Numerische Mathematik 1
Wintersemester 2020, Übungsblatt 6

Ausarbeitung über Moodle bis 13. November 2020. Nach diesem Datum erscheinen die nachträglichen Kommentare und die Lösungen der Teilnehmer.

1. Drei Varianten vom Householder Algorithmus werden untersucht. Bei Variante 1 gibt es keine Änderung von dem Algorithmus auf Seite 106 im Skriptum. Bei Variante 2 wird mit $\|\boldsymbol{x}\|_\infty$ nicht skaliert, d.h. es gelten $\boldsymbol{v} = \boldsymbol{x}$ und $\sigma = \rho$, und sonst sind Varianten 1 und 2 gleich. Bei Variante 3 wird das Vorzeichen umgekehrt, d.h. es gilt $\rho = -s(v_1)\|\boldsymbol{v}\|_2$, und sonst sind Varianten 1 und 3 gleich. Für $n \in \mathbb{N}$ werden diese Varianten der für die folgenden Vektoren angewendet:

```
e1 = zeros(n,1); e1(1) = 1;
x  = e1 + 1.0e-7*sin(1:n)'/sqrt(n);
y  = realmax*sin(1:n)'/sqrt(n);
```

   Kreuzen Sie bei den wahren Behauptungen an.

   (a) Bei Variante 1 gilt $\|P\boldsymbol{x} + \sigma\hat{\boldsymbol{e}}_1\|_2 \leq 10^{-14}\|\boldsymbol{x}\|_2$ für $n = 2, \ldots, 1000$.
   (b) Bei Variante 1 gilt $\|P\boldsymbol{y} + \sigma\hat{\boldsymbol{e}}_1\|_2 \leq 10^{-14}\|\boldsymbol{y}\|_2$ für $n = 2, \ldots, 1000$.
   (c) Bei Variante 2 gilt $\|P\boldsymbol{y} + \sigma\hat{\boldsymbol{e}}_1\|_2 \leq 10^{-14}\|\boldsymbol{y}\|_2$ für $n = 2, \ldots, 1000$.
   (d) Bei Variante 3 gilt $\|P\boldsymbol{x} + \sigma\hat{\boldsymbol{e}}_1\|_2 \leq 10^{-14}\|\boldsymbol{x}\|_2$ für $n = 2, \ldots, 1000$.

   Kommentare: Sehen Sie diesen Code. Für `x` findet Auslöschung bei Variante 3 stattt. Für `y` kann die 2-Norm bei Variante 2 nicht berechnet werden, da die Potenzen nicht speicherbar sind.

2. Für das Randwertproblem auf Seite 15 im Skriptum sei die Neumann Randbedingung $u' = 0$ durch die periodische Randbedingung $u(0) = u(1)$ ersetzt. Für $N \in \mathbb{N}$, $h = 1/N$, $\mu > 0$ und $\boldsymbol{v} \in \mathbb{R}^N$ sei das neue Randwertproblem diskretisiert durch:

```
C = spdiags([-ones(N,1),ones(N,1)],[0,1],N,N); C(N,1) = 1;
I = speye(N); A = I + mu*C'*C/h^2; u = A \ v;
```

   Die Matrix $A \in \mathbb{R}^{N\times N}$ soll mit $A = QHQ^{\mathrm{T}}$ zerlegt werden, wobei die Matrizen Q und H orthogonal bzw. tridiagonal sind. Implementieren Sie den Algorithmus auf Seite 111 im Skriptum mit $N = 5$ und $\mu = h^2$. Kreuzen Sie bei den wahren Behauptungen an.

   (a) Die berechnete Matrix H ist zu den angegebenen dezimalen Stellen

3.0000	1.4142	0	0	0
1.4142	3.0000	-1.0000	0.0000	0.0000
0	-1.0000	2.0000	0.0000	-0.0000
0	0.0000	0.0000	4.0000	-1.0000
0	0.0000	-0.0000	-1.0000	3.0000

   (b) Die berechnete Matrix Q ist zu den angegebenen dezimalen Stellen

1.0000	0	0	0	0
0	-0.7071	0.0000	0.0000	-0.7071
0	0	-0.7071	-0.7071	-0.0000
0	0	-0.7071	0.7071	-0.0000
0	-0.7071	-0.0000	-0.0000	0.7071

(c) Die berechnete Matrix Q ist zu den angegebenen dezimalen Stellen

1.0000	0	0	0	0
0	0	-0.7071	-0.7071	-0.0000
0	-0.7071	0.0000	0.0000	-0.7071
0	-0.7071	-0.0000	-0.0000	0.7071
0	0	-0.7071	0.7071	-0.0000

(d) Die berechnete Matrix $Q^{\mathrm{T}}A$ ist zu den angegebenen dezimalen Stellen die obere Hessenberg Matrix,

3.0000	-1.0000	0	0	-1.0000
1.4142	-2.1213	0.7071	0.7071	-2.1213
0	2.3452	-1.9188	1.0660	-1.9188
0	0	2.6112	-1.7060	-0.4526
0	0	-0.0000	-2.5403	1.2702

Kommentare: Sehen Sie diesen Code mit der Funktion house. Für dieses Beispiel entsteht ein Vektor `h` im Householder-Aufruf, der Null oder fast Null ist. Deswegen muss der Test ($\xi \approx 0$) in der Householder Funktion verwendet werden. Je nachdem ob Matlab oder Octave verwendet wird, kann eine sinnvolle Zerlegung ohne den Test ausgegeben werden, aber die Antworten in Teilen (a) und (b) mit dem Test sind die allgemein zuverlässen Ergebnisse. Antwort (d) ergibt sich mit `Asym = false` im Pseudo-Code, aber wie auf Seite 110 im Skriptum angedeutet, lässt sich dieses Ergebnis von der tridiagonalen Zerlegung nicht direkt berechnen.

3. Für $N \in \mathbb{N}$, $h = 1/N$, $\mu > 0$ und $\boldsymbol{v} \in \mathbb{R}^N$ sei das Randwertproblem auf Seite 15 im Skriptum diskretisiert durch:

```
C = spdiags([-ones(N,1),ones(N,1)],[0,1],N,N);
I = speye(N); A = I + mu*C'*C/h^2; u = A \ v;
```

Die Matrix $A \in \mathbb{R}^{N\times N}$ soll durch den QR-Algorithmus iterativ diagonalisiert werden: $A^{(1)} = A$, $A^{(k)} = Q^{(k)}R^{(k)}$, $A^{(k+1)} = R^{(k)}Q^{(k)}$, $k \in \mathbb{N}$. Implementieren Sie den Algorithmus auf Seite 121 im Skriptum mit $N = 5$ und $\mu = h^2$. Kreuzen Sie bei den wahren Behauptungen an.

(a) Die berechnete Matrix $Q^{(5)}$ ist zu den angegebenen dezimalen Stellen

0.9917	-0.1287	0	0	0
0.1258	0.9697	-0.2095	0	0
0.0263	0.2031	0.9559	-0.2103	0
0.0056	0.0432	0.2034	0.9666	-0.1496
0.0008	0.0065	0.0308	0.1462	0.9888

(b) Die berechnete Matrix $A^{(10)}$ ist zu den angegebenen dezimalen Stellen

```
   4.6049   -0.2467         0         0         0
  -0.2467    3.8839   -0.1671         0         0
        0   -0.1671    2.7374   -0.0510         0
        0         0   -0.0510    1.6914   -0.0292
        0         0         0   -0.0292    1.0824
```

(c) Die berechnete Matrix $R^{(15)}$ ist zu den angegebenen dezimalen Stellen

```
   4.6723   -0.1769    0.0006         0         0
        0    3.8385   -0.0500    0.0000         0
        0         0    2.7158   -0.0077    0.0000
        0         0         0    1.6903   -0.0051
        0         0         0         0    1.0810
```

(d) Die berechnete Matrix $A^{(20)}$ ist zu den angegebenen dezimalen Stellen die tridiagonale Matrix,

```
   4.6825   -0.0000         0         0         0
  -0.0000    3.8308   -0.0000         0         0
        0   -0.0000    2.7154   -0.0000         0
        0         0   -0.0000    1.6903   -0.0000
        0         0         0   -0.0000    1.0810
```

Kommentare: Sehen Sie diesen Code. Antwort (a) ist offenbar falsch, weil $Q^{(k)}$ eine obere Hessenberg Matrix sein muss. Die Eigenwerte der Matrix A sind auf der Diagonal im Teil (d), aber nach 20 Iterationen ist $A^{(k)}$ noch nicht so nah bei dem End-Ergebnis.

4. Für $N \in \mathbb{N}$, $h = 1/N$, $\mu > 0$ und $\boldsymbol{v} \in \mathbb{R}^N$ sei das Randwertproblem auf Seite 15 im Skriptum diskretisiert durch:

```
C = spdiags([-ones(N,1),ones(N,1)],[0,1],N,N);
I = speye(N); A = I + mu*C'*C/h^2; u = A \ v;
```

Die Matrix $A \in \mathbb{R}^{N\times N}$ soll durch den QR-Algorithmus mit Verschiebung iterativ diagonalisiert werden: $A^{(1)} = A$, $A^{(k)} - qI = Q^{(k)}R^{(k)}$, $A^{(k+1)} = R^{(k)}Q^{(k)} + qI$, $k \in \mathbb{N}$, wobei die Verschiebung q noch auszuwählen ist. Führen Sie ein Verschiebung in den Algorithmus auf Seite 121 im Skriptum ein, und implementieren Sie ihn mit $N = 5$ und $\mu = h^2/10$. Kreuzen Sie bei den wahren Behauptungen an.

(a) Mit $q = 1$ ist die berechnete Matrix $Q^{(5)}$ zu den angegebenen dezimalen Stellen

```
   0.9921   -0.1254         0         0         0
   0.1236    0.9777   -0.1696         0         0
   0.0212    0.1680    0.9839   -0.0568         0
   0.0012    0.0096    0.0560    0.9984   -0.0004
   0.0000    0.0000    0.0000    0.0004    1.0000
```

(b) Mit $q = 1$ ist die berechnete Matrix $A^{(10)}$ zu den angegebenen dezimalen Stellen

```
   1.3656   -0.0148         0         0         0
  -0.0148    1.2856   -0.0040         0         0
        0   -0.0040    1.1717   -0.0001         0
        0         0   -0.0001    1.0690   -0.0000
        0         0         0   -0.0000    1.0081
```

(c) Mit $q = 1$ ist die berechnete Matrix $R^{(15)}$ zu den angegebenen dezimalen Stellen

1.3263	-0.1147	0.0042	0	0
0	1.2337	-0.1837	0.0067	0
0	0	1.1593	-0.1702	0.0035
0	0	0	1.1118	-0.0910
0	0	0	0	1.0510

(d) Mit $q = 0$ ist die berechnete Matrix $A^{(20)}$ zu den angegebenen dezimalen Stellen

1.3448	-0.0427	0	0	0
-0.0427	1.2679	-0.0652	0	0
0	-0.0652	1.1657	-0.0657	0
0	0	-0.0657	1.0837	-0.0366
0	0	0	-0.0366	1.0379

Kommentare: Sehen Sie diesen Code. Wie auf Seite 120 im Skriptum angedeutet ist die Konvergenz schneller, wenn die Quotienten zwischen den benachbarten Eigenwerten kleiner sind. Mit $q = 1$ für dieses Beispiel sind diese Verhältnisse möglichst vorteilhaft. Deswegen ist das (wahre) Ergebnis im Teil (b) näher zum Endergebnis als das (wahre) Ergebnis im Teil (d).

5. Für $m = 10$ sei $\boldsymbol{x} = \{x_i\}_{i=1}^m$ mit $x_i = (i - \frac{1}{2})/m$, $i = 1, \ldots, m$. Weiters sei $\boldsymbol{x}^k = \{x_i^k\}_{i=1}^m$ für $k \in \mathbb{N}_0$. Für $n = 5$ sei $V = [\boldsymbol{x}^0, \boldsymbol{x}, \ldots, \boldsymbol{x}^{n-1}]$. Gesucht wird eine vereinfachende Transformation $PVQ = (B; Z)$, wobei $P \in \mathbb{R}^{m\times m}$ und $Q \in \mathbb{R}^{n\times n}$ orthogonal sind, $Z = 0 \in \mathbb{R}^{(m-n)\times n}$ und $B \in \mathbb{R}^{n\times n}$ eine obere bidiagonale Matrix ist. Zu diesem Zweck schreiben Sie einen Matlab-Code, um den Algorithmus auf Seiten 130-131 im Skriptum zu implementieren. Kreuzen Sie bei den wahren Behauptungen an.

(a) Zu den angegebenen dezimalen Stellen gilt zum Schluss $B =$

-3.1623	2.1489	0	0	0
0	-1.6599	-0.5381	0	0
0	0	-0.3454	-0.0748	0
0	0	0	0.0526	-0.0074
0	0	0	0	-0.0049

(b) Zu den angegebenen dezimalen Stellen gilt zum Schluss $B =$

1.7783	-1.4659	0	0	0
0	1.2884	0.7336	0	0
0	0	0.5877	0.2736	0
0	0	0	-0.2293	0.0863
0	0	0	0	0.0703

(c) Zu den angegebenen dezimalen Stellen gilt zum Schluss $Q =$

1.0000	0	0	0	0
0	-0.7358	0.6212	-0.2638	-0.0559
0	-0.4893	-0.2299	0.7375	0.4047
0	-0.3661	-0.4881	0.0392	-0.7913
0	-0.2919	-0.5683	-0.6204	0.4548

(d) Zu den angegebenen dezimalen Stellen gilt zum Schluss $Q =$

```
1.0000         0         0         0         0
     0   -0.7358   -0.4893   -0.3661   -0.2919
     0    0.6212   -0.2299   -0.4881   -0.5683
     0   -0.2638    0.7375    0.0392   -0.6204
     0   -0.0559    0.4047   -0.7913    0.4548
```

Kommentare: Sehen Sie diesen Code mit der Funktion house. Mit Z = `zeros(5,5)` gilt `norm(P*A*Q-[B;Z],1) = 2.2044e-15` mit den (wahren) Ergebnisse aus Teilen (a) und (c), wobei P nicht verraten worden ist. Es gelten auch

`svd(B) = [3.9533,1.4319,0.3291,0.0517,0.0049] = svd(V)`

zu den angegebenen dezimalen Stellen. Die Werte im Teil (b) sind `sqrt` der Werte im Teil (a). Die Matrix in Teil (d) ist die Transponierte der Matrix in Teil (c).

6. Für $m = 7$ sei $\boldsymbol{x} = \{x_i\}_{i=1}^m$ mit $x_i = (i - \frac{1}{2})/m$, $i = 1, \ldots, m$. Weiters sei $\boldsymbol{x}^k = \{x_i^k\}_{i=1}^m$ für $k \in \mathbb{N}_0$. Für $n = 5$ sei $V = [\boldsymbol{x}^0, \boldsymbol{x}, \ldots, \boldsymbol{x}^{n-1}]$ die Vandermonde Matrix, die durch $PVQ = (B; Z)$ transformiert wird, wobei $P \in \mathbb{R}^{m \times m}$ und $Q \in \mathbb{R}^{n \times n}$ orthogonal sind, $Z = 0 \in \mathbb{R}^{(m-n) \times n}$, und B ist gegeben annährungsweise durch

```
-2.6458    1.7942         0         0         0
      0   -1.3761   -0.4435         0         0
      0         0   -0.2805   -0.0617         0
      0         0         0    0.0407   -0.0063
      0         0         0         0   -0.0034
```

Sei B durch diese Matrix gegeben. Schreiben Sie einen Matlab-Code, um den Algorithmus auf Seiten 136-137 im Skriptum zu implementieren, um die Singulärwert-Zerlegung von B zu bestimmen. Kreuzen Sie bei den wahren Behauptungen an.

(a) Zu den angegebenen dezimalen Stellen gilt $B^{(2)} =$

```
3.3029   -0.0373         0         0         0
     0    1.1875    0.0012         0         0
     0         0    0.2677   -0.0000         0
     0         0         0    0.0401    0.0000
     0         0         0         0    0.0034
```

(b) Zu den angegebenen dezimalen Stellen gilt $Q^{(3)} =$

```
-0.7713   -0.5989    0.2100    0.0473   -0.0074
 0.6354   -0.7052    0.3065    0.0698   -0.0109
 0.0365   -0.3794   -0.8982   -0.2162    0.0339
 0.0001   -0.0047   -0.2348    0.9598   -0.1539
-0.0000    0.0000    0.0008   -0.1582   -0.9874
```

(c) Zu den angegebenen dezimalen Stellen gilt $P^{(4)} =$

```
 0.9630   -0.2696   -0.0031    0.0000    0.0000
 0.2688    0.9590    0.0899   -0.0002   -0.0000
-0.0212   -0.0873    0.9953   -0.0357   -0.0000
-0.0007   -0.0030    0.0356    0.9993    0.0134
 0.0000    0.0000   -0.0005   -0.0134    0.9999
```

(d) Zu den angegebenen dezimalen Stellen gilt $B^{(5)} =$

3.3031	-0.0001	0	0	0
0	1.1874	0.0000	0	0
0	0	0.2677	-0.0000	0
0	0	0	0.0401	0.0000
0	0	0	0	0.0034

Kommentare: Sehen Sie diesen Code mit der Funktion house. Antworten (b) und (c) sind $P^{(8)}$ bzw. $Q^{(8)}$ mit `tol = 1/2^23` im Abbruchskriterium `if (norm(d) < tol*norm(g))`. Wenn B nicht abgerundet wird, wie in der Angabe gesehen, ergibt sich das Ergebnis

$$B^{(l)} \stackrel{l\to\infty}{\longrightarrow} \mathrm{diag}\{\texttt{3.3031}, \texttt{1.1875}, \texttt{0.2677}, \texttt{0.0401}, \texttt{0.0034}\} = \mathrm{diag}\{\texttt{svd}(V)\}.$$

7. Für $m = 10$ sei $\boldsymbol{x} = \{x_i\}_{i=1}^m$ mit $x_i = (i - \frac{1}{2})/m$, $i = 1, \ldots, m$. Weiters sei $\boldsymbol{x}^k = \{x_i^k\}_{i=1}^m$ für $k \in \mathbb{N}_0$. Für $n = 5$ sei $V = [\boldsymbol{x}^0, \boldsymbol{x}, \ldots, \boldsymbol{x}^{n-1}]$ mit Singulärwert-Zerlegung $V = U\Sigma W^\top$. Seien $\{c_{j-1}\}_{j=1}^n = \boldsymbol{c}$ die Koeffizienten des Polynoms $P(x; \boldsymbol{c}) = c_0 + c_1 x + \cdots c_{n-1}x^{n-1}$. Mit $\boldsymbol{c}^\star = \boldsymbol{1} \in \mathbb{R}^n$ erfüllt $\boldsymbol{y}^\star = \{P(x_i; \boldsymbol{c}^\star)\}_{i=1}^m$ das System $V\boldsymbol{c}^\star = \boldsymbol{y}^\star$ und so die Systeme

$$V^\top V \boldsymbol{c}^\star = V^\top \boldsymbol{y}^\star \in \mathbb{R}^n, \quad [I, Z]\,\Sigma\, W^\top \boldsymbol{c}^\star = [I, Z]\, U^\top \boldsymbol{y}^\star \in \mathbb{R}^n,$$

wobei $I = \mathrm{diag}\{1\} \in \mathbb{R}^{n\times n}$ und $Z = 0 \in \mathbb{R}^{n\times(m-n)}$. Für Störungen $\boldsymbol{\epsilon}^\ddagger, \boldsymbol{\epsilon}^\dagger \in \mathbb{R}^n$ seien die fehlerhaften Lösungen $\boldsymbol{c}^\ddagger, \boldsymbol{c}^\dagger \in \mathbb{R}^n$ definiert durch

$$V^\top V \boldsymbol{c}^\ddagger = V^\top \boldsymbol{y}^\star + \boldsymbol{\epsilon}^\ddagger, \quad [I, Z]\,\Sigma\, W^\top \boldsymbol{c}^\dagger = [I, Z]\, U^\top \boldsymbol{y}^\star + \boldsymbol{\epsilon}^\dagger.$$

Sei die Eigenraum-Zerlegung von $V^\top V$ gegeben durch

$$V^\top V \boldsymbol{e}_i = \lambda_i \boldsymbol{e}_i, \quad \|\boldsymbol{e}_i\|_2 = 1, \quad i = 1, \ldots, n, \quad 0 < \lambda_1 < \cdots < \lambda_n.$$

Kreuzen Sie bei den wahren Behauptungen an.

(a) Für $\boldsymbol{\epsilon}^\dagger = \boldsymbol{\epsilon}^\ddagger = \boldsymbol{e}_1/100$ gilt

$$\frac{\|\boldsymbol{c}^\star - \boldsymbol{c}^\dagger\|_2}{\|\boldsymbol{c}^\star\|_2} < \frac{\|\boldsymbol{c}^\star - \boldsymbol{c}^\ddagger\|_2}{\|\boldsymbol{c}^\star\|_2}$$

(b) Für $\boldsymbol{\epsilon}^\dagger = \boldsymbol{\epsilon}^\ddagger = \boldsymbol{e}_n/100$ gilt

$$\frac{\|\boldsymbol{c}^\star - \boldsymbol{c}^\dagger\|_2}{\|\boldsymbol{c}^\star\|_2} < \frac{\|\boldsymbol{c}^\star - \boldsymbol{c}^\ddagger\|_2}{\|\boldsymbol{c}^\star\|_2}$$

(c) Für $\boldsymbol{\epsilon}^\dagger = \boldsymbol{\epsilon}^\ddagger = \boldsymbol{e}_1/100$ gilt

$$\left[\frac{\|\boldsymbol{c}^\star - \boldsymbol{c}^\dagger\|_2}{\|\boldsymbol{c}^\star\|_2} \bigg/ \frac{\|\boldsymbol{e}^\dagger\|_2}{\|[I, Z]U^\top \boldsymbol{y}^\star\|_2}\right]^2 < \left[\frac{\|\boldsymbol{c}^\star - \boldsymbol{c}^\ddagger\|_2}{\|\boldsymbol{c}^\star\|_2} \bigg/ \frac{\|\boldsymbol{e}^\ddagger\|_2}{\|V^\top \boldsymbol{y}^\star\|_2}\right]$$

(d) Für $\boldsymbol{\epsilon}^\dagger = \boldsymbol{\epsilon}^\ddagger = \boldsymbol{e}_n/100$ gilt

$$\max_{x\in[0,1]} |P(x; \boldsymbol{c}^\star) - P(x; \boldsymbol{c}^\dagger)| < \max_{x\in[0,1]} |P(x; \boldsymbol{c}^\star) - P(x; \boldsymbol{c}^\ddagger)|.$$

Kommentare: Sehen Sie diesen Code. Zu 3 signifikanten Ziffern sind linken und rechten Seiten der Ungleichungen gegeben durch die Paaren

`(3.51e-01, 1.87e+02)`, `(1.91e-01, 2.86e-04)`,
`(1.91e-01, 2.86e-04)` bzw. `(9.44e-03, 1.28e-03)`.

Da $\boldsymbol{e}_1$ und $\boldsymbol{e}_n$ zu den betragsmäßig kleinsten bzw. größten Eigenwerten gehören, wird $(V^\top V)^{-1}\boldsymbol{e}_1$ viel größer als $(V^\top V)^{-1}\boldsymbol{e}_n$. Deswegen werden die Ungleichungen in (a) und (c) verletzt, weil der Ansatz mit Singulärwert-Zerlegung genauer ist. Die linke und rechte Seiten im Teil (c) entsprechen der Beziehung zwischen Konditionszahlen $\kappa_2(\hat{\Sigma})^2 = \kappa_2(V^\top V)$ auf Seite 127 im Skriptum.

8. Für $N \in \mathbb{N}$, $64 \leq N \leq 512$, wird ein periodisches Signal $\boldsymbol{u}^\star \in \mathbb{R}^{2N-1}$ mit hoher Auflösung grob gemessen, und das Ergebnis ist ein verschwommendes Signal $A\boldsymbol{u}^\star = \boldsymbol{v} \in \mathbb{R}^N$ mit niedrigerer Auflösung. Sei $A \in \mathbb{R}^{N\times(2N-1)}$ gegeben durch

```
t = 1./(1:N); t = [t(N:-1:2),t]; t = t/sum(t);
A = zeros(N,2*N-1);
for i=1:N
    A(i,:) = circshift(t,[0,N + 2*(i-1)]);
end
```

Mit $\boldsymbol{x} = \{x_i\}_{i=1}^{2N-1}$, $x_i = (i - \frac{1}{2})/(2N-1)$, seien $\boldsymbol{u}^\star = \cos(4\pi\boldsymbol{x})$ und $\boldsymbol{v} = A\boldsymbol{u}^\star$. Mit dem Signal $\boldsymbol{v}$ niedrigerer Auflösung wird das Signal höherer Auflösung durch Minimierung von

$$J(\boldsymbol{u}) = \|A\boldsymbol{u} - \boldsymbol{v}\|_2^2 + \mu\|Q\boldsymbol{u}\|_2^2, \quad \mu \geq 0, \quad Q = \begin{bmatrix} -1 & +1 & & 0 \\ & \ddots & \ddots & \\ 0 & & -1 & +1 \end{bmatrix} \in \mathbb{R}^{(2N-2)\times(2N-1)}$$

abgeschätzt. Kreuzen Sie bei den wahren Behauptungen an.

(a) Mit $\mu = 0$ ist $\boldsymbol{u} = A^\dagger \boldsymbol{v}$ eindeutig minimierend für J.
(b) Wenn $\boldsymbol{u}$ für J mit $\mu = 0$ minimierend ist, folgt $\|\boldsymbol{u} - \boldsymbol{u}^\star\|_2 > 0.1\|\boldsymbol{u}^\star\|_2$.
(c) Mit $\mu > 0$ ist $\boldsymbol{u} = (A^\top A + \mu Q^\top Q)^{-1}A^\top \boldsymbol{v}$ eindeutig minimierend für J.
(d) Wenn $\boldsymbol{u}$ für J mit $\mu = 2^{-23}$ minimierend ist, folgt $\|\boldsymbol{u} - \boldsymbol{u}^\star\|_2 < 0.001\|\boldsymbol{u}^\star\|_2$.

Kommentare: Sehen Sie diesen Code. Da die Signale periodisch sind, werden sie ersten und die letzten Komponenten der Vektoren als Nachbarn betrachtet. Die Wirkung der Matrix A ist, die Komponenten von $\boldsymbol{u}$ zu mischen, wobei die Gewichtung stärker ist, wenn der Abstand zwischen Indexen kleiner ist, als ob ein Signal mit niedrigerer Auflösung gemessen wird. Ohne das Vorwissen dass $\boldsymbol{v} = A\boldsymbol{u}^\star$ gilt, ist $\boldsymbol{u} = A^\dagger \boldsymbol{v}$ die vernünftige Lösung des Minimierungsproblems für den Fall $\mu = 0$. Sonst gilt natürlich $J(\boldsymbol{u}^\star) = 0$ mit $\mu = 0$. Wenn Q mit $\tilde{Q} = I$ ersetzt werden sollte, ist $\boldsymbol{u} = A^\dagger \boldsymbol{v}$ auch der Limus der Lösungen mit $\mu \to 0$. Jedoch entstehen unerwünschte Schwingungen in dieser Lösung, und deswegen wird Q verwendet, um diese Schwingungen effektiv zu dämpfen. Wenn diese Dämpfung durch Q verwendet wird, ist $\boldsymbol{u}^\star$ der Limus der Lösungen mit $\mu \to 0$. Die Lösungsformel im Teil (c) ergibt sich aus

$$\begin{aligned} J(\boldsymbol{u}) &= (A\boldsymbol{u} - \boldsymbol{v})^\top(A\boldsymbol{u} - \boldsymbol{v}) + \mu\boldsymbol{u}^\top Q^\top Q\boldsymbol{u} \\ &= \boldsymbol{u}^\top A^\top A\boldsymbol{u} - 2\boldsymbol{v}^\top A\boldsymbol{u} + \boldsymbol{v}^\top\boldsymbol{v} + \mu\boldsymbol{u}^\top Q^\top Q\boldsymbol{u} \end{aligned}$$

$$\nabla_{\boldsymbol{u}} J(\boldsymbol{u}) = 2A^\top A\boldsymbol{u} - 2A^\top \boldsymbol{v} + 2\mu Q^\top Q\boldsymbol{u}$$

Die Matrix $A^\top A + \mu Q^\top Q$ ist SPD, weil mit $\boldsymbol{u}^\top(A^\top A + \mu Q^\top Q)\boldsymbol{u} = 0$ oder $\|A\boldsymbol{u}\|_2 = \|Q\boldsymbol{u}\|_2 = 0$ folgen $\boldsymbol{u} = c\mathbf{1}$ aus $Q\boldsymbol{u} = 0$ und $c = 0$ aus $A\boldsymbol{u} = cA\mathbf{1} = c(\boldsymbol{t}^\top\mathbf{1})\mathbf{1} = 0$.