Iterative Lösung großer Gleichungssysteme,
Übung 2, Aufgabe 1
Lukas Kogler

Ein MATLAB-File mit der Implementation des Algorithmus und einem performance-Test im Vergleich zu **fft** bzw LU-Zerlegung befindet sich im Anhang.

Für die Implementation der **dst** habe ich jedoch einen etwas anderen Weg gewählt, als er im Skript angegeben ist. Anstatt **fft** mit dem Vektor $(0, v, 0, \ldots, 0)$ der Länge $2n+1$ zu machen, der einer Funktion auf $[-1, 1]$, welche auf $[-1, 0]$ 0 ist entspricht, wende ich **fft** auf den Vektor $(-v_n, -v_{n-1}, \ldots, -v_1, 0, v_1, v_2, \ldots, v_n)$ an, der einer **ungeraden** Funktion entspricht. Dadurch erhält man auf natürliche Weise eine Darstellung durch reine **sin**-Terme und man kann sich theoretisch sparen, am Ende den Imaginärteil nehmen zu müssen. Durch Rundungsfehler musste ich (um die MATLAB-Warnung loszuwerden) dies allerdings trotzdem machen.

```
% --- first, verify that all produce the right result ---
n = 10;
%[xi,A,D,L] = make_pois_sparse(n);
[xi,A,D,L] = make_pois_sparse(n);
b = 8*ones(n,1); % solve -u''=8 -> u = 4*x*(1-x)
xi = [0;xi;1];

uref = [4*xi.*(1-xi)];
u_my_dst = [0;solve_my_dst(b,D);0];
u_dst = [0;solve_dst(b,D);0];
u_chol = [0;solve_chol(b,L);0];
u_chol2 = [0;solve_chol2(b,A);0];

subplot(2,1,1);
hold off
plot(xi,uref, 'r');
hold on;
plot(xi, [0;A\b;0], 'c--');
plot(xi, u_chol, 'm');
plot(xi, u_dst, 'b');
plot(xi, u_my_dst, 'g--');
hold off;
title('4*x*(1-x)');
legend('ref','backslash','chol', 'dst','my-dst');

% --- performance tests ---
%ns = []
ns = [10:10:90,100:100:900,1000:1000:9000,10000:10000:90000,100000:↙
100000:1000000];
t = zeros(size(ns,2),4);
nreps = 10;
for k = [1:size(ns,2)]
    n = ns(k);
    [xi,A,D,L] = make_pois_sparse(n);
    b = ones(n,1);
    tic;
    for j=[1:nreps]
        y = solve_dst(b,D);
    end
    t(k,1) = toc;
    tic;
    for j=[1:nreps]
        y = solve_my_dst(b,D);
```

```
    end
    t(k,2) = toc;
    tic;
    for j=[1:nreps]
        y = solve_chol(b,L);
    end
    t(k,3) = toc;
    tic;
    for j=[1:nreps]
        y = solve_chol2(b,A);
    end
    t(k,4) = toc;
end
t = t./nreps;
subplot(2,1,2)
xi2 = 1:size(ns,1);
cols = ['r','g','b','c'];
hold off
plot(ns,t(:,1), cols(1));
hold on
for k=[2:4]
    plot(ns,t(:,k),cols(k));
end
xlabel('n')
ylabel('t')
title('timing');
lgd = legend('dst','my-dst','chol', 'chol+factor');
lgd.Location='northwest';
hold off



% --- functions ---

function [xi,A,D,L] = make_pois_mat(n)
h = 1/(1+n);
xi = (h:h:1-h)';
A = 1/(h^2)*(2*eye(n)-diag(ones(n-1,1),1)-diag(ones(n-1,1),-1));
D = diag(4*sin(pi*xi/2).^2)/(h^2);
L = chol(sparse(A));
end

function [xi,A,D,L] = make_pois_sparse(n)
h = 1/(1+n);
```

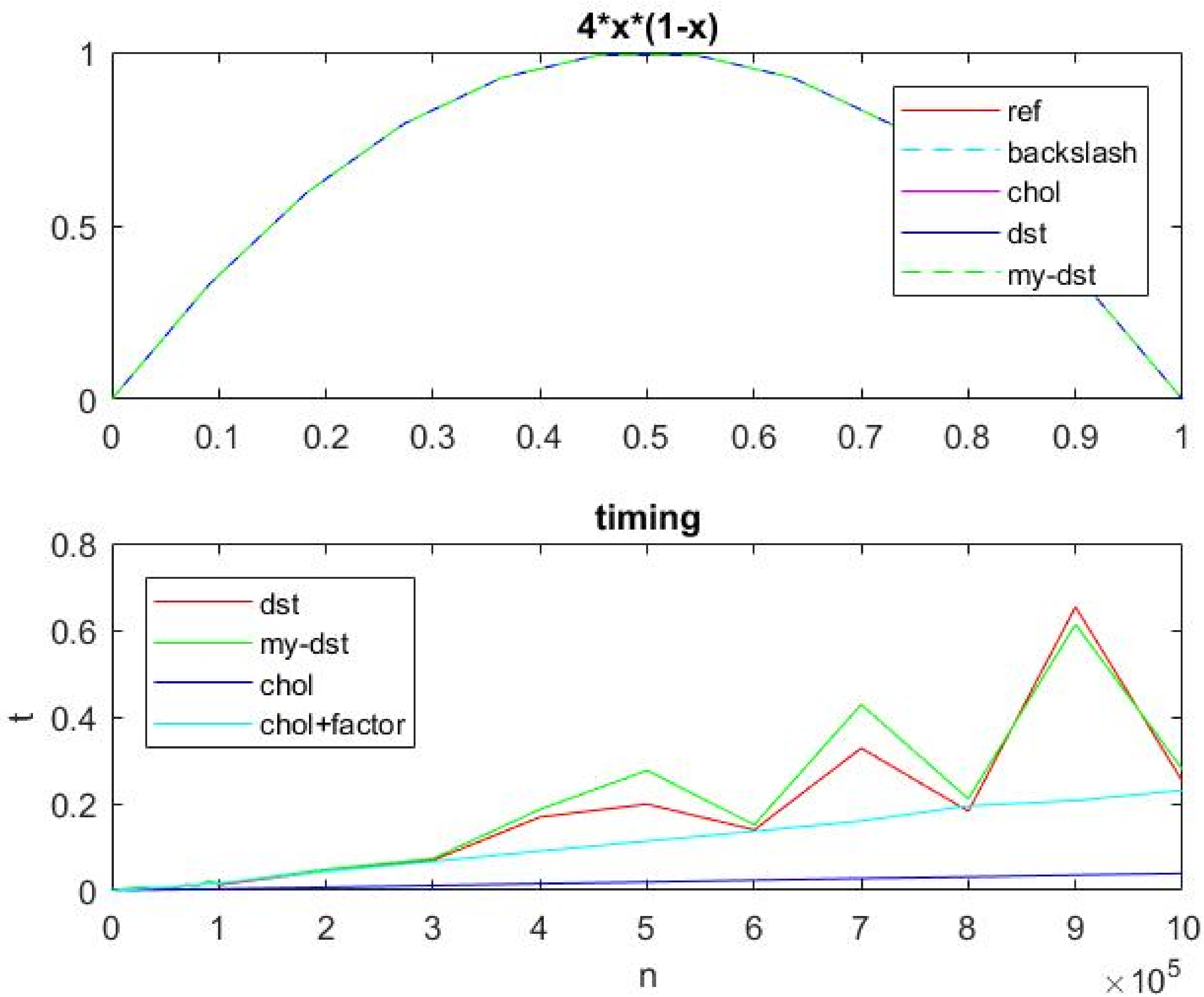
```
xi = (h:h:1-h)';
row = [2,-1,zeros(1,n-2)];
e = 1/(h^2)*ones(n,1);
A = spdiags([-e 2*e -e], -1:1, n, n);
D = spdiags((4*sin(pi*xi/2).^2)/(h^2),0,n,n);
L = chol(A);
end

function x = solve_dst(b,D)
x = dst(b);   % sine-transform
x = D\x;      % eigenvalues
x = idst(x);  % inv. sin-trans
end

function x = solve_my_dst(b,D)
y2 = [0;b;0;-b(end:-1:1)];
y2 = fft(y2);
y3 = D\y2(2:size(b,1)+1);
y2 = ifft([0;y3;0;-y3(end:-1:1)]);
x = real(y2(2:size(b,1)+1)); %only necessary bc of rounding errors??
%x = y2(2:size(b,1)+1);
end

function x = solve_chol(b,L)
x = L'\b;    %L^T y1 = b
x = L\x;     %L x = y1
end

function x = solve_chol2(b,A)
L = chol(A); % L^T * L = A
x = L'\b;    %L^T y1 = b
x = L\x;     %L x = y1
end
```


4*x*(1-x)
ref
backslash
chol
dst
my-dst
0
0.5
1
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
timing
dst
my-dst
chol
chol+factor
t
0
0.2
0.4
0.6
0.8
0
1
2
3
4
5
6
7
8
9
10
n
×10^5

2. Beispiel

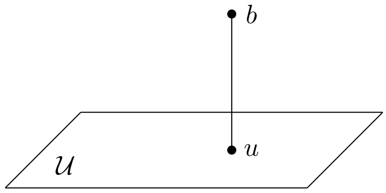
a)

Geg.: $b \in \mathbb{R}^n$, $\mathcal{U} \subset \mathbb{R}^n$ m-dimensionaler Unterraum
Z.z.: $u = \text{argmin}_{v \in \mathcal{U}} \|v - b\|_2 \iff u - b \perp \mathcal{U}$
Beweis. Betrachte den m-dimensionalen Unterraum $\mathcal{U}$ als Bild einer Matrix $A \in \mathbb{R}^{n \times m}$. Es gilt mit $u = Ax$

$$\begin{aligned} &Ax - b \perp \text{Bild}(A) \\ \iff &Ax - b \in \ker(A^T) \\ \iff &A^T(AX - b) = 0 \\ \iff &A^T Ax = A^T b \end{aligned}$$

Die Lösung x der Normalengleichung minimiert aber genau $||Ax - b||_2$ (unter anderem Skriptum Kapitel 7.2) also

$$\text{argmin}_{v \in \text{Bild}(A)} ||v - b|| = Ax = u$$



b)

Geg.: Orthonormalbasis $\{q_1 ... q_m\}$ von $\mathcal{U}$, $Q := (q_1, ..., q_m) \in \mathbb{R}^{n \times m}$

- **Z.z.:** $u = QQ^T b$
  Sei $v \in \mathcal{U}$ fest aber beliebig, dann gilt:

  $$\langle QQ^T b - b, v \rangle = \langle (QQ^T - I)b, v \rangle = \langle b, (QQ^T - I)v \rangle$$

  Stelle $v = Qw$ als Linearkombination in seiner Basis dar mit, so gilt weiters:

  $$\begin{aligned} \langle b, (QQ^T - I)v \rangle &= \langle b, (QQ^T - I)Qw \rangle \\ &= \langle b, QQ^T Qw - Qw \rangle \\ &= \langle b, Qw - Qw \rangle = 0 \end{aligned}$$

  Gemeinsam mit a) folgt die Behauptung.

- **Z.z.:** $P := QQ^T \implies P = P^T = P^2$

  $$\begin{aligned} P^T &= (QQ^T)^T = (Q^T)^T Q^T = QQ^T = P \\ P^2 &= (QQ^T)(QQ^T) = QIQ^T = QQ^T = P \end{aligned}$$

- **Ges.:** Koeffizienten von $u = QQ^T x$ in der Form $u = \sum_{i=1}^{m} \xi_i q_i$

  $$QQ^T (x_1 ... x_m)^T = Q(\langle q_1, x \rangle, ..., \langle q_m, x \rangle)^T = \sum_{i=1}^{m} \langle q_i, x \rangle q_i$$

c)

Geg.: $\{u_1 ... u_m\}$ beliebige Basis von $\mathcal{U}$
Ges.: Orthogonalprojektor P auf U mittels einer Faktorisierung der Matrix $U = (u_1 ... u_m)$.
$P := QQ^T$ leistet gemäß b) das Gewünschte, wobei Q jene orthogonale Matrix aus der reduzierten QR-Zerlegung von U ist, also $U = QR$ mit $Q \in \mathbb{R}^{n \times m}$ und $R \in \mathbb{R}^{m \times m}$ obere Dreiecksmatrix.

d)

Geg.: $A \in \mathbb{R}^{n \times m}$ mit $m < n$ und $\text{rg}A = m$
Ges.: Lösung des linearen Ausgleichsproblems $\|Ax - b\|_2 \longrightarrow$ minimal unter Verwendung der reduzierten QR-Zerlegung von A.
Das lineare Ausgleichsproblem ist äquivalent zur Gauß'schen Normalengleichung $A^T A x = A^T b$. Einsetzen der reduzierten QR-Zerlegung liefert:

$$\begin{aligned} (QR)^T QRx = (QR)^T b &\iff R^T Q^T QRx = R^T Q^T b \\ &\iff R^T Rx = R^T Q^T b \\ &\iff Rx = Q^T b \end{aligned}$$

Es genügt also, das kleinere Gleichungssystem $Rx = Q^T b$ zu lösen.

e)

Im Folgenden führen wir einen Pseudocode für die Aufgabenstellung an.

Algorithm 1 Reduced QR-Decomposition

```
%Erstelle 8x5 Zufallsmatrix:
A = rand(8,5);

%Zufallsvektor:
b = rand(8,1);

% Überprüfe, ob Matrix vollen Rang besitzt
while rank(A) ≠ 5 do
    A = rand(8,5);
end while

% Reduzierte QR-Zerlegung
[Q,R] = qr(A,0);

%Löse Rx = Q^T b
c = transpose(Q)*b;
x = R\c;

%Vergleiche Ergebnisse
display('QR-Verfahren:');
display(x);
display('Matlab-Implementierung:');
display(A\b);
```

3. Beispiel

Geg.: $A \in \mathrm{GL}_n(\mathbb{R})$, $b \in \mathbb{R}^n$, $\mathcal{U}$ ein m-dimensionaler Unterraum des $\mathbb{R}^n$
Ges.: Approximation für die Lösung x_* des Ausgleichsproblems $Ax - b$ in dem Unterraum $\mathcal{U}$ gemäß Galerkin-Approximation.

a)

Geg.: $\{u_1 ... u_m\}$ Basis von $\mathcal{U}$ mit Ansatz $u = U\xi = \sum\limits_{i=1}^{m} \xi_i u_i$
Ges.: Lineares Gleichungssystem, dessen Lösung die Koeffizienten ξ_i liefert.
Es soll also für alle $j = 1...m$ gelten:

$$\begin{aligned}\langle b - \sum_{i=1}^{m} \xi_i A u_i, u_j \rangle = 0 &\iff \langle \sum_{i=1}^{m} \xi_i A u_i, u_j \rangle = \langle b, u_j \rangle \\ &\iff \sum_{i=1}^{m} \xi_i \langle A u_i, u_j \rangle = \langle b, u_j \rangle\end{aligned}$$

In Matrixschreibweise liefert das:

$$\begin{pmatrix} \langle Au_1, u_1 \rangle & \cdots & \langle Au_m, u_1 \rangle \\ \vdots & & \vdots \\ \langle Au_1, u_m \rangle & \cdots & \langle Au_m, u_m \rangle \end{pmatrix} \begin{pmatrix} \xi_1 \\ \vdots \\ \xi_m \end{pmatrix} = \begin{pmatrix} \langle b, u_1 \rangle \\ \vdots \\ \langle b, u_m \rangle \end{pmatrix} \iff U^T A U \xi = U^T b$$

wobei $G := U^T A U$ die Gramm-Matrix ist.
Ist das Gleichungssystem eindeutig lösbar? Nein, da zum Beispiel das Gleichungssystem

$$\begin{pmatrix} 1 & 0 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} \stackrel{!}{=} \begin{pmatrix} 1 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} \iff 0 = 1$$

nicht einmal eine Lösung besitzt.

b)

Sei im folgenden A symmetrisch und positiv definit.
Z.z.: A SPD $\implies G := U^T A U$ SPD
Aufgrund der positiven Definitheit von A gilt:

$$\langle Gx, x \rangle = \langle U^T A U x, x \rangle = \langle A \underbrace{Ux}_{y}, \underbrace{Ux}_{y} \rangle > 0$$

womit die Definitheit folgt. Die Symmetrie von G ist klar.
Ges.: Die zu u gehörige Minimierungsaufgabe.
Wir rufen folgende Resultate über lineare Ausgleichsprobleme aus der Numerik in Erinnerung: Ein Vektor $x_* \in \mathbb{R}^n$ löst genau dann das lineare Ausgleichsproblem

$$\|Ax - b\|_2^2 = \min!$$

wenn x_* die Gauß'sche Normalengleichung

$$A^* A x = A^* b$$

erfüllt. Es existiert immer mindestens eine Lösung des linearen Ausgleichsproblems, und die Menge aller Lösungen ist gegeben durch

$$x_* + \ker(A).$$

Im Folgenden werden wir obiges Gleichungssystem in den Kontext der Normalengleichung setzen. Da A und damit auch insbesondere $A^{\frac{1}{2}}$ symmetrisch sind, folgt

$$\begin{aligned} U^T A U x = U^T b &\iff U^T \underbrace{(A^{\frac{1}{2}})^T A^{\frac{1}{2}}}_{A} U x = U^T \underbrace{A^{\frac{1}{2}} A^{-\frac{1}{2}}}_{I} b \\ &\iff \|A^{\frac{1}{2}} U x - A^{-\frac{1}{2}} b\|_2^2 = \min! \\ &\iff \|A^{\frac{1}{2}} (U x - \underbrace{A^{-1} b)}_{=x_*}\|_2^2 = \min! \\ &\iff \|U x - x_*\|_A^2 = \min! \end{aligned}$$

Über diese Beziehung kann x als orthogonale Projektion von x_* aufgefasst werden.

c)

Im Folgenden führen wir einen Pseudocode für die Aufgabenstellung an.

Algorithm 1 Galerkin Approximation

```
%Symmetrische 8x8 Integer-Zufallsmatrix
A = randi(8,8);
A = A*transpose(A);

%Lösungsvektor:
x = transpose([1 1 1 0 0 0 0 0]);

%Rechte Seite
b = A*x;

%Basis des Unterraums: R^2-Ebene
U = transpose([1 0 0 0 0 0 0 0; 0 1 0 0 0 0 0 0]);

%Reduziertes Gleichungssystem
G = transpose(U)*A*U;
c = transpose(U)*b;

%Koeffizienten:
Xi = G\c;

%Linearkombination der Lösung
u = Xi(1)*U(:,1) + Xi(2)*U(:,2);

%Matlab-Implementiertung
display('Galerkin-Approximation:');
display(u);
```

Iteratives Lösen linearer Gleichungssysteme - Übung 2

Michael Neunteufel

11. April 2017

5. Beispiel

Geg.: Seien A und N symmetrisch und positiv definit mit der zugehörigen Iteration:

$$x_{k+1} = x_k + N(b - Ax_k)$$

und der Iterationsmatrix $M = I - W^{-1}A$ mit $W = N^{-1}$. Ferner gelte:

$$2W > A > 0$$

a)

Z.z.: Die adjungierte Matrix M^A von M in Bezug auf das Skalarprodukt $\langle .,. \rangle_A$ erfüllt:

$$M^A = A^{-1}M^TA$$

Beweis: Sei x beliebig aber fest gewählt, dann gilt:

$$\begin{aligned}
\langle Mx, x\rangle_A &= \langle x, M^TA^Tx\rangle \\
&= \langle x, AA^{-1}M^TAx\rangle \\
&= \langle Ax, A^{-1}M^TAx\rangle \\
&= \langle x, A^{-1}M^TAx\rangle_A = \langle x, M^Ax\rangle_A
\end{aligned}$$

b)

Z.z.: $\rho(M) = \|M\|_A < 1$
Beweis: Wir zeigen zuerst die Ungleichung. Da $\rho(M) < 1$ äquivalent zu $\sigma(M) \subset (-1, 1)$ ist, gilt nach Lemma 5.2 (iv)

$$\rho(M) < 1 \iff I > I - W^{-1}A \quad \text{und} \quad I - W^{-1}A > -I$$

Nach Voraussetzung sind A und W, und damit auch W^{-1} positiv definit, womit die erste Ungleichheit trivialerweise erfüllt ist und es genügt $2I - W^{-1}A > 0$ zu zeigen. Diese Ungleichung ist erfüllt, da mit $2W > A$ folgt:

$$2I - W^{-1}A > 2I - 2W^{-1}W = 0$$

Bleibt noch die Gleichheit $\rho(M) = ||M||_A$ zu zeigen. Aufgrund der Symmetrie von $A^{\frac{1}{2}}MA^{-\frac{1}{2}}$ gilt:

$$\|M\|_A \overset{S.31}{=} \|A^{\frac{1}{2}}MA^{-\frac{1}{2}}\|_2 = \rho(A^{\frac{1}{2}}MA^{-\frac{1}{2}}) \overset{!}{=} \rho(M)$$

Letztere Gleichheit ist erfüllt, da unter Berücksichtigung des Determinantenmultiplikationssatzes folgende Identitäten gelten:

$$\begin{aligned}
\det(A^{\frac{1}{2}}MA^{-\frac{1}{2}} - \lambda I) &= \det(A^{\frac{1}{2}}(M - \lambda I)A^{-\frac{1}{2}}) \\
&= \underbrace{\det(A^{\frac{1}{2}})\det(A^{-\frac{1}{2}})}_{=1}\det(M - \lambda I)
\end{aligned}$$

c)

Z.z.: $\lambda \in (0, \Lambda]$ und $0 < \lambda W \leq A \leq \Lambda W \Longrightarrow \sigma(M) \subset [1-\Lambda, 1-\lambda]$
Beweis:

$$\sigma(M) \subset [1-\Lambda, 1-\lambda] \Longleftrightarrow I - \Lambda I \leq M \leq I - \lambda I$$

Die erste Ungleichung liefert:

$$I - \Lambda I \leq M \Longleftrightarrow \Lambda I - W^{-1}A \geq 0$$

Diese Ungleichung ist erfüllt, da:

$$\Lambda I - W^{-1}A \geq \Lambda I - \Lambda W^{-1}W = 0$$

Analog gilt für die zweite Ungleichung:

$$M \leq I - \lambda I \Longleftrightarrow -W^{-1}A \leq -\lambda W^{-1}W = -\lambda I$$

womit die Behauptung folgt.

d)

Z.z.: $\omega \in (0, 1] \Longrightarrow$ SSOR konvergiert für alle SPD-Matrizen A. Es gilt:

$$M_\omega^{SSOR} = I - W^{-1}A$$

wobei $W^{-1} = \omega(2-\omega)(D+\omega U)^{-1}D(D+\omega L)^{-1}$
Beweis.: Wegen **b)** genügt es zu zeigen, dass $2W > A$, oder dazu äquivalent $2W - A > 0$, gilt.

$$\begin{aligned}
2W - A &= \frac{2}{\omega(2-\omega)}(D+\omega L)D^{-1}(D+\omega U) - A \\
&= \frac{2}{\omega(2-\omega)}(I+\omega LD^{-1})(D+\omega U) - A \\
&= \frac{2}{\omega(2-\omega)}(D+\omega L+\omega U+\omega^2 LD^{-1}U) - (L+D+U) \\
&= \Bigg(\underbrace{\frac{2}{\omega(2-\omega)}}_{>\frac{2}{2-\omega}} - 1\Bigg)D + \left(\frac{2}{2-\omega} - 1\right)L + \left(\frac{2}{2-\omega} - 1\right)U + \frac{2\omega}{2-\omega}LD^{-1}L^T \\
&> \underbrace{\left(\frac{2}{2-\omega} - 1\right)}_{>1} A + \underbrace{\frac{2\omega}{2-\omega}}_{>0}\underbrace{LD^{-1}L^T}_{\geq 0} \geq A > 0
\end{aligned}$$

e)

Z.z.: W sei nur noch positiv definit und es gelte $W + W^T > A > 0$. Dann folgt $(\rho(M) \leq)\ \|M\|_A < 1$.
Beweis:

$$N^T + N = W^{-T}(W+W^T)W^{-1} > W^{-T}AW^{-1} = N^TAN$$

Sei x mit $||x||_A = 1$ beliebig. Dann gilt mit **a)**:

$$\begin{aligned}
||Mx||_A^2 &= \langle Mx, Mx\rangle_A \\
&= \langle x, M^AMx\rangle_A \\
&= \langle x, A^{-1}(I-AN^T)A(I-NA)x\rangle_A \\
&= \langle x, (I - N^TA - NA + N^TANA)x\rangle_A \\
&= \langle x, (I - (N^T+N)A + N^TANA)x\rangle_A \\
&< \langle x, (I - N^TANA + N^TANA)x\rangle_A \\
&= ||x||_A^2 = 1,
\end{aligned}$$

womit sofort $||M||_A < 1$ folgt.

6. Beispiel

Geg.: A SPD und $M_\omega^{SOR} = I - \omega(D+\omega L)^{-1}A$ die Iterationsmatrix des vorwärts SOR-Verfahrens, $\tilde{M}_\omega^{SOR} = I - \omega(D+\omega L^T)^{-1}A$ Iterationsmatrix des rückwärts SOR-Verfahrens, $M_\omega^{SSOR} = \tilde{M}_\omega^{SOR} M_\omega^{SOR}$ Iterationsmatrix des gedämpften SSOR-Verfahrens mit:

$$M_\omega^{SSOR} = I - \omega(2-\omega)(D+\omega L^T)^{-1}D(D+\omega L)^{-1}A$$

a)

Z.z.: $\tilde{M}_\omega^{SOR} = (M_\omega^{SOR})^A$
Beweis: Nach **Beispiel 5 a)** gilt $(M_\omega^{SOR})^A = A^{-1}(M_\omega^{SOR})^T A$ und damit

$$\begin{aligned} A^{-1}(M_\omega^{SOR})^T A &= A^{-1}(I - \omega(D+\omega L)^{-1}A)^T A \\ &= I - \omega A^{-1}A(D+\omega L^T)^{-1}A \\ &= \tilde{M}_\omega^{SOR} \end{aligned}$$

Damit kann unmittelbar auf die nicht-Negativität des zugehörigen Spektrums geschlossen werden.
Z.z.: $\sigma(M_\omega^{SSOR}) \subset \mathbb{R}_0^+$

$$\langle M_\omega^{SSOR}x, x\rangle_A = \langle \tilde{M}_\omega^{SOR} M_\omega^{SOR}x, x\rangle_A \overset{a)}{=} \langle M_\omega^{SOR}x, M_\omega^{SOR}x\rangle_A = \|M_\omega^{SOR}x\|_A^2 \geq 0$$

Damit ist M_ω^{SSOR} positiv semidefinit, womit die Behauptung folgt.

b)

Z.z.: $\|M_\omega^{SSOR}\|_A = \|M_\omega^{SOR}\|_A^2$
Beweis. Nach der Definition der Matrixnorm, **a)** und der Symmetrie des euklidschen Skalarprodukts gilt:

$$\|M_\omega^{SOR}\|_A = \sup_{\|x\|_A = \|y\|_A = 1} \langle M_\omega^{SOR}x, y\rangle_A = \sup_{\|x\|_A = \|y\|_A = 1} \langle x, \tilde{M}_\omega^{SOR}y\rangle_A = \|\tilde{M}_\omega^{SOR}\|_A$$

Damit folgt mit der Submultiplikativität der Matrixnorm:

$$\|M_\omega^{SSOR}\|_A = \|\tilde{M}_\omega^{SOR} M_\omega^{SOR}\|_A \leq \|\tilde{M}_\omega^{SOR}\|_A \|M_\omega^{SOR}\|_A = \|M_\omega^{SOR}\|_A^2$$

Andererseits gilt ebenso für jene x mit $\|x\|_A = 1$ nach der Cauchy-Schwarz Ungleichung und der Submultiplikativität:

$$\|M_\omega^{SOR}x\|_A^2 = \langle M_\omega^{SOR}x, M_\omega^{SOR}x\rangle_A \overset{a)}{=} \langle \tilde{M}_\omega^{SOR} M_\omega^{SOR}x, x\rangle_A \leq \|\tilde{M}_\omega^{SOR} M_\omega^{SOR}\|_A \|x\|_A^2 = \|M_\omega^{SSOR}\|_A$$

womit die Gleichheit bewiesen wäre.