Numerische Mathematik 1
Wintersemester 2020, Übungsblatt 5

Ausarbeitung über Moodle bis 6. November 2020. Nach diesem Datum erscheinen die nachträglichen Kommentare und die Lösungen der Teilnehmer.

1. Für $n \in \mathbb{N}$, $n > 1$, sei die invertierbare Matrix A durch $A = D+L+U$ mit einer invertierbaren diagonalen Matrix D, einer streng unterdreieckigen Matrix L und einer streng oberdreieckigen Matrix U zerlegt. Für $\boldsymbol{x}, \boldsymbol{b} \in \mathbb{R}^n$ wird die Jacobi-Methode $\boldsymbol{x}^{(k)} = T_{\mathrm{J}}\boldsymbol{x}^{(k-1)} + M_{\mathrm{J}}^{-1}\boldsymbol{b}$, $k \in \mathbb{N}$, auf Seite 79 im Skriptum verwendet, um das lineare Gleichungssystem $A\boldsymbol{x} = \boldsymbol{b}$ iterativ zu lösen. Kreuzen Sie bei den wahren Behauptungen an.

   (a) Wenn A streng diagonal dominant ist, folgt $\|T_{\mathrm{J}}\|_\infty < 1$.

   (b) Wenn A streng diagonal dominant ist, folgt $\|T_{\mathrm{J}}\|_1 < 1$.

   (c) Die Jacobi Iterierten $\{\boldsymbol{x}^{(k)}\}_{k=0}^{\infty}$ können divergieren, wenn die folgende Iteration konvergiert
   $$\boldsymbol{y}^{(k)} = -(L+U)D^{-1}\boldsymbol{y}^{(k-1)} + \boldsymbol{b}, \quad k \in \mathbb{N}.$$

   (d) Wenn $A^\top$ streng diagonal dominant ist, müssen die Jacobi Iterierten $\{\boldsymbol{x}^{(k)}\}_{k=0}^{\infty}$ zu $\boldsymbol{x} = A^{-1}\boldsymbol{b}$ konvergieren.

   Kommentare: Die Kombination von (c) und (d) ist der Beweis, dass die Jacobi Iteration konvergiert, nicht nur wenn $\|T_{\mathrm{J}}\|_\infty < 1$ gilt, sondern auch wenn $\|T_{\mathrm{J}}\|_1 < 1$ gilt.

2. Für $n \in \mathbb{N}$, $n > 1$, sei die streng diagonal dominante Matrix $A = \{a_{i,j}\}_{i,j=1}^{n} \in \mathbb{R}^{n\times n}$ durch $A = D + L + U$ mit einer diagonalen Matrix D, einer streng unterdreieckigen Matrix L und einer streng oberdreieckigen Matrix U zerlegt. Mit $\boldsymbol{x}, \boldsymbol{b} \in \mathbb{R}^n$ wird die Gauß-Seidel-Methode $\boldsymbol{x}^{(k)} = T_{\mathrm{GS}}\boldsymbol{x}^{(k-1)} + M_{\mathrm{GS}}^{-1}\boldsymbol{b}$, $k \in \mathbb{N}$, auf Seite 82 im Skriptum verwendet, um das lineare Gleichungssystem $A\boldsymbol{x} = \boldsymbol{b}$ iterativ zu lösen. Sei $(\lambda, \boldsymbol{x})$ ein Eigenwert-Eigenvektor-Paar für die Iterationsmatrix T_{GS} mit $|x_k| = \|\boldsymbol{x}\|_\infty = 1$. Kreuzen Sie bei den wahren Behauptungen an.

   (a) Es gilt $-U\boldsymbol{x} = \lambda(A-U)\boldsymbol{x}$.

   (b) Es gilt
   $$-\sum_{j=i}^{n} a_{i,j}x_j = \lambda \sum_{j=1}^{i} a_{i,j}x_j, \quad 1 \le i \le n.$$

   (c) Es gilt
   $$\lambda a_{k,k}x_k = -\sum_{j=k}^{n} a_{k,j}x_j - \lambda \sum_{j=1}^{k} a_{k,j}x_j$$

   (d) Es gilt
   $$|\lambda| \le \sum_{j=k+1}^{n} |a_{k,j}| \Big/ \left[|a_{k,k}| - \sum_{j=1}^{k-1} |a_{k,j}| \right] < 1.$$

Kommentare: Seien Teile (c) und (d) mit

$$-\sum_{j=i+1}^{n} a_{i,j}x_j = \lambda \sum_{j=1}^{i} a_{i,j}x_j, \quad 1 \le i \le n.$$

bzw.

$$\lambda a_{k,k}x_k = -\sum_{j=k+1}^{n} a_{k,j}x_j - \lambda \sum_{j=1}^{k-1} a_{k,j}x_j$$

korrigiert. Dann aus der letzten Gleichung folgt

$$|\lambda||a_{k,k}| \le \sum_{j=k+1}^{n} |a_{k,j}| + |\lambda| \sum_{j=1}^{k-1} |a_{k,j}|$$

was zu (d) führt. So sind die Schritte (a) – (d) der Beweis, dass die Gauß-Seidel Iteration konvergiert, wenn A streng diagonal dominant ist.

3. Basierend auf Beispiel 8 auf dem 2. Übungsblatt seien die Matrix A und die Vektoren $\boldsymbol{v}$ und $\boldsymbol{u}^\star$ gegeben durch den folgenden Matlab Code.

```
N  = 50; N1 = N-1; h  = 1/N;
mu = 0.01;
Q  = spdiags([-ones(N1,1),ones(N1,1)],[0,1],N1,N); i  = speye(N);
Dx = kron(i,Q); Dy = kron(Q,i); D  = Dx'*Dx + Dy'*Dy; I  = speye(N*N);
v  = sin(1:N*N)'; A  = I + mu*D/h^2; ustar  = A \ v;
```

Das System $A\boldsymbol{u} = \boldsymbol{v}$ soll nun mit der SSOR Methode $\boldsymbol{u}^{(k+1)} = T_{\text{SSOR}}^{(\omega)}\boldsymbol{u}^{(k)} + {M_{\text{SSOR}}^{(\omega)}}^{-1}\boldsymbol{v}$, $k \in \mathbb{N}$, iterativ gelöst werden. Seien $\{\boldsymbol{u}^{(k)}\}_{k=0}^{M}$ die Iterierten der SSOR Methode mit $\boldsymbol{u}^{(0)} = \boldsymbol{v}$. Der Dämpfungsparameter ω ist noch auszuwählen, aber der optimale Wert ist

$$\omega^\star = \operatorname*{argmin}_{\omega \in (0,2)} \rho(T_{\text{SSOR}}^{(\omega)}).$$

Kreuzen Sie bei den wahren Behauptungen an.

(a) Sei $\rho(T_{\text{J}})$ der Spektralradius der Iterationsmatrix T_{J} der Jacobi Methode. Dann ist $\omega^\star$ gegeben durch

$$\frac{2}{1 + \sqrt{1 - [\rho(T_{\text{J}})]^2}}.$$

(b) Wenn die Vektoriteration auf Seite 100 im Skriptum für die Matrix $T_{\text{SSOR}}^{(\omega^\star)}$ und mit dem Startvektor `ones(N*N,1)` angewendet wird, ergibt sich zu 4 signifikanten Ziffern der Grenzwert $\rho(T_{\text{SSOR}}^{(\omega^\star)}) = 0.8594$ als Abschätzung des dominanten Eigenwerts.

(c) Mit $\omega = 0.5$ ist $M = 465$ die kleinste Anzahl der SSOR Iterationen, mit der der relative Fehler $\|\boldsymbol{u}^{(k)} - \boldsymbol{u}^\star\|_2 / \|\boldsymbol{u}^\star\|_2$ unterhalb der Schwelle von einfacher Genauigkeit liegt.

(d) Mit $\omega = 1.5$ ist $M = 119$ die kleinste Anzahl der SSOR Iterationen, mit der der relative Fehler $\|\boldsymbol{u}^{(k)} - \boldsymbol{u}^\star\|_2 / \|\boldsymbol{u}^\star\|_2$ unterhalb der Schwelle von einfacher Genauigkeit liegt.

Kommentare: Sehen Sie diesen Code. Wie auf Seite 466 im Lehrbuch gesehen, ist die Formel im Teil (a) bekannt für $T_{\text{SOR}}^{(\omega)}$ (nicht SSOR) in dem Fall, dass A SPD und tridiagonal ist. Dies lässt sich für das reduzierte Beispiel `A = speye(N) + mu*Q'*Q/h^2` bestätigen. Weitere Details und eine grafische Darstellung der Wirkung des Dämpfungsparameters befinden sich hier. Für das obige 2D Beispiel gelten $\omega^\star = 1.679$ und $\rho(T_{\text{SSOR}}^{(\omega^\star)}) = 0.8594$ zu 4 signifikanten Ziffern. Die richtig Anzahl der Iterationen für Teil (c) ist $M = 865$.

4. Basierend auf Beispiel 8 auf dem 2. Übungsblatt seien die Matrix A und die Vektoren $\boldsymbol{v}$ und $\boldsymbol{u}^\star$ gegeben durch den folgenden Matlab Code.

```
N  = 50; N1 = N-1; h  = 1/N;
mu = 0.01;
Q  = spdiags([-ones(N1,1),ones(N1,1)],[0,1],N1,N); i  = speye(N);
Dx = kron(i,Q); Dy = kron(Q,i); D  = Dx'*Dx + Dy'*Dy; I  = speye(N*N);
v  = sin(1:N*N)'; A  = I + mu*D/h^2; ustar  = A \ v;
```

Das System $A\boldsymbol{u} = \boldsymbol{v}$ soll nun mit der Methode der konjugierten Gradienten iterativ gelöst werden. Seien $\{\boldsymbol{u}^{(k)}\}_{k=0}^{\infty}$ die Iterierten des Verfahrens mit $\boldsymbol{u}^{(0)} = \boldsymbol{v}$. Seien das A-Skalarprodukt und die A-Norm durch $(\boldsymbol{x}, \boldsymbol{y})_A = \boldsymbol{y}^\top A\boldsymbol{x}$ bzw. $\|\boldsymbol{x}\|_A = (\boldsymbol{x}, \boldsymbol{x})_A{}^{1/2}$ definiert. Kreuzen Sie bei den wahren Behauptungen an.

(a) Seien die Matrizen Q und D im Code definiert. Seien $\{(\lambda_j, \boldsymbol{v}_j\}_{i=1}^{N}$ die Eigenwert-Eigenvektor-Paaren für $Q^\top Q$, wie in den Kommentaren für Beispiel 5 auf Übungsblatt 3 detailliert. Dann sind die Eigenwerte der Matrix D gegeben durch $\{\lambda_i + \lambda_j\}_{i,j=1}^{N^2}$.

(b) Es gilt $\kappa_2(A) = 1 + 8\mu/h^2$.

(c) Es gelten

$$\|\boldsymbol{u}^{(k)} - \boldsymbol{u}^\star\|_A \leq 2\|\boldsymbol{u}^{(0)} - \boldsymbol{u}^\star\|_A \left[\frac{\sqrt{\kappa_2(A)} - 1}{\sqrt{\kappa_2(A)} + 1}\right]^k, \quad k \in \mathbb{N}.$$

(d) Es gelten

$$\|\boldsymbol{u}^{(k)} - \boldsymbol{u}^\star\|_A \leq 2\|\boldsymbol{u}^{(0)} - \boldsymbol{u}^\star\|_A \left[\frac{\sqrt{\kappa_2(A)} - 1}{\sqrt{\kappa_2(A)} + 1}\right]^{2k}, \quad k \in \mathbb{N}.$$

Kommentare: Sehen Sie diesen Code. Anhand vom Teil (a) ist $\{1 + \mu(\lambda_i + \lambda_j)/h^2\}_{i,j=1}^{N^2}$ das Spektrum von A. Mit $\lambda_i = 4\sin^2(\pi(i-1)/(2N))$ liegt das Spektrum im Intervall $[1, 1+8\mu/h^2]$. Jedoch gilt $\lambda_i + \lambda_j = 8$ für kein Paar (i, j), und daher gilt $\kappa_2(A) < 1+8\mu/h^2$. Weitere Details über die Methode der konjugierten Gradienten befinden sich hier, und die Fehlerabschätzung im Teil (c) erscheint auf Seite 17.

5. Basierend auf Beispiel 8 auf dem 2. Übungsblatt seien die Matrix A und die Vektoren $\boldsymbol{v}$ und $\boldsymbol{u}^\star$ gegeben durch den folgenden Matlab Code.

```
N  = 50; N1 = N-1; h  = 1/N;
mu = 0.01;
Q  = spdiags([-ones(N1,1),ones(N1,1)],[0,1],N1,N); i  = speye(N);
Dx = kron(i,Q); Dy = kron(Q,i); D  = Dx'*Dx + Dy'*Dy; I  = speye(N*N);
v  = sin(1:N*N)'; A  = I + mu*D/h^2; ustar  = A \ v;
```

Das System $A\boldsymbol{u} = \boldsymbol{v}$ soll nun mit der präkonditionierten Methode der konjugierten Gradienten iterativ gelöst werden. Wie auf Seite 94 im Skriptum angedeutet ist eine SPD Matrix C mit $A \approx C^2$ auszuwählen. Seien $\{\boldsymbol{u}^{(k)}\}_{k=0}^{\infty}$ die Iterierten des Verfahrens mit $C = I$ und $\boldsymbol{u}^{(0)} = \boldsymbol{v}$. Kreuzen Sie bei den wahren Behauptungen an.

(a) Mit $B = C^2$ lässt sich die präkonditionierte Methode der konjugierten Gradienten zur Lösung des Systems $A\boldsymbol{u} = \boldsymbol{v}$ mit `kmax` Iterationen folgendermaßen implementieren.

```
u  = v;
r  = v - A*u;
z  = B \ r;
r1 = z'*r; r2 = r1; a = 0*r;
for k=1:kmax
    s  = r1/r2;
    a  = z + s*a;
    w  = A*a;
    t  = r1/(a'*w);
    u  = u + t*a;
    r  = r - t*w;
    z  = B \ r;
    r2 = r1;
    r1 = z'*r;
end
```

(b) Die approximierten Inversen für die Jacobi Methode und die Gauß-Seidel Methode seien durch M_{J}^{-1} bzw. M_{SGS}^{-1} bezeichnet. Es gelten

$$\kappa_2(M_{\mathrm{J}}^{-1/2} A M_{\mathrm{J}}^{-1/2}) < \kappa_2(M_{\mathrm{SGS}}^{-1/2} A M_{\mathrm{SGS}}^{-1/2}) < \kappa_2(A).$$

(c) Seien $\{\boldsymbol{u}_{\mathrm{J}}^{(k)}\}_{k=0}^{\infty}$ die Iterierten des Verfahrens mit $C^2 = M_{\mathrm{J}}$ und $\boldsymbol{u}_{\mathrm{J}}^{(0)} = \boldsymbol{v}$. Es gelten

$$\|\boldsymbol{u}_{\mathrm{J}}^{(k)} - \boldsymbol{u}^{\star}\|_2 < \|\boldsymbol{u}^{(k)} - \boldsymbol{u}^{\star}\|_2, \quad k = 1, \ldots, N.$$

(d) Seien $\{\boldsymbol{u}_{\mathrm{SGS}}^{(k)}\}_{k=0}^{\infty}$ die Iterierten des Verfahrens mit $C^2 = M_{\mathrm{SGS}}$ und $\boldsymbol{u}_{\mathrm{SGS}}^{(0)} = \boldsymbol{v}$. Es gelten

$$\|\boldsymbol{u}_{\mathrm{SGS}}^{(k)} - \boldsymbol{u}^{\star}\|_2 < \|\boldsymbol{u}^{(k)} - \boldsymbol{u}^{\star}\|_2, \quad k = 1, \ldots, N.$$

Kommentare: Sehen Sie diesen Code. Die Methode der konjugierten Gradienten auf Seite 93 im Skriptum hat die folgende Form für das präkonditionierte System $(C^{-1}AC^{-1})(C\boldsymbol{x}) = C^{-1}\boldsymbol{b}$ auf Seite 94,

$$\begin{aligned}
& C\boldsymbol{x}^{(0)} \text{ sei gegeben} \\
\boldsymbol{v}^{(1)} &= (C^{-1}\boldsymbol{r}^{(0)}) = (C^{-1}\boldsymbol{b}) - (C^{-1}AC^{-1})(C\boldsymbol{x}^{(0)}) \\
\\
\text{Für } k &= 1, 2, \ldots \\
t^{(k)} &= (C^{-1}\boldsymbol{r}^{(k-1)}) \cdot (C^{-1}\boldsymbol{r}^{(k-1)}) / \boldsymbol{v}^{(k)} \cdot (C^{-1}AC^{-1})\boldsymbol{v}^{(k)} \\
(C\boldsymbol{x}^{(k)}) &= (C\boldsymbol{x}^{(k-1)}) + t^{(k)}\boldsymbol{v}^{(k)} \\
(C^{-1}\boldsymbol{r}^{(k)}) &= (C^{-1}\boldsymbol{r}^{(k-1)}) - t^{(k)}(C^{-1}AC^{-1})\boldsymbol{v}^{(k)} \\
s^{(k)} &= (C^{-1}\boldsymbol{r}^{(k)}) \cdot (C^{-1}\boldsymbol{r}^{(k)}) / (C^{-1}\boldsymbol{r}^{(k-1)}) \cdot (C^{-1}\boldsymbol{r}^{(k-1)}) \\
\boldsymbol{v}^{(k+1)} &= (C^{-1}\boldsymbol{r}^{(k)}) + s^{(k)}\boldsymbol{v}^{(k)}
\end{aligned}$$

Mit $\tilde{\boldsymbol{v}}^{(k)} = C^{-1}\boldsymbol{v}^{(k)}$ folgt der Algorithmus, der im Teil (a) implementiert ist,

$$\begin{aligned} & \boldsymbol{x}^{(0)} \text{ sei gegeben} \\ \tilde{\boldsymbol{v}}^{(1)} &= C^{-2}\boldsymbol{r}^{(0)}, \quad \boldsymbol{r}^{(0)} = \boldsymbol{b} - A\boldsymbol{x}^{(0)} \\ \\ \text{Für } k &= 1, 2, \dots \\ t^{(k)} &= \boldsymbol{r}^{(k-1)} \cdot C^{-2}\boldsymbol{r}^{(k-1)} / \tilde{\boldsymbol{v}}^{(k)} \cdot A\tilde{\boldsymbol{v}}^{(k)} \\ \boldsymbol{x}^{(k)} &= \boldsymbol{x}^{(k-1)} + t^{(k)}\tilde{\boldsymbol{v}}^{(k)} \\ \boldsymbol{r}^{(k)} &= \boldsymbol{r}^{(k-1)} - t^{(k)}A\tilde{\boldsymbol{v}}^{(k)} \\ s^{(k)} &= \boldsymbol{r}^{(k)} \cdot C^{-2}\boldsymbol{r}^{(k)} / \boldsymbol{r}^{(k-1)} \cdot C^{-2}\boldsymbol{r}^{(k-1)} \\ \tilde{\boldsymbol{v}}^{(k+1)} &= \boldsymbol{r}^{(k)} + s^{(k)}\tilde{\boldsymbol{v}}^{(k)} \end{aligned}$$

mit dem nur die Matrix C^{-2} erscheint und nicht ihre Wurzel. Dieser Algorithmus befindet sich auf Seite 23 hier.

Für eine SPD Matrix A ist die Konditionszahl $\kappa_2(A)$ gegeben durch den Quotienten $\max\{\sigma(A)\}/\min\{\sigma(A)\}$, wobei $\sigma(A)$ das Spektrum der Matrix A bezeichnet. Für eine SPD Matrix $M = V\Lambda V^\top$, Λ diagonal und V orthogonal, ist die $p$te Potenz der Matrix durch $M^p = V\Lambda^p V^\top$ definiert. Jedoch wird das Spektrum der Matrix $M^{-\frac{1}{2}}AM^{-\frac{1}{2}}$ bequem durch die verallgemeinerten Eigenwerte λ des Systems $A\boldsymbol{x} = \lambda M\boldsymbol{x}$ bestimmt. Folglich sind Konditionszahlen für Teil (b) gegeben durch

```
E_A = eig(full(A));
K_A = max(E_J)/min(E_J);            % = 200.8027
                                    % ist auch cond(full(A))
L   = tril(A,-1);
U   = triu(A,+1);
D   = A - L - U;
M_J = D;
E_J = eig(full(A),full(M_J));
K_J = max(E_J)/min(E_J);            % = 197.2977
                                    % ist auch cond(full(Mh*A*Mh)) mit
                                    % Mh = D^(-0.5)
M_SGS = (D + U);
M_SGS = D \ MSGS;
M_SGS = (D + L)*M_SGS;
E_SGS = eig(full(A),full(M_SGS));   %
K_SGS = max(E_SGS)/min(E_SGS);      % = 25.4398
                                    % ist auch cond(full(Mh*A*Mh)) mit
                                    % Mh = V*Md^(-0.5)*V' wobei
                                    % [V,Md] = eig(full(M_SGS))
```

Da $\kappa_2(M_{\text{J}}^{-1/2}AM_{\text{J}}^{-1/2}) < \kappa_2(A)$ gilt, ist zu erwarten, dass (c) richtig ist. Dies ist aber nicht der Fall, wie die Rechnungen zeigen. Der Unterschied zwischen den Konditionszahlen `200.8027` und `197.2977` ist nicht ausreichend. Jedoch ist $\kappa_2(M_{\text{SGS}}^{-1/2}AM_{\text{SGS}}^{-1/2}) =$ `25.4398` viel kleiner als $\kappa_2(A) =$ `200.8027`, und die Rechnungen zeigen, dass (d) richtig ist.

6. Basierend auf Beispiel 7 auf dem 3. Übungsblatt seien $\{\boldsymbol{W}^k\}_{k=0}^N \subset \mathbb{R}^N$ gegeben durch die folgende Diskretisierung der Wärmegleichung.

```
N = 50; N1 = N-1; h = 1/N; ta = h; ka = 1;
x = linspace(0,1,N+1); x = x(1:N)' + h/2;
Q = spdiags([-ones(N1,1),ones(N1,1)],[0,1],N1,N);
D = Q'*Q;  B = -ka*D/h^2; I = speye(N); A = I - ta*B;
W = sign(x-0.5);
for k=1:N
    W  = A\W;
end
```

Die Systeme $A\boldsymbol{W}^k = \boldsymbol{W}^{k-1}$, $k \in \mathbb{N}$, sollen nun mit der Methode der konjugierten Gradienten iterativ gelöst werden, und die iterativ berechneten Lösungen seien mit $\{\tilde{\boldsymbol{W}}^k\}_{k=0}^N$ bezeichnet. Anfänglich gilt $\tilde{\boldsymbol{W}}^0 = \boldsymbol{W}^0$. Weiters seien die Iterierten beim $k$ten Zeitschritt mit $\{\tilde{\boldsymbol{W}}^{k,l}\}_{l=0}^M$ bezeichnet. Diese Iteration endet mit $\tilde{\boldsymbol{W}}^k = \tilde{\boldsymbol{W}}^{k,M}$ und startet mit $\tilde{\boldsymbol{W}}^{k,0}$, das noch auszuwählen ist. Kreuzen Sie bei den wahren Behauptungen an.

(a) Mit $\tilde{\boldsymbol{W}}^{k,0} = \tilde{\boldsymbol{W}}^{k-1}$ gelten für jede Anzahl $M = 1, \ldots, 10$ der Iterationen,

$$|(\boldsymbol{W}^k)_i| < |(\tilde{\boldsymbol{W}}^k)_i|, \quad i = 1, \ldots, N, \quad k = 1, \ldots, N.$$

(b) Mit $\tilde{\boldsymbol{W}}^{k,0} = \tilde{\boldsymbol{W}}^{k-1}$ gelten für jede Anzahl $M = 1, \ldots, 10$ der Iterationen,

$$(\boldsymbol{W}^k)_i < (\boldsymbol{W}^k)_{i+1}, \quad (\tilde{\boldsymbol{W}}^k)_i < (\tilde{\boldsymbol{W}}^k)_{i+1}, \quad i = 1, \ldots, N-1, \quad k = 0, \ldots, N.$$

(c) Mit $\tilde{\boldsymbol{W}}^{k,0} = \tilde{\boldsymbol{W}}^{k-1}$ ist $M = 25$ die kleinste Anzahl der Iterationen, mit der der relative Fehler $\|\{\tilde{\boldsymbol{W}}^k - \boldsymbol{W}^k\}_{k=0}^N\|_\infty / \|\{\boldsymbol{W}^k\}_{k=0}^N\|_\infty$ unterhalb der Schwelle von einfacher Genauigkeit liegt.

(d) Mit $\tilde{\boldsymbol{W}}^{k,0} = \texttt{ones(N,1)}$ ist $M = 25$ die kleinste Anzahl der Iterationen, mit der der relative Fehler $\|\{\tilde{\boldsymbol{W}}^k - \boldsymbol{W}^k\}_{k=0}^N\|_\infty / \|\{\boldsymbol{W}^k\}_{k=0}^N\|_\infty$ unterhalb der Schwelle von einfacher Genauigkeit liegt.

Kommentare: Sehen Sie diesen Code. Der Diffusionsprozess in der Wärmegleichung ist dämpfend, und daher ist die Abweichung des Zustandes $\boldsymbol{W}^k$ von seinem Fliessgleichgewicht eine monton fallende Funktion von k. Weiters bleiben die Komponenten $\boldsymbol{W}^k = \{W_i^k\}_{i=1}^N$ eine steigende Funktion von i für jedes k. Insofern die iterativ berechneten Zustände $\tilde{\boldsymbol{W}}^k$ akkurat sind, widerspieglen sie diese qualitativen Eigenschaften der Lösung der Wärmegleichung. Obwohl gewisse iterative Verfahren ihre eigenen dämpfenden Eigenschaften genießen, ist die Methode der konjugierten Gradienten nicht derartig.

Mit $M = 25$ Iterationen gilt die Ungleichung im Teil (a) nicht mehr. Mit $M = 5, 6, 7$ gelten die Ungleichungen im Teil (b) nicht. Da die Methode der konjugierten Gradienten ein System in einer endlichen Anzahl von Iterationen mit exakter Arithmetik lösen soll, sinkt der relative Fehler schlagartig von `5.84081e-02` für $M = 24$ auf `3.88578e-15` für $M = 25$ im Teil (c). Da die Zustände $\boldsymbol{W}^k$ und $\boldsymbol{W}^{k-1}$ nicht sehr unterschiedlich sind, ist $\boldsymbol{W}^{k-1}$ ein sinnvoller Startvektor für die Iteration im $k$ten Zeitschritt. Wenn dies nicht verwendet wird, sind $M = 26$ Iterationen im Teil (d) notwendig.

7. Basierend auf Beispiel 8 auf dem 3. Übungsblatt seien $\{\boldsymbol{W}^k\}_{k=0}^N \subset \mathbb{R}^N$ gegeben durch die folgende Diskretisierung der Wellengleichung.

```
N = 50; N1 = N-1; h = 1/N; ta = h; nu = 1;
x = linspace(0,1,N+1); x = x(1:N)' + h/2;
Q = spdiags([-ones(N1,1),ones(N1,1)],[0,1],N1,N);
B = (nu/h)*[sparse(N1,N1),Q;-Q',sparse(N,N)];
I = speye(2*N-1); A = I - (ta/2)*B; C = I + (ta/2)*B;
a = cos(pi*(2*x-1));
W = [(nu/h)*Q*a;0*a];
for k=1:N
    Ws = W;
    W = A\(C*Ws);
    a = a + (ta/2)*(W(N:(2*N-1)+Ws(N:2*N-1)));
end
```

Die Systeme $A\boldsymbol{W}^k = C\boldsymbol{W}^{k-1}$, $k \in \mathbb{N}$, sollen nun mit der symmetrischen Gauß-Seidel Methode gelöst werden, und die iterativ berechneten Lösungen seien mit $\{\tilde{\boldsymbol{W}}^k\}_{k=0}^N$ bezeichnet. Anfänglich gilt $\tilde{\boldsymbol{W}}^0 = \boldsymbol{W}^0$. Die Auslenkungen seien durch

$$\boldsymbol{a}^k = \boldsymbol{a}^{k-1} + (\tau/2)\{(\boldsymbol{W}^k + \boldsymbol{W}^{k-1})_i\}_{i=N}^{2N-1}, \quad k = 1, \ldots, N$$

und

$$\tilde{\boldsymbol{a}}^k = \tilde{\boldsymbol{a}}^{k-1} + (\tau/2)\{(\tilde{\boldsymbol{W}}^k + \tilde{\boldsymbol{W}}^{k-1})_i\}_{i=N}^{2N-1}, \quad k = 1, \ldots, N$$

gegeben. Weiters seien die Iterierten beim $k$ten Zeitschritt durch $\{\tilde{\boldsymbol{W}}^{k,l}\}_{l=0}^M$ bezeichnet. Diese Iteration endet mit $\tilde{\boldsymbol{W}}^k = \tilde{\boldsymbol{W}}^{k,M}$ und startet mit $\tilde{\boldsymbol{W}}^{k,0}$, das noch auszuwählen ist. Kreuzen Sie bei den wahren Behauptungen an.

(a) Mit $\tilde{\boldsymbol{W}}^{k,0} = \tilde{\boldsymbol{W}}^{k-1}$ gelten für jede Anzahl $M = 1, \ldots, 10$ der Iterationen zu 3 signifikanten Ziffern,
$$\|\boldsymbol{W}^k\|_2 = \|\tilde{\boldsymbol{W}}^k\|_2 = 31.4, \quad k = 0, \ldots, N.$$

(b) Mit $\tilde{\boldsymbol{W}}^{k,0} = \tilde{\boldsymbol{W}}^{k-1}$ gelten für jede Anzahl $M = 1, \ldots, 10$ der Iterationen,
$$|(\boldsymbol{a}^k)_i| < |(\tilde{\boldsymbol{a}}^k)_i|, \quad i = 1, \ldots, N, \quad k = 1, \ldots, N.$$

(c) Mit $\tilde{\boldsymbol{W}}^{k,0} = \tilde{\boldsymbol{W}}^{k-1}$ ist $M = 4$ die kleinste Anzahl der Iterationen, mit der der relative Fehler $\|\{\tilde{\boldsymbol{W}}^k - \boldsymbol{W}^k\}_{k=0}^N\|_\infty / \|\{\boldsymbol{W}^k\}_{k=0}^N\|_\infty$ unterhalb der Schwelle von einfacher Genauigkeit liegt.

(d) Mit $\tilde{\boldsymbol{W}}^{k,0} = 10^{-5}\text{sign}(\tilde{\boldsymbol{W}}^{k-1})$ ist $M = 780$ die kleinste Anzahl der Iterationen, mit der der relative Fehler $\|\{\tilde{\boldsymbol{W}}^k - \boldsymbol{W}^k\}_{k=0}^N\|_\infty / \|\{\boldsymbol{W}^k\}_{k=0}^N\|_\infty$ unterhalb der Schwelle von einfacher Genauigkeit liegt.

Kommentare: Sehen Sie diesen Code. Die Methode auf Seiten 228-229 im Skriptum für die Wellengleichung ist im obigen Code mit "\" implementiert. Wie im Skriptum angedeutet, entspricht $\{W_i^k\}_{i=1}^{N-1} = \nu \boldsymbol{a}_x^k$ der potentiellen Energie der Schwingungen und $\{W_i^k\}_{i=N}^{2N-1} \approx \boldsymbol{a}_t^k$ entspricht der kinetischen Energie. Also ist $\|\boldsymbol{W}^k\|_2$ die gesamte Energie zur Zeit $t^k = k\tau$, die im Lauf der Zeit erhalten bleiben soll. Diese Energieerhaltung gelingt der Methode im obigen Code, und deswegen gilt $\|\boldsymbol{W}^k\|_2 = 31.4$, $k = 0, \ldots, N$ Jedoch wenn nur $M = 1$ Iteration der symmetrischen Gauß-Seidel Methode anstatt "\" verwendet wird, gilt $\|\tilde{\boldsymbol{W}}^k\|_2 = 31.4$, $k = 0, \ldots, N$, nicht mehr. Obwohl die symmetrische Gauß-Seidel

Methode eine dämpfende Wirkung für Teil (b) im Beispiel 5 genießen könnten, gibt es kein Gegenstück für Teil (b) hier.

Da die Zustände $\boldsymbol{W}^k$ und $\boldsymbol{W}^{k-1}$ nicht sehr unterschiedlich sind, ist $\boldsymbol{W}^{k-1}$ ein sinnvoller Startvektor für die Iteration im $k$ten Zeitschritt. Wenn dies nicht verwendet wird, sind $M = 780$ Iterationen im Teil (d) notwendig. So viele Iterationen sind auch notwendig, weil der Startvektor $\tilde{\boldsymbol{W}}^{k,0} = \{\tilde{W}_i^{k,0}\}_{i=1}^{2N-1}$ keine glatte Funktion von i ist. Mit einem glatten Startvektor $\tilde{\boldsymbol{W}}^{k,0} = \boldsymbol{0}$ sind die relativen Fehler weniger mit `1.54786e-06` für $M = 3$ und `6.05741e-09` für $M = 4$. Jedoch mit $\tilde{\boldsymbol{W}}^{k,0} = \tilde{\boldsymbol{W}}^{k-1}$ sind die relativen Fehler noch besser mit `1.92030e-07` für $M = 3$ und `7.57108e-10` für $M = 4$.

8. Die folgenden Matrizen seien definiert:

$$A = \begin{bmatrix} +1 & +1 & +1 \\ -1 & +1 & +1 \\ -1 & -1 & +1 \end{bmatrix}, \quad B = \begin{bmatrix} +8 & -2 & +4 \\ -1 & +7 & -2 \\ -3 & +3 & 0 \end{bmatrix}, \quad C = \begin{bmatrix} +2 & -1 & -1 \\ -1 & +2 & +1 \\ +1 & +1 & +2 \end{bmatrix}.$$

Seien $\{\mu^{(k)}\}_{k=1}^{\infty}$ und $\{\boldsymbol{x}^{(k)}\}_{k=1}^{\infty}$ durch die Vektoriteration auf Seite 100 oder die inverse Vektoriteration auf Seite 102 im Skriptum gegeben, wobei $\boldsymbol{x}^{(0)}$ durch

```
x = randn(3,1);
x = x/norm(x,inf);
```

bestimmt wird. Kreuzen Sie bei den wahren Behauptungen an.

(a) Für A führt die Vektoriteration zum Ergebnis $\mu^{(k)} \overset{k\to\infty}{\longrightarrow} \rho(A)$.

(b) Für B führt die Vektoriteration zum Ergebnis, dass $\{\boldsymbol{x}^{(k)}\}_{k=1}^{\infty}$ zu einem Vielfachen von $(1, 1, 0)$ konvergiert.

(c) Für C führt die Vektoriteration zum Ergebnis $|\mu^{(k)} - 3|/(2/3)^k \overset{k\to\infty}{\longrightarrow} c$, $c \in (0, \infty)$.

(d) Für C führt die inverse Vektoriteration mit $q = 0$ zum Ergebnis $\mu^{(k)} \overset{k\to\infty}{\longrightarrow} 1$.

Kommentare: Sehen Sie diesen Code. Die Matrix A hat Eigenwerte $\{1, 1 \pm \sqrt{3}\imath\}$ und daher wird $\rho(A) = |1 \pm \sqrt{3}\imath|$ gegeben durch 2 betragsmäßig gleiche Eigenwerte. Daher sind die Voraussetzungen der Vektoriteration nicht erfüllt, und die Rechnungen mit den Anfangsbedingungen zeigen, dass die Werte $\{\mu^{(k)}\}$ zum Spektralradius von A nicht konvergieren.

Die Matrix B hat die Eigenwerte $\{3, 6\}$ und einen 2-dimensionalen Eigenraum für den größten Eigenwert. Daher sind die Voraussetzungen der Vektoriteration nicht erfüllt, und die Rechnungen mit den Anfangsbedingungen zeigen, dass die Vektoren $\{\boldsymbol{x}^{(k)}\}$ nicht eindeutig konvergieren.

Die Matrix C hat die Eigenwerte $\{1, 2, 3\}$, die Voraussetzungen der Vektoriteration sind erfüllt und die Konvergenzrate der Werte $\{\mu^{(k)}\}$ hängt vom Quotienten $2/3$ der zwei größten Eigenwerte ab. Die inverse Vektoriteration mit $q = 0$ ist die Vektoriteration für die Matrix C^{-1} mit Eigenwerten $\{1/3, 1/2, 1\}$. Die Voraussetzungen der Vektoriteration sind erfüllt, und die Werte $\{\mu^{(k)}\}$ konvergieren zum Wert $\rho(C^{-1}) = 1$.