij} &= \begin{cases} -1 & \text{if } j = i+1 \\ -(1+s) & \text{if } j = i-1 \\ 0 & \text{else} \end{cases} \end{aligned}$$

Let us number the grid points from left to right. Then, for $-1 \leq s$, we get

$$\rho = \frac{\frac{1}{2+s} q^{-1}}{1 - \frac{1+s}{2+s} q} = \frac{1}{2+s} \frac{1}{q - q^2 \frac{1+s}{2+s}}$$

$f(q) = q - q^2 \frac{1+s}{2+s}, f'(q) = 1 - 2q\frac{1+s}{2+s} \stackrel{!}{=} 0 \Rightarrow q_0 = \frac{1}{2}\frac{2+s}{1+s} \Rightarrow f(q_0) = \frac{1}{4}\frac{2+s}{1+s}$.
Observe $1 - \frac{1+s}{2+s} q\big|_{q=q_0} = \frac{1}{2} > 0$. For $-1 \leq s$:

$$\begin{aligned} \rho(C_{GS}) &\leq \frac{1}{2+s} \frac{1}{\frac{1}{4}\frac{2+s}{1+s}} = 4\frac{1+s}{(2+s)^2} \\ \lim_{s\to\infty} 4\frac{1+s}{(2+s)^2} &= 0 \\ 4\frac{1+s}{(2+s)^2}\bigg|_{s=0} &= 1 \\ 4\frac{1+s}{(2+s)^2}\bigg|_{s=-1} &= 0 \end{aligned}$$

The function $4\frac{1+s}{(2+s)^2}$ is depicted in Figure 1.

2D case

$$||(x_{i,j})||_{up,(q_i)} := \max_{(i,j)} \left| q^i x_{(i,j)} \right|$$

Theorem 3. *Assume that*

$$\sum_{k<i} \frac{|a_{(i,j),(k,l)}|}{|a_{(i,j),(i,j)}|} q^{i-k} < 1$$

Then

$$\rho(C_{GS}) \leq \max_{(i,j)} \frac{\sum_{(k,l)\neq(i,j),k\geq i} \left| \frac{a_{(i,j),(k,l)}}{a_{(i,j),(i,j)}} \right| q^{i-k}}{1 - \sum_{k<i} \left| \frac{a_{(i,j),(k,l)}}{a_{(i,j),(i,j)}} \right| q^{i-k}}$$

Example. FD - 2D convection diffusion

$$-\Delta u + b\frac{\partial u}{\partial x} = f$$

Stencil:

$$\frac{1}{h^2} \begin{pmatrix} & -1 & \\ -1-s & 4+s & -1 \\ & -1 & \end{pmatrix}$$

$$\begin{aligned} a_{(i,j),(i,j)} &= 4+s \\ a_{(i,j),(k,l)} &= \begin{cases} -1 & \text{if } k \geq i, (i,j) \neq (k,l) \\ -(1+s) & \text{if } k < i \end{cases} \end{aligned}$$

Let us number the grid points first from left to right and then from down to up. Then, for $-1 \leq s$, we get:

$$\rho \leq \frac{\frac{1}{4+s}(1+1+q^{-1})}{1-\frac{1}{4+s}q(1+s))} = \frac{\frac{1}{4+s}(2+q^{-1})}{1-\frac{q(1+s)}{4+s}}$$

Numerically, one can calculate an optimal parameter q such that $\frac{\frac{1}{4+s}(2+q^{-1})}{1-\frac{q(1+s)}{4+s}}$ is as small as possible. The resulting estimation of the convergence rate is depicted in Figure 2.

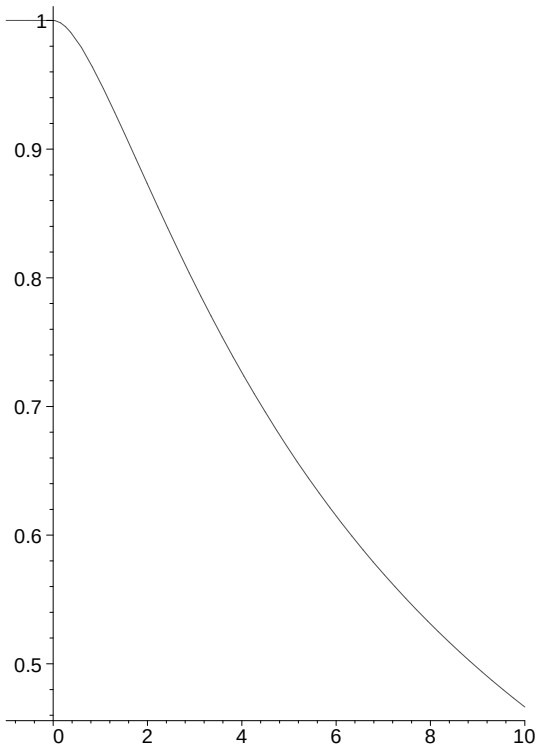


Figure 2: Estimation of the spectral radius of upwind Gauss-Seidel in 2D

1.10.3 FE analysis, Variational approach

We want to solve the following problem:

Find $u_h \in V_h$ such that

$$a(u_h, v_h) = f(v_h) \quad \forall v_h \in V_h \tag{9}$$

where V_h is the space of *bilinear finite elements*.

For example,

$$a(u_h, v_h) = \int_\Omega \nabla u_h \nabla v_h d(x,y) \quad \text{and} \quad f(v_h) = \int_\Omega f_s v_h d(x,y) \tag{10}$$

Relaxation
Let $(V_p)_{p\in\Omega_h}$ be the nodal basis of V_h. Now what means a *relaxation* step?

Let u_h^{old} be an old approximation. Then, let $\mu \in \mathbb{R}$ such that,

$$a(u_h^{old} + \mu v_p, v_p) = f(v_p) \quad \text{and let} \tag{11}$$

$$u_h^{new} = u_h^{old} + \mu v_p \tag{12}$$

This is the relaxation at the grid point p. Furthermore, we can calculate μ in the following way:

$$\mu = \frac{1}{a(v_p, v_p)} \Big(f(v_p) - a(u_h^{old}, v_p) \Big) \tag{13}$$

$$u_h^{new} = u_h^{old} + v_p \frac{1}{a(v_p, v_p)} \Big(f(v_p) - a(u_h^{old}, v_p) \Big) \tag{14}$$

Implementation by EXPDE
Let f_s be the vector describing the right hand side and let

$$f(v) = \int_\Omega f_s v d(x,y) \quad \text{and} \quad a(u,v) = \int_\Omega \nabla u \nabla v d(x,y).$$

The operator corresponding to $a(v,v)$ is the Laplace operator. Now define the variables as

```
Variable f(grid);
Variable u(grid);
Variable f_s(grid);
Variable v(grid);

f=Helm_FE(f_s);
```

Now, for the *Jacobi* method, we get

```
nu = (f-Laplace_{FE}(u)) / Diag_Laplace_FE();
u = u+nu ;
```

and for the *Gauss–Seidel* method we have

```
u = u + (f-Laplace_{FE}(u)) / Diag_Laplace_FE();
```

Lemma 1 (Variational approach)**.** *Let us assume that $a(u, v)$ is symmetric positive definite. Then*
Find $u_h \in V_h$ such that

$$a(u_h, v_h) = f(v_h) \quad \textit{for all} \quad v_h \in V_h \tag{15}$$

is equivalent to

$$u_h \in V_h \textit{ minimizes} \quad \frac{1}{2}a(v_h, v_h) - f(v_h) \quad \textit{for } v_h = u_h. \tag{16}$$

Proof.

Let us assume that u_h satisfies (15). Then, we have to show that

$$\mu \longrightarrow \frac{1}{2}a(u_h + \mu v_h, u_h + \mu v_h) - f(u_h + \mu v_h) = h(\mu) \tag{17}$$

has a minimum at $\mu = 0$ for every $v_h \in V_h$. We achieve this by differentiating (15).

$$\begin{aligned} 0 \quad = h'(\mu) \quad &= \mu a(v_h, v_h) + a(u_h, v_h) - f(v_h) \\ &= \mu a(v_h, v_h) + a(u_h, v_h) - a(u_h, v_h)) \\ &= \mu a(v_h, v_h) \\ &\Rightarrow \mu = 0 \end{aligned}$$

By differentiating the above equation again, we get

$$h''(\mu) = a(v_h, v_h) \Rightarrow h''(\mu) > 0$$

Let us assume that u_h satisfies (16), then $h(\mu) \mid_{\mu=0} = 0$ implies that $a(u_h, v_h) = f(v_h)$. □

A similar calculation shows that the problem

$$\text{Find } \mu \in \mathbb{R} \text{ such that } a(u_h^{old} + \mu v_p, v_p) = f(v_p)$$

is equivalent to

$$\text{Minimize } \tfrac{1}{2} a(u_h^{old} + \mu v_p, u_h^{old} + \mu v_p) - f(u_h^{old} + \mu v_p) \text{ for all } \mu \in \mathbb{R}$$

Conclusion
A relaxation step is a minimizing step.

Gauss–Seidel minimizes $\frac{1}{2} a(\hat{u}_h, \hat{u}_h) - f(\hat{u}_h)$ in several directions. Therefore *divergence* is very unlikely, if $a(u, y)$ is symmetric positive definite.

Example 4. *Let τ_h be a triangulation of a given polygon domain. Then, discretize Poisson's equation by finite elements on this triangulation. A Gauss-Seidel iteration with respect to the nodal basis converges. However the corresponding stiffness matrix is not diagonal dominant in general.*

Example 5 (Linear Elasticity)**.** *Let $E > 0$ and $0 < \nu < \frac{1}{2}$. Define the symmetric derivative*

$$\begin{aligned} \epsilon_{ij} &:= \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) \\ Du &:= \begin{pmatrix} \epsilon_{11} \\ \epsilon_{22} \\ \epsilon_{33} \\ \epsilon_{12} \\ \epsilon_{13} \\ \epsilon_{23} \end{pmatrix} \end{aligned}$$

and the matrix

$$\mathcal{C}^{-1} \frac{1}{E} \begin{pmatrix} 1 & -\nu & -\nu & & & \\ -\nu & 1 & -\nu & & 0 & \\ -\nu & -\nu & 1 & & & \\ & & & 1+\nu & & \\ & 0 & & & 1+\nu & \\ & & & & & 1+\nu \end{pmatrix},$$

where E and ν are physical constants. The bilinear form corresponding to the problem of linear elasticity is

$$\begin{aligned} a : (H^1(\Omega))^3 \times (H^1(\Omega))^3 &\to \mathbb{R} \\ (u, v) &\mapsto \int_\Omega (Du)^T \mathcal{C} Dv \; d(x, y, z) \end{aligned}$$

Using suitable boundary conditions, this matrix is symmetric positive definite. Thus, Gauss-Seidel iteration with respect to the nodal basis on the finite element grid converges.

1.10.4 Analysis of the Convergence of the Jacobi Method

Model problem : Finite difference discretization of the Poisson equation

Both, Gauss–Seidel and the Jacobi method, converge if the coefficient matrix L_h of the finite difference discretization is

- weak diagonally dominant and
- irreducible.

Now, we want to estimate the convergence rate in more detail for Poisson's equation.

To solve the linear system $Ax = b$, the iteration matrix of the Jacobi method is $C_J = D^{-1}(L + R)$. Then, for the model problem, we have

$$A = D - L - R \Longrightarrow C_j = D^{-1}(D - A) = -D^{-1}A + E + E - \frac{h^2}{4}A = E - \frac{h^2}{4}L_h \tag{18}$$

It follows from exercise (1.2) that

$$C_j e_{\nu\mu} = \left(1 - \frac{h^2}{4}\lambda_{\nu\mu}\right) e_{\nu\mu}. \tag{19}$$

Here $\lambda_{\nu,\mu}$ are the eigenvalues

$$\lambda_{\nu,\mu} = \frac{4}{h^2}\left(\sin^2\left(\frac{\pi\nu h}{2}\right) + \sin^2\left(\frac{\pi\mu h}{2}\right)\right)$$

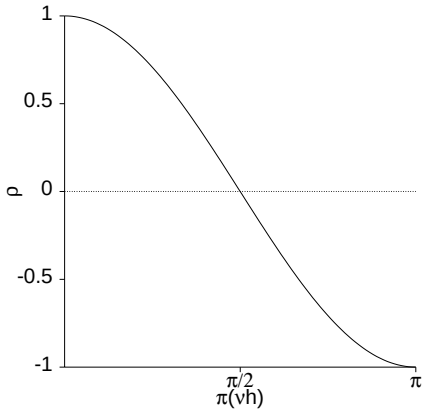
for $\nu, \mu = 1 \ldots (m-1)$, where $h = \frac{1}{m}$. Thus, the iteration matrix C_j has the eigenvalues

$$(\rho_J)_{\nu\mu} = 1 - \sin^2\left(\frac{\pi\nu h}{2}\right) - \sin^2\left(\frac{\pi\mu h}{2}\right) \tag{20}$$

Here, J denotes the Jacobi method. In case of $\nu = \mu$ we have,

$$(\rho_J)_{\nu\nu} = 1 - \sin^2\left(\frac{\pi\nu h}{2}\right) - \sin^2\left(\frac{\pi\nu h}{2}\right) = 1 - 2\sin^2\left(\frac{\pi\nu h}{2}\right) = \cos(\pi\nu h) \quad (21)$$

The following graph depicts the eigenvalues $(\rho_J)_{\nu\nu}$ with respect to the parameter $\pi\nu h$ in (21).



$\Longrightarrow$ Bad convergence for high and low frequencies.

$\Longrightarrow$ Good convergence for middle frequencies.

In particular, one can show that the spectral radius of the iteration matrix is

$$\rho(C) = 1 - O(h^2) \quad (22)$$

Now, the effective convergence rate for the Jacobi method is

$$G_{eff} = sOp(c)/n = \frac{\ln(\frac{1}{2})}{\ln(\rho(c))}.\frac{n}{n} = O(h^{-2}) = O(n). \quad (23)$$

$\Longrightarrow$ The convergence rate for the Jacobi method $(O(n))$ is better than that of the direct solver the Gauss elimination $\big(O(n^2)\big)$.

1.10.5 Iteration Method with Damping Parameter

Let us assume that $x^k \longrightarrow x^{k+1}$ is an iteration. The iteration can be written as $x^k \longrightarrow x^k + (x^{k+1} - x^k)$. The term $(x^{k+1} - x^k)$ can be treated as a correction term. Now a damped iteration is $x^k \longrightarrow \omega(x^{k+1} - x^k)$, where

- ω is called the damping factor or the relaxation parameter and $\omega \in]0, 2[$.

- $\omega > 1$ is called over relaxation.

- $\omega < 1$ is called under relaxation.

SOR(Successive Over Relaxation) method is obtained by performing the Gauss-Seidel method with over relaxation. But SOR has disadvantages for e.g like,

- It is very difficult to find ω for certain class of problems.

1.10.6 Damped Jacobi Method

The Jacobi method with relaxation parameter $\omega = 1$ is

$$x_{Jacobi}^{k+1} = D^{-1}(L + R)x_{Jacobi}^{k} + D^{-1}b \tag{24}$$

The Jacobi method with damping parameter ω is

$$\begin{aligned} x_{\omega}^{k+1} &= x_{\omega}^{k} + \omega(D^{-1}(L + R)x_{\omega}^{k} + D^{-1}b - x_{\omega}^{k}) \\ &= \left\{E(1 - \omega) + \omega D^{-1}(L + R)\right\} x_{\omega}^{k} + \omega D^{-1}b \qquad (25) \\ \implies & C_{\omega} = E(1 - \omega) + \omega D^{-1}(L + R) \qquad (26) \end{aligned}$$

This is the iteration matrix of the damped Jacobi method.

1.10.7 Analysis of the Damped Jacobi method

The iteration matrix of the damped Jacobi method can be written as

$$C_{J,\omega} = E(1 - \omega) + wD^{-1}(D - A) = E - \omega D^{-1}A = E - \omega\frac{h^2}{4}A \tag{27}$$

Furthermore, by (26), the iteration matrix of the damped Jacobi method is

$$C_{J,\omega} = [E + \omega C_j - \omega E] = (1 - \omega)E + \omega C_j \tag{28}$$

where C_j is the iteration matrix of the Jacobi method. The eigenvalues of

the iteration matrix of the Jacobi method are

$$(\rho_J)_{\nu,\mu} = 1 - \left[\sin^2\left(\frac{\pi\nu h}{2}\right) + \sin^2\left(\frac{\pi\mu h}{2}\right)\right]$$

Thus, the eigenvalues of the iteration matrix of the damped Jacobi method are

$$(\rho_{J,\omega})_{\nu,\mu} = 1 - \omega\left[\sin^2\left(\frac{\pi\nu h}{2}\right) + \sin^2\left(\frac{\pi\mu h}{2}\right)\right] \tag{29}$$

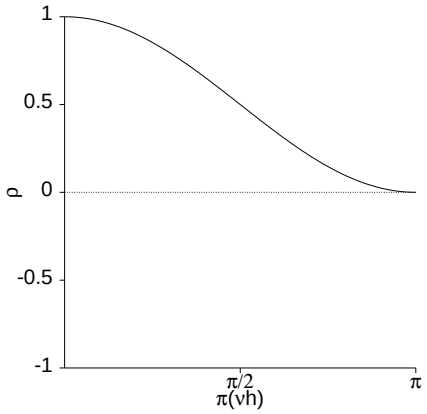
Now, for $\nu = \mu$, we have

$$(\rho_{J,\omega})_{\nu,\nu} = 1 - 2\omega\left[\sin^2\left(\frac{\pi\nu h}{2}\right)\right] \tag{30}$$

Thus, if $\omega = \frac{1}{2}$

$$(\rho_{J,\omega})_{\nu,\nu} = 1 - \left[\sin^2\left(\frac{\pi\nu h}{2}\right)\right] \tag{31}$$

The following graph depicts the eigenvalues $(\rho_{J,\omega})_{\nu\nu}$ with respect to the parameter $\pi\nu h$ in (31).

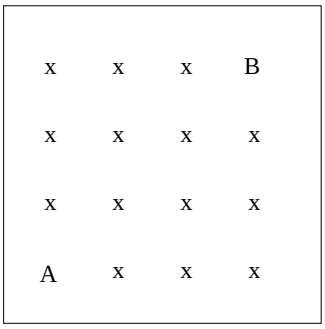


This shows that the damped Jacobi method with $\omega = \frac{1}{2}$ has the properties

- Bad convergence for low frequencies.
- Good convergence for high frequencies.

The Gauss–Seidel method has similar properties as the damped Jacobi method with $\omega = \frac{1}{2}$.

1.10.8 Heuristic approach



By single step methods we require $O(\sqrt{n}) = O(h^{-1})$ operations for a correction in B due to a change in A. The idea is to achieve faster correction by using a coarser grid.

2 Multigrid Algorithm

2.1 Multigrid algorithm on a Simple Structured Grid

2.1.1 Multigrid

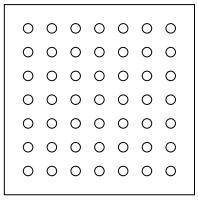


Figure 3: l=3

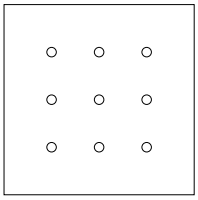


Figure 4: l=2

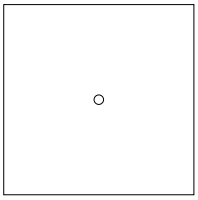


Figure 5: l=1

Let l be the number of levels such that $l_{max} \in \mathbb{N}$ and

$$m_l = 2^l$$
$$n_l = (m_l - 1)^2$$
$$h_l = 2^{-l}$$

for $l = 1 \ldots l_{max}$.

Let us assume that a PDE (e.g. Poisson's equation) is given. Discretize this equation by the grids $\Omega_l := \Omega_{h_l}$ where $l = 1, \ldots, l_{max}$. This leads to the discrete matrix equations

$$A_l x_l = b_l \tag{32}$$

where $b_l, x_l \in S_l$ and $S_l = \mathbb{R}^{n_l}$. The matrix A_l is an invertible matrix of order $n_l \times n_l$.

Let an iterative solver for (32) be given as

$$x_l{}^{k+1} \quad = \quad C_l{}^{relax} x_l{}^k + N_l b_l \quad = \quad \mathcal{S}_{l,b_l}(x_l^k) \tag{33}$$

2.1.2 Idea of Multigrid Algorithm

Let $\widetilde{x}_l$ be an approximate solution for (32). The algebraic $\widetilde{e}_l$ is defined as

$$\widetilde{e}_l = x_l - \widetilde{x}_l. \tag{34}$$

Now $\widetilde{e}_l$ has to be calculated in order to find x_l. The following residual equation is valid for $\widetilde{e}_l$,

$$A_l \widetilde{e}_l = r_l \tag{35}$$

where r_l is called the residual and is given by

$$r_l = b_l - A_l \widetilde{x}_l \tag{36}$$

The aim is to find an approximate solution of the residual equation by solving the equation approximately on a coarse grid Ω_{l-1}. To this end, we need the following matrix operators

- Restriction operator
$$I_l^{\,l-1} : S_l \mapsto S_{l-1}$$
- Prolongation operator
$$I_{l-1}^{l} : S_{l-1} \mapsto S_l$$

2.1.3 Two–grid Multigrid Algorithm

Two–grid Multigrid algorithm with parameters v_1 and v_2
Let x_l^k be an approximate solution of (32) and v_1 and v_2 the parameters of pre–smoothing and post–smoothing.

1. Step 1 (Pre–smoothing)

$$x_l^{k,1} = \mathcal{S}_{l,b_l}^{\,v_1} x_l^k \tag{37}$$

2. Step 2 (Coarse grid correction)

Residual calculation :

$$r_l = b_l - A_l x_l^{k,1} \tag{38}$$

Restriction :

$$r_{l-1} = {I_l}^{l-1} r_l \tag{39}$$

Solve on coarse grid:

$$e_{l-1} = {A_{l-1}}^{-1} r_{l-1} \tag{40}$$

Prolongation :

$$e_l = I_{l-1}^l e_{l-1} \tag{41}$$

Correction :

$$x_l^{k,2} = x_l^{k,1} + e_l \tag{42}$$

3. Step 3 (Post–smoothing)

$$x_l^{k+1} = \mathcal{S}_{l,b_l}^{\,v_2}(x_l^{k,2}) \tag{43}$$

2.1.4 Restriction and Prolongation Operators

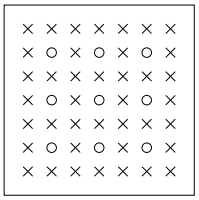


Figure 6: O–Coarse grid point and X–Fine grid point

Let us abbreviate $x_{i,j} = x_{(ih_{l-1},jh_{l-1})}$ and set $x_{i,j} = 0$ for $i = 0$ or $j = 0$ or $i = m_{l-1}$ or $j = m_{l-1}$.

2.1.5 Prolongation or Interpolation

The interpolation or prolongation of $x_{i,j}$ given by $w_{i,j} = \{I_{l-1}^{l}(x)\}_{(ih_l,jh_l)}$ is defined by the following equations

$$w_{2i,2j} = \frac{1}{2}x_{i,j} \tag{44}$$

$$w_{2i+1,2j} = \frac{1}{4}(x_{i,j} + x_{i+1,j}) \tag{45}$$

$$w_{2i,2j+1} = \frac{1}{4}(x_{i,j} + x_{i,j+1}) \tag{46}$$

$$w_{2i+1,2j+1} = \frac{1}{8}(x_{i,j} + x_{i+1,j} + x_{i,j+1} + x_{i+1,j+1}) \tag{47}$$

2.1.6 Pointwise Restriction

Piecewise restriction is rarely applied and defined by

$$\{\dot{I}_l^{\,l-1}(x)\}_{(ih_{l-1},jh_{l-1})} = x_{2i,2j} \tag{48}$$

The quality of this restriction operator is not very good.

2.1.7 Weighted Restriction

Weighted restriction or full weighting is defined by

$$\begin{aligned}\{{I_l}^{l-1}(x)\}_{(ih_{l-1},jh_{l-1})} &= \frac{1}{8}(x_{2i+1,2j+1}+x_{2i-1,2j+1}+x_{2i+1,2j-1}+x_{2i-1,2j-1})+\\ &\quad \frac{1}{4}(x_{2i+1,2j}+x_{2i-1,2j}+x_{2i,2j+1}+x_{2i,2j-1})+\\ &\quad \frac{1}{2}x_{2i,2j}\end{aligned}$$

Remark

$$({I_l}^{l-1})^T = I_{l-1}^l \tag{49}$$

2.2 Iteration Matrix of the Two–Grid Multigrid Algorithm

Theorem 4. *The iteration matrix of a two–grid Multigrid algorithm is*

$$C_l^{two\_grid} = \left(C_l^{relax}\right)^{v_2}\left(E - I_{l-1}^l(A_{l-1})^{-1}{I_l}^{l-1}A_l\right)\left(C_l^{relax}\right)^{v_1} \tag{50}$$

Proof

The coarse grid correction is

$$\begin{aligned}{x_l}^{k,2} &= {x_l}^{k,1} + I_{l-1}^l(A_{l-1})^{-1}{I_l}^{l-1}(b_l - A_l{x_l}^{k,1})\\ &= \left(E - I_{l-1}^l(A_{l-1})^{-1}{I_l}^{l-1}A_l\right){x_l}^{k,1} + I_{l-1}^l(A_{l-1})^{-1}{I_l}^{l-1}b_l\end{aligned}$$

Therefore the iteration matrix of the coarse grid correction of the two–grid Multigrid algorithm is

$$\left(E - I_{l-1}^l(A_{l-1})^{-1}{I_l}^{l-1}A_l\right)$$

A short calculation shows that the iteration matrix of two linear iteration algorithms is the product of the iteration matrices of these algorithms.

2.3 Multigrid Algorithm

Multigrid algorithm $MGM(x_l^k, b_l, l)$ with parameters (v_1,v_2,μ)

Let x_{lmax}^k be an approximate solution of (32). Then,

$$x_{lmax}^{k+1} = MGM(x_{lmax}^k)$$

is the approximate solution of (32) by the multigrid algorithm with an initial vector x_{lmax}^k. The multigrid algorithm can then be described as

If $l = 1$ then $MGM(x_l^k, b_l, l) = A_l^{-1} b_l$

If $l > 1$ then

Step 1 (v_1-pre–smoothing)

$x_l^{k,1} = \mathcal{S}_{l,b_l}^{v_1}(x_l^k)$

Step 2 (Coarse grid correction)

Residual : $r_l = b_l - A_l x_l^{k,1}$

Restriction : $r_{l-1} = I_l^{\ l-1} r_l$

Recursive call:

$e_{l-1}^0 = 0$

for $i = 1 \dots \mu$

$e_{l-1}^i = MGM(e_{l-1}^{i-1}, r_{l-1}, l-1)$

$e_{l-1} = e_{l-1}^{\mu}$

Prolongation : $e_l = I_{l-1}^l e_{l-1}$

Correction : $x_l^{k,2} = x_l^{k,1} + e_l$

Step 3 (v_2-post–smoothing)

$MGM(x_l^k, b_l, l) = \mathcal{S}_{l,b_l}^{v_2}(x_l^{k,2})$

The algorithm $\mu = 1$ is called V-cycle. The algorithm $\mu = 2$ is called W-cycle.

Homework: Describe the multigrid algorithm as a finite state machine, where every state is smoothing step and an operation is a restriction or prolongation. Then, the finite state machine of a V-cycle looks like a “V” and the finite state machine of a W-cycle looks like a “W”.

Let N be the number of unknowns. The computational amount of the V-cycle and W-cycle is $O(N)$.

The theory of multigrid algorithms shows that there is a constant ρ such that the convergence rate of the multigrid algorithm satisfies

$$\rho(C_{MGM,l}) \leq \rho < 1$$

independent of l. Thus, the effective convergence rate of the multigrid algorithm is:

$$G_{eff}(MGM, \mu) = O(1)$$

for $\mu = 1, 2$.

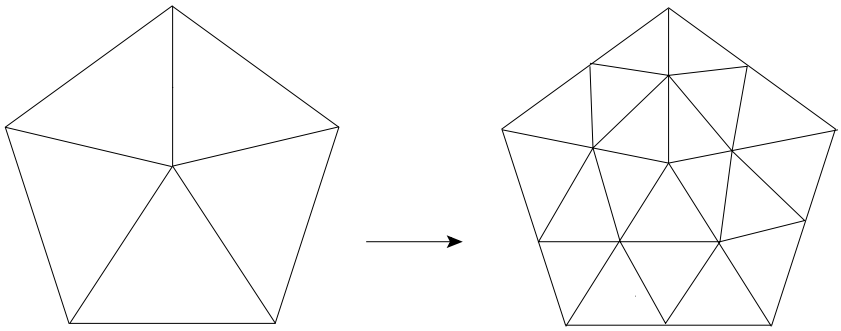
2.4 Multigrid Algorithm for Finite Elements

2.4.1 Model Problem

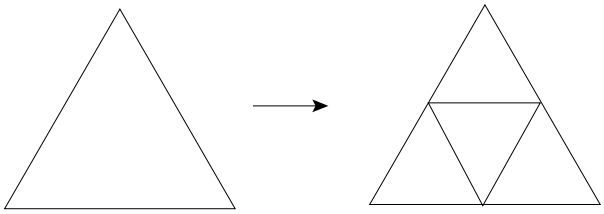
Let $\tau_{h_1} \cdots, \tau_{h_{l_{max}}}$ be a sequence of quasi-uniform subdivisions, where $h_l = 2^{-l}$ such that

$$V_{h_i} \subset V_{h_{i+1}} \quad \text{(This means} \quad V_{2h} \subset V_h)$$

2.4.2 Example



Every triangle is divided into four triangles



We want to solve the problem

$$\begin{aligned}&\text{Find } u_{h_{l_{max}}} \in V_{h_{l_{max}}} \text{ such that}\\ &\quad a(u_{h_{l_{max}}}, v_h) = f(v_h) \quad \forall \quad v_h \in V_{h_{l_{max}}}\end{aligned} \tag{51}$$

To this end, let us study all problems of type

$$\begin{aligned}&\text{Find } u_{h_l} \in V_{h_l} \text{ such that}\\ &\quad a(u_{h_l}, v_h) = f_l(v_h) \quad \forall \quad v_h \in V_{h_l}\\ &\text{for every } l = 0, \cdots, l_{max}\end{aligned} \tag{52}$$

where f_l is a suitable coarse grid right hand side.

2.4.3 The Nodal Basis

Let $(v_i^k)_{k \in \mathring{\Omega}_h}$ be the nodal basis for V_{h_i}. Now (52) can be defined in matrix form as follows:

$$A_i x_i = b_i \tag{53}$$

where

$$A_i = (a_{kj})_{kj \in \mathring{\Omega}_{h_i}}, a_{kj} = a(v_i^k, v_i^j) \tag{54}$$

$$x_i = (x_i^k)_{k \in \mathring{\Omega}_h} \tag{55}$$

$$b_i = (b_i^k)_{k \in \mathring{\Omega}_h} \tag{56}$$

and the solution vector u_h is given by

$$u_{h_i} = \sum x_i^k v_i^k \tag{57}$$

2.4.4 Prolongation Operator for Finite Elements

The natural inclusion is the prolongation operator

$$\begin{array}{ccc} u & \in & V_{h_i} \\ \downarrow & & \\ u & \in & V_{h_{i+1}} \end{array}$$

To implement this operator, we have to describe this operator in a matrix form.

By $V_{h_i} \subset V_{h_{i+1}}$, there are coefficients $\gamma_k^{k'}$ such that

$$v_i^{k'} = \sum_k \gamma_k^{k'} v_{i+1}^k \tag{58}$$

Thus, we get

$$\begin{aligned} u_{h_i} &= \sum_{k'} x_i^{k'} v_i^{k'} = \sum_{k'} \sum_k \gamma_k^{k'} v_{i+1}^k x_i^{k'} && (59) \\ &= \sum_k \left(\sum_{k'} \gamma_k^{k'} x_i^{k'} \right) v_{i+1}^k && (60) \end{aligned}$$

Now the matrix version of the prolongation operator is

$$\begin{array}{c} I_i^{i+1} \left(x_i^{k'} \right)_{k'} = \left(\sum_{k'} \gamma_k^{k'} x_i^{k'} \right)_k \\ \Downarrow \\ I_i^{i+1} = \left(\gamma_k^{k'} \right)_{(k,k')} \end{array}$$

2.4.5 Restriction Operator for Finite Elements

Observe that $F_i \in (V_{h_i})'$.
This means that $F_i : V_{h_i} \longrightarrow \mathbf{R}$ is a linear mapping. The natural inclusion is the restriction operator.

$$\begin{array}{rcll} F_{i+1} & \in & (V_{h_{i+1}})' & \\ \downarrow & & & \\ F_i & \in & (V_{h_i})' & \\ F_i(w) & := & F_{i+1}(w) & \forall \quad w \in V_{h_i} \end{array}$$

The matrix version of the restriction operator can be obtained as follows

$$\begin{aligned} b_i^{k'} &= F_i(v_i^{k'}) = \sum_k \gamma_k^{k'} F_i(v_{i+1}^k) && (61) \\ &= \sum_k \gamma_k^{k'} b_{i+1}^k && (62) \\ I_{i+1}^i &= \left(\gamma_k^{k'} \right)_{(k',k)} && (63) \end{aligned}$$

2.5 Fourier Analysis of the Multigrid method

2.5.1 Local Fourier analysis

A multigrid algorithm consists of several parameters that have to be properly tuned such that the algorithm converges rapidly.The parameters are,

μ	:	recursion parameter.
ν_1, ν_2	:	smoothing parameter.
$\mathcal{S}_{l,b_l}$	:	choice of smoother.
I_{l-1}^{l}	:	choice of the prolongation operator.
I_{l}^{l-1}	:	choice of the restriction operator
$A_l \quad \text{for} \quad l < l_{max}$	:	choice of the stiffness matrix on the courser grid.
		($A_{l_{max}}$is determined by the discretisation.)

The following simplification is made in order to analyze the convergence of the two–grid method more easily and exactly.

Omission of the boundary conditions – transition to an infinite dimensional grid

Instead of the finite dimensional grid

$$\Omega_h^d := \left\{ (j_1 h, j_2 h, \ldots, j_d h) \quad | \quad j_1, j_2, \ldots, j_d \in \left\{ 0, \ldots, \frac{1}{h} \right\} \right\} \tag{64}$$

we apply an infinite dimensional grid

$$\overset{\infty}{\Omega}{}_h^d := \{ (j_1 h, j_2 h, \ldots, j_d h) \quad | \quad j_1, j_2, \ldots, j_d \in \mathbb{Z} \} \tag{65}$$

The operators A_l, I_l^{l-1}, $\mathcal{S}_{l,b_l}$ have to be extended to the infinite dimensional grid in a suitable manner.

Remark

- The operators A_l etc. are stencil operators, e.g a nine point stencil.
- The operators A_l etc. depend on the spatial coordinates.

Then, we define the operators on the infinite grid as follows,

> Let Q_h^d be a stencil operator on the grid Ω_h^d. For every inner point x_0 in the grid Ω_h^d of the stencil $\mathcal{S}(x_0)$,a corresponding stencil operator $\overset{\infty}{Q}{}_h^d$ is defined.

Example

Let $d = 1$. The stiffness matrix obtained by the finite difference discretization of the operator $-\frac{d^2}{dx^2}$ on the grid Ω_h^1 is

$$\mathbf{A_h^1} = \begin{pmatrix} 2 & -1 & & & & \\ -1 & 2 & -1 & & & \\ & -1 & 2 & -1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & -1 & 2 & -1 \\ & & & & -1 & 2 \end{pmatrix} \frac{1}{h^2} \tag{66}$$

The operator on the infinite grid $\overset{\infty}{A}{}_h^1$ is now

$$\overset{\infty}{A}{}_h^1 = \begin{pmatrix} \ddots & & \ddots & & \ddots & \\ & -1 & & 2 & & -1 \\ & & \ddots & & \ddots & \ddots \end{pmatrix} \frac{1}{h^2} \tag{67}$$

which implies

$$\overset{\infty}{A}{}_h^1(u)(x) = (-u(x-h) + 2u(x) - u(x+h))\frac{1}{h^2} \qquad \forall x \in \overset{\infty}{\Omega}{}_h^1 \tag{68}$$

By the extension of the operators on the infinite dimensional grid, we can construct a two–grid method on the infinite dimensional grid $\overset{\infty}{\Omega}{}_h^d$. To analyze the convergence of the two–grid method, we need to know the iteration matrix of the method. By Lemma 3, the iteration matrix for the two–grid method is

$$\left(C_h^{relax}\right)^{\nu_2}\left(E_h - I_H^h (A_H)^{-1} I_h^H A_h\right)\left(C_h^{relax}\right)^{\nu_1}, \text{where } H = 2h. \tag{69}$$

where,

C_h^{relax}	:	iteration matrix of the smoothening step.
E_h	:	extended unit matrix.
I_H^h	:	extended prolongation operator.
I_h^H	:	extended Restriction operator.
A_h, A_H	:	extended stiffness matrices on the coarser grid.

Example

The operators for d=1 are as follows.

$$A_h = \begin{pmatrix} \ddots & \ddots & \ddots & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \end{pmatrix} \frac{1}{h^2} \tag{70}$$

$$A_H = \begin{pmatrix} \ddots & \ddots & \ddots & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \end{pmatrix} \frac{1}{4h^2} \tag{71}$$

$$I_h^H = \begin{pmatrix} \ddots & & & & & \\ & 1 & 2 & 1 & & \\ & & 1 & 2 & 1 & \\ & & & & \ddots & \end{pmatrix} \frac{1}{4} \qquad \left(or \;\; factor \;\; \frac{1}{2\sqrt{2}} \right) \tag{72}$$

$$I_h^H = \begin{pmatrix} \ddots & & & \\ & 1 & & \\ & 2 & & \\ & 1 & 1 & \\ & & 2 & \\ & & 1 & \\ & & & \ddots \end{pmatrix} \frac{1}{2} \qquad \left(or \;\; factor \;\; \frac{1}{2\sqrt{2}} \right) \tag{73}$$

$$C_h^{relax} = \begin{pmatrix} \ddots & & \ddots & & \ddots & \\ & \frac{1}{2}\omega & & 1-\omega & & \frac{1}{2}\omega \\ & & \ddots & & \ddots & & \ddots \end{pmatrix} \tag{74}$$

$$\overset{\omega=\frac{1}{2}}{=} \begin{pmatrix} \ddots & & \ddots & & \ddots & \\ & \frac{1}{4} & & \frac{1}{2} & & \frac{1}{4} \\ & & \ddots & & \ddots & & \ddots \end{pmatrix}$$

We allow these operators to act on the following functional spaces.

$$V_h := \left\{ exp\Big(i\theta\frac{x}{h}\Big)_{x \in \overset{\infty}{\Omega}_h^d} \quad | \quad -\pi \leq \theta \leq \pi \right\} \tag{75}$$

$$V_H := \left\{ exp\Big(i\theta\frac{x}{H}\Big)_{x \in \overset{\infty}{\Omega}_H^d} \quad | \quad -\pi \leq \theta \leq \pi \right\} \tag{76}$$

For reasons of simplicity, let us restrict ourselves to the 1–D case.

2.5.2 Definition

The harmonic frequency of $\exp\left(i\theta\frac{x}{h}\right)$ is $\exp\left(i\tilde{\theta}\frac{x}{h}\right)$ where,

$$\begin{aligned} \tilde{\theta} &:= \theta - \pi \quad \text{for} \quad \theta \geq 0 \\ \tilde{\theta} &:= \theta + \pi \quad \text{for} \quad \theta < 0 \end{aligned}$$

2.5.3 Local Fourier analysis of the smoother

Definition

The functions $\exp\left(i\theta\frac{x}{h}\right)$ are the eigenfunctions of the iteration matrix C of the smoother $\mathcal{S}$with eigenvalues $\mu(\theta)$. This implies that

$$C\exp\left(i\theta\frac{x}{h}\right) = \mu(\theta)\exp\left(i\theta\frac{x}{h}\right)$$

We then have the smoothening factor of $\mathcal{S}$ as

3 Gradient Method and cg

3.1 Gradient Method

Let A a symmetric positive definite $n \times n$ matrix and $b \in \mathbb{R}^n$. Find $x \in \mathbb{R}^n$ such that

$$A \cdot x = b \tag{77}$$

Theorem 5. *The solution of the linear system (77) is solution of the minimization problem:*

$$\min_{x \in \mathbb{R}^n} \frac{1}{2} x^T A x - b^T x \tag{78}$$

Proof. Let $0 \neq v \in \mathbb{R}^n$ and $\lambda \in \mathbb{R}$. If x solves the minimization problem, then

$$\begin{aligned} & \frac{d}{d\lambda} \frac{1}{2} (x + \lambda v)^T A (x + \lambda v) - b^T (x + \lambda v) \Big|_{\lambda = 0} = 0 \\ \Rightarrow \quad & \frac{1}{2} (v^T A x + x^T A v) - b^T v = 0 \\ \Rightarrow \quad & v^T A x = v^T b \qquad \forall v \in \mathbb{R}^n \Rightarrow Ax = b \end{aligned}$$

Let $Ax = b$ and $v \in \mathbb{R}^n$. Then, we get

$$\begin{aligned} & \frac{1}{2} (x + v)^T A (x + v) - b^T (x + v) = \\ & = \frac{1}{2} (x + v)^T b + \frac{1}{2} v^T A v - b^T (x + v) + \frac{1}{2} x^T A v \\ & = \frac{1}{2} v^T A v - \frac{1}{2} (x + v)^T b + \frac{1}{2} b^T v \\ & = \frac{1}{2} v^T A v - \frac{1}{2} x^T b \Rightarrow \text{ minimum at } v = 0 \end{aligned}$$

Gradient method

1. Choose direction for seeking

$$d_k = -\nabla f(x_k),$$

where $f(x_k) = \frac{1}{2} x_k^T A x_k - b^T x_k$. This implies

$$d_k = b - A x_k \tag{79}$$

2. Choose $\alpha_k \in \mathbb{R}$, such that $f(x_{k+1})$ is minimal, where

$$x_{k+1} = x_k + \alpha_k d_k$$

Theorem 6. *α_k of the gradient method can be computed by*

$$\alpha_k = \frac{d_k^T d_k}{d_k^T A d_k} \tag{80}$$

Proof.

$$\begin{aligned}
&\frac{d}{d\alpha_k}\frac{1}{2}(x_k + \alpha_k d_k)^T A(x_k + \alpha_k d_k) - b^T(x_k + \alpha_k d_k) = 0 \\
&\frac{1}{2}(d_k^T A x_k + x_k^T A d_k) + \alpha_k d_k^T A d_k - b^T d_k = 0 \\
&-d_k^T d_k + \alpha_k d_k^T A d_k = 0 \\
\Rightarrow \quad &\alpha_k = \frac{d_k^T d_k}{d_k^T A d_k}
\end{aligned}$$

3.2 Analysis of the Gradient Method

Let x^* the accurate solution of

$$f(x) = \frac{1}{2}x^T A x - b^T x \rightarrow \text{ minimum}$$

This implies

$$Ax^* = b.$$

The energy norm is defined by $||x||_A = \sqrt{x^T A x}$.

Lemma 2.

$$f(x) = f(x^*) + \frac{1}{2}||x - x^*||_A^2$$

Proof.

$$\begin{aligned}
f(x) - f(x^*) &= \frac{1}{2}x^T A x - b^T x - \frac{1}{2}x^{*T} A x^* + b^T x^* = \\
&= \frac{1}{2}x^T A x - x^{*T} A x + \frac{1}{2}x^{*T} A x^* = \\
&= \frac{1}{2}||x - x^*||_A^2
\end{aligned}$$

Lemma 3.

$$||x_{k+1} - x^*||_A^2 = ||x_k - x^*||_A^2 \left\{1 - \frac{(d_k^T d_k)^2}{d_k^T A d_k d_k^T A^{-1} d_k}\right\}$$

Proof. By (79) and (80), we get:

$$\begin{aligned} f(x_{k+1}) &= f(x_k + \alpha_k d_k) \\ &= \frac{1}{2}(x_k + \alpha_k d_k)TA(x_k + \alpha_k d_k) - b^T(x_k + \alpha_k d_k) \\ &= f(x_k) + \alpha_k d_k^T(Ax_k - b) + \frac{1}{2}\alpha_k^2 d_k^T A d_k \\ &= f(x_k) + \frac{1}{2}\frac{(d_k^T d_k)^2}{d_k^T A d_k} \end{aligned}$$

Now, by Lemma 2, we obtain

$$||x_{k+1} - x^*||_A^2 = ||x_k - x^*||_A^2 - \frac{(d_k^T d_k)^2}{d_k^T A d_k}$$

By $d_k = -A(x_k - x^*)$, it is

$$||x_{k+1} - x^*||_A^2 = (A^{-1}d_k)^T A(A^{-1}d_k) = d_k^T A^{-1} d_k$$

and so we get

$$||x_{k+1} - x^*||_A^2 = ||x_k - x^*||_A^2 \left\{1 - \frac{(d_k^T d_k)^2}{d_k^T A d_k d_k^T A^{-1} d_k}\right\}$$

.

Lemma 4 (Inequality of Kantorowitsch)**.** *Let A symmetric positive definite and κ the condition number of A. Then, the following inequality holds*

$$\frac{(x^T Ax)(X^T A^{-1}x)}{(x^T x)^2} \leq \left(\frac{1}{2}\sqrt{\kappa}^{-1} + \frac{1}{2}\sqrt{\kappa}\right)^2$$

Proof. Let the eigenvalues of A $a = \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n = b$ and $\frac{b}{a} = \kappa$. By principal axis transformation, one gets

$$\frac{(x^T Ax)(X^T A^{-1}x)}{(x^T x)^2} = \frac{\sum \lambda_i y_i^2 \sum \lambda_i^{-1} y_i^2}{(\sum y_i^2)^2} = \sum_i \lambda_i z_i \sum_i \lambda_i^{-1} z_i$$

where $x = \sum_i y_i e_i$, e_i are the orthogonal eigenvector, and $z_i = \frac{y_i^2}{\sum_j y_j^2}$. Observe, that $\sum_i z_i = 1$. Thus, $P = (\sum_i \lambda_i z_i, \sum_i \lambda_I^{-1} z_i) = z_i \sum_i (\lambda_i, \lambda_i^{-1})$ is a convex combination.

Now, define

$$\begin{aligned} P &= (\lambda, \mu) \\ \lambda &= \sum_i \lambda_i z_i \\ \mu &= \sum_i \lambda_i^{-1} z_i \end{aligned}$$

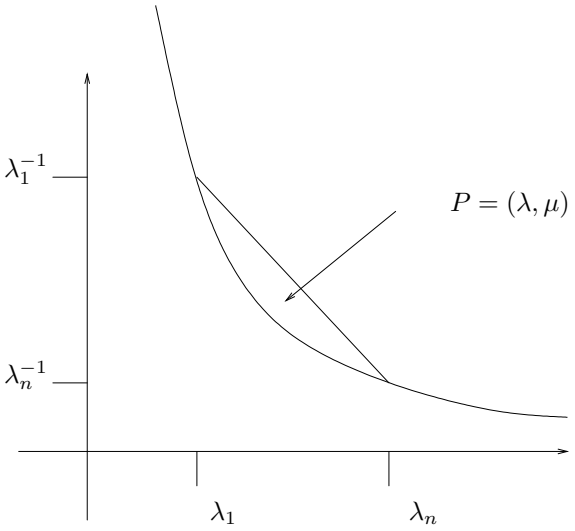


Figure 7: The convex function $x \to x^{-1}$.

Then, by Figure 7, we get

$$\mu = \sum_i \lambda_i^{-1} z_i \leq (\lambda_1 + \lambda_n - \lambda)\lambda_1^{-1}\lambda_n^{-1},$$

since the point P is below the line between $(\lambda_1, \lambda_1^{-1})$ and $(\lambda_n, \lambda_n^{-1})$. Now,

$$\max_{\lambda_1 \leq \lambda \leq \lambda_n} \lambda \frac{\lambda_1 + \lambda_n - \lambda}{\lambda_1 \lambda_n} = \frac{\lambda_1 + \lambda_n)^2}{4\lambda_1 \lambda_n}$$

completes the proof.

Theorem 7. *The gradient method converges as follows:*

$$||x_k - x^*||_A \leq \left(\frac{\kappa - 1}{\kappa + 1}\right)^k ||x_0 - x^*||_A$$

Proof. Apply Lemma 2 to 4 and observe that

$$1 - \frac{1}{\left(\frac{1}{2}\sqrt{\kappa} + \frac{1}{2}\sqrt{\kappa}^{-1}\right)^2} = 1 - \frac{4\kappa}{(\kappa + 1)^2} = \frac{(\kappa - 1)^2}{(\kappa + 1)^2}.$$

3.3 The Method of Conjugate Directions

If A is positive definite, then $x^T Ay$ defines a scalar product. Let $d_0, d_1, \ldots, d_{n-1}$ be A-orthogonal (conjugated) vectors. Then, the set $\{d_0, d_1, \ldots, d_{n-1}\}$ is a basis of $\mathbb{R}^n$.

Observe, that any vector $y \in \mathbb{R}^n$ can be written as:

$$y = \sum_{k=0}^{n-1} \alpha_k d_k$$

Lemma 5. *Let $x_0 \in \mathbb{R}^n$ be a start vector. Then, define*

$$x_{k+1} = x_k + \alpha_k d_k \tag{81}$$

where

$$\alpha_k = -\frac{g_k^T d_k}{d_k^T A d_k}, \qquad g_k = Ax_k - b \tag{82}$$

This sequence leads to the exact solution after at most n iterations: $x_n = A^{-1}b$.

Proof. There exists α_i such that

$$x^* - x_0 = \sum_i \alpha_i d_i$$

Thus, we get

$$d_i^T A(x^* - x_0) = \alpha_i d_i^T A d_i$$

$$\alpha_i = \frac{d_i^T A(x^* - x_0)}{d_i^T A d_i} = -\frac{d_i^T (Ax_0 - b)}{d_i^T A d_i}$$

By induction we show that equation (82) holds.
$k = 0$ follows from the upper equation.
By $x_i = \sum_{k<i} \alpha_k d_k + x_0$ it is $d_i^T A x_i = d_i^T A x_0$. Thus, we get

$$\alpha_i = -\frac{d_i^T (Ax_0 - b)}{d_i^T A d_i}$$

Lemma 6. *x_k in Lemma 5 minimizes*

$$f(x) = \frac{1}{2} x^T A x - b^T x$$

on $x_0 + V_k$, where $V_k = span\{d_0, d_1, \ldots, d_{k-1}\}$.

Furthermore, the following orthogonalization property holds:

$$d_i^T g_k = 0 \qquad \text{for } i < k. \tag{83}$$

Proof. Let us show, that it is enough to prove (83).

Let $d \in V_k \Rightarrow d^T g_k = 0$

$$\begin{aligned} f(x_k + d) &= \frac{1}{2}(x_k + d)^T A(x_k + d) - b(x_k + d) \\ &= \frac{1}{2}(x_k^T A x_k + \frac{1}{2} d^T A d + d^T A x_k - b^T x_k - b^T d \\ &= \frac{1}{2}(x_k^T A x_k + d^T A d) + d^T g_k - b^T x_k \\ &= \frac{1}{2}(x_k^T A x_k + d^T A d) - b^T x_k \end{aligned}$$

$\Rightarrow$ Minimization by $d \in V_k$ not possible!
Proof of (83) by induction:
$\underline{i = k-1:}$
$d_{k-1}^T g_k = d_{k-1}^T \left(A \left(x_{k-1} - \frac{{g_k - 1}^T d_{k-1}}{d_{k-1}^T A d_{k-1}} d_{k-1} \right) - b \right) = 0 \qquad (*)$
$\underline{i < k-1}$ By $x_k - x_{k-1} = \alpha_{k-1} d_{k-1}$, it holds

$$g_k - g_{k-1} = A(x_k - x_{k-1}) = \alpha_{k-1} A d_{k-1}$$

$$\Rightarrow \quad d_i^T (g_k - g_{k-1}) = 0 \qquad \text{for } i < k-1$$

By the induction hypothesis, (83) follows for $i \leq k-2$.

3.4 cg-Method (Conjugate Gradient Algorithm)

The directions $d_0, \ldots, d_{k+1}$ are computed by an orthogonalization of the gradients:

$$\begin{aligned} d_0 &= -g_0 \\ d_{k+1} = -g_{k+1} + \beta_k d_k & \\ & \text{where } \beta_k = \frac{g_{k+1}^T A d_k}{d_k^T A d_k} \text{ if } g_k \neq 0 \end{aligned}$$

Theorem 8. *If $g_{k+1} \neq 0$ then it holds*
(I) It is $d_{k-1} \neq 0$
(II) $V_k = span[g_0, Ag_0, \ldots, A^{k-1} g_0] = span[g_0, g_1, \ldots, g_{k-1}] = span[d_0, d_1, \ldots, d_{k-1}]$
(III) The vectors $d_0, d_1, \ldots, d_{k-1}$ are pairwise A-orthogonal
(IV) It is $f(x_k) = \min_{z \in V_k} f(x_0 + z)$
(V) $\alpha_k = \frac{g_k^T g_k}{d_k^T A d_k}, \qquad \beta_k = \frac{g_{k+1}^T g_{k+1}}{g_k^T g_k}$

Proof by Induction from (I) to (V)
$\underline{k = 1:}$ obvious
$\underline{k \to k+1:}$ First observe, that

$$\text{span}[g_0, g_1, \ldots, g_{k-1}] = \text{span}[d_0, d_1, \ldots, d_{k-1}]$$

follows by $d_{k+1} = -g_{k+1} + \beta_k d_k$. Thus,

$$g_k = g_{k-1} + A(x_k - x_{k-1}) = g_{k-1} + \alpha_{k-1} A d_{k-1}$$

implies that

$$\begin{aligned} & g_k \in \text{span}[g_0, Ag_0, \ldots, A^k g_0] \\ \Rightarrow \quad & \text{span}[g_0, g_1, \ldots, g_k] \subset \text{span}[g_0, Ag_0, \ldots, A^k g_0]. \end{aligned}$$

By induction hypothesis $d_0, d_1, \ldots, d_{k-1}$ are linear independent. By (83), the optimization property of x_k, it is

$$d_i^T g_k = 0 \qquad \text{for } i < k. \qquad (*)$$

Since $g_k \neq 0$, the linear independence holds for $d_0, \ldots, d_{k-1}, g_k$ and therefore we get that $g_0, \ldots, g_k$ is linear independent. Thus, it follows by a dimension argument:

$$\text{span}[g_0, \ldots, g_k] = \text{span}[g_0, Ag_0, \ldots, A^{k-1} g_0].$$

This completes the proof of (II). Since d_{k-1}, g_k are linear independent, it follows $d_k \neq 0$, (I). Now, we proof (III):

$$d_i^T A d_k = -d_I^T A g_k + \beta_{k-1} d_i^T A d_{k-1}$$

By the construction of β_{k-1}, the orthogonality (III) holds for $i = k - 1$:

$$d_i^T A d_k = 0.$$

For $i < k - 1$ the induction hypothesis $d_i^T A d_{k-1} = 0$ implies

$$d_i^T A d_k = -d_i^T A g_k$$

Since $A d_i \subset \text{span}[d_0, \ldots, d_{i+1}]$ and $i + 1 < k$, $(*)$ leads to

$$d_i^T A d_k = 0$$

(IV) is a implication from Lemma 6. Proof of (V): $d_k = -g_k + \beta_{k-1} d_{k-1}$ and $(*)$ imply that

$$\alpha_k = -\frac{g_k^T d_k}{d_k^T A d_k} = \frac{g_k^T g_k}{d_k^T A d_k}.$$

Now, observe that $g_{k+1}^T g_k = 0$ follows by $(*)$ and (II). Thus, we get

$$\beta_k = \frac{g_{k+1}^T A d_k}{d_k^T A d_k} = \frac{g_{k+1}^T \alpha_k A d_k}{g_k^T g_k} = \frac{g_{k+1}^T (g_{k+1} - g_k)}{g_k^T g_k} = \frac{g_{k+1}^T g_{k+1}}{g_k^T g_k}.$$

To implement the cg-algorithm in an efficient way, we apply (V) in Theorem (8). Furthermore, observe that

$$g_{k+1} = A(x_k + \alpha_k d_k) - b = g_k + \alpha_k A d_k$$

and let us introduce the auxiliary vector

$$h := Ad.$$

cg algorithm

$$\begin{array}{lll}
x & = x^0 & \\
g & = Ax - b & \\
\delta_0 & = g^T g & \\
\text{if} & \delta_0 \leq \epsilon \qquad & \text{stop} \\
d & = -g & \\
\underline{\text{recursion:}} & k = 0, 1, \ldots & \\
& h = Ad & \\
& \alpha = \dfrac{\delta_0}{d^T h} & \\
& x := x + \alpha d & \\
& g := g + \alpha h & \\
& \delta_1 = g^T g (= \delta_{k+1}) & \\
& \text{if} \quad \delta_1 \leq \epsilon \qquad & \text{stop} \\
& \beta = \dfrac{\delta_1}{\delta_0} \left(= \beta = \dfrac{\delta_{k+1}}{\delta_k} \right) & \\
& d = -g + \beta d & \\
& \delta_0 := \delta_1 &
\end{array}$$

3.5 Analysis of the cg algorithm

cg is a direct and an iterative method!

Lemma 7. *Let $p \in P_k$ be a polynomial such that*

$$p(0) = 1, \qquad |p(z)| \leq r \qquad \text{for all } z \in \sigma(A)$$

Then, for the cg algorithm, the following inequality holds

$$||x_k - x^*||_A \leq r||x_0 - x^*||_A$$

Proof. Let $q(z) = \frac{p(z)-1}{z}$ and $y := x_0 + q(A)g_0$. Then, by $g_0 = A(x_0 - x^*)$, we get

$$y - x^* = x_0 - x^* + y - x_0 = x_0 - x^* + q(A)g_0 = p(A)(x_0 - x^*)$$

$$\Rightarrow ||y - x^*||_A \leq ||p(A)||_A \cdot ||x_0 - x^*||_A$$

Let $w = \sum_j c_j e_j$, where e_j are the orthonormal eigenvectors of A such that $Ae_j = \lambda_j e_j$. Then, it holds

$$\begin{aligned} ||p(A)w||_A^2 &= ||\sum_j c_j p(\lambda_j) e_j||_A^2 = \sum_j \lambda_j |c_j p(\lambda_j)|^2 \\ &\leq r^2 \sum_j \lambda_j |c_j|^2 \leq r^2 ||\sum_j c_j e_j||_A^2 = r^2 ||w||_A^2 \\ \Rightarrow ||p(A)||_A &\leq r \end{aligned}$$

This shows $||y - x^*||_A \leq r||x_0 - x^*||_A$. By Theorem 8 and Lemma 2, we conclude

$$||x_k - x^*||_A \leq r||x_0 - x^*||_A$$

Lemma 8. *Let*

$$T_k(x) = \frac{1}{2}\left[(x + \sqrt{x^2 - 1})^k + (x - \sqrt{x^2 - 1})^k\right]$$

for $k = 0, 1, \ldots$. Then it holds
a) $T_k(x)$ is a real-valued polynomial of degree $\leq k$.
b) $|T_k(x)| \leq 1$ for $-1 \leq x \leq 1$.
c) $T_k(x) \geq \frac{1}{2}(x + \sqrt{x^2 - 1})^k$ for $x \geq 1$.
d) $T_k(1) = 1$

Proof. a) Using the binomial formula, one can see that the terms with odd powers cancel. Thus, of T_k is a real-valued polynomial.
b) $|T_k(x)| \leq \frac{1}{2}\left(|x + i\sqrt{1 - x^2}|^k + |x - i\sqrt{1 - x^2}|^k\right) \leq 1$ for $|x| \leq 1$.
c) and d) are obvious.

Remark: T_k is called Tschebyscheff polynomial. One can prove

$$T_k(x) = \cos(k \arccos x)$$

Theorem 9.

$$||x_k - x^*||_A \leq \frac{1}{T_k\left(\frac{\kappa+1}{\kappa-1}\right)}||x_0 - x^*||_A \leq 2\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^k ||x_0 - x^*||_A$$

Proof. Let a, b be the extremal eigenvalues of A. Set

$$p(x) = \frac{T_k\left(\frac{b+a-2x}{b-a}\right)}{T_k\left(\frac{b+a}{b-a}\right)}$$

Then, $p(0) = 1$ and by b) in Lemma 8.

$$p(x) \leq \frac{1}{T_k\left(\frac{b+a}{b-a}\right)}$$

for every $x \in \rho(A)$. Furthermore, by Lemma 7:

$$||x_k - x^*||_A \leq \frac{1}{T_k\left(\frac{b+a}{b-a}\right)}||x_0 - x^*||_A$$

Furthermore, observe that

$$\frac{b+a}{b-a} = \frac{\kappa+1}{\kappa-1}$$

and

$$\frac{\kappa+1}{\kappa-1} + \sqrt{\left(\frac{\kappa+1}{\kappa-1}\right)^2 - 1} = \frac{\kappa+1+\sqrt{4\kappa}}{\kappa-1} = \frac{\sqrt{\kappa}+1}{\sqrt{\kappa}-1}.$$

Now c) in Lemma 8 completes the proof.

3.6 Preconditioned cg Algorithm

Let A be a symmetric positive definite $n \times n$-matrix and C a symmetric positive definite $n \times n$-matrix, such that C is an approximation of A.

Example 6.

- *C is the diagonal of A .*
- *C is the tridiagonal part of A . Then, C^{-1} can be computed be a LR-decomposition.*
- *C^{-1} is the result of a suitable symmetric multigrid algorithm.*

Instead of the equation

$$Ax = b$$

we try to solve the equation

$$C^{-1}Ax = C^{-1}b,$$

if $C^{-1}A$ has a smaller condition number than A.

The problem is that in general $C^{-1}A$ is not symmetric positive definite. Therefore, we apply the following Lemma.

Lemma 9. *Define*

$$\langle x, y\rangle_C := x^T C y.$$

$C^{-1}A$ is symmetric positive definite with respect to $\langle\cdot,\cdot\rangle_C$.

This leads to the algorithm

precondition cg algorithm (bad version)

$$\begin{array}{rl}
x & = x^0 \\
g & = C^{-1}(Ax - b) \\
\delta_0 & = g^T C g \\
\text{if} & \delta_0 \le \epsilon \qquad \text{stop} \\
d & = -g \\
\underline{\text{recursion:}} & k = 0, 1, \ldots \\
& h = C^{-1}Ad \\
& \alpha = \dfrac{\delta_0}{d^T C h} \\
& x := x + \alpha d \\
& g := g + \alpha h \\
& \delta_1 = g^T C g \\
& \text{if} \quad \delta_1 \le \epsilon \qquad \text{stop} \\
& \beta = \dfrac{\delta_1}{\delta_0} \left(= \beta = \dfrac{\delta_{k+1}}{\delta_k} \right) \\
& d = -g + \beta d \\
& \delta_0 := \delta_1
\end{array}$$

In several cases C^{-1} can be computed, but C cannot be computed. Therefore, one applies the following more efficient version of the

precondition cg algorithm, which does not require the computation of C. In this version we introduce the new variables w and r by

$$\begin{aligned} Cg &=: r \\ Ch &=: w \end{aligned}$$

and omit the variable h.

precondition cg algorithm (efficient version)

$$\begin{aligned} x \quad & = x^0 \\ r \quad & = Ax - b \\ g \quad & = C^{-1}r \\ \delta_0 \quad & = g^T r \\ \text{if} \quad & \delta_0 \leq \epsilon \qquad \text{stop} \\ d \quad & = -g \\ \underline{\text{recursion:}} \quad & k = 0, 1, \ldots \\ & w = Ad \\ & \alpha = \frac{\delta_0}{d^T w} \\ & x := x + \alpha d \\ & r := r + \alpha w \\ & g := C^{-1}r \\ & \delta_1 = g^T r \\ & \text{if} \quad \delta_1 \leq \epsilon \qquad \text{stop} \\ & \beta = \frac{\delta_1}{\delta_0} \left(= \beta = \frac{\delta_{k+1}}{\delta_k} \right) \\ & d = -g + \beta d \\ & \delta_0 := \delta_1 \end{aligned}$$

4 GMRES

Let A be an invertible $n \times n$ matrix. Furthermore, let $b \in \mathbb{R}^n$ and $x_0 \in \mathbb{R}^n$ a starting vector.

Problem: Find $x \in \mathbb{R}^n$ such that

$$Ax = b$$

Let us consider the Krylov space K_m defined by

$$r_0 = b - Ax_0, \qquad K_m = \text{span}\{r_0, \ldots, A^{m-1} r_0\}$$

4.1 Minimal residual method

Find $x_m \in x_0 + K_m$ such that

$$||b - Ax_m||_2 \qquad \text{is minimal.} \tag{84}$$

A stable basis of K_m can be obtained by the Arnoldi-algorithm. The Arnoldi-algorithm is based on the orthogonalization algorithm of Gram-Schmidt:
Arnoldi Algorithm

$$\text{Let } q_1 \text{ with} \qquad ||q_1||_2 = 1, \qquad q_1 = \frac{r_0}{||r_0||_2}.$$

For $j = 1, \ldots, m-1$:

$$\tilde{q}_{j+1} = Aq_j, \qquad h_{ij} =< \tilde{q}_{j+1}, q_i > \qquad \text{for } i = 1, 2, \ldots, j$$

$$\begin{aligned} \tilde{\tilde{q}}_{j+1} &= \tilde{q}_{j+1} - \sum_{i=1}^{j} h_{ij} q_i \\ h_{j+1,j} &= ||\tilde{\tilde{q}}_{j+1}||_2 \\ q_{j+1} &= \frac{\tilde{\tilde{q}}_{j+1}}{h_{j+1,j}} \end{aligned}$$

Observe, that one has to stop the Arnoldi algorithm, if $h_{j+1,j} = 0$.

To analyze the properties of the Arnoldi algorithm, let us define the matrices

$$Q_k = (q_1, \ldots, q_k), \qquad H_{k+1,k} = \begin{pmatrix} h_{11} & h_{12} & & \\ h_{21} & h_{22} & \ddots & * \\ & h_{32} & \ddots & \\ & & \ddots & h_{k-1,k} \\ & & & h_{k,k} \\ & & & h_{k+1,k} \end{pmatrix}$$

$H_{k+1,k}$ is a Hessenberg matrix.

Then, we get the following lemma.

Lemma 10.
(i) $AQ_k = Q_{k+1}H_{k+1,k}$
(ii) Q is an orthogonal matrix.
(iii) $K_m = span\{q_1, \ldots, q_m\}$

Proof: (i) :

$$h_{j+1,j}q_{j+1} = Aq_j - \sum_{i=1}^{j} h_{ij}q_i, \qquad Aq_j = \sum_{i=1}^{j+1} h_{ij}q_i$$

for $j = 1, ..., k$. (ii) - (iii) is trivial.

By this lemma, we get:

$$x_m = x_0 + Q_m y, \qquad y \in \mathbb{R}^m$$

$$\begin{aligned} ||b - Ax_m||_2 &= ||r_0 - AQ_m y||_2 \\ &= ||r_0 - Q_{m+1}H_{m+1,m}y||_2 \\ &= ||Q_{m+1}\left(\beta\xi_1 - H_{m+1,m}y\right)||_2 \\ &= ||\beta\xi_1 - H_{m+1,m}y||_2, \end{aligned}$$

where $\beta = ||r_0||_2$ and $\xi_1 = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$. For solving the minimization problem (84) it is enough to resolve the problem:

Minimization Problem:
Find $y \in \mathbb{R}^m$ such that

$$||\beta\xi_1 - H_{m+1,m}y||_2 \rightarrow \text{ minimal} \tag{85}$$

The standard approach to solve this problem is to apply the QR-algorithm and Givens rotations.

4.2 Solution of the Minimization Problem of GMRES

$$\min_{y \in \mathbb{R}^m} ||\beta\xi_1 - H_{m+1,m}y||_2, \tag{86}$$

where $H_{m+1,m}$ is a the Hessenberg-matrix:

$$H_{k+1,k} = \begin{pmatrix} h_{11} & h_{12} & & \\ h_{21} & h_{22} & \ddots & * \\ & h_{32} & \ddots & \\ & & \ddots & h_{k-1,k} \\ & & & h_{k,k} \\ & & & h_{k+1,k} \end{pmatrix}$$

Problem (86) can be solved by the QR decomposition. To this end, let F be a unitary matrix $(m+1) \times (m+1)$ matrix and $R_{m+1,m}$ an upper triangular matrix, where the last row is 0 and

$$H_{m+1,m} = F^H R_{m+1,m}$$

Then, one gets

$$\min_{y \in \mathbb{R}^m} ||\beta\xi_1 - H_{m+1,m} y||_2 = \min_{y \in \mathbb{R}^m} ||\beta F \xi_1 - R_{m+1.m} y||_2$$

The solution of this problem is:

$$y = \widetilde{R_{m,m}}^{-1} \beta \widetilde{F\xi_1},$$

where the operator $\tilde{}$ omits the last row. $\widetilde{R_{m,m}}^{-1}$ can easily be computed, since $\widetilde{R_{m,m}}$ is an upper triangular matrix.

4.3 Computation of QR-Decomposition with Givens Rotation

$$\begin{aligned} F &= F_m F_{m-1} \cdots F_1 \\ F_i &= \begin{pmatrix} I & & & \\ & c_i & s_i & \\ & -\overline{s_i} & c_i & \\ & & & I \end{pmatrix} \text{ in the real case } \overset{=}{} \begin{pmatrix} I & & & \\ & c_i & s_i & \\ & -s_i & c_i & \\ & & & I \end{pmatrix}, \end{aligned}$$

where $c_i = cos\theta_i$, $s_i = sin\theta_i$.

It is easy to verify that

$$\begin{pmatrix} c_i & s_i \\ -s_i & c_i \end{pmatrix} \begin{pmatrix} c_i & -s_i \\ s_i & c_i \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

So F is unitary. Construct $F_1, \ldots, F_{m-1}$ such that

$$(F_{m-1} F_{m-2} \cdots F_1) H_{m+1,m} = \begin{pmatrix} * & & & & \\ & * & & * & \\ & & \ddots & & \\ & & & * & * \\ & & & 0 & d \\ & & & 0 & h \end{pmatrix}$$

where $h = h_{m+1,m}$ and F_m satisfies the equation

$$\begin{pmatrix} c_m & s_m \\ -s_m & c_m \end{pmatrix} \begin{pmatrix} d \\ h \end{pmatrix} = \begin{pmatrix} * \\ 0 \end{pmatrix} \tag{87}$$

To obtain (87), consider the following to cases:

1. $d = 0$: $\quad c_m = 0, \quad s_m = 1$
2. $d \neq 0$: $\quad s_m = c_m \frac{h}{d}, \quad c_m = \frac{|d|}{\sqrt{|d|^2+|h|^2}}$

Furthermore, observe

$$s_m^2 + c_m^2 = \left(\frac{h^2}{d^2} + 1\right) \frac{d^2}{d^2 + h^2} \tag{88}$$

4.4 The GMRES Algorithm

(1) Let x_0 be given. Compute $r_0 = b - Ax_0, \quad q_1 = \frac{r_0}{||r_0||_2}$. Set
$\xi = (1, 0, \ldots, 0)^T, \beta = ||r_0||_2$

For $k = 1, 2, \ldots$
(2) Compute q_{k+1} and $h_{i,k}, \quad i = 1, \ldots, k+1$ by the Arnoldi Algorithm.
Set $H(i,k) := h_{i,k}, \quad i = 1, \ldots, k+1$

(3) Apply $F_1, \ldots, F_{k-1}$ to the last column of H, that means for
$i = 1, \ldots, k-1$

$$\begin{pmatrix} c_i & s_i \\ -s_i & c_i \end{pmatrix} \begin{pmatrix} H(i,k) \\ H(i+1,k) \end{pmatrix} \rightarrow \begin{pmatrix} H(i,k) \\ H(i+1,k) \end{pmatrix}$$

(4) Compute the rotation s_k, c_k to get $H(k+1,k)$ to 0.

(5) Compute

$$\begin{pmatrix} \xi(k) \\ \xi(k+1) \end{pmatrix} \leftarrow \begin{pmatrix} c_k & s_k \\ -s_k & c_k \end{pmatrix} \begin{pmatrix} \xi_k \\ 0 \end{pmatrix}$$

$$H(k,k) \leftarrow c_k H(k,k) + s_k H(k+1,k)$$

$$H(k+1,k) \leftarrow 0$$

(6) If the residual $\beta|\xi(k+1)|$ is small enough, stop with the following solution:

- Solve $H_{k,k}y_k = \beta\xi_{k+1}$
- $x_k = x_0 + Q_k y_k$

The computational amount of one GMRES step is

$O(kn)$.

4.5 Convergence of the GMRES method

Theorem 10. *Let $A \in \mathbb{R}^{n\times n}$ positive definite, that means $x^T Ax > 0$ for every $x \in \mathbb{R}^n, x \neq 0$. Let $r_m = b - Ax_m$. Then, it holds*

$$||r_m||_2 \leq \left(1 - \frac{\lambda_{min}^2\left(\frac{A^T+A}{2}\right)}{\lambda_{max}(A^TA)}\right)^{\frac{m}{2}} ||r_0||_2$$

Proof. $x_m = x_0 + \tilde{p}(A)r_0$, where $\tilde{p} \in P_{m-1}$

$$\begin{aligned} ||r_m||_2 &= ||b - Ax_m||_2 = ||b - A(x_0 + \tilde{p}(A)r_0)||_2 = \\ &= ||r_0 - A\tilde{p}(A)r_0||_2 = ||(1 - A\tilde{p}(A))r_0||_2 \\ &= ||p(A)r_0||_2 \end{aligned}$$

whereat $p \in P_m$ with $p(0) = 1$. Now, define $q(A) = 1 - \alpha A, \quad \alpha > 0$. By the minimization property it holds

$$||r_m||_2 = ||p(A)r_0||_2 \leq ||q^m(A)r_0||_2 \leq ||q(A)||_2^m ||r_0||_2 \tag{89}$$

$$\begin{aligned} ||q(A)||_2^2 &= \sup_{x\neq 0} \frac{||(I - \alpha A)x||_2^2}{||x||_2^2} = \\ &\sup_{x\neq 0}\left(1 - 2\alpha\frac{(x, Ax)_2}{(x,x)_2} + \alpha^2\frac{(Ax, Ax)_2}{(x,x)_2}\right) \end{aligned}$$

Sine A is positive definite, it follows

$$\begin{aligned} \frac{(Ax, Ax)_2}{(x,x)_2} &= \frac{(x, A^TAx)_2}{(x,x)_2} \leq \lambda_{max}(A^TA) =: \tilde{\lambda}_{max} \\ \frac{(x, Ax)_2}{(x,x)_2} &= \frac{(x, \frac{A^T+A}{2}x)_2}{(x,x)_2} \geq \lambda_{min}(\frac{A^T + A}{2}) =: \tilde{\lambda}_{min} > 0 \end{aligned}$$

This shows:

$$||q(A)||_2^2 \leq 1 - 2\alpha\tilde{\lambda}_{min} + \alpha^2\tilde{\lambda}_{max}$$

The minimization of the right hand side leads to:

$$\alpha_{min} = \frac{\tilde{\lambda}_{min}}{\tilde{\lambda}_{max}} > 0$$

and thus:

$$\begin{aligned} 0 \leq ||q(A)||_2^2 &\leq 1 - 2\frac{\tilde{\lambda}_{min}^2}{\tilde{\lambda}_{max}} + \frac{\tilde{\lambda}_{min}^2}{\tilde{\lambda}_{max}} \\ &= 1 - \frac{\tilde{\lambda}_{min}^2}{\tilde{\lambda}_{max}} < 1 \end{aligned} \tag{90}$$

(89) and (90) show the assertion.

5 Eigenvalue Problems

5.1 Rayleigh Quotient

Let A, B symmetric, positive definite $n \times n$ matrices. The general eigenvalue problem is:
Find $\lambda \in \mathbb{R}$ and $x \in \mathbb{R}^n, x \neq 0$ such that

$$Ax = \lambda Bx$$

Example 7. *1. Let V_h be a finite element space and let $V_h \subset H_0^1(\Omega)$. Find $\lambda \in \mathbb{R}$ and $u_h \in V_h$ such that*

$$\int_\Omega \nabla u_h \nabla v_h dz = \lambda \int_\Omega u_h v_h dz$$

for every $v_h \in V_h$.

2. Eigenmodes of waveguides

$$\left[\Delta + k_0^2 \epsilon\right] u = \lambda u$$

Theorem 11. *Let λ_{min} the smallest and λ_{max} the largest eigenvalue of $B^{-1}A$. Then, it holds*

$$\begin{aligned} \min_{x \neq 0} \frac{x^T Ax}{x^T Bx} &= \lambda_{min}, \\ \max_{x \neq 0} \frac{x^T Ax}{x^T Bx} &= \lambda_{max}. \end{aligned}$$

$\frac{x^TAx}{x^TBx}$ *is called Rayleigh quotient. If*

$$\frac{\tilde{x}^TA\tilde{x}}{\tilde{x}^TB\tilde{x}} = \lambda_{min}$$

then $\tilde{x}$ *is an eigenvector with eigenvalue* λ_{min}*. If*

$$\frac{\tilde{x}^TA\tilde{x}}{\tilde{x}^TB\tilde{x}} = \lambda_{max}$$

then $\tilde{x}$ *is an eigenvector with eigenvalue* λ_{max}*.*

Proof. Choose the inner product

$$\langle x, y\rangle := x^TBy.$$

$B^{-1}A$ is symmetric with respect to $\langle\cdot,\cdot\rangle$, since

$$\left\langle B^{-1}Ax, y\right\rangle = x^TA^T\left(B^{-1}\right)^TBy = x^TAy = \left\langle x, B^{-1}Ay\right\rangle.$$

Thus, there exist $\langle\cdot,\cdot\rangle$ -orthogonal eigenvectors $e_1,\ldots,e_n$ such that eigenvalues $\lambda_1 \le \lambda_2 \le \ldots \le \lambda_n$. This means:

$$\begin{aligned} B^{-1}Ae_i &= \lambda_i e_i, \\ e_i^TBe_j &= \delta_{ij}. \end{aligned}$$

Let $x = \sum c_ie_i$. Then, we get

$$\frac{x^TAx}{x^TBx} = \frac{\sum c_i^2\lambda_i}{\sum c_i^2}. \tag{91}$$

This implies

$$\lambda_{min} \le \frac{x^TAx}{x^TBx} \le \lambda_{max}.$$

Furthermore, (91) implies that the Rayleigh quotient is maximal or minimal if and only if x is an eigenvector.
This completes the proof.

Corollary. *Let* $V \subset \mathbb{R}^n$ *be a vector space. Then,*

$$\begin{aligned} \min_{x\in V, x\neq 0}\frac{x^TAx}{x^TBx} &\ge \lambda_{min} \\ \max_{x\in V, x\neq 0}\frac{x^TAx}{x^TBx} &\le \lambda_{max} \end{aligned}$$

5.2 Method of Conjugate Gradients

Let

$$\lambda(x) := \frac{x^T Ax}{x^T Bx}.$$

Lemma 11. *The gradient and Hesse-matrix of $\lambda(x)$ are*

$$\begin{aligned} g(x) &= \frac{2}{x^T Bx}(Ax - \lambda(x)Bx) \\ H(x) &= \frac{2}{x^T Bx}(A - \lambda(x)B - Bxg(x)^T - g(x)x^T B). \end{aligned}$$

Proof.

$$\begin{aligned} g(x) &= 2Ax\frac{1}{x^T Bx} - \frac{x^T Ax}{(x^T Bx)^2}2Bx \\ &= \frac{2}{x^T Bx}(Ax - \lambda(x)Bx). \end{aligned}$$

$$\begin{aligned} H(x) &= -\frac{2}{(x^T Bx)^2}2Bx(Ax - \lambda(x)Bx)^T \\ &\quad + \frac{2}{x^T Bx}(A - g(x)(Bx)^T - B\lambda(x)) \\ &= \frac{2}{x^T Bx}(A - \lambda(x)B - Bxg(x)^T - g(x)x^T B). \end{aligned}$$

□

Lemma 12. *Let λ_1 be the maximal eigenvalue of $B^{-1}A$ and λ_n be the minimal eigenvalue of $B^{-1}A$. Then*
$H(e_1)$ is positive definite and
$H(e_n)$ is negative definite.

Proof. By Theorem 11, $g(e_1) = 0$ and $g(e_n) = 0$. Then, a simple calculation completes the proof. □

One can consider

$$\tilde{\lambda}(x + h) := \lambda(x) + g(x)^T h + \frac{1}{2}h^T H(x)h$$

as an approximation of the functional $\lambda(x + h)$.

We are looking for an approximation of e_1 and e_n, where we assume

$$\lambda_1 < \lambda_2 \leq ... \leq \lambda_{n-1} < \lambda_n.$$

are the eigenvalues corresponding e_i. Starting with x_0, we construct a sequence (x_k) which converges to e_1 and e_n. By Theorem 11, we have to find the extreme values of $\lambda(x)$. Thus, let us define

$$x_{k+1} = x_k + \alpha_k s_k$$

such that

$$\frac{\partial \lambda(x_{k+1})}{\partial \alpha_k} = g(x_{k+1})^T s_k = 0,$$

where s_k is a search direction.

Lemma 13. *The equation $g(x_{k+1})^T s_k = 0$ leads to a quadratic equation with respect to α_k.*

Proof.

$$\begin{aligned} g(x_{k+1})^T s_k &= 0 \\ &\Downarrow \\ s_k^T(A(x_k + \alpha_k s_k) - \lambda(x_k + \alpha_k s_k)B(x_k + \alpha_k s_k)) &= 0 \\ (x_k + \alpha_k s_k)^T B(x_k + \alpha_k s_k) s_k^T A(x_k + \alpha_k s_k) & \\ -s_k^T B(x_k + \alpha_k s_k)(x_k + \alpha_k s_k)^T A(x_k + \alpha_k s_k) &= 0 \end{aligned}$$

The term α_k^3 in this equation cancels.

Construction of the search direction s_k:
Let us assume, we are looking for an approximation of $\lambda_{\min} = \lambda_1$.

1.Choice of s_k: Gradient method:

$$s_k := -g(x_k)^T.$$

To find a better search direction, let us construct s_k such that

$$s_k := v + \beta w \quad \text{such that: } g(x_k)^T w = 0.$$

Now, consider the $\tilde{\lambda}(x + \alpha_k s_k)$

$$\begin{aligned}
\tilde{\lambda}(x_k + \alpha_k s_k) &= \lambda(x_k) + g(x_k)^T \alpha_k s_k + \frac{1}{2}(\alpha_k s_k)^T H(x)(\alpha_k s_k) \\
&= \lambda(x_k) + g(x_k)^T (v + \beta w)\alpha_k + \\
&\quad \frac{1}{2}\alpha_k^2 (v + \beta w)^T H(x)(v + \beta w) \\
&= \lambda(x_k) + g(x_k)^T v\alpha_k + \\
&\quad \frac{1}{2}\alpha_k^2 \left(v^T H(x)v + \beta 2 v^T H(x)w + \beta^2 w^T H(x)w\right)
\end{aligned}$$

To minimize $\tilde{\lambda}(x_k + \alpha_k s_k)$, let us choose β such that

$$\frac{\partial \tilde{\lambda}(x_k + \alpha_k s_k)}{\partial \beta} = 0.$$

This implies

$$\begin{aligned}
v^T Hw + \beta w^T Hw &= 0 \\
&\Downarrow \\
(v + \beta w)HW &= 0 \\
&\Downarrow \\
\beta &= -\frac{v^T Hw}{w^T Hw}.
\end{aligned}$$

By Lemma 11 and $g(x_k)^T w = 0$, we obtain

$$\beta = \frac{v^T (A - \lambda(x_k)B)w - v^T g(x_k) x_k^T Bw}{w^T (A - \lambda(x_k)B)w}.$$

By Lemma 13 we can choose v and w as follows:
2.Choice of s_k: Conjugate gradient method:

$$\begin{aligned}
s_k &:= v + \beta w \\
v &= -g(x_k) \\
w &= s_{k-1}.
\end{aligned}$$

3.Choice of s_k: Another method:

$$\begin{aligned}
s_k &:= v + \beta w \\
v &= r = Ax_k - \lambda(x_k)Bx_k \\
w &= s_{k-1} - \frac{x_k^T Bs_{k-1}}{x_k^T Bx_k} x_k.
\end{aligned}$$

Here: $x_k^T Bw = 0$ but not $g(x_k)^T w = 0$.

5.3 Simple Vector Iteration

Let A be a $n \times n$ matrix. For reasons of simplicity, we assume that A is diagonalizable. Similar results hold for general matrices.

Let us assume that $\lambda_1, \ldots, \lambda_n$ are eigenvalues of A with orthonormal eigenvectors e_i. Furthermore, let us assume that:
(i) $|\lambda_1| = \ldots = |\lambda_r| > |\lambda_{r+1}| \geq \ldots \geq |\lambda_n|$
(ii) $\lambda_1 = \ldots = \lambda_r$
(iii) Let x_0 be a start vector such that :
$x_0 = \sum_{i=1}^{n} c_i e_i, \quad E := c_1 e_1 + \ldots + c_r e_r \neq 0.$

Algorithm: Vector Iteration:

$$\begin{aligned} x_{i+1} &= Ax_i \\ \tilde{x_i} &= \frac{x_i}{||x_i||} \end{aligned}$$

Theorem 12. *Let us assume that A is symmetric positive definite.*

$$\left|\left| \frac{x_i}{\lambda_1^i} - E \right|\right| \leq \left| \frac{\lambda_{r+1}}{\lambda_1} \right|^i ||x - E||$$

$\lim_{i\to\infty} \frac{x_i}{\lambda_1^i} = E, \quad \lim_{i\to\infty} \frac{\|\tilde{x}_{i+1}\|_2}{\|\tilde{x}_i\|_2} = |\lambda_1| \text{ and } |\frac{\lambda_{r+1}}{\lambda_1}| < 1.$

Proof. First, observe that

$$\frac{x_i}{\lambda_1^i} = c_1 e_1 + \ldots + c_r e_r + c_{r+1} \left(\frac{\lambda_{r+1}}{\lambda_1} \right)^i e_{r+1} + \ldots + c_n \left(\frac{\lambda_n}{\lambda_1} \right)^i e_n.$$

This implies

$$\begin{aligned} \left\| \frac{x_i}{\lambda_1^i} - E \right\|_2 &= \left\| c_{r+1} \left(\frac{\lambda_{r+1}}{\lambda_1} \right)^i e_{r+1} + \ldots + c_n \left(\frac{\lambda_n}{\lambda_1} \right)^i e_n \right\|_2 \\ &= \left(\sum_{j=r+1}^{n} |c_j|^2 \left| \frac{\lambda_j}{\lambda_1} \right|^{2i} \right)^{\frac{1}{2}} \\ &\leq \left(\sum_{j=r+1}^{n} |c_j|^2 \left| \frac{\lambda_{r+1}}{\lambda_1} \right|^{2i} \right)^{\frac{1}{2}} \\ &= \left| \frac{\lambda_{r+1}}{\lambda_1} \right|^i \|x - E\|_2 . \end{aligned}$$

Modification of the vector iteration

- $x_{i+1} = A^{-1} x_i$ $\rightarrow$ leads to the smallest eigenvalue
- $x_{i+1} = (2E\lambda_{max} - A)\, x_i$ For symmetric positive definite matrices this also leads to the smallest eigenvalue
- $x_{i+1} = (A - \lambda I)^{-1} x_i$ only works, if λ is close to an eigenvalue λ_j, that means

$$|\lambda_j - \lambda| << |\lambda_k - \lambda|$$

One obtains numerical problems, if the eigenvalues are very close to each other (cluster). In this case, one has to find a group of orthogonal eigenvectors.

5.4 Computation of Eigenvalues using the Rayleigh Quotient

Let V_k be a sub-vector space of the $\mathbb{R}^n$ and let A be symmetric positive definite. Then,

$$\mu_1 = \min_{x \in V_k} \frac{x^T A x}{x^T x} \tag{92}$$

is an approximation of the smallest and

$$\mu_2 = \max_{x \in V_k} \frac{x^T A x}{x^T x} \tag{93}$$

an approximation of the largest eigenvalue. If $k << n$, then the eigenvalue problem (92) is less difficult to solve than the original eigenvalue problem eigenvalue problem

$$\min_{x \in V_n} \frac{x^T A x}{x^T x} \tag{94}$$

(92) can be solved by vector iteration, a direct solver, QR-algorithm, or any other direct solver.

Theorem 13. *Let $V_k = span\{d_0, Ad_0, \ldots, A^k d_0\}$ and let us assume that the eigenvalues of A are numbered as follows*

$$\lambda_1 = \lambda_2 = ... = \lambda_{r-1} < \lambda_r \leq \lambda_{r+1} \leq \ldots \lambda_n,$$

where $r \geq 2$. Let e_i be the corresponding eigenvectors. Now, define

$$Z_1 = \text{span}\{e_1, \ldots, e_{r-1}\}.$$

Then, it holds

$$0 \leq \mu_1 - \lambda_1 \leq (\lambda_n - \lambda_1) \left(\frac{\tan \phi_1}{T_k\left(\frac{\kappa_r + 1 - 2\frac{\lambda_1}{\lambda_r}}{\kappa_r - 1} \right)} \right)^2,$$

where $\kappa_r = \frac{\lambda_n}{\lambda_r}$, $\frac{\kappa_r + 1 - 2\frac{\lambda_1}{\lambda_r}}{\kappa_r - 1} > 1$,

$$T_k(x) \geq \frac{1}{2}\left(x + \sqrt{x^2 - 1}\right)^k,$$

and

$$\cos \phi_1 = \max_{z_1 \in Z_1} \frac{|d_0^T A z_1|}{||d_0||_A ||z_1||_A}.$$

So ϕ_1 is the angle between d_0, Z. To calculate the largest eigenvalue, let us use the following abbreviation:
$\lambda_1 \leq \ldots \leq \lambda_r < \lambda_{r+1} = \ldots = \lambda_n$ *and* $\kappa_r = \frac{\lambda_r}{\lambda_1}$. *Then, we get:*

$$\lambda_n - \mu_n \leq (\lambda_n - \lambda_1) \left(\frac{\tan \phi_1}{T_k\left(\frac{2\kappa - \kappa_r - 1}{\kappa - \kappa_r}\right)} \right)^2$$

Proof. Let e_i be the normalized eigenvectors of A, $Ae_i = \lambda_i e_i$. Then, it follows

$$\begin{aligned} d_0 &= \sum_{i=1}^{n} c_i e_i \\ \mu_1 &= \min_{x \in V_k} \frac{x^T A x}{x^T x} = \min_{p \in P_k} \frac{(p(A)d_0)^T A p(A) d_0}{(p(A)d_0)^T (p(A)d_0)} = \\ &= \min_{p \in P_k} \frac{\sum_{i=1}^n c_i^2 \lambda_i p(\lambda_i)^2}{\sum_{i=1}^n c_i^2 p(\lambda_i)^2} \end{aligned}$$

This implies

$$\begin{aligned} 0 &\leq \mu_1 - \lambda_1 \leq \frac{\sum_{i=1}^n c_i^2 (\lambda_i - \lambda_1) p(\lambda_i)^2}{\sum_{i=1}^n c_i^2 p(\lambda_i)^2} = \\ &= \frac{\sum_{i=r}^n c_i^2 (\lambda_i - \lambda_1) p(\lambda_i)^2}{\sum_{i=1}^n c_i^2 p(\lambda_i)^2} \leq \\ &\leq (\lambda_n - \lambda_1) \frac{\sum_{i=r}^n c_i^2 p(\lambda_i)^2}{\sum_{i=1}^n c_i^2 p(\lambda_i)^2} = \\ &= (\lambda_n - \lambda_1) \frac{1}{1 + \frac{p(\lambda_1)^2 \sum_{i=1}^{r-1} c_i^2}{\sum_{i=r}^n c_i^2 p(\lambda_i)^2}} \end{aligned}$$

for every polynomial p. To estimate the smallest eigenvalue choose:

$$p(\lambda) = T_k \left(\frac{\lambda_n + \lambda_r - 2\lambda}{\lambda_n - \lambda_r} \right)$$

Then, it holds

$$|p(\lambda_n)| = 1, \quad |p(\lambda_i)| \leq 1 \text{ for } i = r, r+1, \ldots, n-1$$

Thus, we get

$$\begin{aligned} 0 &\leq \mu_1 - \lambda_1 \leq (\lambda_n - \lambda_1) \frac{1}{1 + \frac{p(\lambda_1)^2 \sum_{i=1}^{r-1} c_i^2}{\sum_{i=r}^n c_i^2}} \\ &\leq (\lambda_n - \lambda_1) \frac{\sum_{i=r}^n c_i^2}{\sum_{i=1}^{r-1} c_i^2} \frac{1}{T_k\left(\frac{\lambda_n + \lambda_r - 2\lambda_1}{\lambda_n - \lambda_r}\right)^2} \end{aligned}$$

$$T_k\left(\frac{\lambda_n+\lambda_r-2\lambda_1}{\lambda_n-\lambda_r}\right) = T_k\left(\frac{\kappa_r+1-2\frac{\lambda_1}{\lambda_r}}{\kappa_r-1}\right)$$

$$\frac{\sqrt{\sum_{i=r}^{n} c_i^2}}{\sqrt{\sum_{i=1}^{r-1} c_i^2}} = \tan\phi_1.$$

To estimate the largest eigenvalue choose:

$$p(\lambda) = T_k\left(\frac{2\lambda-\lambda_r-\lambda_1}{\lambda_n-\lambda_r}\right),$$

where the eigenvalues are $\lambda_1 \leq \ldots \lambda_r < \lambda_{r+1} = \ldots = \lambda_n$. Then, we get

$$\begin{aligned} 0 &\leq \lambda_n - \mu_n \leq \frac{\sum_{i=1}^{n} c_i^2(\lambda_n-\lambda_i)p(\lambda_i)^2}{\sum_{i=1}^{n} c_i^2 p(\lambda_i)^2} \\ &\leq (\lambda_n-\lambda_1)\frac{\sum_{i=1}^{r} c_i^2}{\sum_{i=r+1}^{n} c_i^2}\frac{1}{T_k\left(\frac{2\lambda_n-\lambda_r-\lambda_1}{\lambda_n-\lambda_r}\right)^2} \\ &\leq (\lambda_n-\lambda_1)\tan\phi_1\frac{1}{T_k\left(\frac{2\kappa-\kappa_r-1}{\kappa-\kappa_r}\right)^2}. \end{aligned}$$

Example: Poisson's Equation
Let us discretize Poisson's equation by finite differences. Then, the eigenvalues of the matrix A are:

$$\lambda_{\nu,\mu} = \frac{4}{h^2}\left(\sin^2\left(\frac{\pi\nu h}{2}\right)+\sin^2\left(\frac{\pi\mu h}{2}\right)\right)$$

The smallest eigenvalue is at $\mu=\nu=1$:

$$\begin{aligned} \lambda_1 &= \frac{4}{h^2}\cdot 2\left(\frac{\pi^2h^2}{4}\right) = 2\pi^2 \\ \lambda_2 &= \frac{4}{h^2}\cdot\left(\frac{\pi^2h^2}{4}+4\frac{\pi^2h^2}{4}\right) = 5\pi^2 \end{aligned}$$

Our aim is to find the smallest eigenvalue λ_{min} of A. There are to ways to get an approximation of λ_{min} by minimizing the Rayleigh quotient.

1. Application of the Rayleigh quotient to A^{-1}:

$$\lim_{n\to\infty} T_k\left(\frac{2\kappa-\kappa_r-1}{\kappa-\kappa_r}\right) = T_k(2)$$

$$T_k(2) \geq \frac{1}{2}(2+\sqrt{3})^k \geq \frac{1}{2}(3.7)^k.$$

This implies fast convergence of the smallest eigenvalue of A, by using the inverse iteration applied to A^{-1}.

2. Application of the Rayleigh quotient to A:

$$\lim_{n\to\infty} T_k\left(\frac{\lambda_n + \lambda_r - 2\lambda_1}{\lambda_n - \lambda_r}\right) = T_k(1) = 1.$$

This implies low convergence of the smallest eigenvalue of A for large n! This shows that the first approach is better!!!

5.5 Jacobi-Davidson-Algorithm

5.5.1 The Jacobi-Method

Let the coordinate system be transformed such that

$$e_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

is a "good " approximation of an eigenvector. We want to solve the problem

$$A \begin{pmatrix} 1 \\ z \end{pmatrix} = \lambda \begin{pmatrix} 1 \\ z \end{pmatrix}$$

$z \in \mathbb{C}^{n-1}, \lambda \in \mathbb{C}$.

Idea: Apply Newton method to

$$A \begin{pmatrix} 1 \\ z \end{pmatrix} - \lambda \begin{pmatrix} 1 \\ z \end{pmatrix} =: f \begin{pmatrix} \lambda \\ z \end{pmatrix}.$$

Then, we have to calculate

$$\begin{pmatrix} \lambda_{n+1} \\ z_{n+1} \end{pmatrix} := \begin{pmatrix} \lambda_n \\ z_n \end{pmatrix} - \left(f' \begin{pmatrix} \lambda_n \\ z_n \end{pmatrix} \right)^{-1} f \begin{pmatrix} \lambda_n \\ z_n \end{pmatrix}.$$

The calculation of $\left(f' \begin{pmatrix} \lambda_n \\ z_n \end{pmatrix} \right)^{-1}$ is difficult. Since, e_1 is a "good " approximation of an eigenvector, we can define the following approximative Newton method:

$$\begin{pmatrix} \lambda_{n+1} \\ z_{n+1} \end{pmatrix} := \begin{pmatrix} \lambda_n \\ z_n \end{pmatrix} - \left(f' \begin{pmatrix} \lambda_n \\ 0 \end{pmatrix} \right)^{-1} f \begin{pmatrix} \lambda_n \\ z_n \end{pmatrix}.$$

Let us find a short formula for this iteration. To this end, let $A = \begin{pmatrix} \alpha & c^T \\ b & F \end{pmatrix}$. Then, we get

$$f' \begin{pmatrix} \lambda_n \\ 0 \end{pmatrix} = \begin{pmatrix} 0 & c^T \\ 0 & F \end{pmatrix} - \begin{pmatrix} 1 & 0 \\ 0 & \lambda_n E \end{pmatrix}$$

Let us abbreviate $f \begin{pmatrix} \lambda_n \\ z_n \end{pmatrix} = \begin{pmatrix} p \\ w \end{pmatrix}$. Then, let q, u be such that

$$\left(f' \begin{pmatrix} \lambda_n \\ 0 \end{pmatrix} \right) \begin{pmatrix} q \\ u \end{pmatrix} = \begin{pmatrix} p \\ w \end{pmatrix}.$$

This implies

$$\begin{aligned}
\begin{pmatrix} c^T u - q \\ (F - \lambda_n E)u \end{pmatrix} &= \begin{pmatrix} p \\ w \end{pmatrix} \Rightarrow u = (F - \lambda_n E)^{-1} w, q = c^T u - p \\
\begin{pmatrix} p \\ w \end{pmatrix} &= f \begin{pmatrix} \lambda_n \\ z_n \end{pmatrix} = \begin{pmatrix} \alpha + c^T z_n \\ b + F z_n \end{pmatrix} - \begin{pmatrix} \lambda_n \\ \lambda_n z_n \end{pmatrix} \\
&\Rightarrow \begin{array}{l} p = \alpha + c^T z_n - \lambda_n \\ w = b + F z_n - \lambda_n z_n = b + (F - \lambda_n E) z_n \end{array} \\
u &= (F - \lambda_n E)^{-1} \left(b + (F - \lambda_n E) z_n \right) = (F - \lambda_n E)^{-1} b + z_n
\end{aligned}$$

$$\boxed{z_{n+1} = z_n - u = (F - \lambda_n E)^{-1}(-b)}$$

$$\begin{aligned}
q &= c^T \left((F - \lambda_n E)^{-1} b + z_n \right) - \alpha - c^T z_n + \lambda_n \\
&= c^T \left((F - \lambda_n E)^{-1} b \right) - \alpha + \lambda_n
\end{aligned}$$

$$\boxed{\lambda_{n+1} = \lambda_n - q = c^T z_{n+1} + \alpha}$$

SIAM Review, June 2000, Vol. 42, Number 2.

Instead of inverting $F - \lambda_n E$ exactly, one can approximate $F - \lambda_n E$ by the diagonal. This means we apply the Jacobi iteration for solving $z_{n+1} = (F - \lambda_n E)^{-1}(-b)$ as follows:

$$\begin{aligned}
\text{„}(F - \lambda_n E) z_{n+1} &= -b \Rightarrow \text{“} \\
&\left\{ \begin{array}{c} (D - \lambda_n E) z_{n+1} = (D - F) z_n - b \\ \lambda_{n+1} = c^T z_{n+1} + \alpha \end{array} \right\}
\end{aligned}$$

By changing the notation of λ_n and λ_{n+1} this leads to

$$\boxed{\begin{array}{c} \lambda_n = \alpha + c^T z_n \\ (D - \lambda_n E) z_{n+1} = (D - F) z_n - b \end{array}}$$

5.5.2 Motivation of Davidson's Algorithm

Convergence of Eigenvalues, Eigenvectors.
Let $V \subset H^1(\Omega)$ be a Hilbert space and $a(\cdot, \cdot) : V \times V \to \mathbb{R}$ V-koerziv. Let V_n be spaces such that

$$\lim_{n \to \infty} \inf \left\{ ||u - u^h||_V \middle| u^h \in V_n \right\} = 0 \quad \forall u \in V$$

Let $e \in V$, $e \neq 0$ and $\lambda \in \mathbb{C}$ such that

$$a(e, v) = \lambda \int_\Omega ev \; d\mu \qquad \forall v \in V. \tag{95}$$

Let $e_n \in V_n, \lambda_n \in \mathbb{C}$ such that

$$a(e_n, v_n) = \lambda_n \int_\Omega e_n v_n d\mu \qquad \forall v_n \in V_n. \tag{96}$$

Theorem 14. *If λ is a single eigenvalue, then there is a constant c and a sequence (e_n, λ_n) such that*

$$||e - e_n||_V \leq c d(e, V_n)$$

Connection to the matrix eigenvalue problem
Let A be a matrix which describes $a(\cdot, \cdot)$ with respect to an L^2-orthogonal basis. Then (95) and (96), are equivalent to

$$\begin{aligned} A\vec{e} &= \lambda \vec{e} \\ (A\vec{e_n} - \lambda_n \vec{e_n}) \perp \vec{v_n} \qquad & \vec{e_n} \in \vec{V_n}, \quad \forall v_n \in \vec{V_n} \end{aligned}$$

λ_n is called Ritz value of A with Ritz vector $\vec{e_n} \in \vec{V_n}$. Furthermore λ_n is an eigenvalue of the matrix $B_n = (b_{ij})$, where $b_{ij} = a(v_i, v_j)$ and $(v_i)_i$ a basis of $\vec{V_j}$. There is an eigenvector ξ_n of B_n with eigenvalue λ_n such that

$$\vec{e_n} = \sum_i \xi_n^i v_i.$$

Davidson's Idea:
Choose the optimal eigenvector from the subspace V as a new approximate eigenvector. This is the Ritz vector. By increasing V one gets an approximation of the exact eigenvector.

5.5.3 The concept of the Jacobi-Davidson-Algorithm

Idea A: Compute the optimal eigenvector „Ritz vector“ and „Ritz value“ λ in the subspace V_k
Idea B: Enlarge the subspace $V_k \to V_{k+1}$ by the Idea of „Jacobi“ orthogonal to the old „Ritz vector“ by a Newton step on

$$„A \begin{pmatrix} 1 \\ z \end{pmatrix} - \lambda \begin{pmatrix} 1 \\ z \end{pmatrix} = 0“$$

Orthogonalize the new vector t with respect to V_k and build V_{k+1}.

In the Jacobi method we had

$$\begin{aligned} A\begin{pmatrix} 1 \\ z \end{pmatrix} - \lambda \begin{pmatrix} 1 \\ z \end{pmatrix} &= \begin{pmatrix} 0 \\ 0 \end{pmatrix} \\ A &= \begin{pmatrix} \alpha & c^T \\ b & F \end{pmatrix} \end{aligned}$$

Let us denote

$$A\begin{pmatrix} 1 \\ 0 \end{pmatrix} - \lambda \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} \alpha - \lambda \\ b \end{pmatrix} = r$$

the residual. In the Jacobi method, one has to compute

$$(F - \lambda_n E)^{-1}(-b)$$

.

We have to describe this in suitable spaces.

Let $\hat{u}$ be an approximation of the eigenvalue θ. Let us rotate the coordinate system such that

$$\begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \hat{=} \hat{u}.$$

Now,

$$\left\{ \begin{pmatrix} 0 \\ * \\ \vdots \\ * \end{pmatrix} \right\}$$

corresponds to a space T, which is orthogonal to $\hat{u}$. This means:

$$V = \mathbb{C}\hat{u} \oplus T \quad \text{and} \quad \mathbb{C}\hat{u} \perp T$$

Now we can describe "$\tilde{t} := (F - \lambda_n E)^{-1}(-b)$" in suitable spaces. We have to find a $t \in T$, $\tilde{\epsilon}\mathbb{C}$ such that

$$(A - \theta E)t = -b + \tilde{\epsilon}\hat{u}$$

where $b \perp \hat{u}$ and $r - b \in \mathbb{C}\hat{u}$. This is equivalent to $t \in T$, $\epsilon \in \mathbb{C}$, and

$$(A - \theta E)t = -r + \epsilon\hat{u} \tag{97}$$

This equation can approximatively be solved as follows.

Let M^{-1} be a preconditioner for $A - \theta E$.

This means that M is an approximation of $A - \theta E$.

Thus, instead of solving (97), we are looking for a $\hat{t} \in T$ such that

$$M\hat{t} = -r + \epsilon\hat{u}$$

This leads to $\hat{t} = -M^{-1}r + M^{-1}\epsilon\hat{u}$. Since $\hat{t} \in T$, we obtain:

$$\begin{aligned} 0 &= -\hat{u}M^{-1}r + \hat{u}M^{-1}\epsilon\hat{u} \\ &\Downarrow \\ \epsilon &= \frac{\hat{u}^*M^{-1}r}{\hat{u}^*M^{-1}\hat{u}} \end{aligned}$$

Now, one can solve

$$\hat{t} = M^{-1}(-r + \epsilon\hat{u}).$$

5.5.4 Jacobi-Davidson-Algorithm

Step 1. Start: Choose a non trivial start vector v.
Calculate $v_1 = v_/\|v\|$, $w_1 = Av_1$.
$h_{11} = v_1^* w_1$.
Set $V_1 = \mathbb{R}v_1$, $W_1 = \mathbb{R}w_1$, $H_1 = h_{11}$.
$u = v_1$, $\theta = h_{11}$.
Calculate $v = w_1 - \theta u$.

Step 2. Iterate until convergence:

Step 3. Inner loop: For $k = 1, ..., m-1$:

- Let M be an approximation of $A - \theta E$. Calculate:

$$\epsilon = \frac{\hat{u}^*M^{-1}r}{\hat{u}^*M^{-1}\hat{u}}, \qquad t = M^{-1}(-r + \epsilon u).$$

- Orthogonalize t with respect to V_k by Gram-Schmidt.
  This leads to the vector t^{ortho}.
  Extend V_k by t to obtain V_{k+1}.

- Calculate $w_{k+1} = Av_{k+1}$ and
  extend W_k by w_{k+1} to obtain W_{k+1}.
- Calculate $V_{k+1}^* w_{k+1}$ and $v_{k+1}^* W_{k+1}$.
  Then the whole matrix $H_{k+1} := V_{k+1}^* A V_{k+1}$ is computed.
- Calculate the largest eigenvalue of θ with eigenvector s of H_{k+1}
  (where $\|s\| = 1$).
- Calculate the Ritz vector $u := V_{k+1}s$.
  Calculate $\hat{u} := Au$ (this is $W_{k+1}s$)
  Calculate the residuum $r := \hat{u} - \theta u$.
- Test convergence. Stop if $\|r\|$ is small enough..

Step 4. Restart: If $\|r\|$ can not be reduced any more, then set:

Set $V_1 = \mathbb{R}u$, $W_1 = \mathbb{R}\hat{u}$, $H_1 = \theta$.
Goto Step 3.