Abgabe Übungsblatt 1

Mats Hoffstadt, Phillip Schwarz

23. April 2025

Aufgabe 1

(i)

Beweis. Wir zeigen, dass die Operatornorm auch als Supremum dargestellt werden kann:

$$\|A\|_{op} = \sup_{\|x\|=1} \|Ax\| = \sup_{\|x\|\leq 1} \|Ax\|$$

Ursprüngliche Definition:

$$\|A\|_{op} := \inf\{C > 0 : \|Ax\| \leq C\|x\|\}$$

Schritt 1: Wir zeigen $\|A\|_{op} = \sup_{\|x\|=1} \|Ax\|$.

Für jedes C mit $\|Ax\| \leq C\|x\|$ für alle $x \neq 0$ gilt insbesondere für alle x mit $\|x\| = 1$, dass $\|Ax\| \leq C$. Daher muss $C \geq \sup_{\|x\|=1} \|Ax\|$ sein.

Das Infimum aller solchen C ist also mindestens $\sup_{\|x\|=1} \|Ax\|$, also $\|A\|_{op} \geq \sup_{\|x\|=1} \|Ax\|$.

Andererseits, sei $S := \sup_{\|x\|=1} \|Ax\|$. Dann gilt für beliebiges $x \neq 0$:

$$\begin{aligned}
\|Ax\| &= \left\| A\left(\frac{x}{\|x\|} \cdot \|x\|\right) \right\| \\
&= \|x\| \cdot \left\| A\left(\frac{x}{\|x\|}\right) \right\| \\
&\leq \|x\| \cdot S
\end{aligned}$$

Da dies für alle $x \neq 0$ gilt, ist S ein zulässiges C in der Definition von $\|A\|_{op}$, und somit $\|A\|_{op} \leq S = \sup_{\|x\|=1} \|Ax\|$.

Zusammen erhalten wir $\|A\|_{op} = \sup_{\|x\|=1} \|Ax\|$.

Schritt 2: Wir zeigen $\sup_{\|x\|=1} \|Ax\| = \sup_{\|x\|\leq 1} \|Ax\|$.

Es ist klar, dass $\sup_{\|x\|=1} \|Ax\| \leq \sup_{\|x\|\leq 1} \|Ax\|$, da die Menge $\{x : \|x\| = 1\}$ eine Teilmenge von $\{x : \|x\| \leq 1\}$ ist.

Für die andere Richtung: Sei y mit $\|y\| < 1$. Dann ist $z = \frac{y}{\|y\|}$ ein Vektor mit $\|z\| = 1$, und

$$\begin{aligned}
\|Ay\| &= \left\| A\left(\|y\| \cdot \frac{y}{\|y\|}\right) \right\| \\
&= \|y\| \cdot \|Az\| \\
&< \|Az\| \\
&\leq \sup_{\|x\|=1} \|Ax\|
\end{aligned}$$

Daher wird das Supremum über $\|x\| \leq 1$ immer auf dem Rand $\|x\| = 1$ angenommen, und somit $\sup_{\|x\|\leq 1} \|Ax\| = \sup_{\|x\|=1} \|Ax\|$. □

Aufgabe 2

(i)

Beweis. Wir sollen zeigen, dass die Folge

$$x_{n+1} = \frac{1}{2}\left(x_n + \frac{S}{x_n}\right)$$

für $S > 0$ und $x_0 > 0$ konvergiert. Nach dem Monotoniesatz genügt es zu zeigen, dass die Folge beschränkt und monoton ist.

Schritt 1: Nachweis der Beschränktheit.

Untere Schranke: Wir betrachten die Differenz zwischen x_{n+1} und $\sqrt{S}$:

$$\begin{aligned} x_{n+1} - \sqrt{S} &= \frac{1}{2}\left(x_n + \frac{S}{x_n}\right) - \sqrt{S} \\ &= \frac{x_n^2 + S}{2x_n} - \sqrt{S} \\ &= \frac{x_n^2 + S - 2x_n\sqrt{S}}{2x_n} \\ &= \frac{(x_n - \sqrt{S})^2}{2x_n} \end{aligned}$$

Da $x_n > 0$ und $(x_n - \sqrt{S})^2 \geq 0$ für alle reellen Zahlen x_n, folgt:

$$x_{n+1} - \sqrt{S} = \frac{(x_n - \sqrt{S})^2}{2x_n} \geq 0$$

Somit gilt für alle $n \geq 1$: $x_n \geq \sqrt{S}$. Die Folge ist also nach unten durch $\sqrt{S}$ beschränkt.

Obere Schranke: Wir untersuchen, wann $x_{n+1} \leq x_n$ gilt:

$$\begin{aligned} x_{n+1} \leq x_n &\iff \frac{1}{2}\left(x_n + \frac{S}{x_n}\right) \leq x_n \\ &\iff x_n + \frac{S}{x_n} \leq 2x_n \\ &\iff \frac{S}{x_n} \leq x_n \\ &\iff S \leq x_n^2 \\ &\iff \sqrt{S} \leq x_n \end{aligned}$$

Da wir bereits gezeigt haben, dass $x_n \geq \sqrt{S}$ für alle $n \geq 1$ gilt, folgt $x_{n+1} \leq x_n$ für alle $n \geq 1$.

Falls $x_0 < \sqrt{S}$, so kann $x_1 > x_0$ sein. In diesem Fall ist die Folge durch $\max(x_1, x_0)$ nach oben beschränkt. Falls $x_0 \geq \sqrt{S}$, so ist die Folge durch x_0 nach oben beschränkt.

Insgesamt ist die Folge also durch $\max(x_0, x_1, \sqrt{S})$ nach oben beschränkt.

Schritt 2: Nachweis der Monotonie.

Wir unterscheiden zwei Fälle:

Fall 1: $x_0 \geq \sqrt{S}$

Dann gilt $x_0^2 \geq S$, also $\frac{S}{x_0} \leq x_0$ und somit:

$$x_1 = \frac{1}{2}\left(x_0 + \frac{S}{x_0}\right) \leq \frac{1}{2}(x_0 + x_0) = x_0$$

Da $x_1 \geq \sqrt{S}$ (wie im Nachweis der Beschränktheit gezeigt), folgt induktiv $x_{n+1} \leq x_n$ für alle $n \geq 0$. Die Folge ist in diesem Fall monoton fallend.

Fall 2: $0 < x_0 < \sqrt{S}$

Dann gilt $x_0^2 < S$, also $\frac{S}{x_0} > x_0$ und somit:

$$x_1 = \frac{1}{2}\left(x_0 + \frac{S}{x_0}\right) > \frac{1}{2}(x_0 + x_0) = x_0$$

Wir zeigen nun, dass entweder $x_1 \geq \sqrt{S}$ gilt oder die Folge streng monoton steigend ist, bis ein Folgenglied $x_k \geq \sqrt{S}$ erreicht wird:

Angenommen, für ein $k \geq 0$ gilt $0 < x_k < \sqrt{S}$. Dann ist $x_k^2 < S$, also $\frac{S}{x_k} > x_k$ und somit:

$$x_{k+1} = \frac{1}{2}\left(x_k + \frac{S}{x_k}\right) > \frac{1}{2}(x_k + x_k) = x_k$$

Wir betrachten nun die Funktion $f(x) = \frac{1}{2}(x + \frac{S}{x})$ und ihre Ableitung:

$$f'(x) = \frac{1}{2}\left(1 - \frac{S}{x^2}\right)$$

Diese Ableitung ist negativ für $x < \sqrt{S}$ und positiv für $x > \sqrt{S}$. Das bedeutet, dass $f(x)$ streng monoton fallend für $x < \sqrt{S}$ und streng monoton steigend für $x > \sqrt{S}$ ist. Außerdem gilt $f(\sqrt{S}) = \sqrt{S}$.

Da $f(x) > x$ für $x < \sqrt{S}$ und die Folge durch $x_{n+1} = f(x_n)$ definiert ist, muss die Folge (x_n) streng monoton steigend sein, solange $x_n < \sqrt{S}$ gilt, und sie nähert sich $\sqrt{S}$ an.

Sobald ein Folgenglied $x_k \geq \sqrt{S}$ erreicht wird, gilt nach Fall 1, dass die Folge ab diesem Index monoton fallend ist.

Zusammenfassend lässt sich so folgern, dass die Folge

$$x_{n+1} = \frac{1}{2}(x_n + \frac{S}{x_n})$$

konvergiert. □

Aufgabe 3

(i)

In Julia können wir die drei Normen wie folgt implementieren:

```
# Implementierung der Operatornormen
function norm_1(A::Matrix)
    return maximum([sum(abs.(A[:, j])) for j in 1:size(A, 2)])
end

function norm_inf(A::Matrix)
    return maximum([sum(abs.(A[i, :])) for i in 1:size(A, 1)])
end

function norm_F(A::Matrix)
    return sqrt(sum(abs2.(A)))
end
```

(ii)

Die Matrix H mit den Einträgen $H_{ij} = \frac{1}{i+j-1}$ kann so implementiert werden:

```
# Implementierung der Matrix H
function create_H(n::Int)
    H = zeros(n, n)
    for i in 1:n
        for j in 1:n
            H[i, j] = 1 / (i + j - 1)
        end
    end
    return H
end
```

(iii)

Hier berechnen wir die Normen für verschiedene Werte von N und stellen die Ergebnisse tabellarisch dar:

```
using LinearAlgebra
using DataFrames

# Berechnung der Normen für verschiedene Dimensionen
N_values = [2, 3, 5, 8, 10, 15]
results = []

for N in N_values
    H = create_H(N)

    # Berechnung der Normen
    n1 = norm_1(H)
    n_inf = norm_inf(H)
    nF = norm_F(H)

    push!(results, (N, n1, n_inf, nF))
end

# Ausgabe der Ergebnisse als Tabelle
d = DataFrame(results, [:N, :norm_1, :norm_inf, :norm_F])
```

Ergebnisse und Interpretation

Die Ergebnisse zeigen folgende Werte für die verschiedenen Normen:

N	$\|H\|_1$	$\|H\|_\infty$	$\|H\|_F$
2	1.500000	1.500000	1.269296
3	1.833333	1.833333	1.413624
5	2.283333	2.283333	1.580906
8	2.717857	2.717857	1.722143
10	2.928968	2.928968	1.785527
15	3.318229	3.318229	1.895459

Tabelle 1: Berechnete Normen der Matrix H für verschiedene Dimensionen N

Beobachtungen:

1. Die 1-Norm und die ∞-Norm sind für diese spezielle Matrix H identisch.
2. Die Frobenius-Norm wächst langsamer als die anderen beiden Normen.
3. Die Normen wachsen logarithmisch mit der Dimension N.

Diese Ergebnisse zeigen, dass die Matrix H mit zunehmender Dimension zwar wächst, aber das Wachstum verlangsamt sich logarithmisch. Die Frobenius-Norm wächst deutlich langsamer als die anderen Normen.

Aufgabe 3

(i)

Wie definieren die Funktion `konv` zur Berechnung der Werte $x + 1$ in Abhängigkeit von Startwert x_0, S und Iterationszahl n.
Die Funktion `konv` ist wie folgt definiert:

```
function konv(x0, S, n)
    x = zeros(n+1)
    x[1] = x0

    for i in 1:n
        x[i+1] = 0.5 * (x[i] + S/x[i])
    end

    return x
end
```

(iii)

Werte für Iterationszahl $n = 10$

Iteration	Wert1	Wert2	Wert3	Wert4
0	0.1	0.5	1.0	5.0
1	2.55	0.75	0.75	2.55
2	1.3475	0.7083333333333334	0.7083333333333334	1.3475
3	0.8538050980392157	0.7071428571428572	0.7071428571428572	0.8538050980392157
4	0.7425860126293996	0.7071067811865476	0.7071067811865476	0.7425860126293996
5	0.7137110468787978	0.7071067811865475	0.7071067811865475	0.7137110468787978
6	0.7082898647584593	0.7071067811865475	0.7071067811865475	0.7082898647584593
7	0.7073354335196762	0.7071067811865475	0.7071067811865475	0.7073354335196762
8	0.7071513325157138	0.7071067811865475	0.7071067811865475	0.7071513325157138
9	0.7071145630901789	0.7071067811865475	0.7071067811865475	0.7071145630901789
10	0.7071081765758	0.7071067811865475	0.7071067811865475	0.7071081765758

Werte für Iterationszahl $n = 5$

Iteration	Wert1	Wert2	Wert3	Wert4
0	0.1	0.5	1.0	5.0
1	2.55	0.75	0.75	2.55
2	1.3475	0.7083333333333334	0.7083333333333334	1.3475
3	0.8538050980392157	0.7071428571428572	0.7071428571428572	0.8538050980392157
4	0.7425860126293996	0.7071067811865476	0.7071067811865476	0.7425860126293996
5	0.7137110468787978	0.7071067811865475	0.7071067811865475	0.7137110468787978

Werte für Iterationszahl $n = 3$

Iteration	Wert1	Wert2	Wert3	Wert4
0	0.1	0.5	1.0	5.0
1	2.55	0.75	0.75	2.55
2	1.3475	0.7083333333333334	0.7083333333333334	1.3475
3	0.8538050980392157	0.7071428571428572	0.7071428571428572	0.8538050980392157

Interpretation der Ergebnisse

Die Tabellen zeigen die Konvergenz der Iterationsfolge $x_{i+1} = \frac{1}{2}(x_i + \frac{S}{x_i})$ für verschiedene Startwerte x_0 und den Parameter $S = 0.5$. Folgende Beobachtungen lassen sich machen:

- Unabhängig vom Startwert konvergiert die Folge gegen den Wert 0.7071067811865475, was der Quadratwurzel von 0.5 entspricht.
- Die Konvergenzgeschwindigkeit hängt stark vom Startwert ab. Startwerte, die näher am Grenzwert liegen (wie $x_0 = 0.5$ und $x_0 = 1.0$), konvergieren deutlich schneller als Extremwerte (wie $x_0 = 0.1$ oder $x_0 = 5.0$).
- Interessanterweise zeigen die Startwerte $x_0 = 0.1$ und $x_0 = 5.0$ ein symmetrisches Konvergenzverhalten. Nach der ersten Iteration erreichen beide denselben Wert (2.55) und folgen danach identischen Konvergenzpfaden.
- Bei $n = 3$ Iterationen ist die Konvergenz für die extremen Startwerte noch nicht sehr weit fortgeschritten, während bei $n = 10$ Iterationen alle Startwerte eine gute Approximation der Quadratwurzel erreicht haben.
- Die Methode demonstriert das Heron-Verfahren, eine klassische Methode zur Approximation von Quadratwurzeln. Die Formel $x_{i+1} = \frac{1}{2}(x_i + \frac{S}{x_i})$ konvergiert quadratisch gegen $\sqrt{S}$.