Scientific Computing TU Berlin Winter 2021/22 © Jürgen Fuhrmann
Notebook 13

```
using LinearAlgebra
```

Iterative methods

Iterative methods

Let $V = \mathbb{R}^n$ be equipped with the inner product $(\cdot, \cdot)$. Let A be an $n \times n$ nonsingular matrix.

Solve $Au = b$ iteratively. For this purpose, two components are needed:

- **Preconditioner**: a matrix $M \approx A$ "approximating" the matrix A but with the property that the system $Mv = f$ is easy to solve
- **Iteration scheme**: algorithmic sequence using M and A which updates the solution step by step

Simple iteration scheme

Assume we know the exact solution $\hat{u}$: $A\hat{u} = b$.

Then it must fulfill the identity

$$\hat{u} = \hat{u} - M^{-1}(A\hat{u} - b)$$

$\Rightarrow$ iterative scheme: put the "old" value on the right hand side and the ""new" value on the left hand side:

$$u_{k+1} = u_k - M^{-1}(Au_k - b) \quad (k = 0, 1 \ldots)$$

Obviously, if $u_k = \hat{u}$, the process would be stationary.

Otherwise it leads to a sequence of approximations

$$u_0, u_1, \ldots, u_k, u_{k+1}, \ldots$$

Implementation: solve $Au = b$ with tolerance ε:

1. Choose initial value u_0, set $k = 0$
2. Calculate *residuum* $r_k = Au_k - b$
3. Test convergence: if $||r_k|| < \varepsilon$ set $u = u_k$, finish
4. Calculate *update*: solve $Mv_k = r_k$
5. Update solution: $u_{k+1} = u_k - v_k$, set $k = k + 1$, repeat with step 2.

The Jacobi method

- Let $A = D - E - F$, where D: main diagonal, E: negative lower triangular part F: negative upper triangular part
- Preconditioner: $M = D$, where D is the main diagonal of $A \Rightarrow$

$$u_{k+1,i} = u_{k,i} - \frac{1}{a_{ii}}\left(\sum_{j=1\ldots n} a_{ij}u_{k,j} - b_i\right) \quad (i = 1 \ldots n)$$

- Equivalent to the succesive (row by row) solution of

$$a_{ii}u_{k+1,i} + \sum_{j=1\ldots n, j\neq i} a_{ij}u_{k,j} = b_i \quad (i = 1 \ldots n)$$

- Already calculated results not taken into account
- Variable ordering does not matter

```
jacobi_sweep1! (generic function with 1 method)
```

```
jacobi_sweep1!(unew,uold,A,b)= unew.= uold - inv(Diagonal(A))*(A*uold-b)
```

```
jacobi_sweep2! (generic function with 1 method)
```

```
function jacobi_sweep2!(unew,uold,A,b)
        n=size(A,2)
        for i=1:n
            unew[i]=uold[i]+b[i]/A[i,i]
            for j=1:n
                unew[i]-=A[i,j]*uold[j]/A[i,i]
            end
        end
end
```

```
simple_iteration (generic function with 1 method)
```

```
function simple_iteration(A,b,u0, sweep;tol=1.0e-10,maxiter=100)
    unew=similar(u0)
    uold=copy(u0)
    residual=Inf
    history=zeros(0)
    for i=1:maxiter
        sweep(unew,uold,A,b)
        residual=norm(A*unew-b)
        push!(history,residual)
        if residual<tol
            return unew,history
        end
        uold.=unew
    end
    return unew,history
end
```

```
N = 100
```

```
N=100
```

```
randmatrix (generic function with 1 method)
```

```
randmatrix(N;δ=1.0)=Diagonal(ones(N))-rand(N,N)/(δ*N)
```

```
A=randmatrix(N,δ=1);
```

```
û=rand(N);
```

```
b=A*û;
```

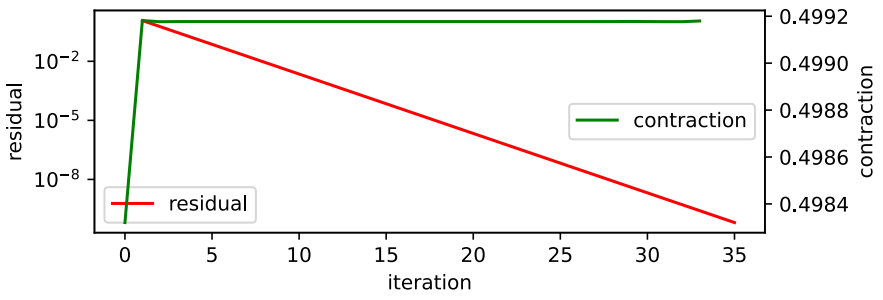
```
0.043894549
```

```
@elapsed (u,history)=simple_iteration(A,b,zeros(N),jacobi_sweep2!)
```

```
1.2915606225074314e-10
```

```
norm(u-û)
```

```
plothistory (generic function with 1 method)
```




```
plothistory(history)
```

The Gauss-Seidel method

-Solve for main diagonal element row by row

- Take already calculated results into account
- Run in ascending order: forward GS

$$a_{ii}u_{k+1,i} + \sum_{j<i} a_{ij}u_{k+1,j} + \sum_{j>i} a_{ij}u_{k,j} = b_i \qquad (i = 1 \dots n)$$
$$(D - E)u_{k+1} - Fu_k = b$$
$$M = D - E$$

- Run in descending order: backward GS

$$a_{ii}u_{k+1,i} + \sum_{j>i} a_{ij}u_{k+1,j} + \sum_{j<i} a_{ij}u_{k,j} = b_i \qquad (i = n \dots 1)$$
$$(D - F)u_{k+1} - Eu_k = b$$
$$M = D - F$$

- May be it is faster ?
- Variable order probably matters

```
gauss_seidel_sweep (generic function with 1 method)
```

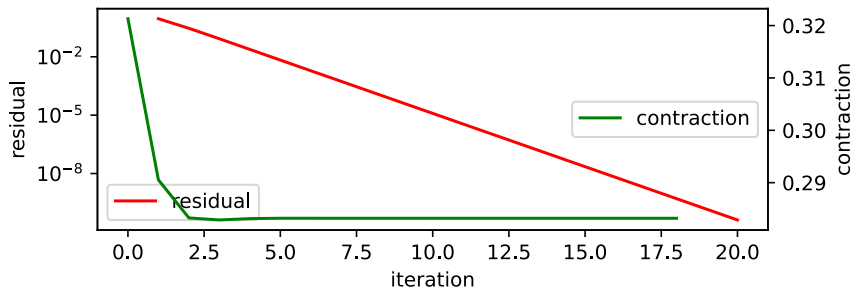
```
function gauss_seidel_sweep(unew,uold,A,b)
        n=size(A,2)
        unew.=uold # we could do this in-place and save space for one vector
        for i=1:n
            unew[i]=b[i]/A[i,i]
            for j=1:i-1
                unew[i]-=A[i,j]*unew[j]/A[i,i]
            end
            for j=i+1:n
                unew[i]-=A[i,j]*unew[j]/A[i,i]
            end
        end
end
```

```
0.067975229
```

```
@elapsed (u_gs,history_gs)=simple_iteration(A,b,zeros(N),gauss_seidel_sweep)
```

```
7.127068274648113e-11
```

```
norm(u_gs-û)
```




```
plothistory(history_gs)
```

Richardson method

$$M = \frac{1}{\alpha} I$$

for some well chosen α

Variants

- SOR: Successive overrelaxation: solve $\omega A = \omega B$ and use splitting

$$\begin{aligned}\omega A &= (D - \omega E) - (\omega F + (1 - \omega D))\\ (D - \omega E)u_{k+1} &= (\omega F + (1 - \omega D))u_k + \omega b\\ M &= \frac{1}{\omega}(D - \omega E)\end{aligned}$$

- SSOR: Symmetric successive overrelaxation

$$\begin{aligned}(D - \omega E)u_{k+\frac{1}{2}} &= (\omega F + (1 - \omega D))u_k + \omega b\\ (D - \omega F)u_{k+1} &= (\omega E + (1 - \omega D))u_{k+\frac{1}{2}} + \omega b\end{aligned}$$

$$M = \frac{1}{\omega(2 - \omega)}(D - \omega E)D^{-1}(D - \omega F)$$

- Block methods
  - Jacobi, Gauss-Seidel, (S)SOR methods can as well be used block-wise, based on a partition of the system matrix into larger blocks,
  - The blocks on the diagonal should be square matrices, and invertible
  - Interesting variant for systems of partial differential equations, where multiple species interact with each other

Convergence

Let $\hat{u}$ be the solution of $Au = b$.

Let $e_k = u_k - \hat{u}$ be the error of the k-th iteration step. Then:

$$\begin{aligned}u_{k+1} &= u_k - M^{-1}(Au_k - b)\\ &= (I - M^{-1}A)u_k + M^{-1}b\\ u_{k+1} - \hat{u} &= u_k - \hat{u} - M^{-1}(Au_k - A\hat{u})\\ &= (I - M^{-1}A)(u_k - \hat{u})\\ &= (I - M^{-1}A)^k(u_0 - \hat{u})\end{aligned}$$

resulting in

$$e_{k+1} = (I - M^{-1}A)^k e_0$$

- So when does $(I - M^{-1}A)^k$ converge to zero for $k \to \infty$?
- Denote $B = I - M^{-1}A$

Jordan canonical form and spectral radius

Notations:

- λ_i $(i = 1 \dots p)$: eigenvalues of B
- $\sigma(B) = \{\lambda_1 \dots \lambda_p\}$: spectrum of B
- μ_i: algebraic multiplicity of λ_i: multiplicity as zero of the characteristic polynomial $\det(B - \lambda I)$
- γ_i geometric multiplicity of λ_i: dimension of $\mathrm{Ker}(B - \lambda I)$
- l_i: index of the eigenvalue: the smallest integer for which $\mathrm{Ker}(B - \lambda I)^{l_i+1} = \mathrm{Ker}(B - \lambda I)^{l_i}$
- $l_i \leq \mu_i$

Theorem (Saad, Th. 1.8) B can be transformed to a block diagonal matrix consisting of p diagonal blocks $D_1 \dots D_p$, each associated with a distinct eigenvalue λ_i.

- Each of the diagonal blocks D_i has itself a block diagonal structure consisting of γ_i *Jordan blocks* $J_{i,1} \dots J_{i,\gamma_i}$.
- Each of the Jordan blocks is an upper bidiagonal matrix of size not exceeding l_i with λ_i on the diagonal and 1 on the first upper diagonal.

$$X^{-1}BX = J = \begin{pmatrix} D_1 & & & \\ & D_2 & & \\ & & \ddots & \\ & & & D_p \end{pmatrix}$$

$$D_i = \begin{pmatrix} J_{i,1} & & & \\ & J_{i,2} & & \\ & & \ddots & \\ & & & J_{i,\gamma_i} \end{pmatrix}$$

$$J_{i,k} = \begin{pmatrix} \lambda_i & 1 & & \\ & \lambda_i & 1 & \\ & & \ddots & 1 \\ & & & \lambda_i \end{pmatrix}$$

Each $J_{i,k}$ is of size $\leq l_i$ and corresponds to a different eigenvector of B.

Definition The spectral radius $\rho(B)$ is the largest absolute value of any eigenvalue of B: $\rho(B) = \max_{\lambda \in \sigma(B)} |\lambda|$.

Theorem (Saad, Th. 1.10):

$$\lim_{k \to \infty} B^k = 0 \Leftrightarrow \rho(B) < 1.$$

Proof $\Rightarrow$:

Let u_i be a unit eigenvector associated with an eigenvalue λ_i. Then

$$\begin{aligned} Bu_i &= \lambda_i u_i \\ B^2 u_i &= \lambda_i B_i u_i = \lambda_i^2 u_i \\ &\vdots \\ B^k u_i &= \lambda_i^k u_i \\ \text{therefore} \quad ||B^k u_i||_2 &= |\lambda_i^k| \\ \text{and} \quad \lim_{k \to \infty} |\lambda_i^k| &= 0 \end{aligned}$$

so we must have $\rho(B) < 1$ □

Proof $\Leftarrow$:

Take the Jordan form $X^{-1}BX = J$. Then $X^{-1}B^kX = J^k$.

Sufficient to regard Jordan block $J_i = \lambda I + E$ where $|\lambda| < 1$ and $E^{l_i} = 0$.

Let $k \geq l_i$. Then

$$J_i^k = \sum_{j=0}^{l_{i-1}} \binom{k}{j} \lambda^{k-j} E^j$$

$$||J_i||^k \leq \sum_{j=0}^{l_{i-1}} \binom{k}{j} |\lambda|^{k-j} ||E||^j$$

But $\binom{k}{j} = \frac{k!}{j!(k-j)!} = \sum_{i=0}^{j} \begin{bmatrix} j \\ i \end{bmatrix} \frac{k^i}{j!} = p_j(k)$ is a polynomial of degree j in k where the Stirling numbers of the first kind are given by

$\begin{bmatrix} 0 \\ 0 \end{bmatrix} = 1, \quad \begin{bmatrix} j \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ j \end{bmatrix} = 0, \quad \begin{bmatrix} j+1 \\ i \end{bmatrix} = j \begin{bmatrix} j \\ i \end{bmatrix} + \begin{bmatrix} j \\ i-1 \end{bmatrix}.$

Thus, $p_j(k)|\lambda|^{k-j} \to 0 \ (k \to \infty)$ as exponential decay beats polynomial growth □

Corollary (Saad, Th. 1.12):

$$\lim_{k\to\infty} ||B^k||^{\frac{1}{k}} = \rho(B)$$

Sufficient condition for iterative method convergence:

$$\rho(I - M^{-1}A) < 1$$

Convergence rate

Assume λ with $|\lambda| = \rho(I - M^{-1}A) < 1$ is the largest eigenvalue and has a single Jordan block of size l. Then the convergence rate is dominated by this Jordan block, and therein by the term with the lowest possible power in λ which due to $E^l = 0$ is $\lambda^{k-l+1}\binom{k}{l-1}E^{l-1}$. Then,

$$||(I - M^{-1}A)^k(u_0 - \hat{u})|| = O\left(|\lambda^{k-l+1}|\binom{k}{l-1}\right)$$

and the "worst case" convergence factor ρ equals the spectral radius:

$$\begin{aligned} \rho &= \lim_{k\to\infty} \left(\max_{u_0} \frac{||(I - M^{-1}A)^k(u_0 - \hat{u})||}{||u_0 - \hat{u}||} \right)^{\frac{1}{k}} \\ &= \lim_{k\to\infty} ||(I - M^{-1}A)^k||^{\frac{1}{k}} \\ &= \rho(I - M^{-1}A) \end{aligned}$$

Depending on u_0, the rate may be faster, though

Optimal parameter α

Assume A has positive real eigenvalues $0 < \lambda_{min} \leq \lambda_i \leq \lambda_{max}$.

E.g. A is symmetric, positive definite (spd).

Let $\alpha > 0$, $M = \frac{1}{\alpha} I \Rightarrow I - M^{-1}A = I - \alpha A$

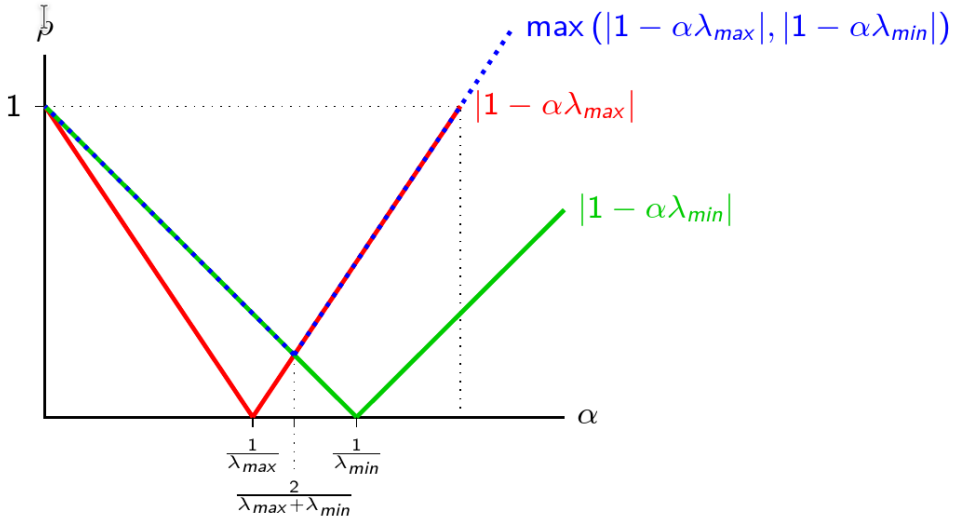
Then for the eigenvalues μ_i of $I - \alpha A$ one has:

$$1 - \alpha\lambda_{max} \leq \mu_i \leq 1 - \alpha\lambda_{min}$$
$$\mu_i < 1$$

We also need $1 - \alpha\lambda_{max} > -1$, so we must have $0 < \alpha < \frac{2}{\lambda_{max}}$.

Theorem. The Richardson iteration converges for any α with $0 < \alpha < \frac{2}{\lambda_{max}}$.

The asymptotic convergence rate is $\rho = \max(|1 - \alpha\lambda_{max}|, |1 - \alpha\lambda_{min}|)$.



- Due to $-(1 - \alpha\lambda_{max}) > -(1 - \alpha\lambda_{min})$ and $+(1 - \alpha\lambda_{min}) > +(1 - \alpha\lambda_{max})$,

$$\begin{aligned}\rho &= \max(|1 - \alpha\lambda_{max}|, |1 - \alpha\lambda_{min}|) \\ &= \max((1 - \alpha\lambda_{max}), -(1 - \alpha\lambda_{min}))\end{aligned}$$

- $1 - \alpha\lambda_{max}$ is monotonically decreasing, the $-(1 - \alpha\lambda_{min})$ increases, so the minimum must be at the intersection

$$\begin{aligned}1 - \alpha\lambda_{max} &= -1 + \alpha\lambda_{min} \\ 2 &= \alpha(\lambda_{max} + \lambda_{min})\end{aligned}$$

Theorem The optimal parameter is

$$\alpha_{opt} = \frac{2}{\lambda_{min} + \lambda_{max}}$$

For this parameter, the convergence factor is

$$\rho_{opt} = \frac{\lambda_{max} - \lambda_{min}}{\lambda_{max} + \lambda_{min}} = \frac{\kappa - 1}{\kappa + 1}$$

where $\kappa = \kappa(A) = \frac{\lambda_{max}}{\lambda_{min}}$ is the spectral condition number of A

```
A_rich =
100×100 Symmetric{Float64, Matrix{Float64}}:
  0.995048     -0.00731417   -0.00126435  -0.000127444  …  -0.00124874   -0.00840498
 -0.00731417    0.992691     -0.0092514   -0.000877535     -0.00498363   -0.000775311
 -0.00126435   -0.0092514     0.996483    -0.00974054      -0.00232105   -0.00964262
 -0.000127444  -0.000877535  -0.00974054   0.9952          -0.00711445   -0.00338844
 -0.00299973   -0.0052567    -0.0016976   -0.00455641      -0.00819544   -0.00648116
 -0.00305736   -0.00175138   -0.00879216  -0.000466981  …  -0.000932459  -0.00316856
 -0.00340156   -0.00655285   -0.00870986  -0.00871876      -0.00844533   -0.00490182
  ⋮                                                      ⋱
 -0.00992772   -0.00783047   -0.00269474  -0.00202993      -0.00199126   -0.0059112
 -0.00522591   -6.0182e-5    -0.00758992  -0.00918784   …  -0.00030877   -0.00100392
 -0.00521487   -0.00412213   -0.0029189   -0.00504286      -0.00581213   -0.00922209
 -0.0080989    -0.00796063   -0.00608553  -0.00852207      -0.00775608   -0.00655473
 -0.00124874   -0.00498363   -0.00232105  -0.00711445       0.996874     -0.00141893
 -0.00840498   -0.000775311  -0.00964262  -0.00338844      -0.00141893    0.993018
```

```
A_rich=Symmetric(randmatrix(N,δ=1))
```

```
(0.506621, 1.05702)
```

```
(λ_min, λ_max)=extrema(eigvals(A_rich))
```

```
κ = 2.086406151490358
```

```
κ=λ_max/λ_min
```

```
α_opt = 1.2790692999218964
```

```
α_opt=2/(λ_min+λ_max)
```

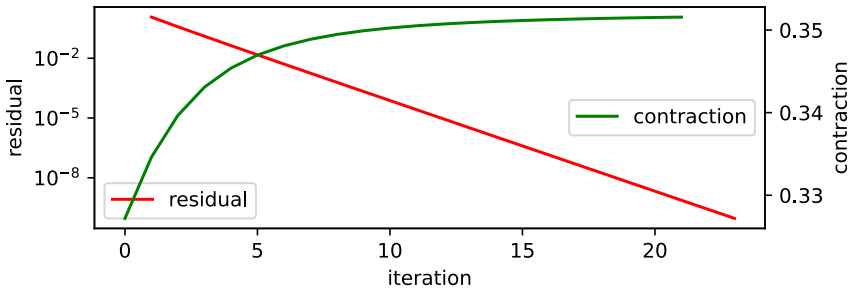
```
ρ_opt = 0.3519971443051188
```

```
ρ_opt=(λ_max- λ_min)/(λ_max + λ_min)
```

```
b_rich=A_rich*û;
```

```
richardson_sweep!(unew,uold, A,b)= unew .= uold - α_opt*(A*uold-b);
```

```
(u_rich,history_rich)=simple_iteration(A_rich,b_rich,zeros(N),richardson_sweep!,maxiter=1000);
```




```
plothistory(history_rich)
```

```
1.762814167338075e-10
```

```
norm(û-u_rich)
```

Parameter for preconditioned iteration

Theorem: M, A spd. Assume the *spectral equivalence estimate*

$$0 < \gamma_{min}(Mu,u) \leq (Au,u) \leq \gamma_{max}(Mu,u)$$

Then for the eigenvalues μ_i of $M^{-1}A$ we have

$$\gamma_{min} \leq \mu_{min} \leq \mu_i \leq \mu_{max} \leq \gamma_{max}$$

and $\kappa(M^{-1}A) \leq \frac{\gamma_{max}}{\gamma_{\min}}$

Proof: Let the inner product $(\cdot,\cdot)_M$ be defined via $(u,v)_M = (Mu,v)$. In this inner product, $C = M^{-1}A$ is self-adjoint:

$$\begin{aligned}(Cu,v)_M &= (MM^{-1}Au,v) = (Au,v) = (M^{-1}Mu,Av) = (Mu,M^{-1}Av)\\ &= (u,M^{-1}A)_M = (u,Cv)_M\end{aligned}$$

Minimum and maximum eigenvalues can be obtained as Ritz values in the $(\cdot,\cdot)_M$ scalar product

$$\mu_{min} = \min_{u\neq 0}\frac{(Cu,u)_M}{(u,u)_M} = \min_{u\neq 0}\frac{(Au,u)}{(Mu,u)} \geq \gamma_{min}$$
$$\mu_{max} = \max_{u\neq 0}\frac{(Cu,u)_M}{(u,u)_M} = \max_{u\neq 0}\frac{(Au,u)}{(Mu,u)} \leq \gamma_{max}$$

Matrix preconditioned Richardson iteration: M, A spd.

Scaled Richardson iteration with preconditioner M

$$u_{k+1} = u_k - \alpha M^{-1}(Au_k - b)$$

Spectral equivalence estimate

$$0 < \gamma_{min}(Mu,u) \leq (Au,u) \leq \gamma_{max}(Mu,u)$$

$\Rightarrow \gamma_{min} \leq \lambda_i \leq \gamma_{max}$

$\Rightarrow$ optimal parameter $\alpha = \frac{2}{\gamma_{max}+\gamma_{min}}$

Relative condition number estimate: $\kappa(M^{-1}A) \leq \frac{\gamma_{max}}{\gamma_{min}}$

Convergence rate with optimal parameter: $\rho \leq \frac{\kappa(M^{-1}A)-1}{\kappa(M^{-1}A)+1}$

Example: Jacobi method and 1D Heat conduction

Regard the $n \times n$ 1D heat conduction matrix with $h = \frac{1}{n-1}$ and $\alpha = \frac{1}{h}$ (easier to analyze).

$$A = \begin{pmatrix} \frac{2}{h} & -\frac{1}{h} & & & & & \\ -\frac{1}{h} & \frac{2}{h} & -\frac{1}{h} & & & & \\ & -\frac{1}{h} & \frac{2}{h} & -\frac{1}{h} & & & \\ & \ddots & \ddots & \ddots & \ddots & & \\ & & -\frac{1}{h} & \frac{2}{h} & -\frac{1}{h} & & \\ & & & -\frac{1}{h} & \frac{2}{h} & -\frac{1}{h} & \\ & & & & -\frac{1}{h} & \frac{2}{h} \end{pmatrix}$$

Eigenvalues (tri-diagonal Toeplitz matrix):

$$\lambda_i = \frac{2}{h}\left(1+\cos\left(\frac{i\pi}{n+1}\right)\right) \quad (i = 1\ldots n)$$

{\tiny Source: A. Böttcher, S. Grudsky: Spectral Properties of Banded Toeplitz Matrices. SIAM,2005}

Express them in h: $n+1 = \frac{1}{h}+2 = \frac{1+2h}{h} \Rightarrow$

$$\lambda_i = \frac{2}{h}\left(1+\cos\left(\frac{ih\pi}{1+2h}\right)\right) \quad (i = 1\ldots n)$$

- For $i = 1 \dots n$, the argument of cos is in $(0, \pi)$
- cos is monotonically decreasing in $(0, \pi)$, so we get λ_{max} for $i = 1$ and λ_{min} for $i = n = \frac{1+h}{h}$
- Therefore:

$$\lambda_{max} = \frac{2}{h}\left(1 + \cos\left(\pi\frac{h}{1+2h}\right)\right) \approx \frac{2}{h}\left(2 - \frac{\pi^2 h^2}{2(1+2h)^2}\right)$$
$$\lambda_{min} = \frac{2}{h}\left(1 + \cos\left(\pi\frac{1+h}{1+2h}\right)\right) \approx \frac{2}{h}\left(\frac{\pi^2 h^2}{2(1+2h)^2}\right)$$

Here, we used the Taylor expansion

$$cos(\delta) = 1 - \frac{\delta^2}{2} + O(\delta^4) \quad (\delta \to 0)$$
$$cos(\pi - \delta) = -1 + \frac{\delta^2}{2} + O(\delta^4) \quad (\delta \to 0)$$

and $\frac{1+h}{1+2h} = \frac{1+2h}{1+2h} - \frac{h}{1+2h} = 1 - \frac{h}{1+2h}$

- The Jacobi preconditioner just multiplies by $\frac{h}{2}$, therefore for $M^{-1}A$:

$$\mu_{max} \approx 2 - \frac{\pi^2 h^2}{2(1+2h)^2}$$
$$\mu_{min} \approx \frac{\pi^2 h^2}{2(1+2h)^2}$$

- Optimal parameter: $\alpha = \frac{2}{\lambda_{max}+\lambda_{min}} \approx 1 \; (h \to 0)$
- **Good news**: α_{opt} is independent of h resp. n
  - No need for spectral estimate in order to work with optimal parameter.
  - Is this true beyond this special case ?

- Condition number + spectral radius

$$\kappa(M^{-1}A) = \kappa(A) \approx \frac{4(1+2h)^2}{\pi^2 h^2} - 1 = O(h^{-2}) \quad (h \to 0)$$
$$\rho(I - M^{-1}A) = \frac{\kappa - 1}{\kappa + 1} = 1 - \frac{\pi^2 h^2}{2(1+2h)^2} = 1 - O(h^2) \quad (h \to 0)$$

- **Bad news:** $\rho \to 1 \quad (h \to 0)$

  - Typical situation with second order PDEs:

$$\kappa(A) = O(h^{-2}) \quad (h \to 0)$$
$$\rho(I - D^{-1}A) = 1 - O(h^2) \quad (h \to 0)$$

  - Mean square error of approximation $||u - u_h||_2 < h^\gamma$, in the simplest case $\gamma = 2$.

Estimating iterative solver complexity

Solve linear system iteratively until $||e_k|| = ||(I - M^{-1}A)^k e_0|| \leq \epsilon$. Estimate the necessary number of iteration steps:

$$\rho^k e_0 \leq \epsilon$$
$$k \ln \rho < \ln \epsilon - \ln e_0$$
$$k \geq k_\rho = \left\lceil \frac{\ln e_0 - \ln \epsilon}{\ln \rho} \right\rceil$$

$\Rightarrow$ we need at least k_ρ iteration steps to reach accuracy ϵ

The **ideal iterative solver**:

- $\rho(I - M^{-1}A) < \rho_0 < 1$ independent of h resp. $N \Rightarrow k_\rho$ independent of N.
- A sparse $\Rightarrow$ matrix-vector multiplication Au has complexity $O(N)$
- Solution of $Mv = r$ has complexity $O(N)$.

$\Rightarrow$ Number of iteration steps k_ρ independent of N Each iteration step has complexity $O(N) \Rightarrow$ Overall complexity $O(N)$

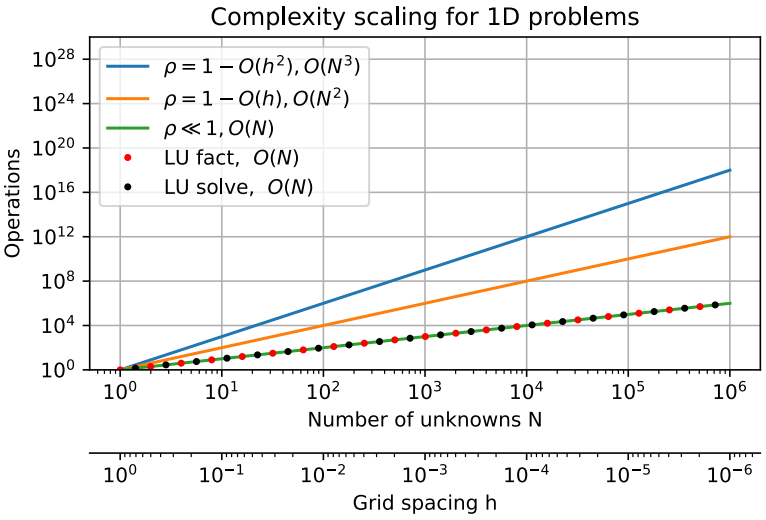
Simple iterative solvers

- $\rho = 1 - h^\delta$
  - $\Rightarrow \ln \rho \approx -h^\delta$
  - $\Rightarrow k_\rho = O(h^{-\delta})$
- d: space dimension:
  - $N \approx n^d$
  - $h \approx \frac{1}{n} \approx N^{-\frac{1}{d}}$
  - $\Rightarrow k_\rho = O(N^{\frac{\delta}{d}})$
- $O(N)$ complexity of one iteration step (e.g. Jacobi, Gauss-Seidel)
- $\Rightarrow$ Overall complexity $O(N^{1+\frac{\delta}{d}})$=$O(N^{\frac{d+\delta}{d}})$
  - Typical scaling for simple iteration schem: $\delta = 2$ (Jacobi, Gauss-Seidel . . .)
  - Does a hypothetical "Improved iteration scheme" with $\delta = 1$ exist ?

Overview on complexity estimates

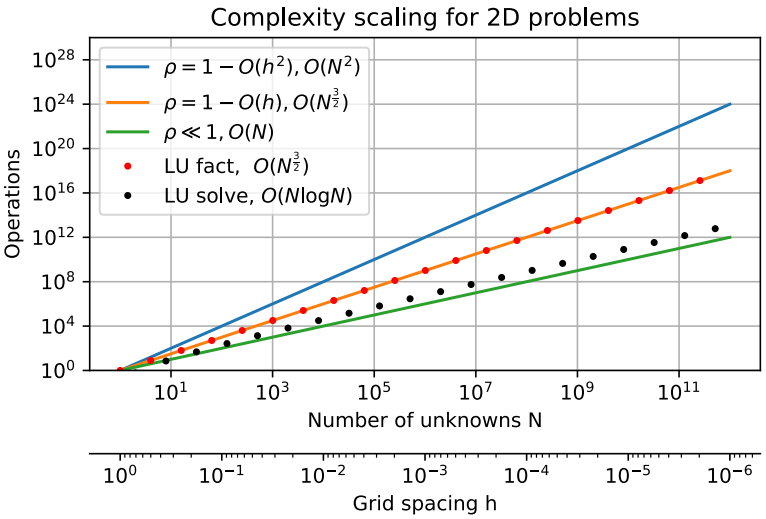
Space dim	$\delta = 2$ $\rho = 1 - O(h^2)$	$\delta = 1$ $\rho = 1 - O(h)$	LU fact	LU solve
1	$O(N^3)$	$O(N^2)$	$O(N)$	$O(N)$
2	$O(N^2)$	$O(N^{\frac{3}{2}})$	$O(N^{\frac{3}{2}})$	$O(N \log N)$
3	$O(N^{\frac{5}{3}})$	$O(N^{\frac{4}{3}})$	$O(N^2)$	$O(N^{\frac{4}{3}})$
Tendency with d ↑	↓	↓	↑↑	↑

Complexity scaling in 1D



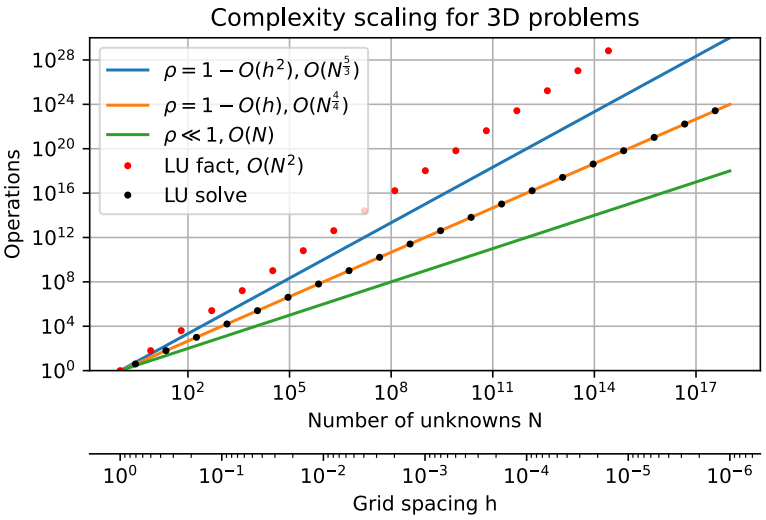
- Sparse direct solvers, tridiagonal solvers are asymptotically optimal
- Non-ideal iterative solvers significantly worse than optimal

Complexity scaling in 2D



- Sparse direct solvers better than simple nonideal iterative solvers ($\delta = 2$) – Jacobi etc.
- Sparse direct solvers on par with hypothetical improved iteration scheme ($\delta = 1$)

Complexity scaling in 3D



- Sparse LU factorization is expensive: going from h to $h/2$ increases N by a factor of 8 and operation count by a factor of 64!
- Sparse LU solve on par with hypothetical improved iteration scheme

What is to be done?

Questions

- Find ideal preconditioner with $\rho \leq \rho_0 < 1$ independent of h, N
- Find "improved preconditioner" with $\kappa(M^{-1}A) = O(h^{-1}) \Rightarrow \delta = 1$
- Find "improved iterative scheme" with $\rho = \frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}$. This would have a similar effect as $\delta = 1$:

  For Jacobi, we had $\kappa = X^2 - 1$ where $X = \frac{2(1+2h)}{\pi h} = O(h^{-1})$.

$$\begin{aligned}
\rho &= 1 + \frac{\sqrt{X^2-1}-1}{\sqrt{X^2-1}+1} - 1 \\
&= 1 + \frac{\sqrt{X^2-1}-1-\sqrt{X^2-1}-1}{\sqrt{X^2-1}+1} \\
&= 1 - \frac{1}{\sqrt{X^2-1}+1} = 1 - \frac{1}{X\left(\sqrt{1-\frac{1}{X^2}}+\frac{1}{X}\right)} \\
&= 1 - O(h) \quad (h \to 0)
\end{aligned}$$

```
true
```

```
begin
    using PlutoUI
    using HypertextLiteral
    using LaTeXStrings
    using PyPlot
    PyPlot.svg(true)
end
```

```
pyplot (generic function with 1 method)
```

```
function pyplot(f;width=300,height=300)
        clf()
        f()
        fig=gcf()
        fig.set_size_inches(width/100,height/100)
        fig
    end
```

```
begin

    highlight(mdstring,color)= htl"""<blockquote style="padding: 10px; background-color: $(color);">$(mdstring)</blockquote>"""

    macro important_str(s)  :(highlight(Markdown.parse($s),"#ffcccc")) end
    macro definition_str(s) :(highlight(Markdown.parse($s),"#ccccff")) end
    macro statement_str(s)  :(highlight(Markdown.parse($s),"#ccffcc")) end


    html"""
    <style>
     h1{background-color:#dddddd;  padding: 10px;}
     h2{background-color:#e7e7e7;  padding: 10px;}
     h3{background-color:#eeeeee;  padding: 10px;}
     h4{background-color:#f7f7f7;  padding: 10px;}
    </style>
"""
end

```