

# Übung 3: Frequenzbereich, Schaltungssimulator und Differentialquotient



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

Einführung in die numerische Berechnung elektromagnetischer Felder

Carsten Wesp, Boian Balouchev,  
Sebastian Schöps

Wintersemester 2025/26  
6. November 2025  
Übungsblatt 3

---

## Aufgabe 3.1: Frequenzbereich

---

In der Elektrotechnik werden oft Probleme behandelt, die sinus-förmig schwingen. Um die Rechnung zu vereinfachen, lassen sich solche Probleme im Frequenzbereich betrachten. Betrachten Sie zunächst die skalare lineare gewöhnliche Differentialgleichung im Zeitbereich

$$x'(t) = f(t, x) := -kx(t) + r(t) \quad (1)$$

mit Startwert  $x(t_0) = x_0$ , wobei  $k > 0$  sein soll. Die Anregung sei durch  $r(t) = \cos(\omega t)$  gegeben, wobei  $\omega$  als Kreisfrequenz bezeichnet wird.

**a)** Berechnen Sie die Lösung der Differentialgleichung im Zeitbereich mithilfe der im Skript angegebenen Formel

$$x(t) = e^{-k(t-t_0)} \left( x_0 + \int_{t_0}^t e^{k(s-t_0)} r(s) \, ds \right).$$

**b)** Berechnen Sie nun die Lösung der Differentialgleichung im Frequenzbereich, indem Sie die Lösung als Kosinus unbekannter Amplitude ansetzen

$$x_{\text{freq}}(t) = \operatorname{Re} \{ \underline{x} e^{j\omega t} \}, \quad (2)$$

wobei  $\underline{x}$  als Phasor bezeichnet wird. Berechnen Sie die Ableitung  $x'_{\text{freq}}(t)$  und setzen Sie alles in die Differentialgleichung (1) ein. Ersetzen Sie auch die Anregung durch den Realteil einer komplexen Größe. Aus der entstandenen Gleichung lässt sich der komplexe Phasor  $\underline{x}$  bestimmen. Nach Bestimmung des Phasors lässt sich aus (2) wiederum die Lösung als zeitabhängige Funktion berechnen.

**c)** Was wurde im Ansatz aus **b)** vernachlässigt im Vergleich zu dem Ansatz aus **a)**? Was könnte der Vorteil vom Ansatz **b)** sein?

**d)** Laden Sie sich die LTSpice-Datei **Beispielschaltung2.asc** über Moodle herunter. Starten Sie die Simulation mit den voreingestellten Parametern. Um was für einen Typ Simulation handelt es sich? Was lässt sich aus den Resultaten ablesen?

Die Schaltung lässt sich durch die Differentialgleichung

$$\frac{d}{dt} i(t) = -\frac{R}{L} i(t) + \frac{1}{L} u_i(t) \quad (3)$$

beschreiben, wobei  $R$  der Widerstand,  $L$  die Induktivität,  $i(t)$  der Strom und  $u_i(t) = \cos(\omega t)$  die anregende Spannung ist.

1. Stellen Sie die Lösung der Differentialgleichung im Zeitbereich dar, indem Sie das Resultat aus **a)** verwenden.
2. Stellen Sie die Lösung nun im Frequenzbereich dar, indem Sie die Differentialgleichung analog zu **b)** als Realteil einer komplexen Gleichung schreiben und nach dem komplexen Phasor des Stroms auflösen.
3. Berechnen Sie den Betrag des Phasors der Spannung, die über dem Widerstand abfällt in Abhängigkeit der Frequenz  $f = \frac{\omega}{2\pi}$ .
4. Plotten Sie Ihr Ergebnis im selben Frequenzbereich, den die LTSpice Simulation auch verwendet hat. Verwenden Sie auch dieselbe Art Diagramm. Was fällt am Frequenzgang auf?

---

### Aufgabe 3.2: Modified Nodal Analysis (MNA)

---

Schreiben Sie einen Schaltungssimulator in Python, der einfache Schaltungen aus den Bauteilen Widerstand (R), Spule (L) und Kondensator (C), sowie konstanten Strom- ( $i$ ) und Spannungsquellen ( $v$ ) mittels modifizierter Knotenanalyse simulieren kann.

**a)** Schreiben Sie eine Python-Routine, die Spice-Netzlisten einlesen kann und daraus alle Matrizen für die modifizierte Knotenanalyse aufstellt, d.h.

- i) die reduzierten Inzidenzmatrizen, d.h. ohne den Masseknoten 0,  $\mathbf{A}_x \in \mathbb{R}^{(n-1) \times b_x}$  für jeden Elementtyp (d.h. Zweig enthält ein Bauteil vom Typ  $x$ )

$$a_{ij} = \begin{cases} +1 & \text{falls Zweig } j \text{ vom Knoten } i \text{ weg führt} \\ -1 & \text{falls Zweig } j \text{ zum Knoten } i \text{ hin führt} \\ 0 & \text{sonst} \end{cases}$$

wobei  $i = 1, \dots, n-1$  und  $j = 1, \dots, b_x$  und  $x \in \{R, L, C, i, v\}$ , dabei ist  $n$  die Anzahl aller Knoten und  $b_x$  die Anzahl der Zweige mit Bauelementen vom Typ  $x$ ; z.B. für einen Kondensator zwischen Knoten 3 und Masse:  $\mathbf{A}_C = [0, 0, 1]^\top \in \mathbb{R}^{3 \times 1}$ ,

- ii) die Diagonalmatrizen  $\mathbf{C} \in \mathbb{R}^{b_C \times b_C}$ ,  $\mathbf{L} \in \mathbb{R}^{b_L \times b_L}$ ,  $\mathbf{G} = \mathbf{R}^{-1} \in \mathbb{R}^{b_R \times b_R}$ , die die numerischen Werte (Induktivitäten in Henry, Kapazitäten in Farad, Leitfähigkeiten in Siemens) enthalten und
- iii) die Spaltenvektoren der Quellen ( $\mathbf{i}_s \in \mathbb{R}^{b_i}$  und  $\mathbf{v}_s \in \mathbb{R}^{b_v}$ ) die konstante Ströme (Ampère) und Spannungen (Volt) enthalten.

#### Tipps:

- Benutzen Sie die Datei `circuit_matrices_vorlage.py`.
- Benutzen Sie zum Einlesen die Funktion `parse_netlist` der zur Verfügung gestellten Datei `parse_spice.py`. Die Netzliste kann über View/SPICE Netlist in LTSpice angezeigt werden.
- Mit `parse_netlist` können nur Schaltungen mit linearen Bauteilen eingelesen werden. Vorsicht: Dioden sind nichtlineare Bauelemente.

**b)** Benutzen Sie die erstellte Routine, um die benötigten Matrizen für folgende Schaltung zu generieren:

Da das System nur Widerstände, Kondensator und Stromquelle enthält, ist es ausreichend, das System mittels der Kirchhoffschen Knotenregel für Ströme zu lösen. Konkret lautet diese:

$$\mathbf{A}_C \mathbf{C} \mathbf{A}_C^\top \dot{\varphi} + \mathbf{A}_R \mathbf{G} \mathbf{A}_R^\top \varphi = -\mathbf{A}_I \mathbf{i}_s$$

Stellen Sie die benötigten Matrizen auf und lösen Sie das Gleichungssystem zunächst für den stationären Fall (d.h.  $\dot{\varphi} = 0$ ) mit  $I = 1 \text{ A}$ . Wie lauten die Ströme und Spannungen an den Widerständen? Gleichen Sie diese mit Ihren Erwartungen ab.

Führen Sie das Vorgehen nun im Frequenzbereich mit dem Ansatz  $\varphi = \Re \{ \underline{\varphi} e^{j\omega t} \}$  sowie  $I = \Re \{ \underline{I} e^{j\omega t} \}$  mit  $\underline{I} = 1 \text{ A}$  und  $\omega = 100\pi/\text{s}$  aus. Stellen Sie die Ströme der Widerstände und des Kondensators graphisch gegenüber.

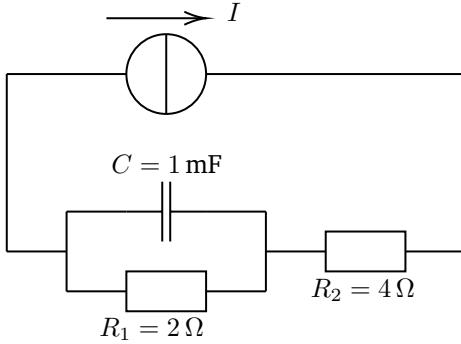


Abbildung 1: Einfache Kondensatorschaltung.

### Aufgabe 3.3: Differenzenquotient

In der Vorlesung wurde die erste Ableitung  $\frac{df}{dx}(x)$  einer (hinreichend glatten) skalaren Funktion  $f : \mathbb{R} \rightarrow \mathbb{R}$  an der Stelle  $x$  durch einen der Differenzenquotienten

$$\begin{aligned}\frac{df}{dx}(x) &= \underbrace{\frac{f(x+h) - f(x)}{h}}_{=: \Delta_1 f(x, h)} + \mathcal{O}(h) \\ \frac{df}{dx}(x) &= \underbrace{\frac{f(x+h) - f(x-h)}{2h}}_{=: \Delta_2 f(x, h)} + \mathcal{O}(h^2)\end{aligned}$$

mit einer Schrittweite  $h \neq 0$  numerisch approximiert.

- a) Welche Konvergenzordnungen  $n_i$  ( $i = 1, 2$ ) haben die obigen Differenzenquotienten? Rechnen Sie das jeweils mittels einer Taylorentwicklung nach.
- b) Konstruieren Sie einen Differenzenquotienten  $\Delta_4 f(x, h)$  vierter Ordnung zur Approximation der ersten Ableitung. Führen Sie dafür die Taylorapproximationen von  $f$  an den Stellen  $f(x+h)$ ,  $f(x-h)$ ,  $f(x+2h)$  und  $f(x-2h)$  bis zur Ordnung  $\mathcal{O}(h^5)$  aus und lösen Sie nach  $f'(x)$  auf.
- c) Gegeben sei eine Polynomfunktion

$$f(x) = Cx^p \quad (4)$$

wobei  $C$  eine Konstante und  $p$  die Polynomordnung sind.

Welche Form hat die Kurve, wenn  $f$  in einem Plot mit doppellogarithmischem Koordinatensystem dargestellt wird?

In einem doppellogarithmischen Koordinatensystem entspricht die sichtbare Steigung von  $f$ , exemplarisch dargestellt in Abb. 2, nicht mehr der Ableitung  $f'(x)$ . Berechnen Sie den Wert der graphischen Steigung  $\frac{\Delta y_{\text{graphisch}}}{\Delta x_{\text{graphisch}}}$  von  $f$  in einem Plot mit doppellogarithmisch skalierten Achsen. Da es sich um ein doppelt logarithmisches Koordinatensystem handelt, sind die Größen  $\Delta x_{\text{graphisch}}$  und  $\Delta y_{\text{graphisch}}$  nicht einfach durch die Differenz der auf den Achsen angegebenen Werte gegeben. Was beschreibt die Steigung stattdessen? Wie lässt sich das Ergebnis im Kontext von Konvergenzanalysen verwenden? Was ändert sich an der Steigung, wenn das logarithmische Koordinatensystem nicht die Basis 10 sondern eine beliebige andere Basis hat?

- d) Schreiben Sie für  $\Delta_i f(x, h)$  ( $i = 1, 2, 4$ ) eine Python-Routine, die einen Konvergenzplot ( $x$ -Achse: Schrittweite  $h$  und  $y$ -Achse: absoluter Fehler) erzeugt (die analytischen Ableitungen der Funktionen, die für die Berechnung des absoluten Fehlers benötigt werden, können in der Routine vorgeben werden). Testen Sie Ihre Routine mit

$$f_1(x) = \exp(x+1) \quad (5)$$

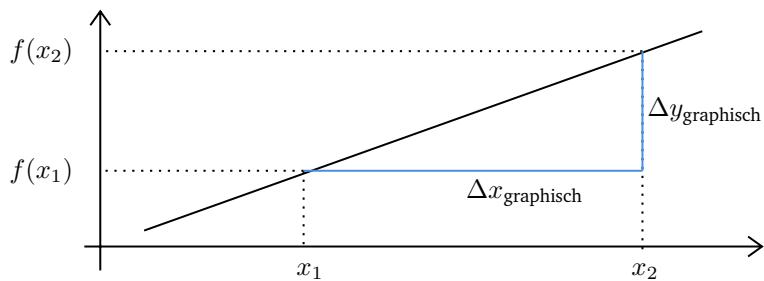


Abbildung 2: Steigungsdreieck in einem doppeltlogarithmischen Plot.

an der Stelle  $x = 1$ ,

$$f_2(x) = \begin{cases} 6x^2 + 4 & x > 1 \\ 4x^3 + 6 & x \leq 1 \end{cases} \quad (6)$$

an der Stelle  $x = 1$  und

$$f_3(x) = \cos(x) \quad (7)$$

an  $x = \pi/2$ .

Wählen Sie  $h \in [10^{-10}, 1]$ . Verwenden Sie zur Darstellung doppelt logarithmische Plots (`pyplot.loglog` aus dem Paket `matplotlib`). Entsprechen die Ergebnisse den Erwartungen? Falls nein, warum?

Ausarbeitung der Aufgaben 3.1, 3.2 und 3.3 und Abgabe der Übung via Moodle bis spätestens 12. November 2025 um 23:59 Uhr.