Numerik

Ingenieurinformatik Teil 2, Sommersemester 2026

David Straub

Gliederung

1. Einführung in Matlab
2. Arbeiten mit Arrays
3. Funktionen und Kontrollstrukturen
4. **Analysis**
5. Lineare Algebra
6. Differentialgleichungen
7. Einführung in Simulink

Fahrplan

Letzte Einheit: Funktionen in Matlab darstellen und damit rechnen → Polynome: `polyval`, `roots`, `polyder`, `polyint`, `polyfit` → Function Handles und Anonymous Functions: `@(x) ...`, `fplot`

Heute: Numerische Integration → Aus Messdaten: `trapz` → Aus einer analytischen Funktion: `integral`

Numerische Integration

Wiederholung: `polyint`

Für **Polynome** gibt es eine exakte Stammfunktion:

```
p = [-80, 500, 0];          % F(x) = -80x² + 500x
P = polyint(p);             % P = [-26.67, 250, 0, 0]

E = polyval(P, 3) - polyval(P, 0)   % bestimmtes Integral von 0 bis 3
```

Aber: Was wenn die Funktion kein Polynom ist?

$$U(z) = 3.6 + 0.3\,\tanh(5z - 2.5) \qquad f(x) = e^{-x^2} \qquad I(t) = \text{Messdaten}$$

Für diese Fälle gibt es keine geschlossene Stammfunktion – man muss **numerisch** integrieren.

Ladung, Strom und State of Charge

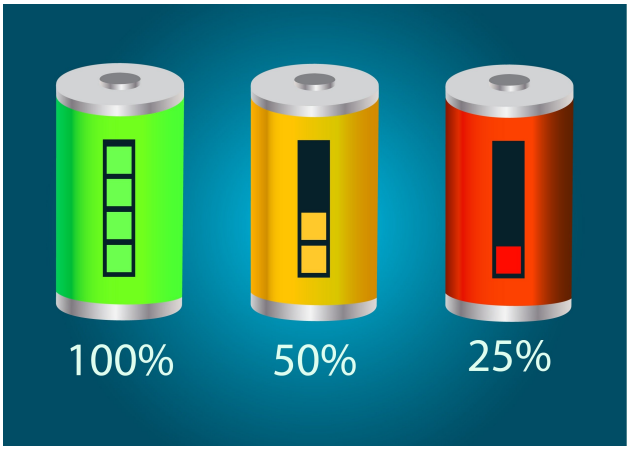
Der **State of Charge** (Ladezustand) z gibt an, wie voll die Batterie ist:

$$z = \frac{Q}{Q_{\max}} \in [0, 1]$$

Der **Strom** ist die zeitliche Änderung der **Ladung**:

$$I(t) = \frac{dQ}{dt} \qquad \Longrightarrow \qquad Q = \int_{t_0}^{t_1} I(t)\,dt$$

Folge: Um z aus Strommessungen zu bestimmen, muss man $I(t)$ integrieren – das ist die Grundlage jedes Batteriemanagementsystems.



Wie viel Ladung wurde gespeichert?

Eine Batterie wird geladen. Der Ladestrom wurde gemessen:

t [s]	0	10	20	30	40	50	60
I [A]	5.0	4.8	4.2	3.5	2.8	1.5	0.5

Frage: Wie viel Ladung $Q = \int_0^{60} I(t)\,dt$ wurde gespeichert?

1. Überlegen, Diskutieren und Berechnen Sie das Ergebnis
2. Wie kann das in Matlab gelöst werden? (Ohne Schleife!)

Rechteckregel

<!– Grafik: Balken/Rechtecke unter der I(t)-Kurve, Höhe = I_i, Breite = Δt -→

$$Q \approx \sum_i I_i \cdot \Delta t \qquad \Delta t = 10\,\text{s}$$

$$Q \approx (5.0 + 4.8 + 4.2 + 3.5 + 2.8 + 1.5) \cdot 10 = 218\,\text{As}$$

Einfach – aber der letzte Messpunkt wird ignoriert und die Kurvenform geht verloren.

Trapezregel

<!– Grafik: gleiche Kurve, aber Trapeze statt Rechtecke – Fläche zwischen Geraden -→

$$Q \approx \sum_{i=1}^{n-1} \frac{I_i + I_{i+1}}{2} \cdot \Delta t$$

$$Q \approx \frac{5.0 + 4.8}{2} \cdot 10 + \frac{4.8 + 4.2}{2} \cdot 10 + \cdots + \frac{1.5 + 0.5}{2} \cdot 10 = 195\,\text{As}$$

Die Gerade zwischen zwei Punkten ist eine bessere Näherung als ein Rechteck.

Grafischer Vergleich

```
t = 0:10:60;
I = [5.0, 4.8, 4.2, 3.5, 2.8, 1.5, 0.5];
Qr = sum(I(1:end-1)*10);
Qt = trapz(t, I);

hold on; grid on;
[xs, ys] = stairs(t, I);
fill([xs; flipud(xs)], [ys; zeros(size(ys))], 'b', 'FaceAlpha', 0.3);
plot(t, I, 'ro');
text(50, 4, sprintf('Qr = %.1f As', Qr));

fill([t, fliplr(t)], [I, zeros(size(I))], 'g', 'FaceAlpha', 0.3);
plot(t, I, 'ro-');
text(50, 3.6, sprintf('Qt = %.1f As', Qt));
```

Was wenn Δt nicht konstant ist?

Messdaten kommen selten mit gleichmäßigem Raster. Die Trapezregel verallgemeinert sich direkt – jedes Intervall hat sein eigenes Δt_i:

$$Q \approx \sum_{i=1}^{n-1} \frac{I_i + I_{i+1}}{2} \cdot (t_{i+1} - t_i)$$

`trapz(t, I)` macht genau das – unabhängig vom Raster.

`trapz` – Diskrete Daten integrieren

```
t = [0, 10, 20, 30, 40, 50, 60];
I = [5.0, 4.8, 4.2, 3.5, 2.8, 1.5, 0.5];

Q = trapz(t, I)      % → 195 As
Q_Ah = Q / 3600      % → Ah
```

- Ungleichmäßige Abstände in `t` sind kein Problem
- `trapz(y)` ohne `t` nimmt $\Delta t = 1$ an

`cumtrapz` – Ladung als Funktion der Zeit

```
Q_kum = cumtrapz(t, I);

plot(t, Q_kum)
xlabel('Zeit [s]')
ylabel('Ladung [As]')
```

`cumtrapz` gibt einen Vektor zurück – die kumulierte Ladung zu jedem Messzeitpunkt:

- `Q_kum(1) = 0` (Startwert)
- `Q_kum(end) = 195` (Gesamtladung)

Anwendung: State-of-Charge-Tracking in Batteriemanagementsystemen.

Übung: `trapz`

Messung des Entladestroms einer Batterie (Kapazität 10 Ah, voll geladen):

```
t = [0,  600, 1200, 1800, 2400, 3000, 3600];  % [s]
I = [8.0, 7.5,  6.8,  5.9,  4.5,  2.8,  0.5]; % [A]
```

1. Berechnen Sie die entnommene Ladung in As und Ah.
2. Wie viel Prozent der Batterie wurden entladen?
3. Plotten Sie den kumulierten Ladungsverlauf mit `cumtrapz`.

Denkaufgabe: Adaptives Sampling 🤔

Ein Sensor speichert einen Messpunkt nur, wenn sich der Strom um mehr als 100 mA ändert. Eine Batterie (10 Ah) wird entladen, der Laststrom ändert sich abrupt bei $t = 30\,\text{min}$:

```
t = [0,   1800,  3600];  % [s]
I = [2.0,  8.0,   8.0];  % [A]
```

`trapz(t, I)` liefert $Q = 23400\,\text{As} = 6.5\,\text{Ah}$.

1. Was war der tatsächliche Stromverlauf zwischen $t = 0$ und $t = 1800\,\text{s}$?
2. Wie viel Ladung wurde wirklich entnommen?
3. Welche Methode wäre hier treffender, und warum?

Funktionen integrieren: `integral`

Von diskreten Daten zur analytischen Funktion

Bei **Messdaten** sind die Stützstellen durch die Messung vorgegeben. - Mehr Genauigkeit → mehr Messpunkte nötig, aber die hat man oft nicht - Lineare Interpolation (Trapez) ist die sicherste Annahme

Bei einer **analytischen Funktion** kann man die Funktion an beliebigen Stellen auswerten. - `integral` wählt Stützstellen selbst, schätzt den Fehler ab, verfeinert wo nötig - Je höher der Polynomgrad der lokalen Näherung, desto genauer – bei gleicher Anzahl Auswertungen

> Man rechnet immer noch numerisch – der Unterschied ist: **wer die Stützstellen wählt**.

Rechteck → Trapez → Simpson: Visueller Vergleich

Historische Anwendung: Kepler'sche Fassregel

$$V = \frac{h}{6}\left(A_1 + 4A_m + A_2\right)$$

Simpson-Regel für die Integration von Querschnittsflächen $A(z)$ über die Höhe h eines Fasses.

Leerlaufspannung

Die **Leerlaufspannung** (engl. *Open Circuit Voltage*, OCV) ist die Spannung einer Batterie ohne Stromfluss – sie hängt vom Ladezustand $z \in [0, 1]$ ab.

Elektrochemische Modelle liefern $U(z)$ als Formel, z.B.:

$$U(z) = 3.6 + 0.3 \tanh(5z - 2.5)$$

```
fplot(@(z) 3.6 + 0.3*tanh(5*z - 2.5), [0, 1])
xlabel('Ladezustand z')
ylabel('Spannung U [V]')
```

Frage: Wie viel Energie $E = Q \int_0^1 U(z)\,dz$ ist in einer Batterie mit $Q = 3\,\text{Ah}$ gespeichert?

`integral` – Funktion integrieren

```
U = @(z) 3.6 + 0.3 * tanh(5 * z - 2.5);

Q = 3;                          % Kapazität [Ah]
E = Q * integral(U, 0, 1)       % Energie [Wh]
```

`integral(f, a, b)` berechnet $\int_a^b f(z)\,dz$ – adaptiv und hochgenau.

`f` muss ein Function Handle sein und **vektorisiert** arbeiten (`.^`, `.*`).

`trapz` oder `integral`?

	`trapz`	`integral`
Eingabe	Messdaten (Vektoren)	Function Handle
Stützstellen	durch Messung vorgegeben	automatisch, adaptiv
Genauigkeit	abhängig von Abtastrate	hoch
Typischer Einsatz	Sensordaten, Messreihen	physikalische Modelle

Übung: `integral`

Leerlaufspannungs-Kurve einer LFP-Batterie (Lithium-Eisenphosphat, $Q = 10\,\text{Ah}$):

$$U(z) = 3.3 + 0.2\,\tanh(5z - 1.5)$$

1. Plotten Sie $U(z)$ mit `fplot`.
2. Wie viel Energie ist noch gespeichert, wenn die Batterie auf 20% entladen ist? ($E = Q\int_{0.2}^{1} U(z)\,dz$)
3. Berechnen Sie den Energieinhalt der voll geladenen Batterie in Wh.