Grundlagen der Modellbildung und Simulation

Peter Bastian
Institut für Parallele und Verteilte Systeme
Universität Stuttgart
Universitätsstraße 38
D-70569 Stuttgart
email: Peter.Bastian@ipvs.uni-stuttgart.de

Version vom 15. Juli 2008

Inhaltsverzeichnis

1 Einführung **7**
- 1.1 Motivation: Signalleitung in Neuronen . . . 7
- 1.2 Grundlegende Begriffe . . . 10

2 Ereignisgesteuerte Simulation **15**
- 2.1 Was ist ereignisgesteuerte Simulation? . . . 15
- 2.2 Formalisierter Ansatz zur ereignisgesteuerten Simulation . . . 17
- 2.3 Praktische Aspekte . . . 21

3 Zelluläre Automaten **23**
- 3.1 Definition eines zellulären Automaten . . . 23
- 3.2 Verkehrssimulation mit zellulären Automaten . . . 27
- 3.3 Strömungssimulation mit zellulären Automaten . . . 28

4 Modellierung elektrischer Bauelemente **33**
- 4.1 Strom und Spannung . . . 33
- 4.2 Ideale Netzwerkelemente . . . 35
- 4.3 Kirchhoffsche Gesetze . . . 37

5 Netzwerkanalyse mit dem Knotenpotentialverfahren **41**
- 5.1 Analyse zweier einfacher Netzwerke . . . 41
- 5.2 Das Knotenpotentialverfahren . . . 43
- 5.3 Weiterführende Aspekte . . . 47

6 Komplexe Wechselstromrechnung **51**
- 6.1 Widerstandsnetzwerke . . . 51
- 6.2 Analyse des Parallelschwingkreises . . . 52
- 6.3 Komplexe Wechselstromrechnung . . . 58

7 Direkte Lösung vollbesetzter linearer Gleichungssysteme **63**
- 7.1 Gauß-Elimination und LU-Zerlegung . . . 63
- 7.2 Performance, ijk-Formen . . . 66

8 Direkte Lösung dünnbesetzter linearer Gleichungssysteme **71**
- 8.1 Was ist das Problem? . . . 71
- 8.2 Anordnungsstrategien zur Fill-In-Minimierung . . . 72
- 8.3 Cholesky-Zerlegung . . . 73
- 8.4 Direkte Lösung dünnbesetzter, symmetrisch positiv definiter Systeme . . . 74

9 Abstiegsverfahren **79**
- 9.1 Der Charakterisierungssatz . . . 79
- 9.2 Line Search . . . 80

9.3 Abstiegsverfahren in algorithmischer Form 81
9.4 Wahl der Suchrichtungen 81
9.5 Verfahren der konjugierten Gradienten 83

10 Einführung in gewöhnliche Differentialgleichungen 89
10.1 Beispiele gewöhnlicher Differentialgleichungen 89
10.2 Charakterisierung von Differentialgleichungen 92
10.3 Zur Theorie gewöhnlicher Differentialgleichungen 93

11 Einschrittverfahren 99
11.1 Eulersches Polygonzugverfahren (explizites Eulerverfahren) 99
11.2 Taylor- und Runge-Kutta-Verfahren 103
11.3 Konvergenz von Einschrittverfahren 106

12 Schrittweitensteuerung für Einschrittverfahren 109
12.1 Ein anderer Zugang zur Konvergenz 109
12.2 Bestimmung der Schrittweite 111
12.3 Schätzung der τ_n^m 112
12.4 Adaptiver Algorithmus 113

13 Stabilität und steife Probleme 115
13.1 Lineare Stabilitätsanalyse 115
13.2 Steife Probleme 118
13.3 Verfahren zur Lösung steifer Probleme 120

14 Modellierung mit partiellen Differentialgleichungen 123
14.1 Begriffe aus der Vektoranalysis 123
14.2 Modellierung des Wärmetransports 124
14.3 Weitere Anwendungen 127

15 Typeinteilung partieller Differentialgleichungen 129
15.1 Allgemeine Definition 129
15.2 Elementare partielle Differentialgleichungen 129
15.3 Sachgemäß gestellte Probleme 132
15.4 Typeinteilung 132

16 Finite Differenzen für die Poissongleichung 135
16.1 Differenzenformeln 135
16.2 Finite Differenzen in einer Raumdimension 136
16.3 Der n-dimensionale Fall 137

Literatur 141

Vorwort

Mathematische Modellbildung und numerische Simulation sind heute ein wichtiger Baustein vom reinen Erkenntnisgewinn in den Natur- und Ingenierwissenschaften bis hin zur Produktion in der Industrie. In dieser Vorlesung werden wir zunächst eine Klassifikation der verschiedenen Modelle kennenlernen und uns dann intensiv vor allem mit kontinuierlichen Modellen und deren numerischer Simulation beschäftigen. Als Beispiel werden wir die elektrische Netzwerkanalyse heranziehen, da dort unterschiedlichste mathematische Methoden von der Graphentheorie über lineare Gleichungssysteme bis hin zu differentiell-agebraischen Systemen zum Einsatz kommen.

Dieses Skript basiert auf einer Ausarbeitung aus dem Sommersemester 2007. Für die Erfassung des Textes in LaTeX danke ich Herrn Simon Zilliken recht herzlich. Alle verbleibenden Fehler gehen natürlich auf mein Konto.

Stuttgart, im April 2008 *Peter Bastian*

1 Einführung

1.1 Motivation: Signalleitung in Neuronen

Zur Motivation ein Beispiel aus der aktuellen Forschung. Um zu einem besseren Verständnis der Vorgänge im Gehirn zu gelangen versucht man die wesentlichen im Gehirn ablaufenden Vorgänge (welches sind das?) auf der Basis möglichst guter experimenteller Daten zu modellieren und zu simulieren. Im Gegensatz zu den in der Informatik bekannten neuronalen Netzen geht es hier wirklich darum die in einem echten Gehirn vorkommenden Objekte möglichst gut zu erfassen.

Für unsere Zwecke betrachten wir drei Ebenen, die unterschiedlichen räumlichen Skalen entsprechen.

Schematischer Aufbau des Gehirns

- *Netzwerkebene*
  - Das Gehirn besteht aus Nervenzellen, den *Neuronen*.
  - Ein Neuron gibt Signale über *Synapsen* an andere Neuronen weiter.
- *Zellebene*

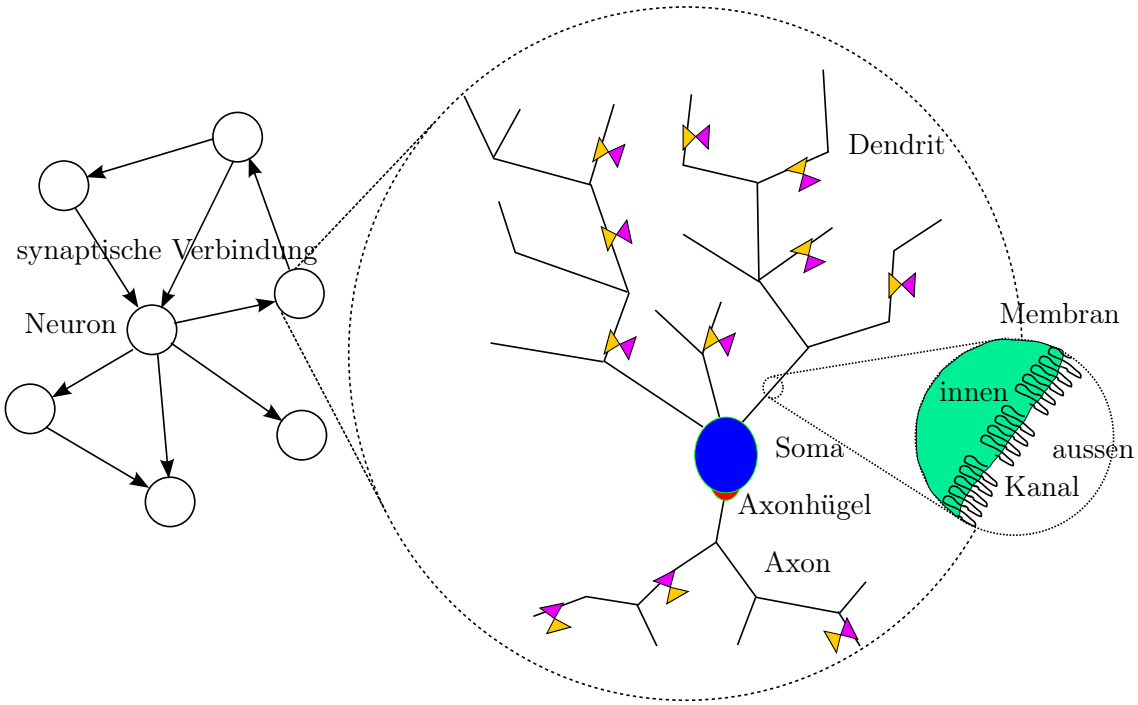


Abbildung 1: Schematischer Aufbau des Gehirns.

  - Ein Neuron besteht aus mehreren *Dendriten*, *Soma* und *Axon*.
  - *Die* Dendriten sind baumartig verzweigt. Über die Synapsen erfolgt eine *elektrische* Eingabe in Form eines Stromes.
  - Eine Potentialänderung breitet sich in Richtung Soma aus.
  - Das Soma ist der *Zellkern*. Übersteigt das Potential am *Axonhügel* eine Schwelle so wird ein *Aktionspotential* (AP) ausgelöst.
  - *Das* Axon ist baumartig verzweigt. Das AP ist eine Potentialänderung entlang des Axons.
  - Synapsen übertragen Signale auf chemischem Weg.
  - Dendriten und Axone sind lokal lange, dünne Zylinder.

- *Subzellebene*
  - Die Zelle ist durch eine Membran (Makromoleküle) begrenzt.
  - Kanäle in der Membran sind Schleusen für bestimme Stoffe.

Für jede der drei Ebenen entwickeln wir nun eine Modellvorstellung. Wir beginnen mit der Subzellebene.

Subzellebene

- Zur Beschreibung des Aktionspotentiales stellen wir uns eine kleine Scheibe des zylinderförmigen Axons vor.
- Aktionspotentiale gibt es normalerweise nur auf dem Axon.
- Zustandsgrößen sind das Potential $v(t)$, sowie die Konzenrationen von Natriumionen $c_{Na}(t)$, Kaliumionen $c_K(t)$ sowie eines Gemisches von Ionen L: $c_L(t)$.
- Zustandsgrößen hängen von der Zeit, aber nicht vom Ort innerhalb der Scheibe ab.
- Für ein Modell, das die zeitliche Entwicklung der Zustandsgrößen beschreibt haben A. L. Hodgkin und A. F. Huxley den Nobelpreis für Medizin 1952 erhalten (s. u.).

Hodgkin-Huxley Modell

- Das Potential und die Konzentrationen gehorchen dem System vier gewöhnlicher Differentialgleichungen für alle $t \in (a, b)$:

$$C_M \frac{\partial v}{\partial t} = I_{ext}(v,t) - I_{Na}(v, c_{Na}, c_L) - I_K(v, c_K) - I_L(v), \tag{1.1}$$

$$\frac{\partial c_k}{\partial t} = \alpha_k(v)(1 - c_k) - \beta_k(v)c_k \quad k \in \{Na, K, L\} \tag{1.2}$$

  (plus Anfangsbedingungen).

- I_{Na}, I_K und I_L beschreiben die Ströme durch drei verschiedene Sorten von Kanälen:

$$I_{Na}(v, c_{Na}, c_L) = g_{Na} c_{Na}^3 c_L (v - v_{Na}), \tag{1.3}$$
$$I_K(v, c_K) = g_K c_K^4 (v - v_K), \tag{1.4}$$
$$I_L(v) = g_L (v - v_L). \tag{1.5}$$

  Die g_i und die v_i sind Konstanten.

- $I_{ext}(v, t)$ beschreibt die externe Erregung (z. B. Synapsen).

- Schließlich die fehlenden Reaktionsparameter:

$$\alpha_{Na}(v) = 0.1 \frac{-v + 25}{e^{(-v+25)/10} - 1} \qquad \beta_{Na}(v) = 4e^{-v/18}, \tag{1.6}$$
$$\alpha_K(v) = 0.01 \frac{-v + 10}{e^{(-v+10)/10} - 1} \qquad \beta_K(v) = 0.125e^{-v/80}, \tag{1.7}$$
$$\alpha_L(v) = 0.07e^{-v/20} \qquad \beta_L(v) = \frac{1}{e^{(-v+30)/10} + 1}. \tag{1.8}$$

Auf der Zellebene (Einzelneuron) betrachten wir nun die Potentialausbreitung entlang des Neurons.

- Dendrit und Axon werden als räumlich eindimensional abstrahiert da Durchmesser (μm) sehr klein gegenüber Länge (mm).

- Zustandsgrößen, z. B. $v(x, t)$, sind nun Funktionen der Position $x \in \Omega$ und der Zeit $t \in (a, b)$.

- Das Potential gehorcht der partiellen Differentialgleichung:

$$C_m \frac{\partial v}{\partial t} = \frac{\partial}{\partial x} \left(D_m \frac{\partial v}{\partial x} \right) - I_{ext}(v, t) - I_L(v) \quad \text{in } \Omega \times (a, b) \tag{1.9}$$

  mit den Anfangsbedingungen

$$v(x, 0) = v^0(x) \tag{1.10}$$

  und den Randbedingungen

$$v(x, t) = v^{\mathrm{D}}(x, t), x \in \Gamma_{\mathrm{D}}, \qquad - \left(D_m \frac{\partial v}{\partial x} \right) \cdot \nu = J(x, t), x \in \Gamma_{\mathrm{N}}. \tag{1.11}$$

- An jedem Punkt $x \in \Omega$ zusätzlich noch der Reaktionsteil aus dem HH-Modell.

Schließlich erreichen wir die Netzwerkebene.

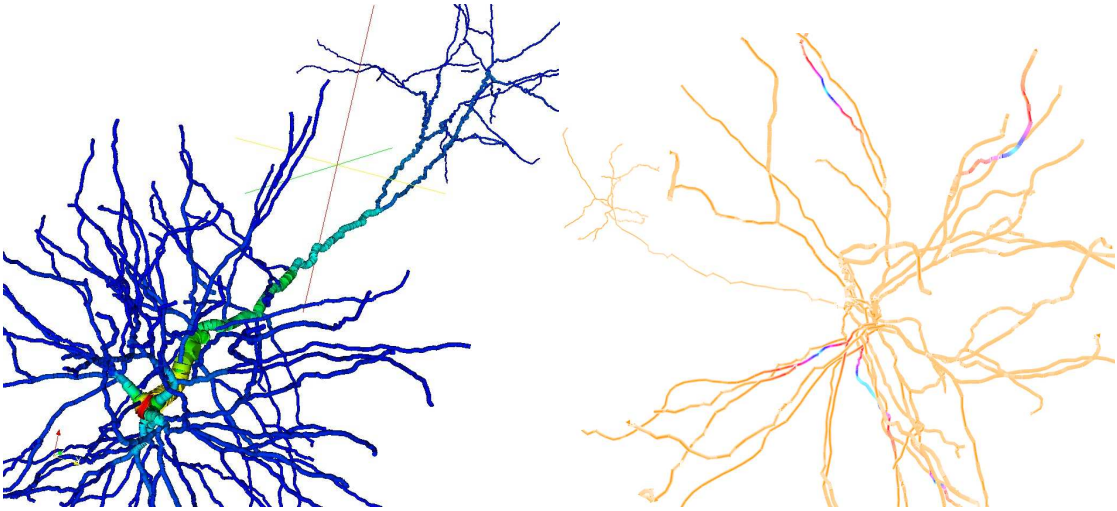


Abbildung 2: Simulation der Potentialverteilung auf einer realistischen Geometrie (Simulation von Stefan Lang, IPVS, Stuttgart, Daten von Christian de Kock, MPImF, Heidelberg).

Netzwerkmodell

- Aktionspotential auf dem Axon erreicht eine Synapse.
- Signalübertragung in der Synapse erfolgt auf chemischem Wege und verschiedene Verläufe $I_{ext}(t)$ stehen zur Verfügung.
- Auch für die Synapse selbst existieren detaillierte Modelle.
- Die Signalleitung auf dem Axon kann durch eine Verzögerung modelliert werden.
- Das Gesamtmodell ist eine hybride Simulation aus ereignisgesteuerter diskreter Simulation und kontinuierlicher Simulation.

1.2 Grundlegende Begriffe

In dieser Vorlesung geht es um

- *Modelle* zur Beschreibung von (realen) Systemen und die
- *Simulation* (Ausführung) dieser Modelle auf einem Computer.

Dabei werden wir uns auf eine relativ kleine Menge von Modellen beschränken in der Annahme, dass sich die dort erlernten Methoden auch auf andere Bereiche übertragen lassen.

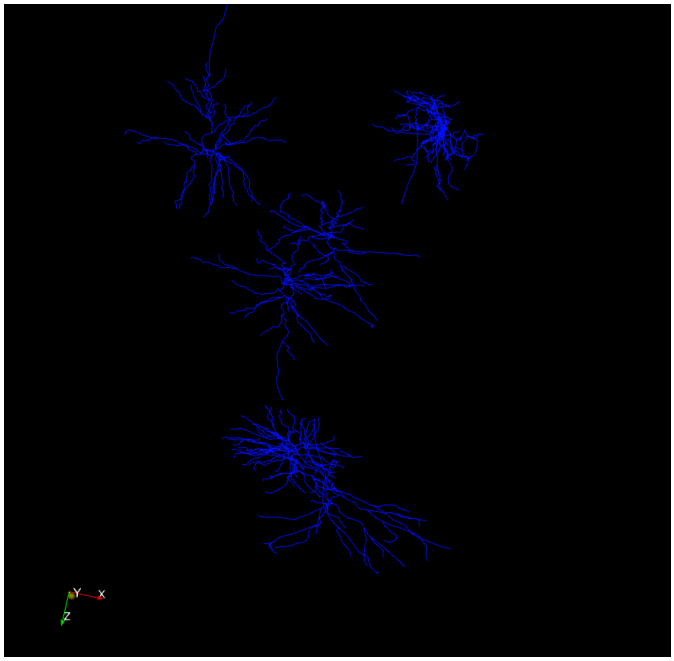


Abbildung 3: Netzwerk bestehend aus 5 Zellen (Simulation von Stefan Lang, IPVS, Stuttgart, Daten MPImF, Heidelberg).

System Unter einem System versteht man allgemein Komponenten, die über Signale interagieren. Dabei unterscheidet man zwischen *offenen* und *geschlossenen Systemen*. *Zustandsgrößen* beschreiben eindeutig den Zustand der Komponenten. Abbildung 4 erläutert den Unterschied zwischem offenem und geschlossenem System.

Modell Ein Modell ist eine Repräsentation eines Systems zum Zweck der Erfassung des Systems. Das Modell ist eine Vereinfachung des Systems, welche aber zur Erfüllung des Zwecks ausreicht. Modelle können hierarchisch sein, da auch das Modell wieder ein System darstellt (Mehrskaligkeit, s.u.)

Mathematisches Modell Das Modell wird formalisiert mit den Methoden der Mathematik, z.B.

1. Prädikatenlogik
2. Regelsysteme (Automaten, formale Sprachen, Petri-Netze, ...)
3. Gleichungen (algebraische Gleichungen, Differentialgleichungen)

In der Physik und den Ingenieurwissenschaften ist dieses Vorgehen natürlich, denn hier werden Sachverhalte bereits mathematisch beschrieben. In anderen Disziplinen, wie Biologie oder Medizin, ist dies schwieriger.

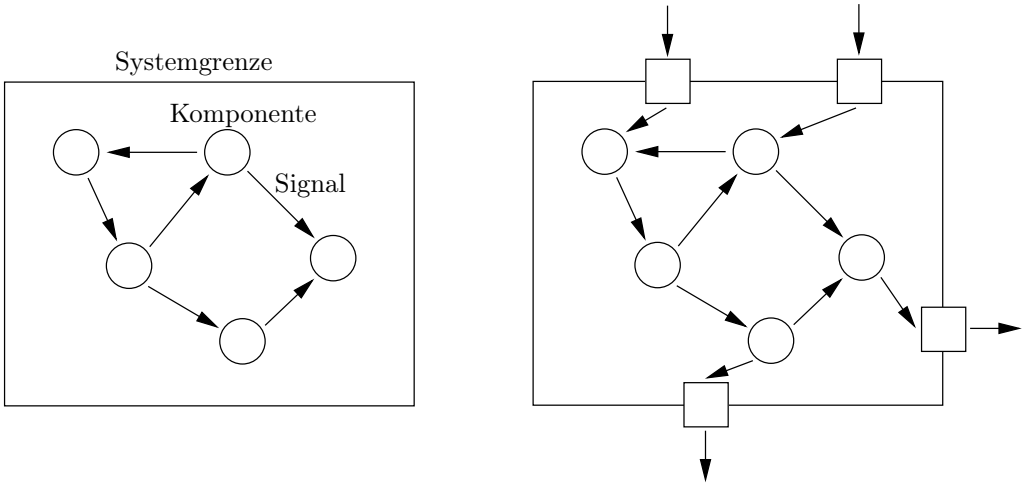
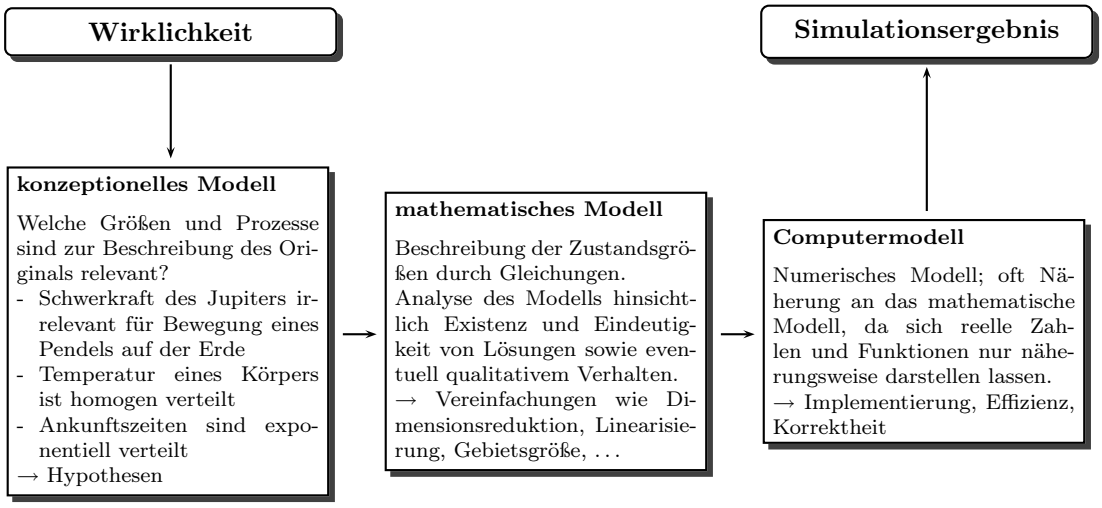


Abbildung 4: Im **geschlossenen System** (links) ist das Systemverhalten nur durch seine Komponenten bestimmt. Das **offene System** (rechts) interagiert mit seiner Umwelt. Zur Beschreibung des Verhaltens ist Zusatzinformation an der Systemgrenze erforderlich.

Allgemeines Vorgehen in der Modellbildung und Simulation



Nach erfolgter Modellbildung und Simulation sind die Simulationsergebnisse mit der Wirklichkeit zu vergleichen.

Entsprechen die Abweichungen nicht den Erwartungen (das Maß für einen akzeptablen Fehler hängt sehr vom Problembereich ab) beginnt die Suche nach dem Grund für die Abweichung. Hierfür sind dann die einzelnen Teilschritte zu hinterfragen. Insbesondere unterscheidet man zwischen

Modellfehler Nicht erfasste physikalische Prozesse, Annahmen über Homogenität, Ver-

teilungen, Form von Abhängigkeiten, etc.

Datenfehler Ungenügend genau erfasste Daten an der Systemgrenze (Rand- und Anfangsbedingungen) oder Parameter.

Simulationsfehler (auch Approximationsfehler) Fehler die durch Approximation des Modells zum Zwecke der numerischen Lösung entstehen.

Modellbildung und Simulation erfordern eine intensive interdisziplinäre Kooperation der beteiligten Fachgebiete. Derzeit findet ein Übergang zur Modellierung einzelner Prozesse auf einer Skala hin zur Modellierung gekoppelter Prozesse auf verschiedenen Skalen statt.

Forderungen an eine mathematische Modellierung nach (Kra97)

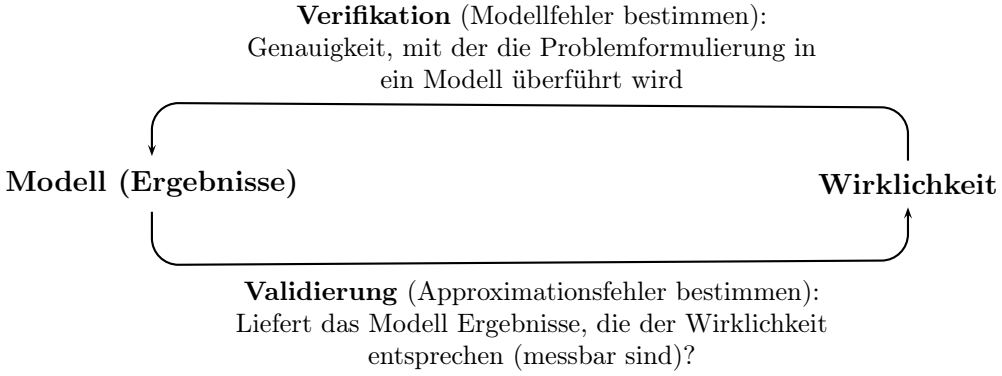
1. Hypothesen zur Aufstellung des Modells sollten einsichtig und möglichst überprüfbar sein.

2. Umsetzung der Problemstellung in die Sprache der Mathematik sollte überzeugend und einsichtig sein
   Beispiel: Bevölkerungszuwachs $= c \cdot$ Bevölkerung — ist das wirklich linear?

3. Lösungsmethode (Computermodell) sollte möglichst eng an das mathematische Modell angepasst sein (kleine Rundungs- und Diskretisierungsfehler, ...)

Wie überprüft man ein Modell?

Das folgende Diagramm soll noch einmal den Unterschied zwischen Modellfehler und Approximationsfehler deutlich machen.



Eigenschaften von mathematischen Modellen

Mathematische Modell können in verschiedene Klassen unterteilt werden.

Dynamisches Modell Zustandsgrößen und Signale hängen von der Zeit ab. Gegenteil: statisches Modell

Deterministisches Modell Zu jedem Zeitpunkt sind die Zustandsgrößen und Signale eindeutig bestimmt. Gegenteil: stochastisches Modell; Größen können zufällige Werte annehmen

Diskretes Modell Werte der Zustandsgrößen sind nur für diskrete Zeitpunkte, etwa $t \in \mathbb{N}$, erklärt. Zusätzlich kann auch die Wertemenge diskret (abzählbar) sein. Gegenteil: kontinuierliches Modell; Werte sind für $t \in [a, b] \subset \mathbb{R}$ erklärt.

In dieser Vorlesung werden hauptsächlich dynamische, deterministische, kontinuierliche Modelle behandelt. Beginnen werden wir jedoch mit dynamischen, diskreten Modellen (deterministisch und stochastisch).

Hierarchische Modelle (Mehrskaligkeit)

Viele Systeme können auf unterschiedlichen *Skalen* in Raum und Zeit betrachtet werden. Dies haben wir schon am Beispiel der Neuronennetzwerke gesehen.

Als ein weiteres Beispiel betrachten wir den Aufbau von Computern:

1. Aufbau aus Baugruppen wie Prozessor, Speicher, ALU, Bussen, Cache, ...
   Modellierung mittels *Hardware Description Language*.

2. Aufbau der Baugruppen aus logischen Grundelementen (UND, ODER, ...), die durch Leitungen verknüpft sind. Spannung auf einer Leitung kann nur die beiden Zustände $\{low, high\}$ annehmen.
   Modellierung durch *ereignisgesteuerte Simulation*.

3. Gatter können durch ein Netzwerk von Widerständen, Spulen, Kondensatoren und Transistoren beschrieben werden. Spannungen und Ströme auf bzw. durch Leitungen sind kontinuierliche Funktionen der Zeit: $I_1(t), U_1(t), \ldots$

4. Tranistoren sind durch Schichten unterschiedlich dotierter Halbleitermaterialien aufgebaut. Potential und Ladungsträgerdichten sind kontinuierliche Funktionen in Raum und Zeit: $\phi(x, y, z, t)$

5. Demnächst werden Strukturen auf dem Chip so klein sein, dass quantenmechanische Effekte eine Rolle spielen werden. Der Transport von Ladungsträgern wird dann unter gewissen Annahmen mit dem Landauer-Formalismus beschrieben.

Welche Skala die Richtige ist, hängt von der Anwendung ab. Makroskopische Systeme können nur mit enormem Aufwand auf der Mikroskala simuliert werden. In *Mehrskalensimulationen* koppelt man Modelle auf verschiedenen Skalen.
Eine Trennung in verschiedene Skalen ist nicht immer möglich (kontinuierliche Skalen, z.B. turbulente Strömungen).

2 Ereignisgesteuerte Simulation

Literatur: (BC84), (Bal96)

2.1 Was ist ereignisgesteuerte Simulation?

Aus der systemtheoretischen Sicht besteht ein System aus *interagierenden Entitäten*. Sein Zustand wird durch eine Menge von *Zustandsvariablen* beschrieben.

Systeme kann man klassifizieren in:

Kontinuierliche Systeme Zustandsvariablen hängen kontinuierlich von der Zeit ab.

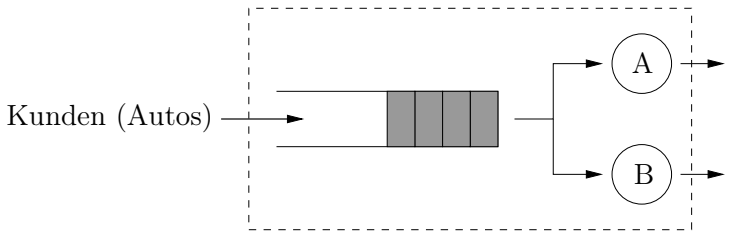
Diskretes System Zustandsvariablen ändern sich nur zu diskreten Zeitpunkten, die prinzipiell bekannt sind (→ *synchrone Simulation*). Üblicherweise ändern sich viele Zustandsvariablen gleichzeitig.

Ereignisgesteuertes System (*engl. Discrete Event Simulation*) Wenige Zustandsvariablen ändern sich zu diskreten, meist zufälligen Zeitpunkten (→ *asynchrone Simulation*). Die ereignisgesteuerte Simulation ist ein Spezialfall der diskreten Simulation.

Anwendungen

- Verkehrssysteme (Ampeln, Kreuzungen, Autobahnsysteme, ...)
- Produktionssysteme (Fertigungslinien, Lagerhaltung, workflow in Krankenhäusern, ...)
- Warteschlangensysteme (Bedieneinheiten, Warteschlangen, Ankunftsrate, ...)
- Chipsimulation auf Logikebene
- Kampfhandlungen
- Kommunikationssysteme (Telefonnetz, ...)

Beispiel 2.1 (Able-Baker Car Hop System, aus (BC84))**.** Beispiel für ein Warteschlangensystem. In einem Drive-In-Restaurant arbeiten die zwei Bedienungen Able und Baker. Able arbeitet etwas schneller als Baker.



Ankunftsrate der Autos		Bedienzeit von Able		Bedienzeit von Baker	
min	**p**	**min**	**p**	**min**	**p**
1	0.25	2	0.30	3	0.35
2	0.40	3	0.28	4	0.25
3	0.20	4	0.25	5	0.20
4	0.15	5	0.17	6	0.20

Ablauf:

- Auto kommt an:
  1. Able ist frei → Job an Able
  2. Baker ist frei → Job an Baker
  3. keiner frei → Warteschlange (unendlich lang)
- Bedienung wird frei: Auftrag aus Warteschlange annehmen, falls sie nicht leer ist

Ereignisse:
- Auto kommt an
- Bedienung beendet Job

Fragen:
- Wieviel Prozent der Arbeitszeit ist Able/Baker beschäftigt?
- Wieviele Autos warten maximal/im Mittel?
- Wie hoch ist die mittlere Wartezeit eines Kunden?
- Lohnt sich eine zusätzliche Bedienung?

Theoretische Betrachtungen zu diesem Beispiel. Ankunftszeit der Autos und die Bedienzeiten sind Zufallsvariablen X_C, X_A und X_B für die sich der Erwartungswert bestimmen lässt:

$$\mathbb{E}[X_C] = 1 \cdot 0.25 + 2 \cdot 0.4 + 3 \cdot 0.2 + 4 \cdot 0.15 = 2.25.$$

Analog bestimmt man

$$\mathbb{E}[X_A] = 3.29, \qquad \mathbb{E}[X_B] = 4.25.$$

Dies bedeutet, dass im Mittel alle 2.25 Minuten ein Auto ankommt und dass Able bzw. Baker im Mittel 3.29 bzw. 4.25 Minuten brauchen um ein Auto zu bedienen (Einheit [Minuten/Auto]

Able und Baker arbeiten aber gleichzeitig. Wieviele Minuten pro Auto brauchen Sie *gemeinsam* im Mittel?

Hierzu betrachte

$$\frac{1}{\mathbb{E}[X_A]} : \text{Autos die Able pro Minute bearbeitet [Auto/Minute], analog}$$
$$\frac{1}{\mathbb{E}[X_B]} : \text{Autos die Baker pro Minute bearbeitet [Auto/Minute].}$$

Zusammen bearbeiten Sie also $\mathbb{E}[X_A]^{-1} + \mathbb{E}[X_B]^{-1}$ Autos pro Minute, was dann eben

$$\frac{1}{\frac{1}{\mathbb{E}[X_A]} + \frac{1}{\mathbb{E}[X_B]}} = \frac{1}{0.30395 + 0.23529} = 1.85444$$

Minuten pro Auto im Mittel von Able und Baker zusammen benötigt werden.

Da nur alle 2.25 Minuten ein Auto ankommt steht genügend Arbeitskapazität zur Verfügung.

Ein Lauf der Simulation liefert nach 1000000 Minuten folgende Zahlen:

2.24985	Minuten mittlere Ankunftszeit,
0.854847	Minuten mittlere Wartezeit pro Auto,
4.52503	Minuten mittlere Verweilzeit pro Auto,
88.1345%	der Zeit ist Able beschäftigt,
74.9959%	der Zeit ist Baker beschäftigt.

□

Es gibt auch analytsische Methoden zur Analyse von Warteschlangenproblemen (Markov-Ketten): Die Lösung mit stochastischen Methoden erlaubt mehr Einblick, beispielsweise in Abhängigkeiten von Parametern wie der Ankunftsrate. Sie erfordert aber einschränkende Annahmen, wie zum Beispiel über die Verteilung der Ankunftsrate. Analytische Methoden sind nicht Gegenstand dieser Vorlesung.

2.2 Formalisierter Ansatz zur ereignisgesteuerten Simulation

Wir wollen nun den den Ablauf einer ereignisgesteuerten Simulation im Sinne eines Automaten formal beschreiben.

Eine ereignisgesteuerte Simulation wird charakterisiert durch

- Das System besteht aus einer endlichen Menge von Entitäten $V = \{v_1, \ldots, v_n\}$ die miteinander interagieren.
- Der Zustand jedes $v_i \in V$ wird durch ein Element der Menge Z_i beschrieben. Der Gesamtzustand des Systems ist somit

  $$z = (z_1, \ldots, z_n) \in Z = Z_1 \times \ldots \times Z_n.$$

- Ein Element der Signalmenge $S = \{s_1, \ldots, s_m\}$ kennzeichnet, welche Aktivität durch ein Ereignis ausgelöst wird. Die Signale seien total geordnet, d. h. es gibt eine Relation $<$ auf S.

  Eine Aktivität dauert ein gewisses Zeitintervall, dessen Länge zu Beginn der Aktivität bekannt ist. Das bedeutet, zu Beginn der Aktivität (a, b) — zur Zeit a — ist $b - a$ bekannt. Bei einer stochastischen Aktivität wird die Dauer zum Zeitpunkt a zufällig ermittelt.
- Die Menge der mögliche Ereignisse ist $E \subseteq \mathbb{N}_0 \times S$. $(t, s) \in E$ bedeutet, dass das Signal s zur Zeit t ausgelöst wird. Im Signal sind auch die beteiligten Entitäten kodiert.
- Zustandsübergangsfunktion $F : E \times Z \to Z$.
- Ereignisfunktion zum auslösen neuer Ereignisse $N : E \times Z \to \mathcal{P}(E)$.

- Anfangsmenge von Ereignissen $Q^0 \subset E$.
- Anfangszustand $z^0 \in Z$.

Algorithmisch läuft dann eine ereignisgesteuerte Simulation folgendermaßen ab:
$t = 0$
$Q = Q^0$
$z = z^0$
while $(Q \neq \emptyset \wedge t \leq T_{end})$ **do**
Bestimme $e = (t_{\min}, s_{\min})$ so dass $\forall (e', s') \in Q : t' > t_{\min} \vee (t' = t_{\min} \wedge s' \geq s_{\min})$
$Q = Q \setminus \{e\} \cup N(e, z)$
$z = F(e, z)$
$t = t_{\min}$
reale Simulation: speichere Daten für Statistik
end while

Abbildung 5 verdeutlicht diesen Ablauf graphisch. Die Implementierung der Ereigniswarteschlange erfolgt in der Praxis mit einer „priority queue“.

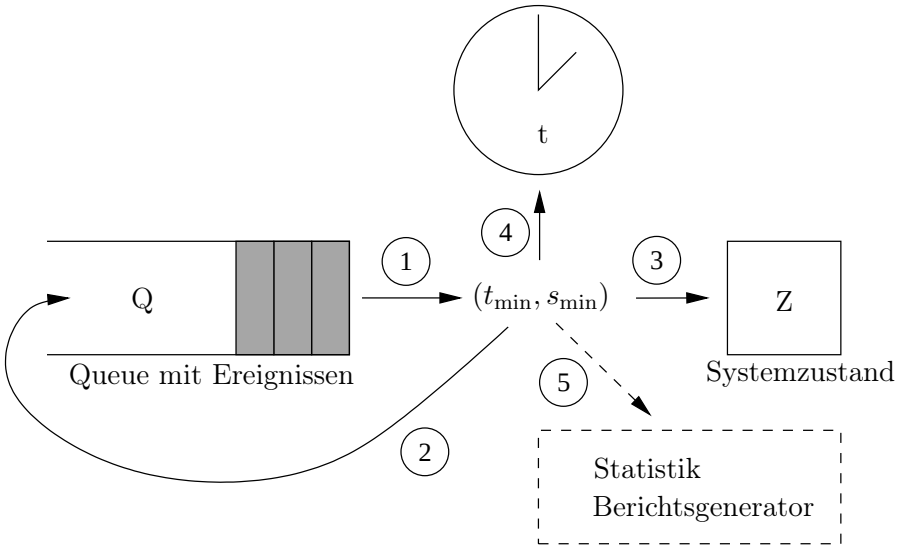


Abbildung 5: ereignisgesteuerte Simulation

Beispiel 2.2 (Fortsetzung von Beispiel 2.1)**.** Das Warteschlangensystem aus Beispiel 2.1 könnten wir folgendermaßen beschreiben:
Entitäten: $V = \{v_A, v_B, v_C\}$.
Zustandsmenge: $Z_A = Z_B = \{0, 1\}, Z_C = \mathbb{N}_0$.
Signalmenge $S = \{s_A, s_B, s_C\}$ wobei s_A bedeutet, dass Able seinen Job beendet, s_B bedeutet, dass Baker seinen Job beendet und s_C bedeutet, dass ein Auto ankommt. Die Signale sind folgendermaßen geordnet: $s_A < s_B < s_C$ (ist eigentlich egal).

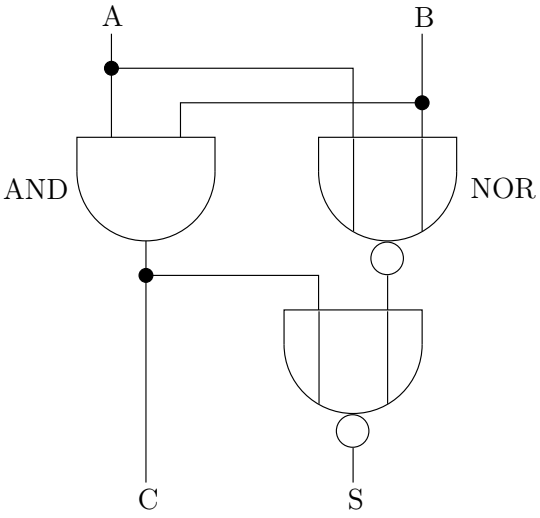
Zustandsübergänge und Folgeereignisse werden durch folgende Tabelle beschrieben:

E	F	N
(t, s_C)	$z_A = 0 \longrightarrow z_A = 1$	$z_A = 0 \longrightarrow \{(t + X_C, s_C), (t + X_A, s_A)\}$
	$z_B = 0 \longrightarrow z_B = 1$	$z_B = 0 \longrightarrow \{(t + X_C, s_C), (t + X_B, s_B)\}$
	sonst $\longrightarrow z_C = z_C + 1$	sonst $\longrightarrow \{(t + X_C, s_C)\}$
(t, s_A)	$z_C > 0 \longrightarrow z_C = z_C - 1$	$z_C > 0 \longrightarrow \{(t + X_A, s_A)\}$
	$z_C = 0 \longrightarrow z_A = 0$	$z_C = 0 \longrightarrow \emptyset$
(t, s_B)	$z_C > 0 \longrightarrow z_C = z_C - 1$	$z_C > 0 \longrightarrow \{(t + X_B, s_B)\}$
	$z_C = 0 \longrightarrow z_B = 0$	$z_C = 0 \longrightarrow \emptyset$

Hier sind X_A, X_B und X_C Zufallsvariablen für die Bedienzeiten und Ankunftszeiten. □

Als ein weiteres, informatikrelevanteres Beispiel für die ereignisgesteuerte Simulation betrachten wir die Simulation von Schaltwerken.

Beispiel 2.3 (Logiksimulation)**.** Es sollen beliebige Schaltwerke, bestehend aus evtl. rückgekoppelten logischen Grundelementen (AND, OR, NOR, ...), simuliert werden. Jedes Grundelement besitzt eine Verzögerungszeit δ, die vergeht, bis sich eine Änderung



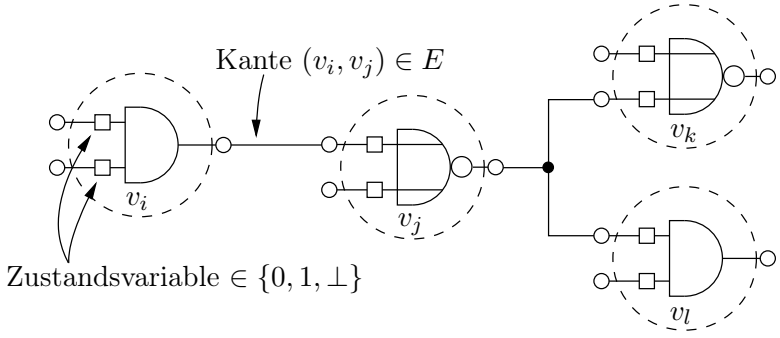
A	**B**	**S**	**C**
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Abbildung 6: Halbaddierer-Schaltnetz mit Wahrheitstabelle

am Eingang am Ausgang bemerkbar macht. Zur Vereinfachung kann man annehmen, dass alle Gatter die selbe Verzögerungszeit besitzen. In der Praxis ist man am Worst-Case-Verhalten der ganzen Schaltung — dem *kritischen Pfad* — interessiert.
Die logischen Grundelemente werden durch Wahrheitstabellen modelliert. Wie wird ihr Zustand gespeichert? → Zwei Möglichkeiten:

1. Jedes Gatter wird durch einen Knoten repräsentiert, in dem der Zustand gekapselt ist.

   Entitäten $V = \{v_1, \ldots, v_n\}$. Jedes $v_i \in V$ ist ein Gatter.



Zustände Es sei $U = \{0, 1, \perp\}$ die Menge der logische Zustände ($\perp$ bezeichnet einen undefinierten Zustand, also entweder 0 oder 1), n_i die Anzahl der Eingänge von Knoten i und $I_i = \{0, \ldots, n_i - 1\}$ die Indexmenge der Eingänge von Knoten i.

$Z_i = U^{n_i}$ ist dann die Zustandsmenge von Knoten i (Wert an jedem Eingang wird gespeichert).

$Z = \bigotimes_i Z_i$ ist die Gesamtzustandsmenge.

Signale $S \subset U \times \mathbb{N}_0 \times V$.

Ereignis $(t, s = (u, k, v_i))$: Zur Zeit t geht der k-te Eingang von v_i auf den Wert u.

Zustandsänderung Die mit einem Eingang assoziierte Zustandsvariable merkt sich das Eingangssignal.

Folgeereignisse *Alle* am Ausgang angeschlossenen Bausteine erhalten zur Zeit $t+\delta$ eine Änderung des entsprechenden Eingangs. In den Folgeereignissen spiegelt sich die Schaltungsstruktur wider!

Gleichzeitige Ereignisse Im Modell ändert ein Ereignis *genau einen* Eingang zu einem Zeitpunkt. In der Realität können sich dagegen *mehrere* Eingänge gleichzeitig ändern. Dies erreichen wir im Modell dadurch, dass mehrere Ereignisse mit der selben Zeit in der Queue sein dürfen, d.h. diese Eingänge werden *einer nach dem anderen* zum selben Zeitpunkt auf ihren neuen Wert gesetzt.

Die dadurch ausgelösten Folgeereignisse müssen in ihrer Erstellungsreihenfolge erhalten werden obwohl sie alle den selben Zeitstempel tragen (stabiles Sortierverfahren in der Prioritätswarteschlange) *oder* vorhandene Ereignisse an einem Eingang mit dem selben Zeitstempel müssen überschrieben werden.

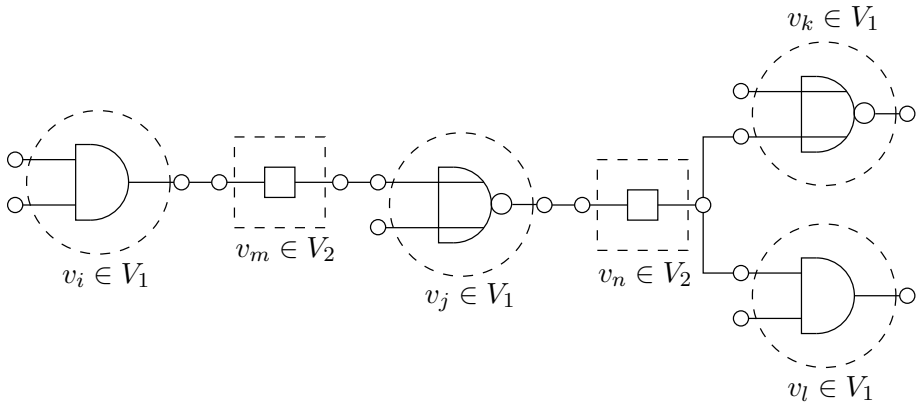
Beispiel: Exklusiv-Oder-Gatter. Zunächst seien alle Eingänge 0, dann werde einer 1, dann werde der zweite 1. Zunächst wird also das Ereignis Ausgang 1 zur Zeit $t + \delta$ aufgenommen, welches kurz darauf von Ausgang 0 zur Zeit $t + \delta$ überschrieben wird.

Problem Diese Modellierung ist relativ ineffizient, da der Wert am Ausgang eines Gatters evtl. mehrfach gespeichert wird.

2. Um die mehrfache Speicherung von Ausgangswerten zu vermeiden, kann man zwei Sorten von Knoten einführen:

- Grundbausteine (Gatter) ohne Zustand: V_1
- Verbindungen, die den Zustand auf der Leitung kapseln: V_2

Dann ist die Knotenmenge $V = V_1 \cup V_2$. Der Graph $G = (V, E)$ ist *bipartit*, da jede Kante einen Knoten aus V_1 mit einem Knoten aus V_2 verbindet. $v_i \in V_1$ löst



zum Zeitpunkt $t + \delta$ bei v_m ein Ereignis aus. Dazu ist ein lesender Zugriff auf die Zustände seiner Eingangsdrähte erforderlich. Es findet keine Zustandsänderung in v_i statt (v_i trägt keinen Zustand)!

$v_m \in V_2$ ändert bei Empfang des Ereignisses seinen Zustand und löst *ohne Verzögerung* ein Ereignis an den Eingängen der angeschlossenen Gatter aus. □

Bemerkung 2.4. Beide Möglichkeiten ordnen Zustandsvariablen eindeutig einen Knoten zu (Kapselung). Die zweite Variante erfordert lesenden Zugriff auf den Zustand einer anderen Komponente, wohingegen dies in der ersten Variante nicht erforderlich ist. Beide Varianten lassen sich gut mit *objektorientierten* Programmiersprachen umsetzen. Unsere allgemeine Beschreibung gibt das nicht zwingend vor, man könnte dort auch globale Zustände erlauben (wie im Able-Baker Beispiel geschehen). □

2.3 Praktische Aspekte

- Wie konstruiert man ein ereignisgesteuertes Modell im Rechner?
  - Schreibe ein Programm (Fortran, C, C++) — das liefert zwar ein sehr effizientes Modell (bzgl. Rechenzeit), ist aber im Allgemeinen sehr aufwändig. Klassenbibliotheken wie em-Plant (ehemals Simple++) für Prozessautomatisierung können den Entwurf vereinfachen.
  - Verwendung allgemeiner Simulationsprogramme mit speziellen Eingabesprachen, z.B. GPSS (IBM, 1960).
  - Anwendungsspezifische Simulationsprogramme, z.B. für Logiksimulation, sind in der Regel effizienter.
- Erzeugung von Zufallszahlen mit vorgegebener Verteilung, z.B. mittels *Inverse Transform Method*.

- Fehler in stochastischen Simulationen: Um zuverlässige Aussagen zu erhalten, benötigt man viele Realisierungen. Nach dem Gesetz der großen Zahlen gilt

$$p\left[\left|\frac{1}{n}\sum X_i - \mathbb{E}[X]\right| \geq \delta\right] \leq \epsilon \text{ für } n \geq \frac{\mathrm{Var}[X]^2}{\epsilon\delta^2}, \text{ d.h. } \delta \sim \sqrt{\frac{1}{n}}$$

  also sehr langsame Konvergenz.

- Simulationsprobleme können *sehr* groß sein (z.B. Design eines Core 2 Duo Prozessors) und sind dann nur noch mit parallelem Höchstleistungsrechnen berechenbar. Die parallele ereignisgesteuerte Simulation ist ein sehr schwieriges Thema, da die Kausalität immer sichergestellt werden muss: Befinden sich in der Queue zwei Ereignisse (e_1, t_1), (e_2, t_2) mit $t_1 < t_2$, so darf man sie nicht parallel berechnen, falls e_1 ein Ereignis (e_3, t_3) generiert mit $t_3 < t_2$. Dies ist aber a priori nicht bekannt!

3 Zelluläre Automaten

Zelluläre Automaten (engl. cellular automata) sind diskrete Modelle spezieller Form. Motivation war zunächst die Frage:

> Wie kann man mit möglichst einfachen, insbesondere lokalen Regeln komplexes Verhalten erzeugen?

3.1 Definition eines zellulären Automaten

Definition 3.1. Ein zellulärer Automat besteht aus

1. Einem Gitter Ω. Ω ist
   - diskret (abzählbar) und — zunächst — unendlich
   - von regelmäßiger Gestalt
   - von einer bestimmten Dimension

   Beispiel: $\Omega = \mathbb{Z} \times \mathbb{Z} = \{(i,j) \,|\, i,j \in \mathbb{Z}\}$.
   Die Elemente des Gitters heißen *Zellen*.

2. Einer *endlichen* Zustandsmenge Z.

3. Jeder Zelle $x \in \Omega$ wird ein Element aus Z zugeordnet, d.h. $z : \Omega \to Z$. Die Menge aller möglichen Zustände ist $G = \{z \,|\, z : \Omega \to Z\}$. $z(x) \in Z$ ist der Zustand der Zelle $x \in \Omega$, kurz z_x.

4. Zustände können sich mit der Zeit ändern. Die Zeit ist ebenfalls diskret ($n \in \mathbb{N}$).

   Der Zustand $z^n \in G$ zur Zeit $n \in \mathbb{N}$ heißt *Generation*.

5. Wir führen ein *endliches* Referenzgebiet $\hat{\Omega}$ ein. Beispiel: $\hat{\Omega} = \{(i,j) \,|\, |i| \leq 1 \wedge |j| \leq 1\}$.

6. Auch auf dem Referenzgebiet gibt es lokale Zustände: $\hat{G} = \{\hat{z} \,|\, \hat{z} : \hat{\Omega} \to Z\}$.

7. Die Abbildung $R_x : G \to \hat{G}$ schneidet aus dem Gesamtzustand einen lokalen Zustand heraus mittels

   $$(R_x(z))(y) = z(x+y) \qquad x \in \Omega, z \in G, y \in \hat{\Omega}.$$

   Die Zellen $N_x = \{y \in \Omega \,|\, y = x + w, w \in \hat{\Omega}\}$ heissen Nachbarschaft von x.

8. Die Regel ist eine Funktion

   $$f : \hat{G} \to Z$$

   , einen die einen lokalen Zustand auf ein Element aus Z abbildet.

9. Der Übergang von einer Generation z^n zur Generation $z^{n+1} = F(z^n)$ findet durch lokale Anwendung der Regel statt:

$$z^{n+1}(x) = f(R_x z^n) \qquad x \in \Omega.$$

$F : G \to G$ heisst Zustandsübergangsfunktion.

Beachte: Der neue Zustand der Zelle x hängt nur von den alten Werten in der Nachbarschaft der Zelle ab (*explizites Zeitschrittverfahren*).

Im Unterschied zur ereignisgesteuerten Simulation finden hier alle Änderungen in den Einzelzellen synchron statt.

Dies ist eine Menge gekoppelter endlicher deterministischer Automaten.

Geschichte der zellulären Automaten:

- Grundidee von J. von Neumann und S. Ulam am Los Alamos National Laboratory in den 1940 er Jahren.
- Neumann wollte "sich selbst reproduzierende Roboter" bauen und entwickelte einen zellulären Automaten mit 29 (!) Zuständen.
- 1969: Konrad Zuse: Physikalische Gesetze sind auf *kleinster Skala* diskret, d.h. die Welt ist ein gigantischer zellulärer Automat („Rechnender Raum").
- 1970: "Game of Life" von J. H. Conway.
- ab 1983: S. Wolfram (Begründer von "Mathematica") forscht über zelluläre Automaten. Ein gewisser Hype setzt ein.
- 2004: Wolframs umstrittenes Buch "A New Kind of Science" setzt die Ideen von Zuse fort.

Beispiel 3.2 (1D-Automaten)**.** Einfachster zellulärer Automat:

$$\Omega = \mathbb{Z} \text{ (eindimensionale Anordnung der Zellen)}$$
$$Z = \{0, 1\} \text{ (binäre Zustände)}$$
$$\hat{\Omega} = \{-1, 0, 1\}$$

Zustandsübergang f wird beschrieben durch Tabelle:

Referenzzustand	111	110	101	100	011	010	001	000	
Folgezustand	0	0	0	1	1	1	1	0	$= 30_{10}$

- Jeder Automat wird durch 8 Bit beschrieben $\Rightarrow$ es existieren 256 verschiedene Automaten.

- Die Automaten werden dezimal durchnummeriert (oben ist der Automat #30 angegeben).
- Untersuche nun das Verhalten dieses Automaten bezüglich der Standardeingabe:

$$\begin{array}{lcccccccccc} z^0 & \ldots & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & \ldots \\ z^1 & \ldots & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & \ldots \\ z^2 & \ldots & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & \ldots \\ usw. & & & & & & & & & & & \end{array}$$

- Bilder aller 256 Automaten gibt es bei (Bur), siehe auch Abbildung 7.
- Der Automat #30 zeigt sehr komplexes Verhalten.
- $z^n(0)$ wird von Mathematica als Zufallszahlengenerator verwendet.

Beispiel 3.3 (Game of Life)**.** Erfunden von J. H. Conway (britischer Mathematiker) im Jahr 1970.

$$\begin{aligned} \Omega &= \mathbb{Z} \times \mathbb{Z} \\ Z &= \{0, 1\} \\ \hat{\Omega} &= \{(i, j) \,|\, |i| \leq 1 \wedge |j| \leq 1\} \end{aligned}$$

Regeln:

1. $z^n(x) = 1$ und weniger als zwei Nachbarn mit Zustand $1 \Rightarrow 0$
2. $z^n(x) = 1$ und mehr als drei Nachbarn mit Zustand $1 \Rightarrow 0$
3. $z^n(x) = 1$ und zwei oder drei Nachbarn mit Zustand $1 \Rightarrow 1$
4. $z^n(x) = 0$ und genau drei Nachbarn mit Zustand $1 \Rightarrow 1$
5. In allen anderen Fällen ist der Folgezustand 0.

Dieser zelluläre Automat kennt Startkonfigurationen, die beliebige Muster konstruieren können (Replikation). Man kann Muster angeben, die eine beliebige Turingmaschine simulieren. Game of Life ist Turing–vollständig!

Erweiterungen von zellulären Automaten:

1. Stochastische Regeln: Aus mehreren möglichen Folgezuständen für einen lokalen Zustand wird zufällig ausgewählt.
2. endliche Gebiete
3. periodische Gebiete

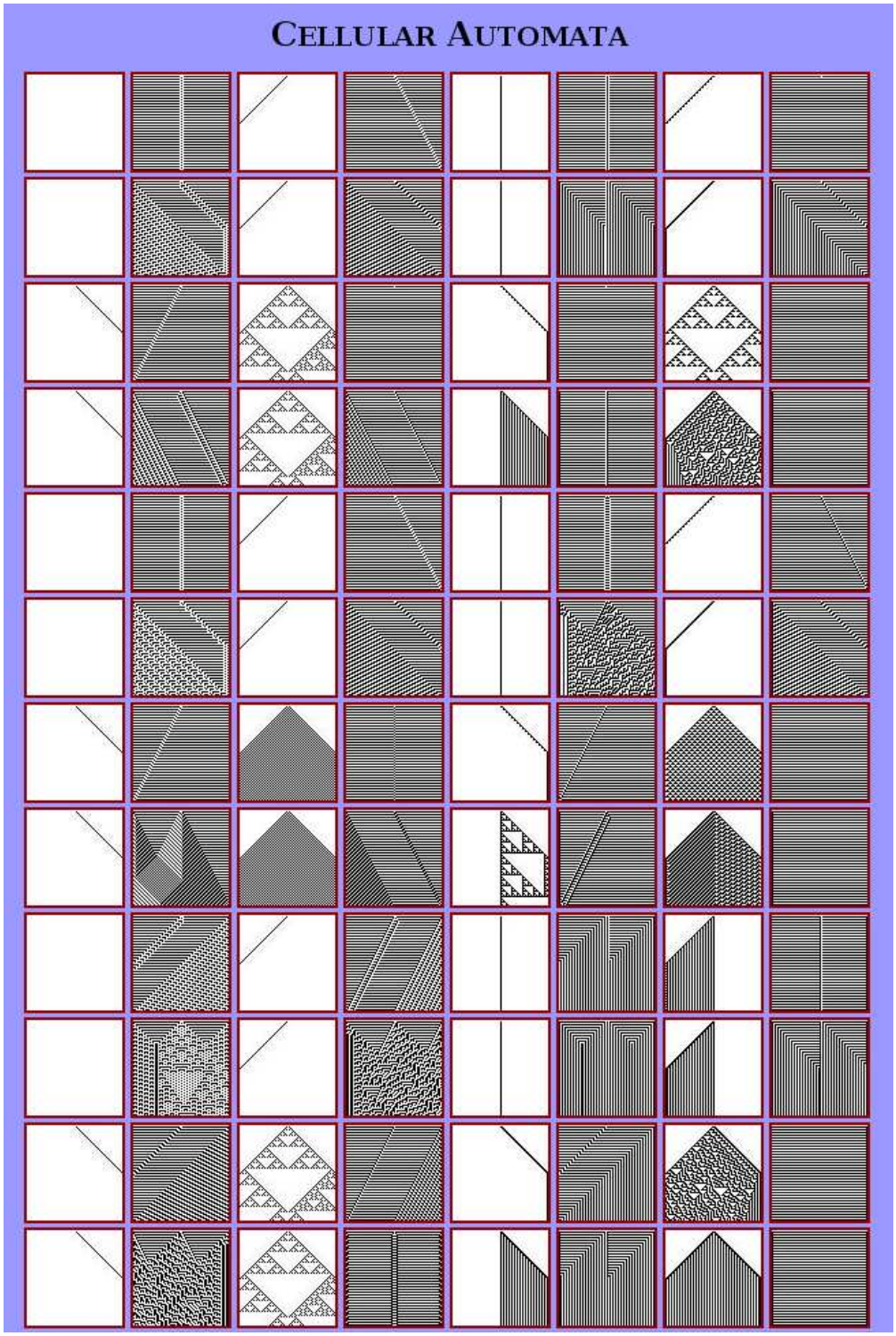


Abbildung 7: Ausschnitt aus der Liste aller Zellulären 1D-Automaten von `http://falconnet.peddie.org/students/2007/nburoojy/projects/cellular/`.

3.2 Verkehrssimulation mit zellulären Automaten (NS92)

$$\Omega = \mathbb{Z}$$
$$Z = \underbrace{\{0,1\}}_{\substack{\text{Auto in}\\ \text{Zelle?}}} \times \underbrace{\{0,1,\ldots,5\}}_{\substack{\text{Geschwindigkeit}\\ \text{in Zellen/Schritt}}}$$
$$\hat{\Omega} = \{0,1,2,3,4,5\}$$

Idee:

- Straße besteht aus Zellen (z.B. 7,5 m = Autolänge + Abstand im Stau) und ist unendlich lang (deshalb $\Omega = \mathbb{Z}$).

- Zustand einer Zelle: $z_i = (b_i, v_i)$
  b_i: Auto vorhanden?
  v_i: diskrete Geschwindigkeit $0, \ldots, 5$, die Einheit ist Zellen pro Zeitschritt. Dies entspricht 0 bis 135 km/h bei $\delta t = 1s$.

- Nachbarschaft: Fahrer sieht nur bis 5 Zellen voraus, d.h. die Strecke, die er in einem Schritt maximal fährt.

Regeln für $b_i = 1$:

1. Erhöhe Geschwindigkeit des Autos i um 1, falls $v_i < 5$ (beschleunigen).

2. Sei d_i der Abstand (in Zellen) zum nächsten Auto in N_i ($d_i = 5$ falls kein anderes Auto in N_i). Setze $v_i = \min(v_i, d_i)$ (Kollision vermeiden).

3. Falls $v_i > 0$, setze $v_i = v_i - 1$ mit einer gewissen Wahrscheinlichkeit p (trödeln) $\rightarrow$ Modell ist stochastisch!

4. $z_i = (0,0)$; $z_{i+v_i} = (1, v_i)$ (fahren).

Motivation:

- Fährt ein Auto nicht maximal schnell und ist der Vordermann weit genug weg, so wird beschleunigt. Der Beschleunigungsschritt kann durch den Trödelschritt wieder rückgängig gemacht werden. Das Auto wird aber nicht langsamer!

- Ein Auto mit Maximalgeschwindigkeit kann zufällig langsamer werden.

- Ein Auto, das wegen seinem Vordermann langsamer wurde, kann durch den Trödelschritt noch langsamer werden. Dieses Überreagieren ist die Ursache für "Staus aus dem Nichts".

Übergang ins Kontinuierliche:

Für eine sehr lange Strasse ist es sehr aufwändig das Verhalten jedes einzelnen Fahrzeuges zu simulieren. Um zu einer effizienteren Modellierung zu gelangen führt man durch Mittelwertbildung eine „Fahrzeugdichte“ ein:

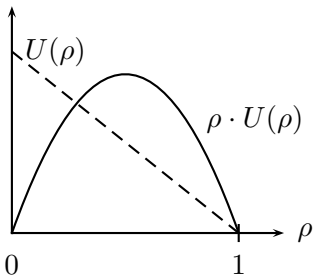
$$\rho(x) = \frac{\sum_{|i-x|<\delta} b_i}{\sum_{|i-x|<\delta} 1} = \left[\frac{\text{Autos}}{\text{Zelle}}\right]$$

- Offensichtlich gilt $0 \leq \rho(x) \leq 1$.
- Für $\delta \to \infty$ nimmt $\rho(x)$ kontinuierliche Werte an.
- Man kann auch raum-zeitliche Mittel definieren.

Für die Fahrzeugdichte kann man eine Differentialgleichungsmodell formulieren:

$$\frac{\partial \rho(x,t)}{\partial t} + \frac{\partial}{\partial x}\Big(\underbrace{U\left(\rho(x,t)\right)}_{\substack{\text{Geschwindigkeit}\\ \text{in Abhängigkeit}\\ \text{der Fahrzeugdichte}}} \cdot \rho(x,t)\Big) = 0$$

$$\underbrace{\quad\quad}_{\substack{[\frac{m}{s}]\cdot[\frac{Autos}{m}]=[\frac{Autos}{s}]=\\ \text{“Fluss” an Autos in einem Punkt}}}$$



Dies ist eine hyperbolische Erhaltungsgleichung (dazu mehr in der Vorlesung Numerik partieller Differentialgleichungen (Bas08)). Zur Analyse des Modells (Existenz und Eindeutigkeit der Lösung) stehen die Methoden der Analysis zur Verfügung.

Diese Gleichung beschreibt den Verkehrsfluss auf einer größeren (räumlichen) Skala, der sogenannten „Makroskala“ als der zelluläre Automat, der die Vorgänge auf der „Mikroskala“ beschreibt. Den Vorgang der Ableitung der makroskaligen Gleichung aus einer mikroskaligen Beschreibung nennt man „Upscaling“.

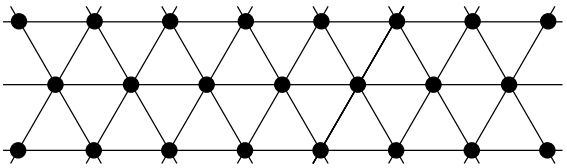
In der Differentialgleichung wird als Parameter die Fahrzeuggeschwindigkeit in Abhängigkeit der Dichte benötigt. Dies nennt man auch einen „effektiven Parameter“ der aus dem mikroskaligen Modell bestimmt werden kann. Dabei würde man für vorgegebene Fahrzeugdichten jeweils die mittlere Geschwindigkeit durch Simulation bestimmen.

3.3 Strömungssimulation mit zellulären Automaten (FHP86)

Lattice gas cellular automata: Modellierung der Strömung idealer Gase. Das heißt, die Gasmoleküle bewegen sich entsprechend der Newtonschen Gesetze und kollidieren ab und an (kinetische Gastheorie).

FHP-Automat

Gitter: Hexagonalgitter aus gleichseitigen Dreiecken



Zustandsmenge: $Z = \mathbb{B}^6$, $\mathbb{B} = \{0, 1\}$, mit folgender Bedeutung:

- Auf den Gitterlinien bewegen sich Teilchen, die alle die selbe Masse m besitzen.
- In einem Schritt bewegt sich ein Teilchen genau von einem Knoten zum Nächsten. $\Rightarrow$ 6 verschiedene Geschwindigkeiten mit jeweils gleichen Beträgen
- Zu einem diskreten Zeitpunkt befinden sich an einem Knoten maximal 6 Teilchen, höchstens eines in jeder Richtung.

Zustandsübergang Der Zustandsübergang besteht aus zwei Teilschritten. Im ersten Teilschritt führen die Teilchen an einem Knoten des Gitters Kollissionen aus. Dann fliegt jedes Teilchen in seiner Richtung zum nächsten Knoten.

Kollisionen Kollisionen werden durch eine Abbildung $f : \mathbb{B}^6 \to \mathbb{B}^6$ beschrieben. Kollisionen finden nur dann statt, wenn sich an einem Ort zwei, drei oder vier Teilchen befinden, deren Gesamtimpuls Null ist. In allen anderen Fällen bleibt der Zustand unverändert.

Bei zwei Teilchen mit Gesamtimpuls Null müssen diese genau aufeinander zu fliegen. Beide Teilchen werden dann entweder um $+60°$ oder um $-60°$ gedreht. Dabei wird die Richtung zufällig ausgewählt. Siehe Abbildung 8.

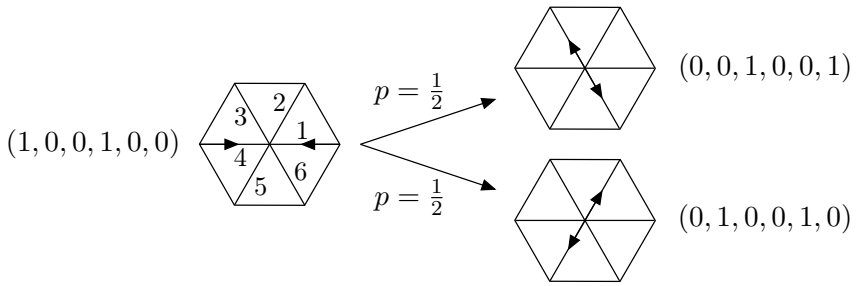


Abbildung 8: Zwei-Partikel-Kollision

Bei drei Teilchen mit Gesamtimpuls Null beträgt der Winkel zwischen diesen $120°$. Diese Konfiguration wird um $+60°$ gedreht. Siehe Abbildung 9.

Bei vier Teilchen mit Gesamtimpuls Null gibt es zwei gegenüberliegende unbesetzte Richtungen. Hier wird die gesamte Konfiguration mit Wahrscheinlichkeit $1/2$ um $+60°$, bzw. mit Wahrscheinlichkeit $1/2$ um $-60°$ gedreht. Siehe Abbildung 10.

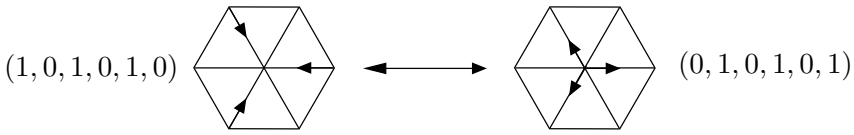


Abbildung 9: Drei-Partikel-Kollision

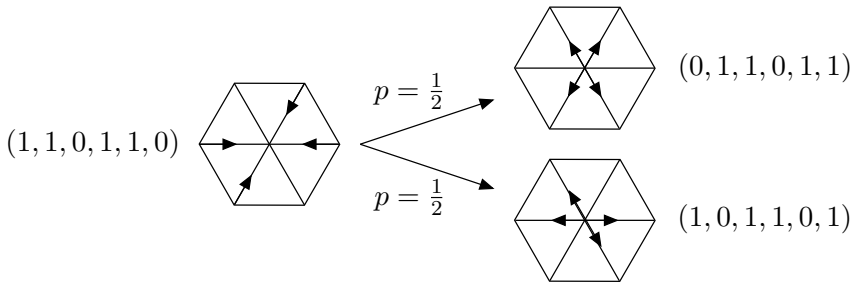


Abbildung 10: Vier-Partikel-Kollision

Wieder kann man durch Mittelung eine makroskopische Dichte und Impuls bzw. Geschwindigkeit definieren:

$$\rho(x) = \frac{\sum\limits_{\|y\| \leq \delta} \sum\limits_{i=1}^{6} z_i(x+y)}{\sum\limits_{\|y\| \leq \delta} 1},$$

$$\vec{v}(x) = \frac{\sum\limits_{\|y\| \leq \delta} \left\| \sum\limits_{i=1}^{6} z_i(x+y) \cdot \vec{e}_i \right\|}{\sum\limits_{\|y\| \leq \delta} 1}$$

Im Grenzübergang gehorchen diese makroskopischen Größen der Navier-Stokes-Gleichung (in 2D).

Eine praktisch nutzbare Weiterentwicklung stellt das Lattice-Boltzmann-Verfahren dar.

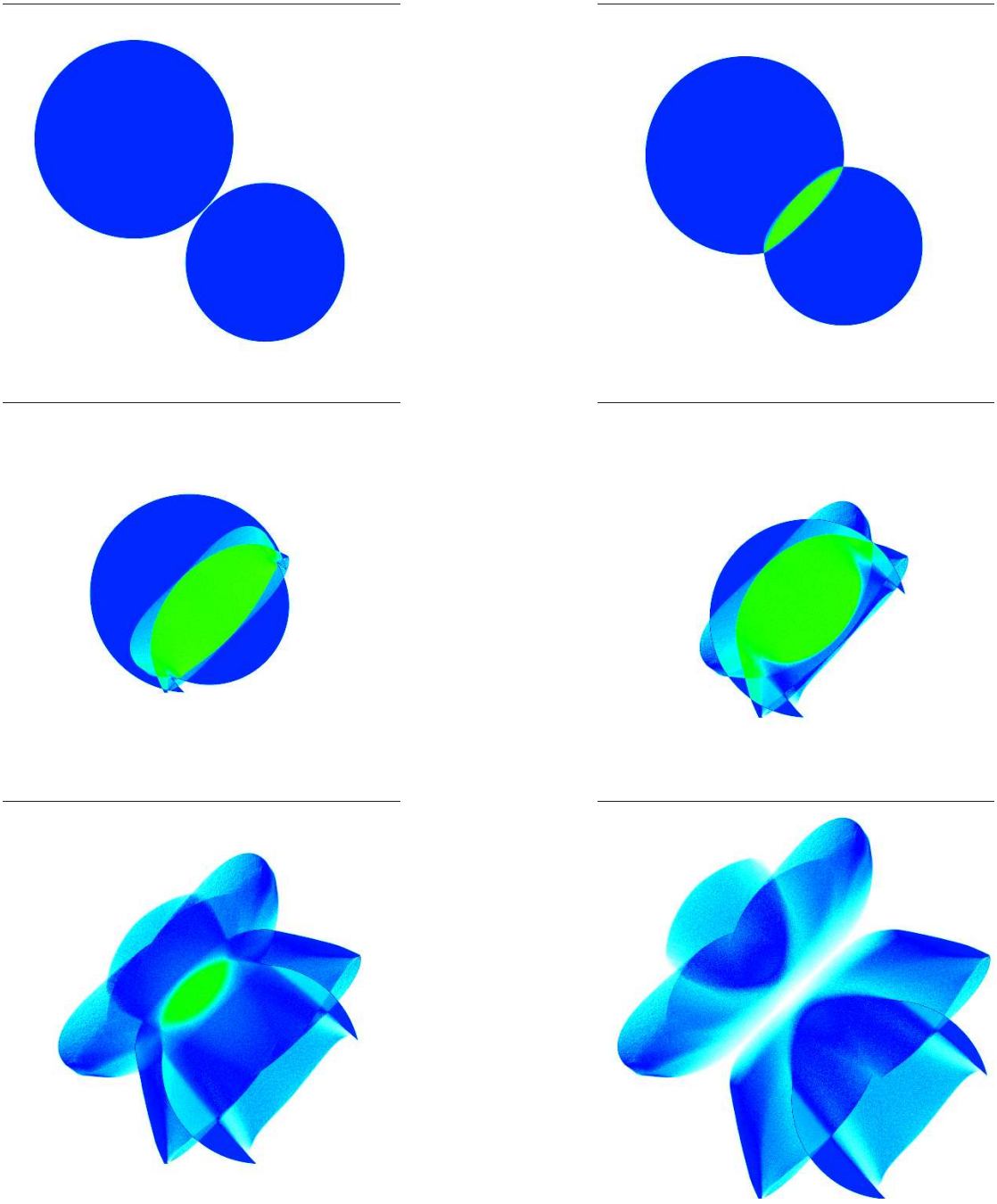


Abbildung 11: Simulation zweier ineinanderfliegender Gaswolken mit dem FHP-Automaten. Die Simulation benutzte 8192^2 Gitterpunkte, dargestellt ist die über je 16^2 Zellen gemittelte Dichte.

4 Modellierung elektrischer Bauelemente

Ziel ist die Modellierung und Simulation elektrischer Schaltungen bestehend aus Widerständen, Kondensatoren, Spulen sowie Quellen und ggf. Halbleiterbauelementen (Dioden, Transistoren).
Wie in Abschnitt 1.2 bereits besprochen, handelt es sich dabei um ein Mehrskalenproblem. Eine mögliche Beschreibungsebene sind die Maxwellschen Gleichungen. Dies ist ein System partieller Differentialgleichungen für das elektrische und magnetische Feld in Raum und Zeit. Daraus leitet man vereinfachte Gleichungen für sogenannte *konzentrierte Bauelemente* ab. Die Einführung unten folgt der ausführlichen und sehr lesenswerten Behandlung der Netzwerkanalyse in (Unb81).

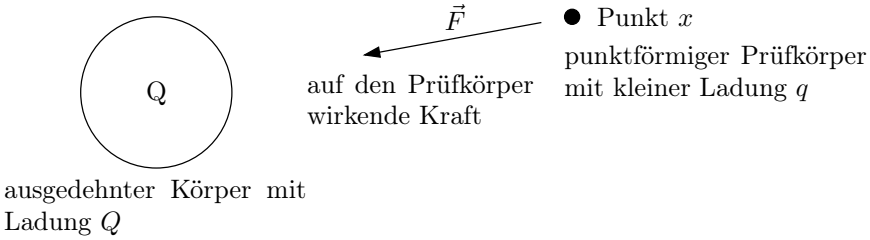
Dieses Anwendungsproblem bietet einen exemplarischen Einblick in die Modellierung auf Kontinuumsebene. Gleichzeit motiviert es eine reiche Auswahl mathematischer Methoden zur numerischen Lösung der Modellgleichungen von linearen Gleichungssystemen bis hin zu differentiell-algebraischen Systemen.

4.1 Strom und Spannung

Wir begeben uns auf eine kontinuumsmechanische Betrachtungsebene.

Spannung

- Körper können elektrische Ladungen tragen.
- Elektrisch geladene Körper ziehen sich je nach Vorzeichen ihrer Ladung an oder stoßen sich ab (*Coulombsches Gesetz*).



- $\vec{F}(\vec{x})$ hängt von $\vec{x}$ und q ab.
- $\vec{E}(\vec{x}) := \frac{\vec{F}(\vec{x})}{q}$ heißt *elektrische Feldstärke* und hat die Einheit $\left[\frac{N}{C}\right] = \left[\frac{V}{m}\right]$.
- Unter der *Spannung* zwischen zwei Punkten x_1 und x_2 in einem elektrischen Feld versteht man die Größe

$$u_{12} = \int_{x_1}^{x_2} \vec{E} \cdot \vec{t}\, ds \tag{4.1}$$

längs eines Weges C von x_1 nach x_2. (4.1) ist ein Kurvenintegral zweiter Art, bei dem $\vec{t}$ der Tangenteneinheitsvektor in jedem Punkt der Kurve bezeichnet. Die Bezeichnung des Linienintegrals zweiter Art ist nicht einheitlich. Oft findet man auch die Schreibweise $\int_{x_1}^{x_2} \vec{E}\, d\vec{r}$ mit $d\vec{r} = \vec{t}\, ds$.

- Ruhen die felderzeugenden Ladungen, so ist u_{12} unabhängig vom gewählten Weg. Dies ist eine elementare Eigenschaft des elektrischen Feldes, die aus den Maxwellschen Gleichungen folgt.

- Die Spannung zwischen zwei Punkten ist gerichtet, da sich die Tangentenrichtung im Kurvenintegral mit der Durchlaufrichtung gerade umdreht:

$$u_{12} = \int_{x_1}^{x_2} \vec{E} \cdot \vec{t}\, ds = -\int_{x_2}^{x_1} \vec{E} \cdot \vec{t}\, ds = -u_{21} \tag{4.2}$$

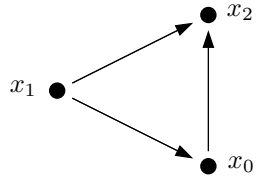
  $\Rightarrow$ Richtung und Vorzeichen sind in diesem Sinne austauschbar.

- Wegen $\vec{F} = q\vec{E}$ ist $qu_{12} = q\int_{x_1}^{x_2} \vec{E} \cdot \vec{t}\, ds = \int_{x_1}^{x_2} \vec{F} \cdot \vec{t}\, ds$ eine Energie! Dies ist völlig analog zur Mechanik (Arbeit ist Kraft mal Weg). Genauer ist qu_{12} eine potenzielle Energie, also die Fähigkeit Arbeit zu verrichten.

- Als *Potential* definiert man die negative Spannung relativ zu einem gewählten Bezugspunkt:

$$\varphi(x) = -\int_{x_0}^{x} \vec{E} \cdot \vec{t}\, ds \tag{4.3}$$

  Man kann jedem Punkt die skalare Größe $\varphi(x)$ zuordnen (Höhe im Gravitationsfeld) und darüber die Spannung berechnen:



$$\begin{aligned} u_{12} &= \int_{x_1}^{x_2} \vec{E} \cdot \vec{t}\, ds = \int_{x_1}^{x_0} \vec{E} \cdot \vec{t}\, ds + \int_{x_0}^{x_2} \vec{E} \cdot \vec{t}\, ds = -\int_{x_0}^{x_1} \vec{E} ds - \left[-\int_{x_0}^{x_2} \vec{E} \cdot \vec{t}\, ds\right] \\ &= \varphi(x_1) - \varphi(x_2) \end{aligned} \tag{4.4}$$

  Hier wurde entscheidend die Wegunabhängigkeit des Linienintegrals im elektrischen Feld benutzt.

Strom

- Bewegen sich elektrische Ladungen, so spricht man von einem *Strom*.

- In Metallen stehen freie Elektronen zur Verfügung (sog. *Leitungselektronen*). Herrscht im Leiter ein elektrisches Feld $\vec{E}$, so wirkt auf ein Elektron die Kraft

$$\vec{F}_{el} = -e\vec{E} \tag{4.5}$$

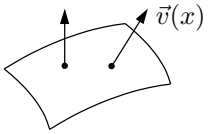
  (Minuszeichen, da Ladung des Elektrons negativ ist).

- Dem entgegen wirkt eine materialabhängige „Reibungskraft“. Dadurch stellt sich eine feste Geschwindigkeit
$$\vec{v} = b\vec{E} \tag{4.6}$$
ein.

- In einem makroskopisch ausgedehnten Körper wollen wir nicht jedes Elektron einzeln betrachten. Daher führt man in der Kontinuumsbetrachtung die *Ladungsdichte* ein:
$$\rho(x,t) = \lim_{\Delta V \to 0} \frac{\Delta Q(x,t)}{\Delta V(x)} = -e \cdot \lim_{\Delta V \to 0} \frac{\Delta N(x,t)}{\Delta V(x)} \tag{4.7}$$
In dieser Definition ist $\Delta N(x,t)$ die Anzahl der Elektronen in dem Volumen $\Delta V(x,t)$ zur Zeit t. Das Volumen $\Delta V(x,t)$ ist um den Punkt x lokalisiert. Die Ladung in einem beliebigen (makroskopischen) Raumvolumen V ist dann gegeben durch das Volumenintegral
$$\int_V \rho(x,t)\, dx.$$

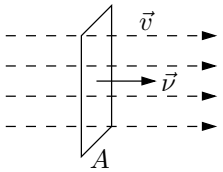
- Zusammen mit dem Geschwindigkeitsfeld ergibt sich die *Stromdichte*:
$$\vec{J}(x,t) = \rho(x,t)\vec{v}(x) \qquad \left[\frac{C}{m^3} \cdot \frac{m}{s} = \frac{C}{m^2 s}\right] \tag{4.8}$$

- Für eine gerichtete Fläche A im Raum erhält man den Strom durch die Fläche als Oberflächenintegral

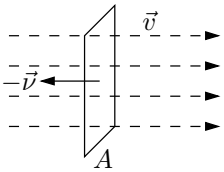


$$i_A(t) = \int_A \vec{J}(x,t) \cdot \vec{\nu}\, ds \qquad \left[\frac{C}{m^2 s} \cdot m^2 = \frac{C}{s} = A\right] \tag{4.9}$$

- Auch der Strom ist bezüglich der Normalenrichtung der Fläche gerichtet:



$i = \vec{J}\vec{\nu}\,|A|$



$i' = -\vec{J}\vec{\nu}\,|A| = -i$

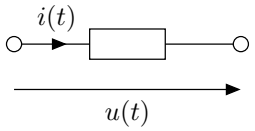
4.2 Ideale Netzwerkelemente

Wir gehen nun zu idealisierten Modellen für elektrische Bauelemente über (Skalenübergang):

- Ein Bauelement hat zwei Anschlüsse. Durch das Element fließt ein Strom $i(t)$ und es liegt eine Spannung $u(t)$ über den beiden Klemmen an. Dem Element ist eine Richtung zugeordnet, die die Zählrichtung von Strom und Spannung festlegt.

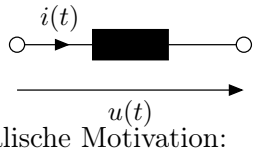
- $u(t)$ und $i(t)$ sind skalare Größen. Vom räumlichen Aufbau der Elemente wird abstrahiert.
  Dies ist nur unter gewissen Annahmen zulässig: $u(t)$ und $i(t)$ dürfen sich nicht zu schnell ändern $\Rightarrow$ Frequenzen < 10 MHz.

Ohmscher Widerstand



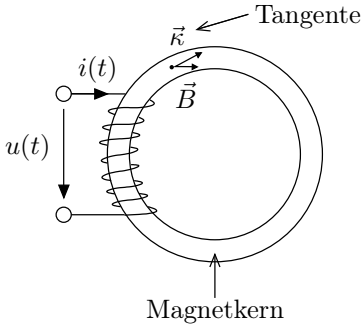
Es gilt: $u(t) = R \cdot i(t)$
$R > 0$: Widerstand in Ohm $\left[\Omega = \frac{V}{A}\right]$

Induktivität (Spule)



Es gilt: $u(t) = L \cdot \frac{di(t)}{dt}$
$L > 0$: Induktivität in Henry $\left[H = \frac{Vs}{A}\right]$

physikalische Motivation:



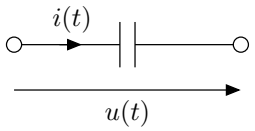
Strom erzeugt magnetisches Feld $\vec{B}$ im Magnetkern (Durchflutungsgesetz):

$$i = c_1 \vec{B} \vec{\kappa}$$

Zeitlich sich änderndes Feld $\vec{B}$ erzeugt Spannung (Induktionsgesetz):

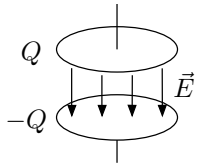
$$u = c_2 \frac{d\left(\vec{B}(t) \cdot \vec{\kappa}\right)}{dt} = \frac{c_2}{c_1} \frac{di(t)}{dt}$$

Kapazität (Kondensator)



Es gilt: $i(t) = C \cdot \frac{du(t)}{dt}$
$C > 0$: Kapazität in Farad $\left[F = \frac{As}{V}\right]$

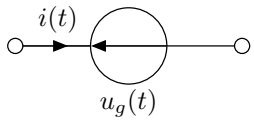
physikalische Motivation: Plattenkondensator



Oben: Ladung Q (Elektronenmangel)
Unten: Ladung $-Q$ (Elektronenüberschuss)
$\Rightarrow$ es entsteht ein elektrisches Feld.
Wichtig: Es fließt kein Strom durch den Kondensator, sondern es werden Ladungen gespeichert, die man wieder abrufen kann.

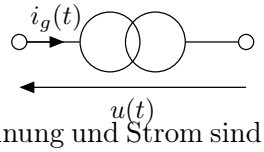
Spulen und Kondensatoren sind Energiespeicher. Nach dem Aufladen können sie Ströme fließen lassen, also Arbeit verrichten.

Spannungsquelle



An den Klammern steht eine Spannung $u_g(t)$ zur Verfügung. Diese ist unabhängig vom Strom $i(t)$. Bei positivem $i(t)$ wäre die rechte Klemme der Pluspol und die linke Klemme der Minuspol. Traditionsgemäß fließt ein positiver Strom vom Pluspol der Spannungsquelle zum Minuspol.

Stromquelle



Die Stromquelle prägt einen Strom $i_g(t)$ auf. Dieser ist unabhängig von der Spannung $u(t)$.

Spannung und Strom sind bei den Quellen entgegengesetzt zu einander orientiert. Diese Wahl sorgt dafür, dass in einer einfachen Schaltung mit einer Spannungsquelle und einem Widerstand keine unatürlichen negativen Vorzeichen auftreten.

gesteuerte Quellen

Hängt der Strom oder die Spannung einer Quelle von einem Strom oder einer Spannung an einem anderen Element ab, so spricht man von einer gesteuerten Quelle (so werden Halbleiterbauelemente modelliert).

reale Bauelemente

Reale Bauelemente lassen sich gut durch Zusammenschaltung idealisierter Bauelemente beschreiben.

4.3 Kirchhoffsche Gesetze

Netzwerke sind Zusammenschaltungen von Bauelementen.

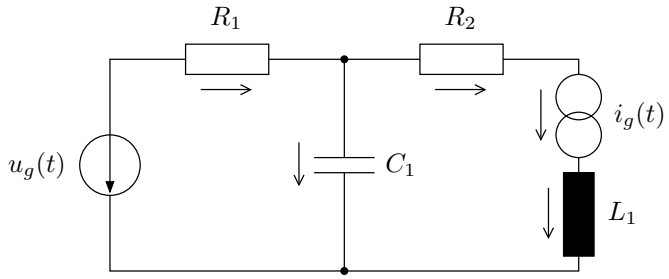


Abbildung 12: Netzwerk

Für jedes Element ist willkürlich eine Richtung festzulegen. Zu bestimmen sind nun die gerichteten Ströme und Spannungen an den einzelnen Elementen. Dazu benötigt man zusätzlich zu den Bauelementbeziehungen noch weitere Gesetze.

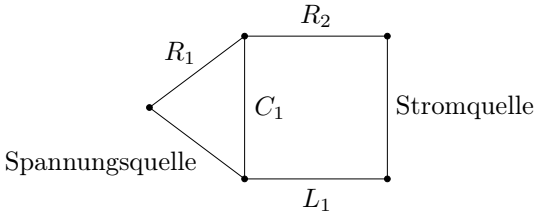
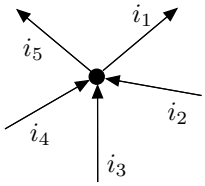


Abbildung 13: Netzwerk als Graph

Knotenregel (1. Kirchhoffsches Gesetz)

Die Summe aller an einem Knoten ein- und abgehenden Ströme ist zu jedem Zeitpunkt Null.

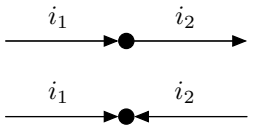
$$\sum_{\mu=1}^{m} \pm i_\mu(t) = 0$$



Vorzeichen:
+ Strom wird zum Knoten hin gezählt
− Strom wird vom Knoten weg gezählt

Begründung: In einem Knoten wird keine Ladung gespeichert.

Beispiel 4.1.

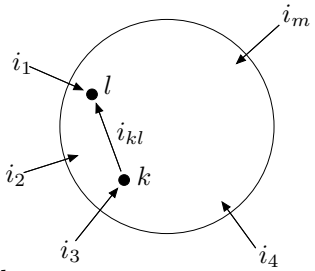


$i_1 - i_2 = 0 \Rightarrow i_1 = i_2$

$i_1 + i_2 = 0 \Rightarrow i_1 = -i_2$

Folgerungen aus der Knotenregel

1. Gegeben ein beliebiges Netzwerk, auf das m Ströme hinführen.



Dann gilt $\sum_{\mu=1}^{m} i_\mu(t) = 0$, denn:

- Netzwerk habe q innere Knoten
- stelle Knotenregel für jeden der q Knoten auf
- summiere über alle q Gleichungen

$$\begin{aligned} \text{Gleichung } k\text{:} \quad \ldots - i_{kl} \ldots + i_3 &= 0 \\ \text{Gleichung } l\text{:} \quad \ldots + i_{kl} \ldots + i_1 &= 0 \\ \sum_{\mu=1}^{m} i_\mu(t) &= 0 \end{aligned}$$

Alle inneren Ströme heben sich weg.

2. Bei einem Netzwerk mit k Knoten muss man die Knotenregel nur für die ersten $k-1$ Knoten aufstellen. Summieren über die ersten $k-1$ Gleichungen gibt:

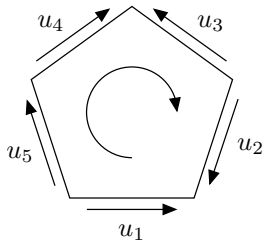
$$\begin{aligned} \text{Gleichung } l\text{:} \quad \ldots + i_{ml} \ldots - i_{lk} &= 0 \\ \text{Gleichung } m\text{:} \quad \ldots - i_{ml} \ldots + i_{mk} &= 0 \\ \sum \pm i_{jk} &= 0 \end{aligned}$$

Alle Ströme zwischen Knoten $l, m < k$ heben sich wieder weg. Es bleibt die Knotenregel für den Knoten k stehen. Die k-te Gleichung ist somit linear abhängig von den ersten $k-1$ Gleichungen.

Maschenregel (2. Kirchhoffsches Gesetz)

Ein geschlossener Weg in einem Netzwerk heißt *Masche*. Einer Masche ist eine Orientierung (Durchlaufrichtung) zuzuordnen.
Die Summe aller Spannungen der Elemente in einer Masche ist zu jedem Zeitpunkt Null.

$$\sum_{\mu=1}^{m} \pm u_\mu(t) = 0$$



Vorzeichen:
+ Spannung hat gleiche Orientierung wie Masche
− Spannung hat entgegengesetzte Orientierung

Begründung: $\int_{x_1}^{x_2} \vec{E} ds = \varphi(x_1) - \varphi(x_2) = 0$, falls $x_1 = x_2$. Die Maschenregel ist also eine Folge der Wegunabhängigkeit.

Eine Masche in einem ebenen Netzwerk (planarer Graph), die keine Elemente umschließt, heißt *Elementarmasche*. In ebenen Netzwerken genügt es, die Maschenregel für alle Elementarmaschen aufzustellen. Alle anderen Gleichungen sind linear abhängig.

5 Netzwerkanalyse mit dem Knotenpotentialverfahren

Zunächst betrachten wir ein Beispiel wie mit den eingeführten Konzepten ein elektrisches Netzwerk analysiert werden kann. Dann leiten wir ein Verfahren zur systematischen Analyse beliebiger Netzwerke her.

5.1 Analyse zweier einfacher Netzwerke

Wir wollen das folgende einfache Netzwerk analysieren.

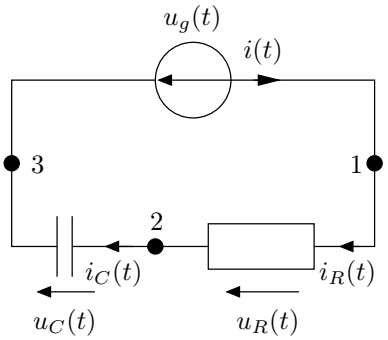


Abbildung 14: RC-Glied

Gleichungen

I Netzwerkelemente: $u_g(t)$ ist gegeben, sowie

$$u_R(t) = R \cdot i_R(t) \tag{a}$$
$$i_C(t) = C \cdot \frac{du_C(t)}{dt} \tag{b}$$

II Knotenregel (Wie oben erläutert sind nur zwei von drei Knoten zu betrachten):

$$-i(t) - i_R(t) = 0 \qquad \text{(Knoten 1)} \tag{c}$$
$$i_R(t) - i_C(t) = 0 \qquad \text{(Knoten 2)} \tag{d}$$

III Maschenregel (die einzige Masche sei im Uhrzeigersinn orientiert):

$$-u_g(t) + u_R(t) + u_C(t) = 0 \tag{e}$$

Es ergeben sich 5 Gleichungen für 5 Unbekannte i, i_R, i_C, u_R, u_C.

Alle Gleichungen zusammen bilden ein *differentiell-algebraisches System* (engl. differential algebraic equation, DAE), da es aus (gewöhnlichen) Differentialgleichungen und algebraischen Gleichungen besteht.

Geschickte Lösung

$$u_R(t) \overset{(a)}{=} R \cdot i_R(t) \overset{(d)}{=} R \cdot i_C(t) \overset{(b)}{=} RC\frac{du_C(t)}{dt}$$

Eingesetzt in die Maschenregel (e) erhält man eine lineare gewöhnliche Differentialgleichung für $u_C(t)$:

$$RC\frac{du_C(t)}{dt} + u_C(t) = u_g(t) \tag{5.1}$$

Eine Anfangsbedingung für $u_C(t_0)$ ist erforderlich.

Lösung der Differentialgleichung

für $u_C(0) = 0$, $u_g(t) = \begin{cases} 0 & \text{falls } t < 0 \\ U_0 & \text{falls } t \geq 0 \end{cases}$

Ansatz: $u_C(t) = c_1 e^{c_2 t} + c_3$. Einsetzen in Gleichung (5.1) ergibt:

$$RCc_1c_2e^{c_2t} + c_1e^{c_2t} + c_3 = U_0 \Leftrightarrow \left[RCc_1c_2 + c_1\right] e^{c_2t} + c_3 = U_0$$

Die linke Seite muss für alle Zeiten $t \geq 0$ gleich U_0 sein. Dies führt per Koeffizientenvergleich auf die beiden Bedingungen

$$RCc_1c_2 + c_1 = 0 \Rightarrow RCc_2 + 1 = 0 \Rightarrow c_2 = -\frac{1}{RC}$$
$$c_3 = U_0$$

Zusätzlich liefert die Anfangsbedingung:

$$c_1e^{c_20} + c_3 = 0 \Rightarrow c_1 = -c_3 = -U_0$$

also:

$$u_c(t) = -U_0e^{-\frac{1}{RC}t} + U_0 = U_0\left(1 - e^{-\frac{t}{RC}}\right)$$
$$i_c(t) = i_R(t) = Ru_R(t) = R\left(u_g(t) - u_C(t)\right) = R\left(U_0 - U_0\left(1 - e^{-\frac{t}{RC}}\right)\right) = RU_0e^{-\frac{t}{RC}}$$

Weiteres Beispiel: gedämpfter Reihenschwingkreis

Gleichungen

I Netzwerkelemente:

$$u_R(t) = Ri_R(t)$$
$$u_L(t) = L\frac{di_L(t)}{dt}$$
$$i_C(t) = C\frac{du_C(t)}{dt}$$

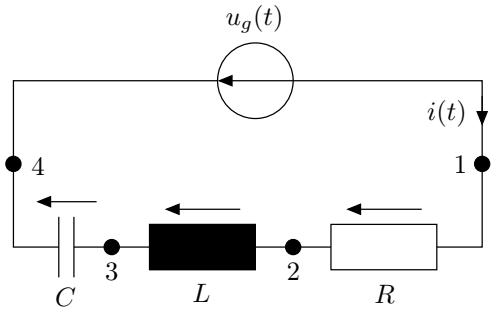


Abbildung 15: gedämpfter Reihenschwingkreis

II Knotenregel (3 Gleichungen):

$$i_R(t) = i_L(t) = i_C(t) = i(t)$$

III Maschenregel (im Uhrzeigersinn orientiert):

$$u_R(t) + u_L(t) + u_C(t) = u_g(t)$$

Aus I und II folgt:

$$u_R(t) = Ri(t) = RC\frac{du_C(t)}{dt}$$
$$u_L(t) = L\frac{di(t)}{dt} = LC\frac{d^2u_C(t)}{dt^2}$$

Mit III erhält man für $u_C(t)$ eine lineare gewöhnliche Differentialgleichung zweiter Ordnung:

$$LC\frac{d^2u_C(t)}{dt^2} + RC\frac{du_C(t)}{dt} + u_C(t) = u_g(t) \tag{5.2}$$

Anfangsbedingungen für $u_C(t_0)$ und $\frac{du_C}{dt}\big|_{t=t_0} = \frac{i_L(t_0)}{C}$ werden benötigt (Für jeden linear unabhängigen Energiespeicher benötigt man eine Anfangsbedingung).

5.2 Das Knotenpotentialverfahren

Sollen beliebige Netzwerke z.B. durch ein Computerprogramm analysiert werden, so benötigen wir ein systematisches (algorithmisches) Vorgehen. Das Knotenpotentialverfahren stellt ein mögliches Verfahren dar.

Lineare Unabhängigkeit

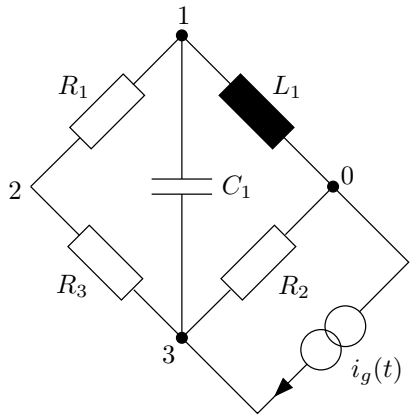
- Knoten- und Maschenregel erzeugen lineare Beziehungen zwischen Zweigströmen bzw. Zweigspannungen. Wie wir gesehen haben, lassen sich diese Gleichungen teilweise auseinander ableiten, d.h. sie sind linear abhängig.

- Folge: Nur eine Teilmenge der Zweigströme bzw. -spannungen ist frei wählbar. Der Rest ergibt sich zwangsläufig aus der Knoten- bzw. Maschenregel.
- Test auf lineare Abhängigkeit: Setze eine Teilmenge der Zweigströme (Zweigspannungen) auf Null und betrachte einen weiteren Strom (eine weitere Spannung). Ist dieser (diese) dann ebenfalls Null, so ist er (sie) linear abhängig von der Teilmenge.

Begriffe aus der Graphentheorie

- Es sei ein RLC-Netzwerk mit k Knoten und l Kanten (Elementen) gegeben.
- Der zugehörige Graph se zusammenhängend und jeder Knoten habe mindestens Grad zwei.
- Zu jeder Kante gebe es mindestens einen geschlossenen Weg in dem Graphen der über diese Kante führt.
- *Baum*: Graph, bei dem es zwischen je zwei Knoten genau einen Weg gibt.
- Ein Baum mit k Knoten hat genau $k-1$ Kanten.
- In jeden zusammenhängenden Graphen lässt sich ein (vollständiger) spannender Baum legen. Dieser ist in der Regel nicht eindeutig bestimmt.
- Die übrigen $m = l - (k-1)$ Kanten nennt man *Baumkomplement*.
- Zu jeder Kante im Baumkomplement gibt es genau einen Weg im spannenden Baum, der die beiden Endknoten verbindet.

Beispiel 5.1. Das Knotenpotentialverfahren erklären wir an folgendem Beispiel:



- Knoten sind ab 0 durchnummeriert.
- Zählrichtung der Ströme und Spannungen ergibt sich durch die Nummerierung (u_{12} ist die Spannung von 1 nach 2).

- Zunächst behandeln wir nur Netzwerke ohne Spannungsquellen! Diese führen wir später ein.

□

Wahl linear unabhängiger Spannungen Wähle einen beliebigen spannenden Baum im Netzwerk. Die $k-1$ Spannungen zu den Baumzweigen stellen ein maximales System linear unabhängiger Spannungen dar, denn:

1. Setze alle Spannungen in Baumzweigen auf Null (Kurzschluss) und betrachte einen Zweig im Komplement. Es gibt genau eine Masche über den Baum, die Spannung in diesem Zweig ist also auch Null.

2. Setze Spannung an genau einer Kante des Baumes ungleich Null und alle anderen auf Null. Dann gibt es mindestens einen Komplementzweig, an dem die Spannung ungleich 0 ist. Weniger als $k-1$ Spannungen genügen also nicht.

Die Spannungen an den Kanten des Baumkomplements werden aus den Spannungen über Baumzweige berechnet. *Dies bedeutet, dass die Maschenregel automatisch erfüllt ist!*

Knotenpotentiale, Ausdrücken der Zweigspannung durch Potentiale Statt der Spannungen über die $k-1$ Baumkanten kann man auch die Potentiale (die Spannungen relativ zu einem Bezugspunkt) an $k-1$ Knoten betrachten. Dazu wählt man einen Knoten als Bezugspunkt. Dies sei hier der Knoten 0, d.h. $\varphi_0 = 0$. Damit gilt:

$$\begin{aligned} u_{12} &= \varphi_1 - \varphi_2 \\ u_{23} &= \varphi_2 - \varphi_3 \\ u_{30} &= \varphi_3 - 0 \\ u_{01} &= 0 - \varphi_1 \\ u_{13} &= \varphi_1 - \varphi_3 \end{aligned} \quad \xrightarrow{\text{Matrixform}} \quad \underbrace{\begin{bmatrix} u_{12} \\ u_{23} \\ u_{30} \\ u_{01} \\ u_{13} \end{bmatrix}}_{u(t)} = \underbrace{\begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \\ -1 & 0 & 0 \\ 1 & 0 & -1 \end{bmatrix}}_{K} \cdot \underbrace{\begin{bmatrix} \varphi_1 \\ \varphi_2 \\ \varphi_3 \end{bmatrix}}_{\varphi(t)} \tag{5.3}$$

Wegen obiger Überlegung sind die Spalten von K linear unabhängig. Der Vektor $u(t)$ enthält *alle* Zweigspannungen (bei parallelgeschalteten Zweigen ist die Indizierung so ungeschickt).

Anwendung der Knotenregel Die Knotenregel muss für $k-1$ Knoten des Netzwerkes gefordert werden.

- Lasse Gleichung für Bezugsknoten 0 weg.

- Zählrichtung: Vom Knoten weg → positives Vorzeichen (dies ist genau entgegengesetzt aber letztlich eine willkührlich Festlegung).

$$\begin{aligned}
&\text{Knoten 1} && i_{12} + i_{13} - i_{01} = 0 \\
&\text{Knoten 2} && -i_{12} + i_{23} = 0 \\
&\text{Knoten 3} && -i_{23} - i_{13} + i_{30} - i_g(t) = 0
\end{aligned}$$

$$\xrightarrow{\text{Matrixform}} \quad \underbrace{\begin{bmatrix} 1 & 0 & 0 & -1 & 1 \\ -1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & -1 \end{bmatrix}}_{K^T} \cdot \underbrace{\begin{bmatrix} i_{12} \\ i_{23} \\ i_{30} \\ i_{01} \\ i_{13} \end{bmatrix}}_{i(t)} = \underbrace{\begin{bmatrix} 0 \\ 0 \\ i_g(t) \end{bmatrix}}_{b(t)} \tag{5.4}$$

Netzwerkelemente Schließlich gilt für jedes Netzwerkelement ein Zusammenhang zwischen Strom und Spannung.

$$\begin{aligned}
i_{12}(t) &= \frac{u_{12}(t)}{R_1} \\
i_{23}(t) &= \frac{u_{23}(t)}{R_3} \\
i_{30}(t) &= \frac{u_{30}(t)}{R_2} \\
i_{01}(t) &= i_{01}(t_0) + \frac{1}{L_1} \int_{t_0}^{t} u_{01}(\tau) d\tau \qquad \left(\text{wegen } u_L(t) = L \frac{di_L(t)}{dt} \right) \\
i_{13}(t) &= C_1 \frac{du_{13}(t)}{dt}
\end{aligned} \tag{5.5}$$

In Matrixform gilt: $i(t) = Yu(t) + y_0$
Y ist ein diagonaler Operator: Multiplikation (R), Differentiation (C), Integration (L). Der Vektor y_0 enthält die Anfangsbedingungen für die Induktivitäten, die sich durch die Integration der Strom-Spannungsbeziehung ergibt.

Kombination aller Teilschritte

$$K^T i(t) \overset{(5.5)}{=} K^T \left(Yu(t) + y_0 \right) \overset{(5.3)}{=} K^T Y K \varphi(t) + K^T y_0 \overset{(5.4)}{=} b(t)$$

$$\Leftrightarrow \quad K^T Y K \varphi(t) = b - K^T y_0 \tag{5.6}$$

Beispiel 5.2 (Anwendung auf das Netzwerk aus Beispiel 5.1)**.** Es ergibt sich das System

$$\begin{aligned}
&\text{Knoten 1} && \frac{\varphi_1 - \varphi_2}{R_1} + C_1 \frac{d\left(\varphi_1 - \varphi_3\right)}{dt} - \left[i_{01} + \frac{1}{L_1} \int_{t_0}^{t} -\varphi_1(\tau) d\tau \right] = 0 \\
&\text{Knoten 2} && -\frac{\varphi_1 - \varphi_2}{R_1} + \frac{\varphi_2 - \varphi_3}{R_3} = 0 \\
&\text{Knoten 3} && -\frac{\varphi_2 - \varphi_3}{R_3} - C_1 \frac{d\left(\varphi_1 - \varphi_3\right)}{dt} + \frac{\varphi_3}{R_2} = i_g(t)
\end{aligned} \tag{5.7}$$

Fasst man die Terme mit φ_1, φ_2, φ_3 zusammen ergibt sich die Form:

$$\begin{aligned}
\left[\frac{1}{R_1} + C_1\frac{d}{dt} + \frac{1}{L}\int_{t_0}^{t}\right]\varphi_1 \qquad & -\frac{1}{R_2}\varphi_2 \qquad & -C_1\frac{d}{dt}\varphi_3 \;&= i_{01}(t_0) \\
-\frac{1}{R_1}\varphi_1 + \left[\frac{1}{R_1} + \frac{1}{R_3}\right]\varphi_2 \qquad & & -\frac{1}{R_3}\varphi_3 \;&= 0 \\
-C_1\frac{d}{dt}\varphi_1 \qquad & -\frac{1}{R_3}\varphi_2 + \left[\frac{1}{R_2} + C_1\frac{d}{dt} + \frac{1}{R_3}\right]\varphi_3 \;&= i_g(t)
\end{aligned} \tag{5.8}$$

□

Zusammenfassung

- Als allgemeines Resultat ergibt sich ein System von linearen Algebro–Integro–Differentialgleichungen mit konstanten Koeffizienten.
- Hat man nur Widerstände (Widerstandsnetzwerk), so erhält man ein lineares Gleichungssystem.
- Kapazitäten ⇒ Ableitung, Induktivitäten ⇒ Integral
- Durch Differenzieren erhält man ein Differentialgleichungssystem zweiter Ordnung. Dieses kann auf ein System erster Ordnung reduziert werden.

  Alternativ (und besser) führt man die Stammfunktion $\Phi_i(t) = \int_{t_0}^{t} \varphi_i(\xi)\, d\xi$ als neue Unbekannte ein und fügt die Differentialgleichung $\frac{d}{dt}\Phi_i(t) = \varphi_i(t)$ mit der Anfangsbedingung $\Phi_i(t_0) = 0$ hinzu.

  Als Resultat erhält man ein *differential-algebraisches System*.
- Zusätzlich sind noch Anfangsbedingungen erforderlich, und zwar soviele wie man Differentialgleichungen hat. Diese sind aus den Anfangszuständen der linear unabhängigen Energiespeicher (Induktivitäten, Kapazitäten) abzuleiten.

5.3 Weiterführende Aspekte

Maschenstromanalyse Dieses Verfahren ist *dual* zum Knotenpotentialverfahren und geht folgender maßen vor:

- Als System linear unabhängiger Größen wählt man hier einen *fiktiven Maschenstrom für jede Kante des Baumkomplements*. Dies sei der Vektor $\psi(t)$. Jede Baumkomplementkante kann in eindeutiger Weise über den Baum zu einem geschlossenen Pfad ergänzt werden. Durch diesen Pfad fließt der Maschenstrom.
- Der Strom durch einen Baumzweig ergibt sich dann als eine (orientierte) Summe der jeweiligen Maschenströme in denen die Baumkante vorkommt

$$i(t) = A\psi(t).$$

Es gilt wieder $a_{ij} \in \{-1, 0, +1\}$. Man kann zeigen, dass für dieses System von Strömen die Knotenregel automatisch erfüllt ist.

- Nun ist die Maschenregel auf ein System auf ein maximales System unabhängiger Maschen anzuwenden. Hierzu wählt man eine Masche je Komplementzweig. Dies resultiert in
$$A^T u(t) = b$$
wobei in b wieder die (hier:) Spannungsquellen zusammengefasst sind.
- Schließlich beschreibt
$$u(t) = Xi(t) + u_0$$
den Zusammenhang zwischen Spannung und Strom für jeden Zweig des Netzwerkes. Für jeden Widerstand erhält man eine algebraische Gleichung, für jede Induktivität eine Differentialgleichung und für jede Kapazität eine Integralgleichung.
- alle drei Schritte zusammen ergeben:
$$\left.\begin{aligned} i(t) &= A\psi(t) \\ A^T u(t) &= b \\ u(t) &= Xi(t) + u_0 \end{aligned}\right\} \Rightarrow A^T \left(XA\psi(t) + u_0\right) = b$$

Zustandsraumverfahren Mischung aus Knotenpotentialverfahren und Maschenstromanalyse.

- Eliminiere sofort die rein algebraischen Gleichungen.
- Resultiert grundsätzlich in einem System linearer gewöhnlicher Differentialgleichungen erster Ordnung mit konstanten Koeffizienten.
$$\begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & & \vdots \\ a_{n1} & \cdots & a_{nn} \end{bmatrix} \cdot \begin{bmatrix} \frac{dz_1(t)}{dt} \\ \vdots \\ \frac{dz_n(t)}{dt} \end{bmatrix} = \begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix} \quad \xrightarrow{\text{Abkürzung}} \quad A\frac{dz}{dt} = b \tag{5.9}$$

Gesteuerte Quellen

Angenommen, eine (Strom-) Quelle sei durch eine andere Zweigspannung oder einen Zweigstrom gesteuert, also
$$i_g(t) = f\left(z_i\left(t\right)\right)$$

Dann hat (5.9) die allgemeinere Form
$$A\frac{dz}{dt} = f\left(z_i\left(t\right)\right) + b$$

f ist im Allgemeinen eine nichtlineare Funktion.

Berücksichtigung von Spannungsquellen im Knotenpotentialverfahren

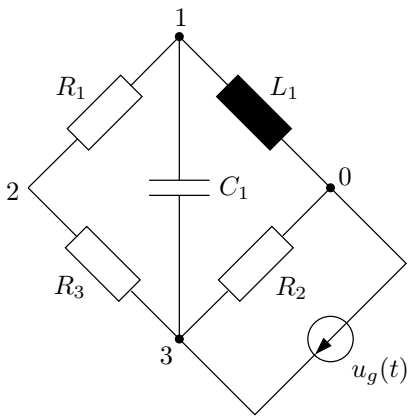
Befindet sich zwischen zwei Knoten eine Spannungsquelle, so ist eines der beiden Knotenpotentiale eine Funktion des Anderen. Betrachte eine Spannungsquelle zwischen Knoten μ und ν. Die Knoten seien $0, \ldots, k-1$ mit Bezugsknoten 0. Es gilt:

$$\varphi_\nu = \varphi_\mu + u_g$$

Unterscheide zwei Fälle:

1. $\mu, \nu \neq 0$ (keiner ist Bezugsknoten):
   - Lasse φ_ν als Unbekannte weg und setze $\varphi_\nu(t) = \varphi_\mu(t) + u_g(t)$.
   - Ersetze Knotenregel in μ durch Knotenregel über Hülle von μ und ν.
2. $\mu = 0$ (oder entsprechend für $\nu = 0$):
   - Lasse φ_ν als Unbekannte weg und setze $\varphi_\nu(t) = \varphi_0 + u_g(t) = u_g(t)$.
   - Streiche die Gleichung für Knoten ν.

Beispiel 5.3. Ersetze in Beispiel 5.1 die Strom- durch eine Spannungsquelle:



Knotenregel:

Knoten 1	$i_{12} + i_{13} - i_{01} = 0$
Knoten 2	$-i_{12} + i_{23} = 0$

Knotenpotentiale:

$$\begin{aligned}
\varphi_3 &= 0 + u_g = u_g \\
u_{12} &= \varphi_1 - \varphi_2 \\
u_{23} &= \varphi_2 - u_g \\
u_{30} &= u_g \\
u_{01} &= -\varphi_1 \\
u_{13} &= \varphi_1 - u_g
\end{aligned}$$

Weiter wie in Beispiel 5.1.

6 Komplexe Wechselstromrechnung

6.1 Widerstandsnetzwerke

Beschreibt man Ströme und Spannungen in einer Schaltung bestehend aus Widerständen, so ergibt sich ein lineares Gleichungssystem. Analog dazu kann man auch den Fluss in einem Rohrleitungssystem betrachten:

$$\text{Spannung} \leftrightarrow \text{Druck}$$
$$\text{Strom} \leftrightarrow \text{Fluss}$$
$$\text{Ohmsches Gesetz} \leftrightarrow \text{Hagen-Poiseuille: } q = \frac{\pi r^4}{8\nu l}\Delta p$$

Hier ist Δp die Druckdifferenz zwischen den beiden Enden des Rohres, r dessen Durchmesser, l dessen Länge und ν die Viskosität des Fluids.

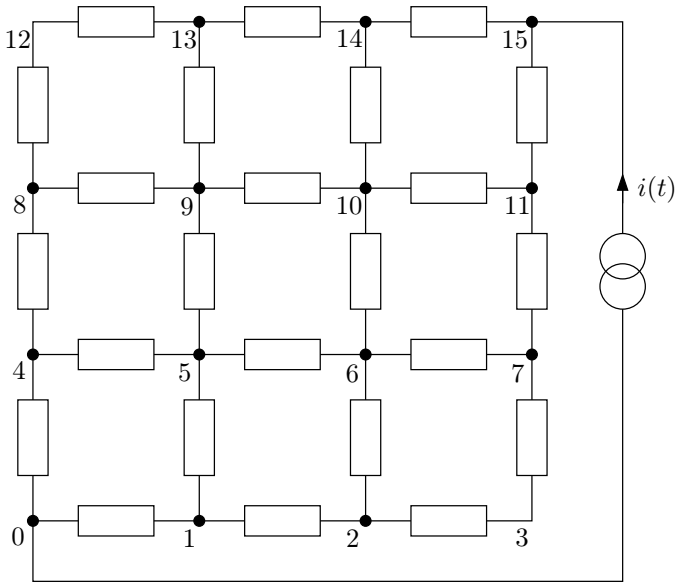


Abbildung 16: Widerstandsnetzwerk

Beispiel 6.1. Betrachte das regelmäßige Netzwerk aus Abbildung 16. Der Einfachheit halber haben alle Widerstände die Größe R.

Knotenpotentialverfahren Unbekannte $\varphi_1, \ldots, \varphi_{15}$; $\varphi_0 = 0$.

Topologie $Nb(i) \subset \{0, \ldots, 15\}$ sei die Menge der Nachbarknoten von Knoten i.

Gleichungen

$$\sum_{j \in Nb(i)} \frac{\varphi_i - \varphi_j}{R} = 0 \qquad 0 < i < 15$$

$$\sum_{j \in Nb(i)} \frac{\varphi_i - \varphi_j}{R} = i(t) \qquad i = 15$$

In Matrixschreibweise:

$$\frac{1}{R} \cdot \left[\begin{array}{ccc|ccc|ccc|ccc}
3 & -1 & & & -1 & & & & & & & \\
-1 & 3 & -1 & & & -1 & & & & & & \\
 & -1 & 2 & & & & -1 & & & & & \\
\hline
 & & & 3 & -1 & & & -1 & & & & \\
-1 & & & -1 & 4 & -1 & & & -1 & & & \\
 & -1 & & & -1 & 4 & -1 & & & -1 & & \\
 & & -1 & & & -1 & 3 & & & & -1 & \\
\hline
 & & & -1 & & & & 3 & -1 & & & \\
 & & & & -1 & & & -1 & 4 & -1 & & \\
 & & & & & -1 & & & -1 & 4 & -1 & \\
 & & & & & & -1 & & & -1 & 3 & \\
\hline
 & & & & & & & -1 & & & 2 & -1 \\
 & & & & & & & & -1 & & -1 & 3 & -1 \\
 & & & & & & & & & -1 & & -1 & 3 & -1 \\
 & & & & & & & & & & -1 & & -1 & 2
\end{array}\right] \cdot \begin{bmatrix} \varphi_1 \\ \\ \\ \\ \\ \\ \vdots \\ \\ \\ \\ \\ \\ \varphi_{15} \end{bmatrix} = \begin{bmatrix} 0 \\ \\ \\ \\ \\ \\ \vdots \\ \\ \\ \\ \\ \\ 0 \\ i(t) \end{bmatrix}$$

Die Matrix hat Bandgestalt. Pro Zeile sind maximal 5 Einträge $\neq 0$, es handelt sich also um eine dünnbesetzte Matrix. □

Nun zeigen wir wie das im Beispiel hergeleitete lineare Gleichungssystem mit dem Programm `octave` berechnet werden kann.

Octave[1] ist eine Skriptsprache speziell für numerische Anwendungen, sozusagen ein Taschenrechner „de Luxe“. Octave ist freie Software unter der GNU-Lizenz und ist weitgehend kompatibel zu MatLab, einem kommerziellem Produkt.

Das Listing in Abbildung 17 zeigt wie man mit Octave sehr einfach ein lineares Gleichungssystem lösen kann.

6.2 Analyse des Parallelschwingkreises

Zuerst analysieren wir noch das Verhalten eines einfachen Netzwerkes. Betrachte den gedämpften Parallelschwingkreis.

[1] `www.octave.org`

```
# (1) Definiere Werte fuer R und I
R = 1E3
I = 0.1

# (2) Definiere die Matrix A
A = [ 3,-1,0,   0,-1,0,0,   0,0,0,0,    0,0,0,0;
     -1,3,-1,   0,0,-1,0,   0,0,0,0,    0,0,0,0;
      0,-1,2,   0,0,0,-1,   0,0,0,0,    0,0,0,0;
      0,0,0,    3,-1,0,0,   -1,0,0,0,   0,0,0,0;
      -1,0,0,   -1,4,-1,0,  0,-1,0,0,   0,0,0,0;
      0,-1,0,   0,-1,4,-1,  0,0,-1,0,   0,0,0,0;
      0,0,-1,   0,0,-1,3,   0,0,0,-1,   0,0,0,0;
      0,0,0,    -1,0,0,0,   3,-1,0,0,   -1,0,0,0;
      0,0,0,    0,-1,0,0,   -1,4,-1,0,  0,-1,0,0;
      0,0,0,    0,0,-1,0,   0,-1,4,-1,  0,0,-1,0;
      0,0,0,    0,0,0,-1,   0,0,-1,3,   0,0,0,-1;
      0,0,0,    0,0,0,0,    -1,0,0,0,   2,-1,0,0;
      0,0,0,    0,0,0,0,    0,-1,0,0,   -1,4,-1,0;
      0,0,0,    0,0,0,0,    0,0,-1,0,   0,-1,4,-1;
      0,0,0,    0,0,0,0,    0,0,0,-1,   0,0,-1,2] * 1/R

# (3) und die rechte Seite (Spaltenvektor)
b = [0;0;0;0;0;0;0;0;0;0;0;0;0;0;I]

# (4) loese LGS
x = A\b

# (5) berechne effektiven Widerstand
Reff = x(15)/I
```

Abbildung 17: Lösung eines linearen Gleichungssystems mit Octave.

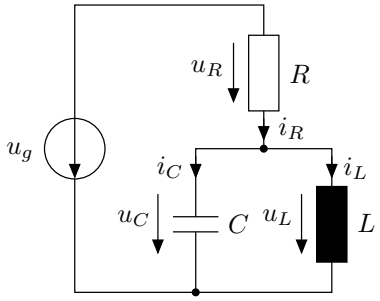


Abbildung 18: gedämpfter Parallelschwingkreis

Maschenregel für die Elementarmaschen:

$$u_R + u_C - u_g = 0 \qquad (a)$$
$$u_C - u_L = 0 \qquad (b)$$

Knotenregel für zwei Knoten:

$$i_R - i_C - i_L = 0 \quad \text{(Knoten 1)} \qquad (c)$$
$$i_g - i_R = 0 \qquad (*)$$

Elemente:

$$u_R = R i_R \qquad (d)$$
$$i_C = C \frac{du_C}{dt} \qquad (e)$$
$$u_L = L \frac{di_L}{dt} \qquad (f)$$

Zustandsgrößen: u_C, i_L

$$u_R \overset{(d)}{=} R i_R \overset{(c)}{=} R\,(i_C + i_L) \overset{(e)}{=} R\left(C \frac{du_C}{dt} + i_L\right) = RC \frac{du_C}{dt} + R i_L$$

Eingesetzt in (a) erhält man:

$$RC \frac{du_C}{dt} + R i_L + u_C = u_g(t) \qquad (6.1a)$$

(f) und (b) liefern:

$$u_C = L \frac{di_L}{dt} \qquad (6.1b)$$

```
x0 = [0; 0];                          # Anfangsbedingungen
t = linspace(0.0, 1.0, 1000)';# von 0 bis 1 in 1000 Schritten
x = lsode ("podef", x0, t);    # loese die DGL "podef"
plot (t,x);                           # Zeige das Ergebnis
```

Abbildung 19: Octave Code zur Lösung eines Systems gewöhnlicher Differentialgleichungen.

Umordnen liefert:

$$\begin{aligned} \frac{du_C}{dt} &= -\frac{1}{RC}u_C - \frac{1}{C}i_L + \frac{1}{RC}u_g(t) \\ \frac{di_L}{dt} &= \frac{1}{L}u_C \end{aligned} \tag{6.2}$$

also ein lineares System gewöhnlicher Differentialgleichung, das noch um die Anfangsbedingungen

$$u_C(t_0) = u_{C,0}, \qquad i_L(t_0) = i_{L,0}$$

zu ergänzen ist. Die gewöhnliche Differentialgleichung hat die Standardform $\frac{dz(t)}{dt} = f(z(t), t)$ bzw. hier speziell:

$$\frac{dz(t)}{dt} = Az(t) + b(t) \quad \text{mit} \quad A = \begin{bmatrix} -\frac{1}{RC} & -\frac{1}{C} \\ \frac{1}{L} & 0 \end{bmatrix} \tag{6.3}$$

Im folgenden zeigen wir wie die Lösung dieses Differentialgleichungssystems mit dem Programm `octave` berechnet werden kann. Später werden wir verschiedene Methoden zur numerischen Lösung von Differentialgleichungssystemen kennenlernen.

Die rechte Seite $f(x(t), t)$ der Differentialgleichung muss wird in der Funktion `podef` spezifiziert. Für die numerische Lösung müssen wir konkrete Werte für R, L und C einsetzen. Wir wählen: $R = 100\,\Omega$, $C = 10^{-4}\,F$, $L = 0.01 \cdot 4R^2C\,H$, $\Delta t = \frac{1}{1000}\,s$, $T = [0, 1]$. Die vollständige Funktion `podef` zeigt die Abbildung 20.

Wir betrachten verschiedene Anregungen $u_g(t)$ und Werte für die Induktivität L. Zunächst sei $u_g(t)$ eine Rechteckfunktion. Dies ist die letzte Zeile im Listing 20 vor `endfunction`.

Es entsteht eine gedämpfte Schwingung (Abbildung 21).

Für $L = 2 \cdot 4R^2C$ und $L = 1 \cdot 4R^2C$ tritt hingegen keine Schwingung mehr auf (Abbildung 22 und Abbildung 23).

Nun betrachten wir eine harmonische Anregung (Abbildung 24): Nach einer unregelmäßigen Einschwingphase sind u_c und i_c harmonisch, mit gleicher Frequenz, aber unterschiedlicher Amplitude und Phase.

```
function xdot = podef (x,t)

  R = 1E2;
  C = 1E-4;
  L = 4*R*R*C*0.01;

  xdot(1) = -1.0/(R*C)*x(1) - 1.0/C*x(2);
  xdot(2) = 1.0/L*x(1);

  # Quelle hinzufügen:
  xdot(1) = xdot(1) + 5*(1-sign(rem(t,0.5)/0.5 - 0.5));

endfunction
```

Abbildung 20: Definition der rechten Seite eines Differentialgleichungssystems.

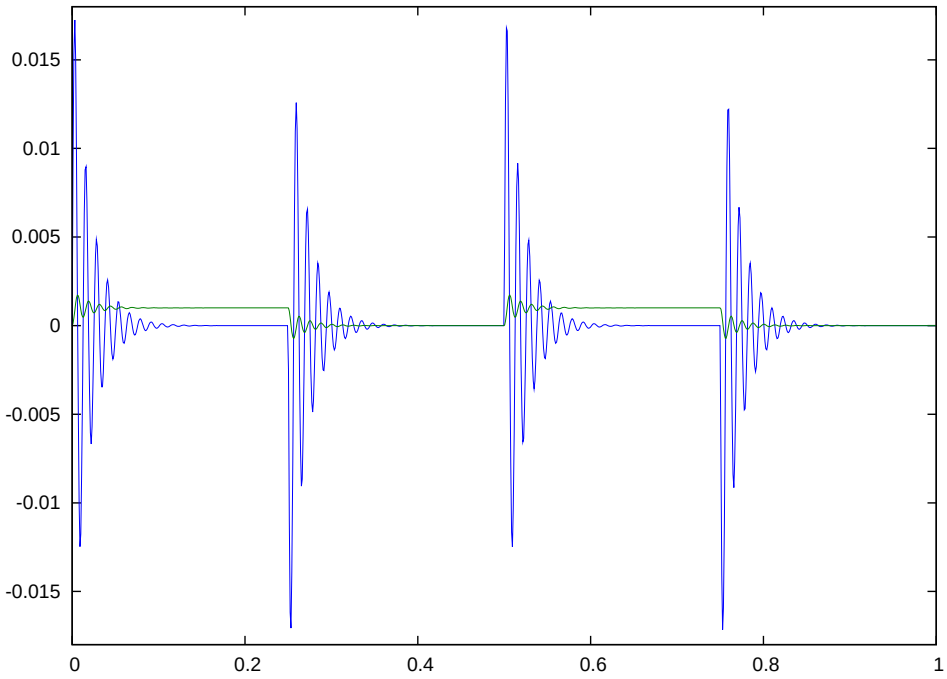


Abbildung 21: Rechteckanregung bei $L = 0.01 \cdot 4R^2C$.

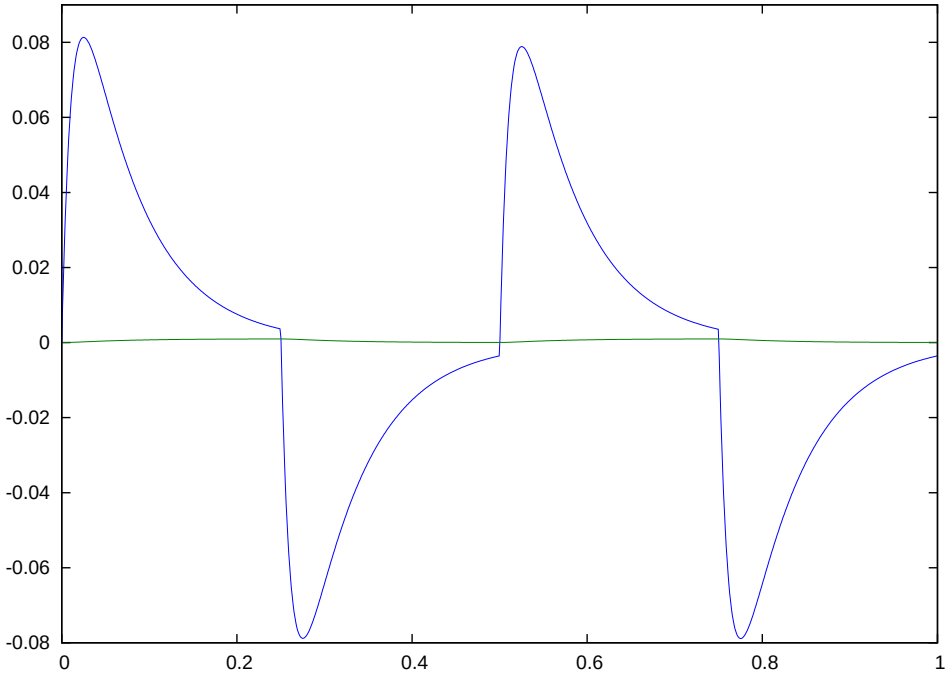


Abbildung 22: Rechteckanregung bei $L = 2 \cdot 4R^2C$.

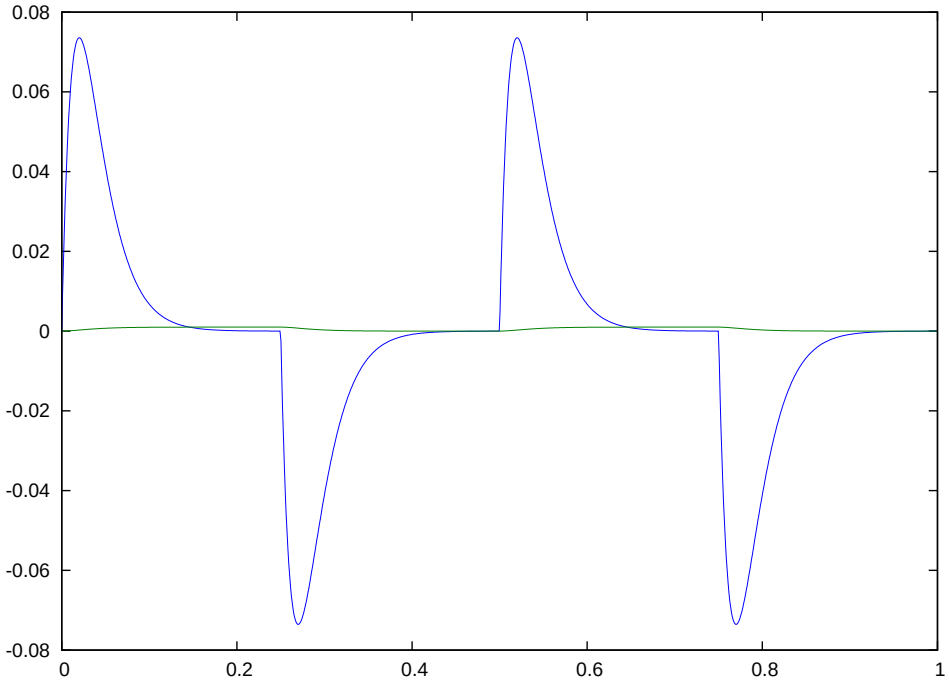


Abbildung 23: Rechteckanregung bei $L = 1 \cdot 4R^2C$.

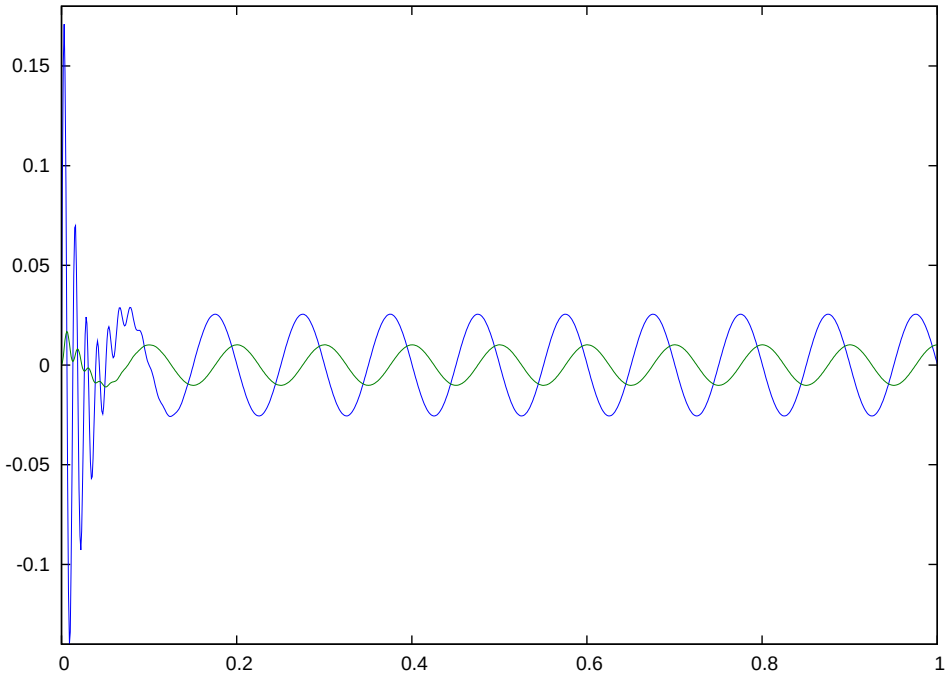


Abbildung 24: Harmonische Anregung bei $L = 0.01 \cdot 4R^2C$.

Eigenwertanalyse Die Ausbildung einer Schwingung für genügend kleine Werte von L kann man durch eine Analyse der Eigenwerte der Matrix A erklären:

$$\mathrm{Det}\,(A - \lambda I) = \mathrm{Det}\begin{bmatrix} -\frac{1}{RC} - \lambda & -\frac{1}{C} \\ \frac{1}{L} & -\lambda \end{bmatrix} = -\lambda\left(-\frac{1}{RC} - \lambda\right) + \frac{1}{LC} = \lambda^2 + \frac{1}{RC}\lambda + \frac{1}{LC} = 0$$

$$\Leftrightarrow \lambda = -\frac{1}{2RC} \pm \sqrt{\frac{1}{4R^2C^2} - \frac{1}{LC}} = -\frac{1}{2RC} \pm \sqrt{\frac{L^2 - 4R^2CL}{4R^2C^2L^2}} = -\frac{1}{2RC} \pm \frac{\sqrt{L^2 - 4R^2LC}}{2RCL}$$

$$L^2 - 4R^2LC \begin{cases} \geq 0 & \text{reelle Eigenwerte} \rightarrow \text{e-Funktion} \\ < 0 & \text{komplexe Eigenwerte} \rightarrow \text{Schwingung} \end{cases}$$

$$\text{Grenze: } L^2 - 4R^2LC \geq 0 \Leftrightarrow L \geq 4R^2C$$

Später werden wir sehen, dass die allgemeine Lösung von $\frac{dz}{dt} = Az + b$ die Berechnung der Eigenwerte von A erfordert.

6.3 Komplexe Wechselstromrechnung

Harmonische Erregung mit *einer* Frequenz führt nach dem Einschwingvorgang zu einer harmonischen Antwort. RLC-Netzwerke im eingeschwungenen Zustand lassen sich also

ebenfalls mit linearen Gleichungssystemen analysieren.

Wir betrachten noch einmal das RC-Glied aus Abschnitt 5.1:

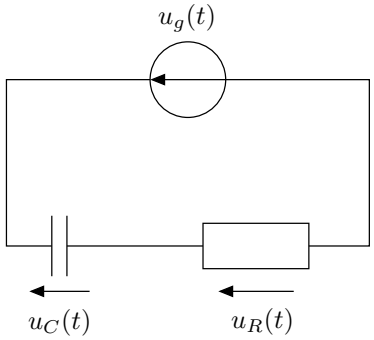


Abbildung 25: RC-Glied

$$RC\frac{du_C(t)}{dt} + u_C(t) = u_g(t) \qquad u_C(0) = u_0 \tag{6.4}$$

$u_g(t)$ sei harmonisch, das heißt

$$u_g(t) = U_g \cos(\omega t + \alpha) \tag{6.5}$$

Nach der Theorie der gewöhnlichen Differentialgleichungen ist dann auch $u_C(t)$ harmonisch, wodurch man folgenden Ansatz erhält:

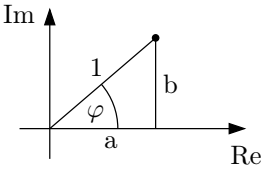
$$u_C(t) = U_C \cos(\omega t + \beta) \tag{6.6}$$

Gesucht sind die Amplitude U_C und die Phase β. Lösungsmöglichkeiten:

1. Setze (6.6) in die Differentialgleichung (6.4) ein und wende Additionstheoreme an.
2. Geschickter ist der Übergang zu komplexen Zahlen.

Eulersche Beziehung:

$$e^{i\varphi} = \cos\varphi + i \cdot \sin\varphi$$



Damit kann man schreiben:

$$\frac{1}{2}\left(e^{i\varphi} + e^{-i\varphi}\right) = \frac{1}{2}\left(\cos\varphi + i \cdot \sin\varphi + \cos(-\varphi) + i \cdot \sin(-\varphi)\right) = \cos\varphi$$

Also mit $\varphi = \omega t + \alpha$:

$$\begin{aligned} u_g(t) &= U_g \frac{1}{2}\left(e^{i(\omega t+\alpha)} + e^{-i(\omega t+\alpha)}\right) \\ &= \frac{1}{2}\left(U_g e^{i\alpha} e^{i\omega t} + U_g e^{-i\alpha} e^{-i\omega t}\right) \\ &= \frac{1}{2}\left(\underline{U}_g e^{i\omega t} + \underline{U}_g^* e^{-i\omega t}\right) \end{aligned}$$

mit einer komplexen Zahl $\underline{U}_g$. Ist die komplexe Zahl $\underline{U}_g$ gegeben, so folgt

$$\underline{U}_g = U_g e^{i\alpha} = U_g \cos\alpha + i \cdot U_g \sin\alpha$$

Aus

$$\begin{aligned} U_g \cos\alpha &= \mathrm{Re}\left(\underline{U}_g\right) \\ U_g \sin\alpha &= \mathrm{Im}\left(\underline{U}_g\right) \end{aligned}$$

folgt:

$$U_g^2\left(\cos^2\alpha + \sin^2\alpha\right) = \left[\mathrm{Re}\left(\underline{U}_g\right)\right]^2 + \left[\mathrm{Im}\left(\underline{U}_g\right)\right]^2$$

$$\Rightarrow U_g = \sqrt{\left[\mathrm{Re}\left(\underline{U}_g\right)\right]^2 + \left[\mathrm{Im}\left(\underline{U}_g\right)\right]^2} \quad (6.7a)$$

$$\frac{\sin\alpha}{\cos\alpha} = \tan\alpha = \frac{\mathrm{Im}\left(\underline{U}_g\right)}{\mathrm{Re}\left(\underline{U}_g\right)} \Rightarrow \alpha = \arctan\frac{\mathrm{Im}\left(\underline{U}_g\right)}{\mathrm{Re}\left(\underline{U}_g\right)}. \quad (6.7b)$$

Hier ist natürlich $\mathrm{Re}\left(\underline{U}_g\right) \neq 0$ Voraussetzung. Ist $\mathrm{Re}\left(\underline{U}_g\right) = 0$ dann ist wegen $\cos\alpha = 0$ entweder $\alpha = \pi/2$ oder $\alpha = -\pi/2$ je nach Vorzeichen vom $\mathrm{Im}\left(\underline{U}_g\right)/U_g$.

Ebenso setzt man an:

$$u_C(t) = \frac{1}{2}\left(\underline{U}_C e^{i\omega t} + \underline{U}_C^* e^{-i\omega t}\right) \quad \text{mit } \underline{U}_C = U_C e^{i\beta}$$

In die Differentialgleichung (6.4) einsetzen:

$$RC\frac{1}{2}\left[\underline{U}_C i\omega e^{i\omega t} + \underline{U}_C^*(-i\omega)e^{-i\omega t}\right] + \frac{1}{2}\left[\underline{U}_C e^{i\omega t} + \underline{U}_C^* e^{-i\omega t}\right] = \frac{1}{2}\left[\underline{U}_g e^{i\omega t} + \underline{U}_g^* e^{-i\omega t}\right]$$

$$\Leftrightarrow \frac{1}{2}\underline{U}_C\left(1 + i\omega RC\right)e^{i\omega t} + \frac{1}{2}\underline{U}_C^*\left(1 - i\omega RC\right)e^{-i\omega t} = \frac{1}{2}\underline{U}_g e^{i\omega t} + \frac{1}{2}\underline{U}_g^* e^{-i\omega t}$$

$e^{i\omega t}$ und $e^{-i\omega t}$ sind unabhängig. Wir beobachten, dass das Bilden der Ableitung in diesem Fall zu einer einfachen Multiplikation wird.

Koeffizientenvergleich ergibt

$$\begin{aligned} \frac{1}{2}\underline{U}_C\left(1 + i\omega RC\right) &= \frac{1}{2}\underline{U}_g \\ \frac{1}{2}\underline{U}_C^*\left(1 - i\omega RC\right) &= \frac{1}{2}\underline{U}_g^* \end{aligned}$$

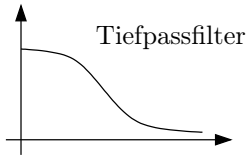
Diese beiden Gleichungen sind aber linear abhängig. Es genügt also, eine der beiden zu erfüllen. Somit erhalten wir

$$\underline{U}_C = \left(\frac{1}{1+i\omega RC}\right)\underline{U}_g \tag{6.8}$$

Daraus lassen sich mit Hilfe von (6.7) Amplitude und Phasenlage bestimmen.

Übertragungsverhalten Abhängigkeit von ω und Frequenzgang

$$\left|\frac{1}{1+i\omega RC}\right| \to \begin{cases} 1 & \text{für } \omega \to 0 \\ 0 & \text{für } \omega \to \infty \end{cases}$$



Verallgemeinerung Wir können jeden harmonischen Strom / jede harmonische Spannung schreiben als

$$i(t) = \frac{1}{2}\left(\underline{I}e^{i\omega t} + \underline{I}^* e^{-i\omega t}\right),$$
$$u(t) = \frac{1}{2}\left(\underline{U}e^{i\omega t} + \underline{U}^* e^{-i\omega t}\right).$$

Die Strom-Spannungsbeziehungen der Netzwerkelemente lauten bei harmonischen Strömen bzw. Spannungen

Widerstand	$u = Ri$	$\underline{U} = R\underline{I}$
Kondensator	$i = C\frac{du}{dt}$	$\underline{I} = i\omega C\underline{U}$
Spule	$u = L\frac{di}{dt}$	$\underline{U} = i\omega L\underline{I}$

Spulen und Kondensatoren können als komplexe Widerstände interpretiert werden!
Die Kirchhoffschen Regeln gelten auch für die komplexwertigen Ströme und Spannungen. Auch die Netzwerkanalyse, z.B. mittels Knotenpotentialverfahren, kann analog durchgeführt werden.

Beispiel 6.2 (Anwendung auf den Schwingkreis)**.** In Beispiel 6.2 war $\underline{U}_g$ gegeben, gesucht waren $\underline{U}_C$ und $\underline{I}_L$. Die beiden Maschenregeln ergeben im Komplexen die Gleichungen

$$RCi\omega\underline{U}_C + R\underline{I}_L + \underline{U}_C = \underline{U}_g, \tag{6.9a}$$
$$\underline{U}_C = Li\omega\underline{I}_L. \tag{6.9b}$$

Umsortieren ergibt ein lineares Gleichungssystem mit komplexwertigen Koeffizienten:

$$\begin{bmatrix} 1+i\omega RC & R \\ -1 & i\omega L \end{bmatrix} \cdot \begin{bmatrix} \underline{U}_C \\ \underline{I}_L \end{bmatrix} = \begin{bmatrix} \underline{U}_g \\ 0 \end{bmatrix}$$

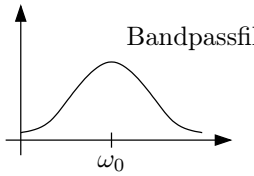
Lösung:

$$\begin{bmatrix} \underline{U}_C \\ \underline{I}_L \end{bmatrix} = \frac{1}{(1+i\omega RC)i\omega L + R} \begin{bmatrix} i\omega L & -R \\ 1 & 1+i\omega RC \end{bmatrix} \cdot \begin{bmatrix} \underline{U}_g \\ 0 \end{bmatrix}$$

also etwa

$$\underline{U}_C = \underbrace{\frac{i\omega L}{R(1-\omega^2 LC) + i\omega L}}_{=1 \text{ für } \omega_0 = \frac{1}{\sqrt{LC}}} \underline{U}_g$$

ω_0: *Resonanzfrequenz*



□

Somit führen RLC-Netzwerke mit harmonischer Erregung auf lineare Gleichungssysteme mit komplexen Koeffizienten. Wir wollen uns nun der numerischen Lösung von linearen Gleichungssystemen zuwenden.

7 Direkte Lösung vollbesetzter linearer Gleichungssysteme

Es geht um die Lösung einer Gleichung der Form

$$Ax = b \tag{7.1}$$

mit $A \in \mathbb{K}^{n\times n}$, $x, b \in \mathbb{K}^n$, $\mathbb{K} = \mathbb{R}$ oder $\mathbb{C}$.
Wir nehmen an:

- A ist quadratisch.
- A ist regulär, d.h. $Ax = b$ hat für jedes b genau eine Lösung.
- A besitze im wesentlichen Elemente $\neq 0$ (für dünnbesetzte Matrizen s. Kapitel 8).

7.1 Gauß-Elimination und LU-Zerlegung

Das lineare Gleichungssystem wird im Rechner mit Maschinenzahlen gelöst. Hier haben wir das folgende Ergebnis.

Konditionierung der Aufgabe (7.1): Auswirkung von Störungen in den Eingaben A und b auf das Ergebnis bei exakter Arithmetik.
Sei $Ax = b$ und $(A + \delta A)(x + \delta x) = (b + \delta b)$ mit $\|\delta A\| < \frac{1}{\|A^{-1}\|}$. Dann gilt:

$$\frac{\|\delta x\|}{\|x\|} \leq \frac{\kappa(A)}{1 - \kappa(A)\frac{\|\delta A\|}{\|A\|}} \left(\frac{\|\delta b\|}{\|b\|} + \frac{\|\delta A\|}{\|A\|} \right)$$

mit einer Vektornorm $\|.\|$ mit verträglicher Matrixnorm sowie der *Konditionszahl* $\kappa(A) = \|A\| \cdot \left\|A^{-1}\right\|$, siehe (Ran06).

Beispiel 7.1. Die Kondition sei $\kappa(A) \approx 10^s$ und der Eingangsfehler betrage

$$\frac{\|\delta A\|}{\|A\|} \approx \frac{\|\delta b\|}{\|b\|} \approx 10^{-k}.$$

Für den Fehler im Ergebnis ergibt sich dann

$$\frac{\|\delta x\|}{\|x\|} \lessapprox 10^{s-k}$$

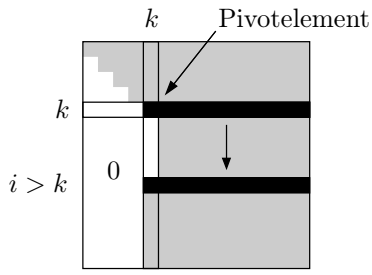
□

Gauß-Elimination

Transformation des linearen Gleichungssystems auf obere Dreiecksgestalt. Elimination von a_{ik}: Subtrahiere das $\frac{a_{ik}}{a_{kk}}$-fache der k-ten Zeile von der i-ten Zeile.

Algorithmus 1 Gauß-Elimination

for $k = 0$ to $n-2$ **do**
  for $i = k+1$ to $n-1$ **do**
    $l = \frac{a_{i,k}}{a_{k,k}}$
    for $j = k+1$ to $n-1$ **do**
      $a_{i,j} = a_{i,j} - l \cdot a_{k,j}$
    end for
    $b_i = b_i - l \cdot b_k$
  end for
end for



Innerste Schleife ist eine *axpy-Operation* ($ax + y$, x, y Vektoren, a Skalar). Unter *axpy* versteht man die Operation $z = ax + y$. Die Bezeichnung stammt aus der Softwarebibliothek LINPACK (dort gibt es die Varianten *saxpy* und *daxpy* für single- bzw. double Genauigkeit).

LU-Zerlegung

Die Elimination der a_{ik} kann als Matrixmultiplikation von links mit der Matrix

$$L'_{ik} = (I - Q_{ik}) \qquad (Q_{ik})_{\alpha,\beta} = \begin{cases} \frac{a_{ik}}{a_{kk}} & \text{falls } \alpha = i, \beta = k \\ 0 & \text{sonst} \end{cases}$$

geschrieben werden. Damit gilt:

$$\underbrace{L'_{n-1,n-2} \cdots L'_{2,1} L'_{1,0}}_{L'} A = U$$

Es zeigt sich, dass $(L)'^{-1} =: L$ eine untere Dreiecksmatrix ist mit

$$(L)_{ik} = \begin{cases} \frac{a_{ik}}{a_{kk}} & i > k \quad \text{(die Zahlen aus dem Gauß-Verfahren!)} \\ 1 & i = k \\ 0 & i < k \end{cases},$$

d. h. $L'^{-1} = I + Q_{ik}$. Somit hat man die Darstellung $A = LU$. Aus Effizienzgründen speichert man L und U *in place*, das heißt A wird überschrieben.

Algorithmus 2 LU-Zerlegung

```
for k = 0 to n − 2 do
  for i = k + 1 to n − 1 do
    l_{i,k} = a_{i,k} / a_{k,k}   /* l_{i,k} nutzt Platz von a_{i,k} */
    for j = k + 1 to n − 1 do
      a_{i,j} = a_{i,j} − l_{i,k} · a_{k,j}
    end for
  end for
end for
```

Das lineare Gleichunggsystem löst man damit folgendermaßen:

1. Berechne L und U mit Algorithmus 2.
2. Löse $Ly = b$ durch "vorwärts Einsetzen".
3. Löse $Ux = y$ durch "rückwärts Einsetzen".

Aufwand:

- Algorithmus 2 benötigt $\frac{2}{3}n^3 + \mathrm{O}(n^2)$ Operationen.
- Vorwärts und rückwärts Einsetzen benötigen je n^2 Operationen.

Da die rechte Seite bei der Zerlegung nicht verändert wird, ist das Verfahren günstig, falls (7.1) für mehrere rechte Seiten gelöst werden soll (z.B. Invertierung einer Matrix).

Pivotisierung: Was macht man, wenn $a_{kk} = 0$ oder sehr klein ist?

- $a_{kk} = 0$ kommt bei symmetrisch positiv definiten Matrizen nicht vor:

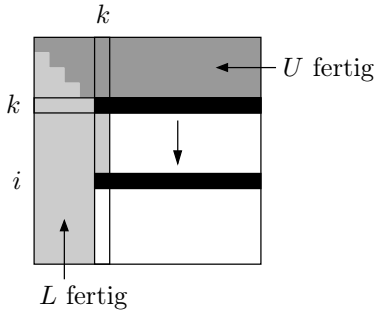
$$\text{s.p.d.:} \quad A = A^T \wedge x^T Ax > 0 \quad \forall x \neq 0$$

- Totales Pivoting:
  1. Finde betragsgrößtes $a_{i,j}$ mit $i, j \in \{k, \ldots, n-1\}$.
  2. Tausche Zeile i mit Zeile k und Spalte j mit Spalte k.
- Spaltenpivoting:
  1. Finde betragsgrößtes $a_{i,k}$ mit $i \in \{k, \ldots, n-1\}$.
  2. Tausche Zeile i mit Zeile k.
- Anstatt Daten zu bewegen kann man auch die Indizes permutieren.

7.2 Performance, ijk-Formen

Frage: Wie verhält sich die LU-Zerlegung auf modernen Rechnerarchitekturen? Hierzu sind insbesondere die Datenzugriffe zu betrachten.

Algorithmus 2 nennt man wegen der Reihenfolge der Schleifen *kij-Form*.



- Innerste (j-Schleife) addiert Vielfaches eines Vektors (k-te Zeile) auf einen Vektor (i-te Zeile) → axpy-Operation.
- Zeilenweise Speicherung von A (wie z.B. in C) führt auf konsekutiven Speicherzugriff.
- Spaltenweise Speicherung von A (Fortran) führt auf nicht-konsekutive Speicherzugriffe ⇒ ineffizient auf modernen Rechnern, die ganze Cachelines laden.
- axpy-Operationen stehen auf Vektorrechnern als Befehl zur Verfügung.
- Die mittlere Länge der Vektoren ist $\frac{2}{3}n + O(1)$).
- Sei $A(i_1 : i_2, j_1 : j_2)$ die Untermatrix von A mit $i_1 \leq i \leq i_2$ sowie $j_1 \leq j \leq j_2$ dann kann man die beiden innersten Schleifen auch schreiben als

  $$A(k{+}1 : n{-}1, k{+}1 : n{-}1) = A(k{+}1 : n{-}1, k{+}1 : n{-}1) - l(k{+}1 : n{-}1)A(k, k{+}1 : n{-}1).$$

  Der zweite Term ist ein Zeilenvektor mal einen Spaltenvektor und wird auch als „äußeres Produkt“ bezeichnet.

Spaltenorientierte Variante: kji-Form

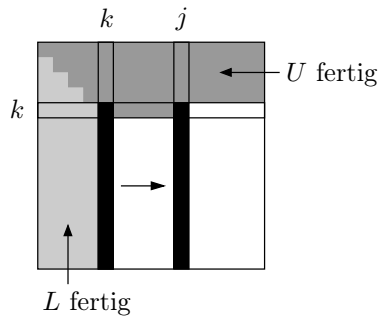
Insgesamt gibt es $3! = 6$ verschiedene Formen der Gauß-Elimination.

Die ikj-Form

- Diese Variante ist wieder zeilenorientiert. Die entsprechende spaltenorientierte Variante ist die *jki-Form*.
- Die innerste Schleife ist wieder eine axpy-Operation.
- Die Variante ist günstig für dünnbesetzte Matrizen.

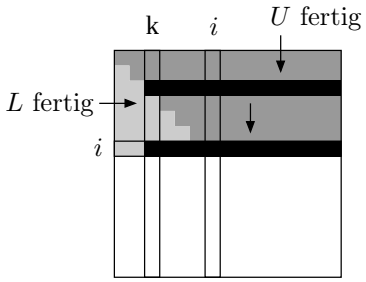
Algorithmus 3 LU-Zerlegung, spaltenorientiert

```
for k = 0 to n − 2 do
  for s = k + 1 to n − 1 do
    l_{s,k} = a_{s,k} / a_{k,k}
  end for
  for j = k + 1 to n − 1 do
    for i = k + 1 to n − 1 do
      a_{i,j} = a_{i,j} − l_{i,k} · a_{k,j}
    end for
  end for
end for
```



Algorithmus 4 LU-Zerlegung, ikj-Form

```
for i = 1 to n − 1 do
  for k = 0 to i − 1 do
    l_{i,k} = a_{i,k} / a_{k,k}
    for j = k + 1 to n − 1 do
      a_{i,j} = a_{i,j} − l_{i,k} · a_{k,j}
    end for
  end for
end for
```



- Andere Bezeichnung: *left* oder *backward-looking*, da immer die aktuelle Zeile mit allen vorherigen eliminiert wird. Noch eine andere Bezeichnung lautet *delayed update*.

Die ijk-Form

Variante der ikj-Form

- Die innerste Schleife ist ein Skalarprodukt.
- Allerdings erfolgt der Zugriff spaltenweise und zeilenweise ⇒ einer der beiden "Vektoren" wird nicht konsekutiv gelesen.

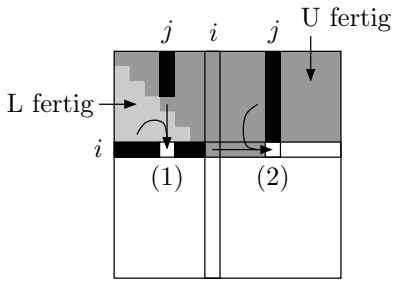
Blockalgorithmen

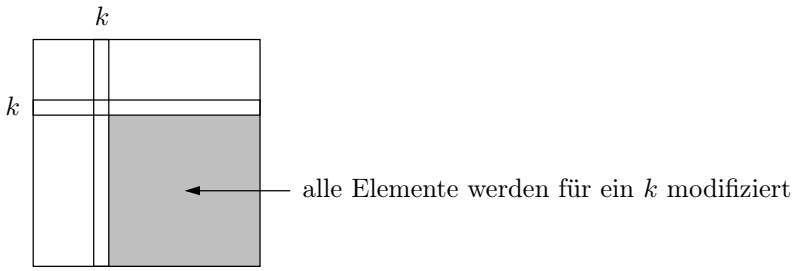
Obige Verfahren nutzen axpy-Operation oder Skalarprodukt in der innersten Schleife. Bisher wurde die Speicherbandbreite voll ausgenutzt, aber nicht auf die Wiederverwendung von Daten geachtet. Übersteigt die Matrix eine bestimmte Größe, so können die zu bearbeitenden Daten nicht im Cache gehalten werden.

Beispiel 7.2.

Algorithmus 5 LU-Zerlegung, ijk-Form

```
for i = 1 to n − 1 do
  for j = 1 to i do
    l_{i,j−1} = a_{i,j−1} / a_{j−1,j−1}
    for k = 0 to j − 1 do
      a_{i,j} = a_{i,j} − l_{i,k} · a_{k,j}
    end for
  end for
  for j = i + 1 to n − 1 do
    for k = 0 to i − 1 do
      a_{i,j} = a_{i,j} − l_{i,k} · a_{k,j}
    end for
  end for
end for
```





□

LU-Zerlegung: $O(n^3)$ Operationen auf n^2 Daten ⇒ Daten werden mehrfach verwendet. Um den Cache effizient ausnutzen zu können, wird eine gegebene $n \times n$-Matrix A mit einer 2×2-Blockstruktur versehen:

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \quad \text{mit } A_{11} \in \mathbb{R}^{b \times b}, A_{12} \in \mathbb{R}^{b \times (n-b)}, A_{21} \in \mathbb{R}^{(n-b) \times b}, A_{22} \in \mathbb{R}^{(n-b) \times (n-b)}$$

Nach Elimination der ersten b Spalten mittels jki- oder kji-Form der LU-Zerlegung gilt die Gleichung

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} = \begin{bmatrix} L_{11} & 0 \\ L_{21} & I \end{bmatrix} \cdot \begin{bmatrix} U_{11} & U_{12} \\ 0 & U_{22} \end{bmatrix}$$

Verfahren:

1. Löse

$$\begin{aligned} L_{11}U_{11} &= A_{11} && LU\text{-Zerlegung von } A_{11}, \\ L_{21}U_{11} &= A_{21} && U_{11} \text{ bekannte obere Dreiecksmatrix.} \end{aligned}$$

   Die zweite Gleichung kann man auch als $U_{11}^T L_{21}^T = A_{21}^T$ schreiben, d.h. es ist eine untere Dreiecksmatrix mit vielen rechten Seiten zu lösen.

2. Berechne U_{12} durch Lösen von $L_{11}U_{12} = A_{12}$ für viele rechte Seiten. Dies ist effizient möglich, da L_{11} eine untere Dreiecksmatrix ist.

3. Berechne U_{22} aus $L_{21} \cdot U_{12} + U_{22} = A_{22} \Rightarrow U_{22} = A_{22} - L_{21} \cdot U_{12}$. Das Matrixprodukt $L_{21} \cdot U_{12}$ wird wiederum blockweise berechnet.

4. Fahre rekursiv mit U_{22} fort.

Cache-effiziente Berechnung von Matrixprodukten: A, B und C sind $p \times p$-Blockmatrizen mit Blöcken der Größe $b \times b$ (ggf. bis auf letzte Zeile und Spalte)

$$\begin{bmatrix} A_{11} & \dots & A_{1p} \\ \vdots & & \vdots \\ A_{p1} & \dots & A_{pp} \end{bmatrix} \cdot \begin{bmatrix} B_{11} & \dots & B_{1p} \\ \vdots & & \vdots \\ B_{p1} & \dots & B_{pp} \end{bmatrix} = \begin{bmatrix} C_{11} & \dots & C_{1p} \\ \vdots & & \vdots \\ C_{p1} & \dots & C_{pp} \end{bmatrix}$$

$$C_{ij} = \sum_{k=1}^{p} \underbrace{A_{ik} \cdot B_{kj}}_{O(b^3) \text{ Operationen auf } b^2 \text{ Daten im Cache}}$$

Bemerkung 7.3.

- Für beliebige Matrixgrößen, variable Blockgrößen entsprechend L1/L2-Cache und Pivoting wird selbst die LU-Zerlegung ganz schön kompliziert!

- BLAS (*Basic Linear Algebra Subroutines*) enthält eine Menge hochoptimierter Funktionen für lineare Algebra

  1. Vektoroperationen (axpy, Skalarprodukt) $\rightarrow O(n)$
  2. Matrix-Vektor-Operationen $(A \cdot x) \rightarrow O(n^2)$
  3. Matrix-Matrix-Operationen $(A \cdot A) \rightarrow O(n^3)$

- LAPACK-Software implementiert diverse Zerlegungen, Eigenwertlöser usw. auf Basis von BLAS Level 3 Blockalgorithmen.

8 Direkte Lösung dünnbesetzter linearer Gleichungssysteme

8.1 Was ist das Problem?

Viele Anwendungen liefern Gleichungssysteme mit vielen Nulleinträgen, wie die Netzwerkanalyse (siehe Beispiel 6.2). Allgemein sagt man:

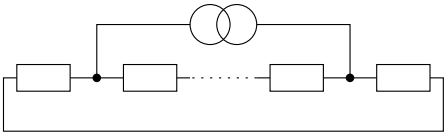
$$A \text{ ist dünnbesetzt} \Leftrightarrow A \text{ hat } O(n) \text{ Nichtnullelemente.}$$

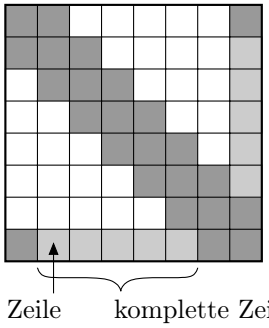
Typischerweise enthält jede Zeile $\leq m$ Nichtnullelemente.

Diese Definition geht davon aus, dass man nicht nur eine einzelne Matrix sondern eine ganze Familie von Matrizen wachsender Größe betrachtet.

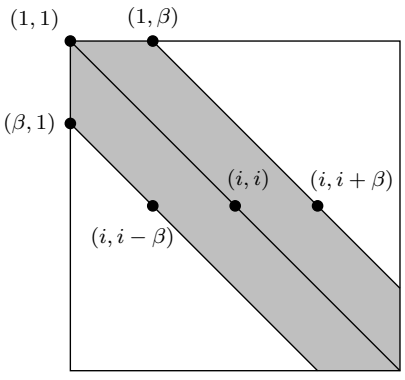
Fill-In

Betrachte die Matrix, die aus folgendem Netzwerk ensteht:



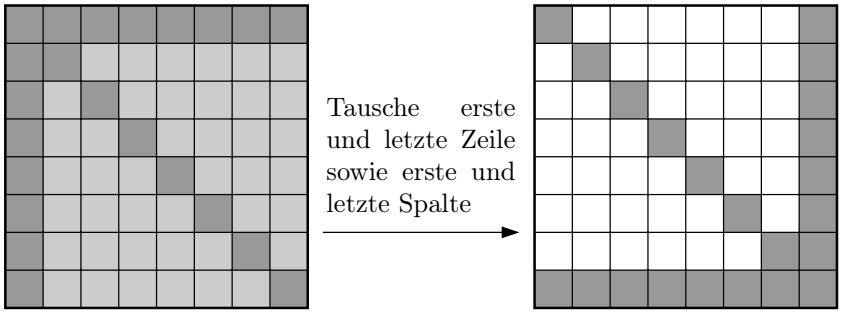


Allgemeine Bandmatrizen



- maximale Anzahl von Einträgen pro Zeile: $2\beta - 1$
- Aufwand: $A(n) \leq n \cdot \beta \cdot 2\beta = O(n\beta^2)$
- 2D-Gitter von Widerständen: $\beta = \sqrt{n}$
  $\Rightarrow A(n) = O(n^2)$
- 3D-Gitter von Widerständen: $\beta = n^{\frac{2}{3}}$
  $\Rightarrow A(n) = O\left(n^{\frac{7}{3}}\right)$

Pfeilmatrizen



Matrix voll kein Fill-In

- Umordnen von Zeilen und Spalten beeinflusst den Fill-In.
- Die Bestimmung der optimalen Anordnung für minimalen Fill-In ist ein NP-vollständiges Problem (Yan81).
- Es gibt schnelle Heuristiken, die gute Ergebnisse erzielen.
- Fill-In-Minimierung kollidiert im Allgemeinen mit Pivoting.

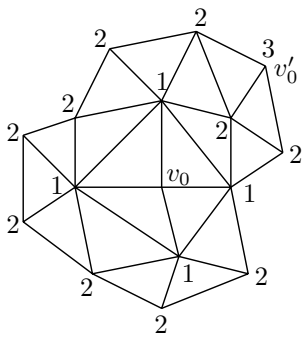
8.2 Anordnungsstrategien zur Fill-In-Minimierung

Graph einer Matrix: Sei $A \in \mathbb{R}^{n \times n}$ eine dünnbesetzte, symmetrisch positiv definite Matrix. Wir setzen

$$V_A = \{0, \ldots, n-1\}$$
$$E_A = \{(i,j) | a_{ij} \neq 0\}$$
$$G_A = (V_A, E_A)$$

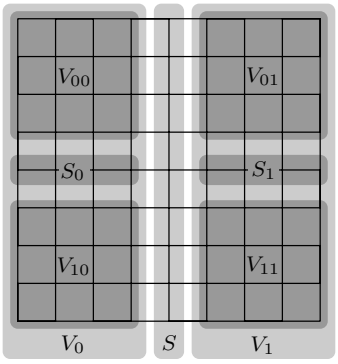
Da A symmetrisch ist, ist auch E_A symmetrisch und G_A stellt einen ungerichteten Graph dar.
Bei Widerstandsnetzwerken entspricht G_A genau wieder dem Graph des Netzwerks.

Reverse Cuthill-McKee Ordering: Minimierung der Bandbreite β einer Bandmatrix



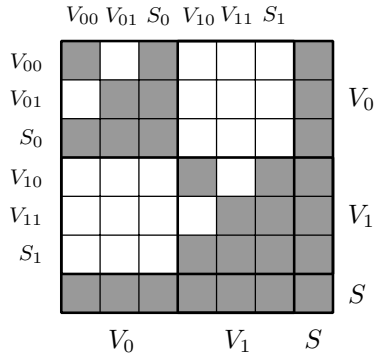
1. wähle Startknoten v_0
2. bestimme einen Knoten v_0' mit maximaler Entfernung (Breitensuche, Dijkstra)
3. nimm v_0' als Startknoten
4. bestimme Entfernungen aller Knoten von v_0'
5. ordne Knoten nach der Entfernung von v_0'
6. tausche Zeilen und Spalten von A entsprechend

Nested Dissection: Speziell geeignet für gitterartige Graphen



1. zerlege $V = V_0 \cup S \cup V_1$ mit $E_A \cap (V_0 \times V_1) = \emptyset$, V_0 und V_1 möglichst gleich groß sowie S möglichst klein
2. ordne zuerst alle Knoten von V_0, dann V_1 und schließlich S
3. fahre rekursiv mit V_0 und V_1 fort

Das Resultat ist eine Matrix in Blockgestalt:



- Nested Dissection erreicht eine Komplexität von $O\left(n^{\frac{3}{2}}\right)$, wenn G_A ein quadratisches 2D-Gitter ist und $O(n^2)$ für kubische 3D-Gitter.
- Finden des optimalen Graphseparatoren ist ein NP-vollständiges Problem. Es gibt aber effiziente Heuristiken, die in $O(n)$ Schritten einen guten Separator konstruieren. Dieselben Methoden verwendet man auch zur Partitionierung von Gittern zum Zwecke der parallelen Bearbeitung.

8.3 Cholesky-Zerlegung

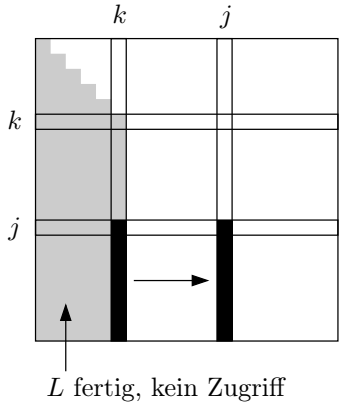
Bei symmetrisch positiv definiten Matrizen lässt sich der Aufwand für die LU-Zerlegung um den Faktor zwei reduzieren, da die Elemente im oberen Dreieck gleich denen im unteren Dreieck sind. Der entstehende Algorithmus heißt Cholesky-Zerlegung:

$$
\begin{aligned}
A &= LU && \text{die normale LU-Zerlegung} \\
&= LD\underbrace{D^{-1}U}_{=L^T} && \text{mit } D = \operatorname{diag}(U) \\
&= \underbrace{LD^{\frac{1}{2}}}_{\tilde{L}}\underbrace{D^{\frac{1}{2}}L^T}_{\tilde{L}^T} && D^{\frac{1}{2}} = \operatorname{diag}\sqrt{d_{ii}} \\
&= \tilde{L}\tilde{L}^T
\end{aligned}
$$

Man berechnet und speichert nur $\tilde{L}$. Es gilt $\tilde{l}_{ii} = \sqrt{u_{ii}}$. Wir untersuchen zwei Formen der Cholesky-Zerlegung, die *kji-Form* und die *jki-Form*.

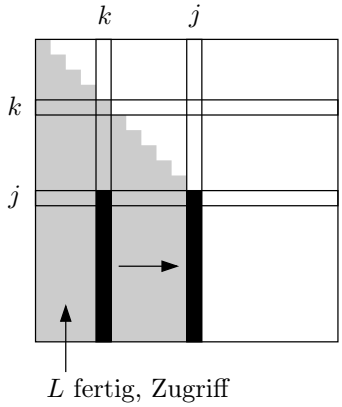
Algorithmus 6 Cholesky-Zerlegung, kji-Form

for $k = 0$ to $n-1$ **do**
  $l_{kk} = \sqrt{l_{kk}}$
  for $s = k+1$ to $n-1$ **do**
    $l_{sk} = \frac{l_{sk}}{l_{kk}}$
  end for
  for $j = k+1$ to $n-1$ **do**
    for $i = j$ to $n-1$ **do**
      $a_{ij} = a_{ij} - l_{ik} l_{jk}$
    end for
  end for
end for



Algorithmus 7 Cholesky-Zerlegung, jki-Form

for $j = 0$ to $n-1$ **do**
  for $k = 0$ to $j-1$ **do**
    for $i = j$ to $n-1$ **do**
      $a_{ij} = a_{ij} - l_{ik} l_{jk}$
    end for
  end for
  $l_{jj} = \sqrt{l_{jj}}$
  for $s = j+1$ to $n-1$ **do**
    $l_{sj} = \frac{l_{sj}}{l_{jj}}$
  end for
end for



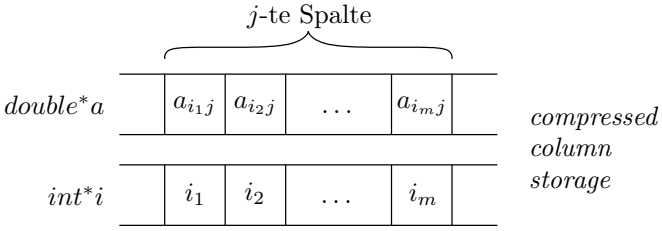
- Beide Verfahren sind spaltenorientiert.
- kji-Form führt Updates auf "späteren" Spalten $j > k$ sofort durch (*immediate update*).
- jki-Form führt Updates der Spalte j für alle "früheren" Spalten $k < j$ verzögert durch (*delayed update*).
- Update in Spalte j ist nur nötig, falls $a_{kj} = l_{jk} \neq 0$.

8.4 Direkte Lösung dünnbesetzter, symmetrisch positiv definiter Systeme

Das allgemeine Vorgehen ist wie folgt:

1. Bestimme gute Nummerierung, die Fill-In minimiert.

2. Führe symbolische Faktorisierung durch um Fill-In exakt zu bestimmen. → Datenstruktur kann allokiert werden. Aus Effizienzgründen bevorzugt man eine konsekutive Speicherung der Nichtnullelemente, z.B. spaltenweise:



3. Durchführung der numerischen Faktorisierung.

4. Lösung der Dreieckssysteme.

Allgemeine, unsymmetrische Systeme sind wesentlich schwieriger zu behandeln und erfordern andere Algorithmen.
Im Folgenden betrachten wir Schritt 3 genauer.

Der Eliminationsbaum

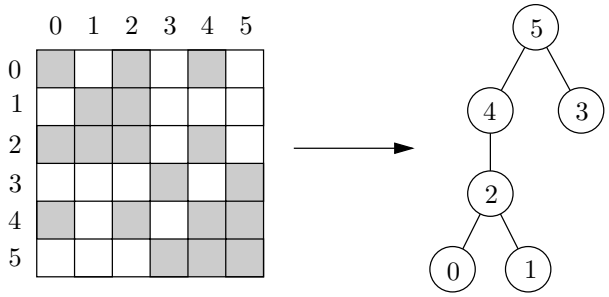
Der Eliminationsbaum ist definiert über den Cholesky-Faktor $\tilde{L}$:

$$\text{parent}(j) = \begin{cases} \min\left\{i | i > j \wedge \tilde{l}_{ij} \neq 0\right\} & \text{falls diese Menge nicht leer ist} \\ j & \text{sonst} \end{cases}$$

Also gilt $\text{parent}(j) \geq j$. Der Eliminationsbaum hat n Knoten $0, \ldots, n-1$, entsprechend den Spaltennummern bei einer $n \times n$-Matrix.

Der Eliminationsbaum ist nach der symbolischen Faktorisierung bekannt, da nur die Struktur von L, nicht aber die konkreten Einträge benötigt werden.

Beispiel 8.1.



Eigenschaften:

- $\text{parent}(j) = i$ bedeutet:

  - Spalte i bekommt von Spalte j ein Update, aber nicht nur! $i > j$ ist die erste Spalte, die von Spalte j ein update bekommen wird (im Sinne von right-looking).
  - Umgedreht ist im left-looking Sinne $j < i$ die letzte Spalte von der Spalte i ein Update bekommen wird.
  - Spalte i kann berechnet werden, sobald die Updates von allen Kindern berücksichtigt sind.

- Der Unterbaum mit Knoten j als Wurzel enthält alle Spalten, von denen Spalte j jemals ein Update bekommt. In Beispiel 8.1 bekommt Spalte 4 Updates von den Spalten 0, 1 und 2, nicht aber von Spalte 3.
- Der Eliminationsbaum stellt die Datenabhängigkeiten dar.
- Disjunkte Teilbäume können parallel eliminiert werden (was wir nicht weiter betrachten).
- Der Eliminationsbaum einer vollbesetzten Matrix ist eine lineare Kette.

Mehrfrontenmethode

Die Mehrfrontenmethode ist die Basis aller effizienten Methoden für symmetrisch positiv definite Matrizen. Zunächst benötigen wir zwei zusätzliche Definitionen:

- $\text{Struct}\,(L_{*j}) = \{i > j | l_{ij} \neq 0\}$, die Indizes der Nichtnullelemente in der j-ten Spalte des L-Faktors.
- $C = \text{extend\_add}\,(A, B)$, die Addition zweier dünnbesetzter Matrizen
  - Nichtnullelemente(C) = Nichtnullelemente$(A) \cup$ Nichtnullelemente(B)
  - $c_{ij} = a_{ij} + b_{ij}$
  - ist effizient in $O\,(|\text{Nichtnullelemente}(A)| + |\text{Nichtnullelemente}(B)|)$ berechenbar

Die Faktorisierung formulieren wir als rekursive Funktion Factor. Diese wird aufgerufen mit der Wurzel des Eliminationsbaumes, also Factor(n-1).

$F_j = \text{Factor}(j)$

a) Sei $\{i_1, \ldots, i_r\} = \text{Struct}\,(L_{*j})$.

b) Initiialisiere

$$F_j = \begin{pmatrix} a_{j,j} & a_{j,i_1} & \ldots & a_{j,i_r} \\ a_{i_1,j} & 0 & \ldots & 0 \\ \vdots & \vdots & & \vdots \\ a_{i_r,j} & 0 & \ldots & 0 \end{pmatrix}$$

F_j ist eine vollbesetzte Matrix, die alle Updates *rechts von Spalte j* enthalten wird, die aus Elimination der Spalte j *und allen j modifizierenden Spalten* entstehen.

c) Für jedes (direkte) Kind i von j im Eliminationsbaum berechne:

- $F_i = \text{Factor}(i)$, $F_i = \left(\begin{array}{c|c} f_{i,i} & f_{*i}^T \\ \hline f_{*i} & U_i \end{array}\right)$. U_i ist also die Untermatrix von F_i welche nicht die erste Zeile und Spalte enthält.
- $F_j = \text{extend\_add}\,(F_j, U_i)$ (akkumuliere Updates von Spalte i und allen Kindeskindern.)

Die erste Spalte von F_j enthält nun alle Modifikationen der Spalte j von früheren Spalten.

d) Führe einen Schritt der Cholesky-Zerlegung für die erste Spalte durch:

$$F_j = \left(\begin{array}{c|c} l_{j,j} & 0 \\ \hline l_{i_1,j} & \\ \vdots & I \\ l_{i_m,j} & \end{array}\right) \cdot \left(\begin{array}{c|c} 1 & 0 \\ \hline 0 & U_j \end{array}\right) \cdot \left(\begin{array}{c|ccc} l_{j,j} & l_{i_1,j} & \dots & l_{i_m,j} \\ \hline 0 & & I & \end{array}\right)$$

$m \geq r$: Durch extend_add können weitere Nichtnullelemente hinzukommen.

return F_j.

Große Vorteile:

- Alle rechenintensiven Operationen in Schritt d) werden auf vollbesetzten Matrizen durchgeführt.
- Sparse-Struktur tritt nur in extend_add und der Rekursion in c) auf.

9 Abstiegsverfahren

In diesem Kapitel betrachten wir Abstiegsverfahren zur iterativen Lösung linearer Gleichungssysteme. Abstiegsverfahren berechnen eine Folge von Iterierten $x^0, x^1, \ldots, x^k, \ldots$, die sich der Lösung von $Ax = b$ annähern.

Vorteile iterativer Verfahren:

- Keine Elimination, also keine Probleme durch Fill-In.
- Der Aufwand für einen Schritt $x^k \to x^{k+1}$ beträgt $O\left(\text{Nichtnullelemente}\left(A\right)\right)$. Abstiegsverfahren lohnen sich also insbesondere bei dünnbesetzten Matrizen.

Nachteil: Die Konvergenz des Verfahrens ist meist nur für gewisse Klassen von Matrizen gewährleistet.

Wir beschränken uns hier weitgehend auf symmetrisch positiv definite Matrizen und führen die folgenden Abkürzungen ein:

- $(x, y) = \sum_{i=1}^{n} x_i y_i;\ \|x\| = \sqrt{(x, x)}$ (Euklidsches Skalarprodukt und Norm)
- $(x, y)_A = (Ax, y) = x^T A y;\ \|x\|_A = \sqrt{(x, x)_A}$ (Energieskalarprodukt, Energienorm)

Symmetrisch positiv definite Matrizen haben ein reelles, positives Spektrum:

- $\sigma(A) = \{\lambda = \lambda_1, \ldots, \lambda_n = \Lambda\}$ mit $\lambda > 0$, $\lambda_i \leq \lambda_{i+1}$, $\lambda_i \in \mathbb{R}^+$
- Spektralradius: $\rho(A) = \max_{\lambda_i \in \sigma(A)} |\lambda_i| = \Lambda$
- $\text{cond}_2(A) = \kappa(A) = \frac{\Lambda}{\lambda}$

9.1 Der Charakterisierungssatz

Satz 9.1 (Charakterisierungssatz)**.** Sei A eine symmetrisch positiv definite Matrix. Für die eindeutige Lösung von $Ax = b$ gilt

$$J(y) > J(x) \quad \forall y \in \mathbb{R}^n, y \neq x$$

mit dem quadratischen Funktional

$$J(y) := \frac{1}{2}(Ay, y) - (b, y). \tag{9.1}$$

Beweis.

1. $Ax = b \Rightarrow J(y) > J(x) \quad \forall y \neq x$

$$\begin{aligned} J(y) - J(x) &= \frac{1}{2}\Big((Ay, y) - 2\,(b, y) - (Ax, x) + 2\,(b, x)\Big) \\ &= \frac{1}{2}\Big((Ay, y) - 2\,(Ax, y) - (Ax, x) + 2\,(Ax, x)\Big) \\ &= \frac{1}{2}\Big((Ay, y) - 2\,(Ax, y) + (Ax, x)\Big) \\ &= \frac{1}{2}\,(A\,[x - y]\,, x - y) \\ &> 0 \text{ für } x - y \neq 0 \end{aligned}$$

2. $Ax = b \Leftarrow J(y) > J(x) \quad \forall y \neq x$

   Notwendig für Minimum ist $\nabla J(x) = \begin{pmatrix} \frac{\partial J}{\partial x_1}(x) \\ \vdots \\ \frac{\partial J}{\partial x_n}(x) \end{pmatrix} = 0.$

   Es gilt:

$$\frac{\partial J}{\partial x_i}(x) = \frac{1}{2}\left[e_i^T A x + x^T A e_i\right] - b^T e_i = e_i^T\,(Ax - b)$$

   wobei e_i der i-te Einheitsvektor ist. Also gilt:

$$\nabla J(x) = Ax - b = 0$$

□

9.2 Line Search

Abstiegsverfahren bestimmen ausgehend von einem Startwert x^0 eine Folge von Iterierten der Form

$$x^{t+1} = x^t + \alpha_t r^t$$

r^t ist dabei eine vorgegebene Suchrichtung. Die Schrittweite α_t wird bestimmt durch die Vorschrift

$$J\left(x^{t+1}\right) = \min_{\alpha \in \mathbb{R}} J\left(x^t + \alpha r^t\right) \tag{9.2}$$

Wegen

$$\begin{aligned} &\frac{\mathrm{d}}{\mathrm{d}\alpha}\left[\frac{1}{2}\left(A\left[x^t + \alpha r^t\right], x^t + \alpha r^t\right) - \left(b, x^t + \alpha r^t\right)\right] \\ &= \frac{1}{2}\left(Ar^t, x^t + \alpha r^t\right) + \frac{1}{2}\left(A\left[x^t + \alpha r^t\right], r^t\right) - \left(b, r^t\right) \\ &= \left(Ax^t, r^t\right) + \alpha\left(Ar^t, r^t\right) - \left(b, r^t\right) \\ &= \left(Ax^t - b, r^t\right) + \alpha\left(Ar^t, r^t\right) \end{aligned}$$

gilt

$$\frac{\mathrm{d}}{\mathrm{d}\alpha} J\left(x^t + \alpha r^t\right) = 0 \quad \Leftrightarrow \quad \alpha = -\frac{\left(Ax^t - b, r^t\right)}{\left(Ar^t, r^t\right)}$$

Wir führen eine Abkürzung ein:

$$g^t := Ax^t - b = \nabla J\left(x^t\right)$$

Die Größe $-g^t$ heißt *Defekt*. Beachte: $x^t = x \Leftrightarrow g^t = 0$.

9.3 Abstiegsverfahren in algorithmischer Form

Algorithmus 8 allgemeines Abstiegsverfahren

Startwert: $x^0 \in \mathbb{R}$
$g^0 := Ax^0 - b$
for $t \geq 0$ **do**
  wähle Suchrichtung r^t
  $q^t = Ar^t$
  $\alpha_t = -\frac{\left(g^t, r^t\right)}{\left(q^t, r^t\right)}$
  $x^{t+1} = x^t + \alpha_t r^t$
  $g^{t+1} = g^t + \alpha_t q^t$
end for

- Aufwand: Im Wesentlichen eine Matrix-Vektor-Multiplikation.
- A muss nicht als Matrix gespeichert sein. Es genügt, wenn man Ar^t effizient berechnen kann („on-the-fly-berechnung“).
- Der Algorithmus ist ohne Abbruchkriterium formuliert. Dies ist in der Praxis natürlich noch zu ergänzen.

9.4 Wahl der Suchrichtungen

- $r^t = e_j$ mit $j = (t \bmod n) + 1$ und e_j dem Einheitsvektor in Richtung j liefert das *Gauß-Seidel-Verfahren*:

$$\begin{aligned} x^{t+1} &= x^t - \frac{\left(Ax^t - b, e_j\right)}{\left(Ae_j, e_j\right)} e_j \qquad j = (t \bmod n) + 1 \\ &= x^t + \begin{cases} \frac{\left(b - Ax^t\right)_j}{A_{jj}} & \text{falls } i = j \\ 0 & \text{sonst} \end{cases} \end{aligned}$$

Algorithmus 9 Gradientenverfahren

Startwert: $x^0 \in \mathbb{R}$
$g^0 := Ax^0 - b$
for $t \geq 0$ **do**
  $q^t = Ag^t$
  $\alpha_t = \frac{(g^t, g^t)}{(q^t, g^t)}$
  $x^{t+1} = x^t - \alpha_t g^t$
  $g^{t+1} = g^t - \alpha_t q^t$
end for

- $r^t = -g^t = -\nabla J\left(x^t\right)$ liefert das *Gradientenverfahren* (siehe Algorithmus 9): Für das Gradientenverfahren gilt

$$\left(g^{t+1}, g^t\right) = \left(g^t - \alpha_t A g^t, g^t\right) = \left(g^t, g^t\right) - \frac{\left(g^t, g^t\right)}{\left(Ag^t, g^t\right)} \left(Ag^t, g^t\right) = 0$$

  Zwei direkt aufeinanderfolgende Suchrichtungen stehen also senkrecht aufeinander.

Satz 9.2 (Konvergenz des Gradientenverfahrens)**.** Wir definieren das Fehlerfunktional

$$E(y) = \|y - x\|_A^2 = (y - x, A[y - x]) \quad \forall y \in \mathbb{R}^n$$

und setzen $e^t := x^t - x$ (der Fehler in der t-ten Iterierten). Dann gilt für das Gradientenverfahren

$$E\left(x^{t+1}\right) \leq \left(1 - \frac{1}{\kappa(A)}\right) E\left(x^t\right)$$

Wegen $\kappa(A) \geq 1$ gilt also $E\left(x^{t+1}\right) < E\left(x^t\right)$.

Beweis. Aus (Ran).

$$\begin{aligned}
\frac{E\left(x^t\right) - E\left(x^{t+1}\right)}{E\left(x^t\right)} &= \frac{\left(e^t, Ae^t\right) - \left(e^{t+1}, Ae^{t+1}\right)}{\left(e^t, Ae^t\right)} \\
&= \frac{\left(e^t, Ae^t\right) - \left(e^t - \alpha_t g^t, A\left[e^t - \alpha_t g^t\right]\right)}{\left(e^t, Ae^t\right)} \\
&= \frac{2\alpha_t \left(e^t, Ag^t\right) - \alpha_t^2 \left(g^t, Ag^t\right)}{\left(e^t, Ae^t\right)} \\
&= \frac{2\frac{\left(g^t, g^t\right)}{\left(Ag^t, g^t\right)} \left(g^t, g^t\right) - \frac{\left(g^t, g^t\right)^2}{\left(Ag^t, g^t\right)^2} \left(g^t, Ag^t\right)}{\left(g^t, A^{-1} g^t\right)} \quad = \frac{\left\|g^t\right\|^4}{\left(g^t, Ag^t\right)\left(g^t, A^{-1} g^t\right)}
\end{aligned}$$

Raleigh-Quotienten liefern

$$\lambda \|y\|^2 \leq (y, Ay) \leq \Lambda \|y\|^2 \quad \text{sowie} \quad \Lambda^{-1} \|y\| \leq \left(y, A^{-1} y\right) \leq \lambda^{-1} \|y\|^2$$

und damit

$$\frac{E\left(x^t\right) - E\left(x^{t+1}\right)}{E\left(x^t\right)} = \frac{\left\|g^t\right\|^4}{\left(g^t, Ag^t\right)\left(g^t, A^{-1}g^t\right)} \geq \frac{\left\|g^t\right\|^4}{\Lambda \left\|g^t\right\|^2 \lambda^{-1} \left\|g^t\right\|^2} = \frac{\lambda}{\Lambda}$$

$$\Leftrightarrow E\left(x^t\right) - E\left(x^{t+1}\right) \geq \frac{\lambda}{\Lambda} E\left(x^t\right) \quad \Leftrightarrow \quad \underbrace{E\left(x^{t+1}\right)}_{\left\|e^{t+1}\right\|_A^2} \leq \left(1 - \frac{1}{\kappa(A)}\right) \underbrace{E\left(x^t\right)}_{\left\|e^t\right\|_A^2}$$

□

Eine bessere Abschätzung liefert

$$E\left(x^{t+1}\right) \leq \left(\frac{\kappa(A) - 1}{\kappa(A) + 1}\right)^2 E\left(x^t\right)$$

9.5 Verfahren der konjugierten Gradienten

Das Gradientenverfahren leidet darunter, dass in einem Schritt zwar in Richtung r^t minimiert wird, diese Optimalität aber im übernächsten Schritt wieder vergessen wird. Das Verfahren der konjugierten Gradienten (conjugate gradients, CG-Verfahren) vermeidet diesen Nachteil.

Mehrfache Suchrichtungen Dies führt zur Idee, in einem höherdimensionalen Raum zu minimieren, d. h. in einem Schritt bezüglich mehrerer Suchrichtungen zu optimieren. Dazu sei

$$B_t := \operatorname{span}\left\{d^0, d^1, \ldots, d^{t-1}\right\}$$

ein t-dimensionaler Suchraum, das heißt, die d^i seien linear unabhängig. Zunächst seien die d^i nicht weiter spezifiziert.

Bestimme nun ausgehend von x^0 ein x^t derart, dass

$$x^t = x^0 + \sum_{i=0}^{t-1} \alpha_j^{t-1} d^j \ \in x^0 + B_t$$

wobei die Koeffizienten α_j^{t-1} so bestimmt werden, dass analog zu (9.2)

$$J\left(x^t\right) = \min_{y \in x^0 + B_t} J\left(y\right) \tag{9.3}$$

Notwendige Bedingung für ein Minimum ist

$$\frac{\partial J\left(x^0 + \sum \alpha_j^{t-1} d^j\right)}{\partial \alpha_i^{t-1}} = 0 \qquad i = 0, \ldots, t-1.$$

Dies führt (ausrechnen der Ableitung mittels Produktregel) auf die sogenannten *Galerkin-Gleichungen*:

$$\left(Ax^t - b, d^i\right) = 0 \qquad \forall i = 0, \ldots, t-1. \tag{9.4}$$

Setzt man den Ansatz für x^t ein führt dies auf ein lineares Gleichungssystem für die Koeffizienten α_j^{t-1}:

$$\left(Ax^0 - b, d^i\right) + \sum_{j=0}^{t-1} \alpha_j^{t-1} \left(Ad^j, d^i\right) = 0 \qquad \forall i = 0, \ldots, t-1. \tag{9.5}$$

Bemerkung 9.3. Die Gleichungen (9.4) ergeben auch für unsymmetrisches A Sinn, jedoch ist (9.3) nur für symmetrisch positiv definites A definiert. (9.4) ist also allgemeiner. □

Dies erfordert ein speichern aller Suchrichtungen und das Lösen eines Gleichungssystems das mit der Anzahl der Suchrichtungen immer größer wird. Für n Suchrichtungen würde man in einem Schritt die Lösung erhalten.

Bis jetzt waren die Suchrichtungen beliebig. Ist A symmetrisch positiv definit so bildet (Ax, y) ein Skalarprodukt und es macht Sinn die Suchrichtungen derart zu wählen, dass

$$(Ad^j, d^i) = 0 \qquad \forall 0 \leq i < j \leq t-1. \tag{9.6}$$

Man sagt dann, dass alle Suchrichtungen *A-orthogonal* bzw. *konjugiert* zueinander sind. Damit reduzieren sich die Galerkin-Gleichungen auf

$$\begin{aligned} &\left(Ax^0 - b, d^i\right) + \alpha_i^{t-1} \left(Ad^i, d^i\right) = 0, \\ \Leftrightarrow \quad &\alpha_i^{t-1} = -\frac{(Ax^0 - b, d^i)}{(Ad^i, d^i)} \qquad \forall i = 0, \ldots, t-1. \end{aligned} \tag{9.7}$$

Die Gleichung für α_i^{t-1} hängt nun nicht mehr von den Suchrichtungen d^j, $j \neq i$ ab. Insbesondere verändert das Hinzufügen weiterer Suchrichtungen nicht den Koeffizienten bezüglich einer alten Suchrichtung. Im A-orthogonalen Fall können wir also den Superskript $t-1$ an α_j^{t-1} weglassen.

Krylovraum Durch die Bedingung der A-orthogonalität sind die Suchrichtungen noch nicht eindeutig festgelegt. Im folgenden bauen wir die Menge der Suchrichtungen *schrittweise* auf.

Sind $d^0, \ldots, d^{t-1}$ und x^t gegeben, wobei x^t durch Lösen der zugehörigen Galerkin-Gleichungen ermittelt wurde, dann berechnen wir die nächste Suchrichtung über den Ansatz

$$d^t = -g^t + \sum_{j=0}^{t-1} \beta_j^{t-1} d^j \tag{9.8}$$

wobei $g^t = Ax^t - b$ wie immer der Gradient ist und die Koeffizienten β_j^{t-1} noch so zu bestimmen sind, dass $\left(d^t, Ad^i\right) = 0$ ist für $0 \leq i < t$.

Bevor wir dies tun zeigen wir noch die Behauptung, dass unabhängig von der Wahl der β_j^{t-1} gilt

$$\operatorname{span}\{d^0, \ldots, d^t\} = \operatorname{span}\{r^0, Ar^0, \ldots, A^t r^0\} \qquad \text{mit } r^0 = b - Ax^0. \tag{9.9}$$

Den Raum $K_t(r^0, A) = \text{span}\{r^0, Ar^0, \ldots, A^t r^0\}$ nennt man einen Krylovraum. Die Behauptung zeigen wir durch Induktion. Zunächst ist für $t = 0$ natürlich $d^0 = -g^0 = b - Ax^0 = r^0$. Die Behauptung sei nun bis $t-1$ erfüllt. Dann gilt

$$\begin{aligned} d^t &= -\left[A\left(x^0 + \sum_{i=0}^{t-1} \alpha_i d^i\right) - b\right] + \sum_{j=0}^{t-1} \beta_j^{t-1} d^j \\ &= b - Ax^0 - \sum_{i=0}^{t-1} \alpha_i A d^i + \sum_{j=0}^{t-1} \beta_j^{t-1} d^j \\ &= \underbrace{b - Ax^0 - \sum_{i=0}^{t-2} \alpha_i A d^i + \sum_{j=0}^{t-1} \beta_j^{t-1} d^j}_{\in \text{span}\,\{r^0,\ldots,A^{t-1}r^0\}} - \underbrace{\alpha_{t-1} A d^{t-1}}_{\in \text{span}\,\{Ar^0,\ldots,A^t r^0\}} . \end{aligned}$$

Ist $A^t r^0$ linear unabhängig von $r^0, \ldots, A^{t-1} r^0$ so vergrößert die neue Richtung die Dimension des Krylovraumes. Da $\alpha_{t-1} \neq 0$ wird in diesem Fall die Richtung $A^t r^0$ auch tatsächlich zur Darstellung von d^t benötigt, es ist also $\text{span}\{d^0, \ldots, d^t\} = \text{span}\{r^0, Ar^0, \ldots, A^t r^0\}$.

Beachte, dass für die Dimension des Krylovraumes gilt $\dim K_t(r^0, A) \leq \min(t+1, n)$. Als Beispiel: Ist r^0 ein Eigenvektor von A so gilt $Ar^0 = \lambda r^0$ und $\dim K_t(r^0, A) = 1$ für alle t. In diesem Fall ist das CG-Verfahren aber auch in einem Schritt fertig.

Im Gradientenverfahren gilt $d^t = -g^t$ und somit ist dieses Verfahren ein Spezialfall von (9.8) mit $\beta_j^{t-1} = 0$. Verfahren bei denen die Suchrichtungen in $K_t(r^0, A)$ sind nennt man *Krylovraumverfahren*. Sowohl das Gradientenverfahren als auch das CG-Verfahren sind Vertreter der Klasse der Krylovraumverfahren.

Berechnung der neuen Suchrichtung Es seien $d^0, \ldots, d^{t-1}$ und x^t gegeben, wobei x^t durch Lösen der zugehörigen Galerkin-Gleichungen ermittelt wurde. Die Suchrichtungen $d^0, \ldots, d^{t-1}$ seien A-orthogonal.

Nun sind die Koeffizienten β_j^{t-1} in (9.8) zu berechnen um die Suchrichtung d^t zu bestimmen.

Die neue Suchrichtung d^t muss wieder A-orthogonal zu allen bisherigen Suchrichtungen sein, also

$$\begin{aligned} 0 = \left(d^t, Ad^i\right) &= \left(-g^t, Ad^i\right) + \sum_{j=0}^{t-1} \beta_j^{t-1} \underbrace{\left(d^j, Ad^i\right)}_{0 \text{ für } j \neq i} \\ &= -\left(g^t, Ad^i\right) + \beta_i^{t-1} \left(d^i, Ad^i\right) \qquad 0 \leq i \leq t-1. \end{aligned} \tag{9.10}$$

Wir unterscheiden die zwei Fälle $i \leq t-2$ und $i = t-1$:

1. $0 \leq i \leq t-2$: $g^t = Ax^t - b$ in den ersten Term eingesetzt liefert $(Ax^t - b, Ad^i)$. Nun wurde x^t ja über die Lösung der Galerkingleichungen $(Ax^t - b, d^i) = 0$ für $0 \leq i \leq t-1$ bestimmt. Wegen $\text{span}\{d^0, \ldots, d^{t-1}\} = \text{span}\{r^0, \ldots, A^{t-1} r^0\}$ gilt damit auch $(Ax^t - b, A^i r^0) = 0$ für $0 \leq i \leq t-1$. Da $Ad^i \in \text{span}\{Ar^0, \ldots, A^{i+1} r^0\}$ gilt $(Ax^t - b, Ad^i) = 0$ solange $i + 1 \leq t - 1 \Leftrightarrow i \leq t-2$.

Damit reduziert sich (9.10) auf

$$\beta_i^{t-1} \underbrace{\left(d^i, Ad^i\right)}_{>0} = 0 \quad \Leftrightarrow \quad \beta_i^{t-1} = 0.$$

2. $i = t-1$: Es ergibt sich

$$-\left(g^t, Ad^{t-1}\right) + \beta_{t-1}^{t-1}\left(d^{t-1}, Ad^{t-1}\right) = 0 \quad \Leftrightarrow \quad \beta_{t-1}^{t-1} = \frac{\left(g^t, Ad^{t-1}\right)}{\left(d^{t-1}, Ad^{t-1}\right)}.$$

Für die neue Suchrichtung gilt damit

$$d^t = -g^t + \beta_{t-1}^{t-1} d^{t-1}.$$

Berechnung der nächsten Iterierten Nachdem nun die Suchrichtung d^t bekannt ist kann die neue Iterierte x^{t+1} berechnet werden. Wegen der A-orthogonalität der Suchrichtungen gilt nach (9.7)

$$\alpha_t = \frac{(g^0, d^t)}{(Ad^t, d^t)} = \frac{(g^t, d^t)}{(Ad^t, d^t)}$$

und damit

$$x^{t+1} = x^t + \alpha_t d^t.$$

Nun können wir das Gesamtverfahren formulieren:

Algorithmus 10 CG-Verfahren

Startwert $x^0 \in \mathbb{R}^n$
$g^0 = Ax^0 - b$
$d^0 = -g^0$
for $t \geq 0$ **do**
  $q^t = Ad^t$
  $\alpha_t = -\frac{\left(g^t, d^t\right)}{\left(q^t, d^t\right)}$
  $x^{t+1} = x^t + \alpha_t d^t$
  $g^{t+1} = g^t + \alpha_t q^t$
  $\beta_t = \frac{\left(g^{t+1}, q^t\right)}{\left(q^t, d^t\right)}$
  $d^{t+1} = -g^{t+1} + \beta_t d^t$
end for

Damit erhalten wir das CG-Verfahren (Algorithmus 10). Der Aufwand entsteht im Wesentlichen aus der Berechnung von Ad^t.

Satz 9.4. In exakter Arithmetik liefert das CG-Verfahren nach spätestens n Schritten die Lösung. □

In der Praxis wird dies wegen Rundungsfehlern nicht erreicht und man benutzt das CG-Verfahren als iteratives Verfahren. Über die Fehlerreduktion pro Schritt gibt der folgende Satz Auskunft.

Satz 9.5 (Konvergenz des CG-Verfahrens)**.** Für das CG-Verfahren gilt die Abschätzung

$$\left\|x^t - x\right\|_A \leq 2\left(\frac{\sqrt{\kappa(A)}-1}{\sqrt{\kappa(A)}+1}\right)^t \left\|x^0 - x\right\|_A$$

Zur Reduzierung des Anfangsfehlers auf $\varepsilon \left\|x^0 - x\right\|_A$ sind höchstens

$$t(\varepsilon) \leq \frac{1}{2}\sqrt{\kappa(A)}\ln\left(\frac{2}{\varepsilon}\right) + 1$$

Iterationen notwendig. □

Vorkonditionierung

Kombiniert ein Krylovraumverfahren mit linearen Iterationsverfahren.

- Wenig Iterationen $\Leftrightarrow \kappa(A)$ klein.
- Sei C eine reguläre Matrix mit $C = KK$.
- Transformiere:
$$Ax = b \quad \Leftrightarrow \quad \underbrace{K^{-1}AK^{-1}}_{\tilde{A}}\underbrace{Kx}_{\tilde{x}} = \underbrace{K^{-1}b}_{\tilde{b}}.$$
$\tilde{A}$ ist symmetrisch positiv definit wenn A symmetrisch positiv definit.
Wegen $\sigma(\tilde{A}) = \sigma(K^{-1}AK^{-1}) = \sigma(K^{-1}K^{-1}AK^{-1}K) = \sigma(C^{-1}A)$ hat $\tilde{A}$ dieselben Eigenwerte wie $C^{-1}A$.
C sei so gewählt, dass $\kappa\left(\tilde{A}\right) \ll \kappa(A)$, d. h. C sollte möglichst gleich A aber leicht invertierbar sein.
- Wende das CG-Verfahren auf das transformierte System an.
- Die Transformation führt man nicht vorab einmal durch, sondern man transformiert jeden Schritt des CG-Verfahrens. ⇒ Pro Schritt ist zusätzlich nur das Lösen eines Systems $Cv^t = d^t$ erforderlich.

10 Einführung in gewöhnliche Differentialgleichungen

10.1 Beispiele gewöhnlicher Differentialgleichungen

Literatur: (Ran; Sim)

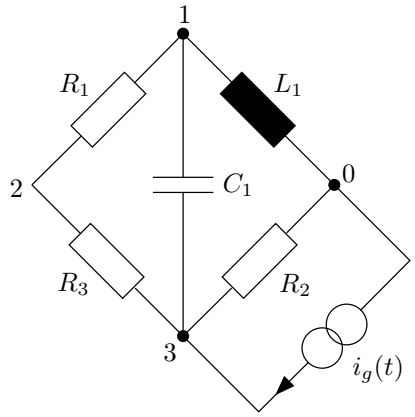


Abbildung 26: Netzwerk aus Beispiel 5.1

Beispiel 10.1 (Elektrische Netzwerke)**.** In Beispiel 5.1 analysierten wir das Netzwerk aus Abbildung 26. Unter Benutzung des Knotenpotenzialverfahrens ergibt sich das folgende Gleichungssystem:

$$\begin{bmatrix} C_1 & 0 & 0 & 0 \\ 0 & L & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \frac{\mathrm{d}u_{13}}{\mathrm{d}t} \\ \frac{\mathrm{d}i_{01}}{\mathrm{d}t} \\ \frac{\mathrm{d}\varphi_2}{\mathrm{d}t} \\ \frac{\mathrm{d}\varphi_3}{\mathrm{d}t} \end{bmatrix} + \begin{bmatrix} \frac{1}{R_1} & -1 & -\frac{1}{R_1} & \frac{1}{R_1} \\ -1 & 0 & 0 & -1 \\ \frac{1}{R_1} & -1 & -\frac{1}{R_1} - \frac{1}{R_3} & \frac{1}{R_1} + \frac{1}{R_2} + \frac{1}{R_3} \\ -\frac{1}{R_1} & 0 & \frac{1}{R_1} + \frac{1}{R_3} & -\frac{1}{R_1} - \frac{1}{R_3} \end{bmatrix} \begin{bmatrix} u_{13} \\ i_{01} \\ \varphi_2 \\ \varphi_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ i_g(t) \\ 0 \end{bmatrix}. \quad (10.1)$$

Zusätzlich benötigt man Anfangsbedingungen:

$$u_{13}(t_0) = u_0 \quad \text{und} \quad i_{01}(t_0) = i_0. \quad (10.2)$$

Allgemein ergibt sich bei der Netzwerkanalyse (mit linearen Bauelementen) ein lineares differentiell-algebraisches System (DAE) mit konstanten Koeffizienten in der Form

$$Au'(t) + Bu(t) = f(t).$$

A, B sind quadratische Matrizen und A ist im Allgemeinen singulär.
Für (10.1) gilt speziell

$$\begin{aligned} A_{11}u_1'(t) + B_{11}u_1(t) + B_{12}u_2(t) &= f_1(t), \\ B_{21}u_1(t) + B_{22}u_2(t) &= f_2(t) \end{aligned}$$

mit $u_1 = (u_{13}, i_{01})^T$, $u_2 = (\varphi_2, \varphi_3)^T$ und reglärem A_{11}, B_{22}. Also kann man (10.1) wegen $u_2(t) = B_{22}^{-1}\left(f_2(t) - B_{21}u_1(t)\right)$ auf ein implizites System gewöhnlicher Differentialgleichungen reduzieren:

$$A_{11}u_1'(t) + \left(B_{11} - B_{12}B_{22}^{-1}B_{21}\right)u_1(t) = f_1(t) - B_{12}B_{22}^{-1}f_2(t).$$

Nun ist auch klar, dass die beiden Anfangsbedingungen (10.2) genügen. Schließlich kann man noch zur sogenannten *expliziten* Form übergehen:

$$\begin{aligned} u_1'(t) &= A_{11}^{-1}\left[f_1(t) - B_{12}B_{22}^{-1}f_2(t) - \left(B_{11} - B_{12}B_{22}^{-1}B_{21}\right)u_1(t)\right], \\ u_1(t_0) &= (u_0, i_0)^T. \end{aligned}$$

Von nun an verwenden wir die explizite Form

$$\begin{aligned} u' &= f\left(t, u(t)\right) \\ u\left(t_0\right) &= u_0 \end{aligned}$$

Beispiel 10.2 (Populationsdynamik, Räuber-Beute-Modell)**.** Sei $f(t)$ die Anzahl der Füchse und $r(t)$ die Anzahl der Kaninchen. Unter der Annahme, dass der Futtervorrat für die Kaninchen unbeschränkt ist und die Füchse sich ausschließlich von Kaninchen ernähren kann man die Anzahl der Füchse und Kaninchen in einem bestimmten Gebiet mit zwei gekoppelten Differentialgleichungen modellieren:

$$\begin{aligned} r'(t) &= 2r(t) - \alpha r(t)f(t) \qquad & r(0) &= r_0 \geq 0, \\ f'(t) &= -f(t) + \alpha r(t)f(t) \qquad & f(0) &= f_0 \geq 0. \end{aligned}$$

1. $\alpha = 0$: $\Rightarrow r(t) = r_0 e^{2t}$, $f(t) = f_0 e^{-t}$.
2. $\alpha > 0$: Das Wachstum von r verlangsamt sich proportional zu $r(t)f(t)$, das Wachstum von f vergrößert sich proportional zu $r(t)f(t)$.

In diesem Beispiel ist die rechte Seite nicht linear!

Beispiel 10.3 (N-Körper Problem, Astronomie)**.** Betrachte die Bewegung von N Körpern mit den Massen m_i unter ihrem eigenen Schwerefeld. Zu bestimmen sind ihre Positionen $x_i(t) \in \mathbb{R}^3$ und Geschwindigkeiten $v_i(t) \in \mathbb{R}^3$.
Die Differentialgleichung für die Position erhält man aus der Definition der Geschwindigkeit:

$$\frac{\mathrm{d}x_i(t)}{\mathrm{d}t} = v_i(t); \qquad x_i\left(t_0\right) = x_{i,0}; \qquad i = 1, \ldots, N$$

Aus dem zweiten Newton'schen Gesetz erhält man die Differentialgleichung:

$$\underbrace{\sum_{\substack{1\le j\le N\\ j\neq i}} \frac{\gamma m_j m_i \left(x_j - x_i\right)}{\left\|x_j - x_i\right\|^3}}_{\text{Gravitation}} = \underbrace{\vec{F}_i(t) = m_i a_i(t) = m_i \frac{\mathrm{d}v_i(t)}{\mathrm{d}t}}_{\text{Newton}}; \qquad i = 1, \ldots, N$$

$$\Rightarrow \frac{\mathrm{d}v_i(t)}{\mathrm{d}t} = \sum_{\substack{1\le j\le N\\ j\neq i}} \frac{\gamma m_j \left(x_j - x_i\right)}{\left\|x_j - x_i\right\|^3}; \qquad v_i\left(t_0\right) = v_{i,0}; \qquad i = 1, \ldots, N.$$

Insgesamt ergibt sich also ein System aus $6N$ gekoppelten, nichtlinearen Differentialgleichungen.

Beispiel 10.4 (Reaktionskinetik, Chemie)**.** Eine Gleichgewichtsreaktion der Form

$$aA + bB \underset{k_2}{\overset{k_1}{\rightleftharpoons}} cC + dD$$

wird modelliert durch das System

$$\begin{aligned} c_A'(t) &= R(t) \\ c_B'(t) &= R(t) \\ c_C'(t) &= -R(t) \\ c_D'(t) &= -R(t) \end{aligned} \quad \text{mit} \quad \begin{aligned} R(t) &= -k_1 \left(c_A(t)\right)^a \left(c_B(t)\right)^b + k_2 \left(c_C(t)\right)^c \left(c_D(t)\right)^d \\ c_i\left(t_0\right) &= C_i; \quad i \in \{A, B, C, D\}\,. \end{aligned}$$

$c_i(t)$ ist die Konzentration von i, z.B. in $\left[\frac{\mathrm{mol}}{\mathrm{l}}\right]$ (also eine Anzahldichte). Im Reaktionsterm $R(t)$ modelliert $(c_A(t))^a (c_B(t))^b$ die Wahrscheinlichkeit, dass sich a Teilchen der Sorte A und b Teilchen der Sorte B *gleichzeitig* an einem Ort befinden um miteinander reagieren zu können.

Im chemischen Gleichgewicht gilt

$$c_i'(t) = 0 \quad \Leftrightarrow \quad R(t) = 0 \quad \Leftrightarrow \quad \frac{c_A^a c_B^b}{c_C^c c_D^d} = \frac{k_2}{k_1} = K_{\mathrm{eq}}.$$

Beispiel 10.5 (Randwertproblem)**.** Als Differentialgleichung zweiter Ordnung beschreibt

$$\frac{\mathrm{d}^2 u(x)}{\mathrm{d}x^2} = q(x)$$

die stationäre Diffusion mit einer Quelle $q(x)$. Es können *Anfangswerte*

$$u\left(x_0\right) = U_0, \quad u'\left(x_0\right) = U_0'$$

oder *Randwerte*

$$u\left(x_0\right) = U_0 \quad \text{und} \quad u\left(x_1\right) = U_1 \qquad \left(x_1 > x_0\right)$$

vorgegeben werden. Wir behandeln hier ausschließlich Anfangswertprobleme.

10.2 Charakterisierung von Differentialgleichungen

Die allgemeine implizite Differentialgleichung der Ordnung m lautet:

$$F\left(t, u(t), \frac{\mathrm{d}u(t)}{\mathrm{d}t}, \ldots, \frac{\mathrm{d}^m u(t)}{\mathrm{d}t^m}\right) = 0; \qquad t \in I.$$

Mittels

$$\left.\begin{aligned} \frac{\mathrm{d}u_0(t)}{\mathrm{d}t} &= u_1(t) \\ &\vdots \\ \frac{\mathrm{d}u_{m-1}(t)}{\mathrm{d}t} &= u_m(t) \end{aligned}\right\} \quad \frac{\mathrm{d}}{\mathrm{d}t}u_{i-1}(t) = u_i(t) = \frac{\mathrm{d}}{\mathrm{d}t}\frac{\mathrm{d}^{i-1}u(t)}{\mathrm{d}t^{i-1}} = \frac{\mathrm{d}^i u(t)}{\mathrm{d}t^i}$$

$$F\left(t, u_0(t), u_1(t), \ldots, u_m(t)\right) = 0$$

also $u_0(t) = u(t)$ und $u_i(t) = \frac{\mathrm{d}^i u(t)}{\mathrm{d}t^i}$, $i = 1, \ldots, m$, lässt sich jede Differentialgleichung m-ter Ordnung auf ein System von $m+1$ Differentialgleichungen erster Ordnung reduzieren. Daher betrachten wir ab sofort nur noch solche Systeme erster Ordnung.

Mit $u : I \to \mathbb{R}^m$ und $F : I \times \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}^m$ ist

$$F\left(t, u(t), u'(t)\right) = 0; \qquad t \in I, \tag{10.3}$$

das allgemeine *implizite* differentiell-algebraische System (*DAE*) erster Ordnung. Ist (10.3) nach $u'(t)$ auflösbar, so heißt

$$u'(t) = f\left(t, u(t)\right), \qquad t \in I,$$

ein System gewöhnlicher Differentialgleichungen in *expliziter Form*. Sind die algebraischen Nebenbedingungen in (10.3) explizit, so nimmt sie die Form

$$\begin{aligned} F\left(t, u_1(t), u_2(t), u_1'(t)\right) &= 0, \\ G\left(t, u_1(t), u_2(t)\right) &= 0, \end{aligned}$$

an. Die *semi-explizite Form* (da explizit in u_1' aber Nebenbedingung implizit in u_2) einer DAE lautet

$$\begin{aligned} u_1'(t) &= f\left(t, u_1(t), u_2(t)\right) \\ 0 &= g\left(t, u_1(t), u_2(t)\right) \end{aligned} \tag{10.4}$$

Index einer DAE: Wir betrachten die semi-explizite Form (10.4) und differenzieren die zweite Gleichung nach t:

$$\begin{aligned} u_1'(t) &= f\left(t, u_1(t), u_2(t)\right), \\ \frac{\partial g}{\partial u_1}\left(t, u_1(t), u_2(t)\right) u_1'(t) + \frac{\partial g}{\partial u_2}\left(t, u_1(t), u_2(t)\right) u_2'(t) &= -\frac{\partial g}{\partial u_t}\left(t, u_1(t), u_2(t)\right). \end{aligned} \tag{10.5}$$

Hier steht $\frac{\partial g}{\partial u_k}$ für die Jacobimatrix von g nach den Variablen u_k, d. h. $(\frac{\partial g}{\partial u_k})_{ij} = \frac{\partial g_i}{\partial u_{k,j}}$. Ist $\frac{\partial g}{\partial u_2}$ regulär, so ist (10.5) eine gewöhnliche Differentialgleichung in impliziter Form.

Falls nicht, so kann man (10.5) mit algebraischen Umformungen wieder auf die Form (10.4) bringen (mit anderen Unbekannten u_1 und u_2) und den Prozess wiederholen, solange, bis man eine gewöhnliche Differentialgleichung erhält. Die Anzahl der benötigten Schritte bezeichnet den *Index*. DAEs mit Index ≥ 2 sind numerisch deutlich schwieriger zu lösen als solche mit Index ≤ 1 (hierbei ist eine DAE vom Index 0 eine gewöhnliche Differentialgleichung in impliziter Form).

Wir werden hier DAEs nicht weiter behandeln und verweisen auf das Buch (BCP96).

10.3 Zur Theorie gewöhnlicher Differentialgleichungen

Definition 10.6 (Anfangswertproblem)**.** Zu einem gegebenen Punkt $(t_0, u_0) \in D = I \times \Omega \subset \mathbb{R} \times \mathbb{R}^m$ ist eine stetig differenzierbare Funktion $u : I \to \mathbb{R}^m$ gesucht, so dass

1. $\text{Graph}(u) := \{(t, u(t)) \mid t \in I\} \subset D$,
2. $u'(t) = f\,(t, u(t))\,; \quad t \in I$,
3. $u\,(t_0) = u_0$. □

Nach dem Hauptsatz der Differential- und Integralrechnung gilt für u:

$$\int_{t_0}^{t} u'(s)ds = \int_{t_0}^{t} f\,(s, u(s))\,ds \quad \Leftrightarrow \quad u(t) = u\,(t_0) + \int_{t_0}^{t} f\,(s, u(s))\,ds \quad \forall t \in I$$

Eine Lösung u der Differentialgleichung ist, wenn sie existiert, mindestens einmal stetig differenzierbar. Die rechte Seite f sollte demnach mindestens stetig sein. Dass umgekehrt aus der Stetigkeit auch die Existenz einer Lösung folgt zeigt der folgende Satz.

Satz 10.7 (Existenzsatz von Peano)**.** $f(t, x)$ sei stetig auf der Menge

$$D = \left\{(t, x) \in \mathbb{R} \times \mathbb{R}^m \middle| \, |t - t_0| \leq \alpha; \|x - u_0\| \leq \beta\right\}.$$

Dann existiert eine Lösung $u(t)$ des Anfangswertproblems auf dem Intervall $I = [t_0 - T, t_0 + T]$, wobei

$$T := \min\left(\alpha, \frac{\beta}{M}\right) \quad \text{mit} \quad M := \max_{(t,x)\in D} \|f\,(t, x)\|\,.$$

Beweis. Siehe (Ran) □

Satz 10.8 (Fortsetzungssatz)**.** Sei $f(t, x)$ stetig auf einem beschränkten Zylinder $D \subset \mathbb{R} \times \mathbb{R}^m$ mit $t_0, u_0) \in D$. Dann lässt sich die Lösung aus Satz 10.7 bis auf den Rand von D fortsetzen.

Beispiel 10.9. Wir betrachten zwei Beispiele:

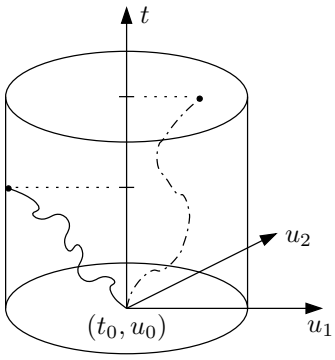
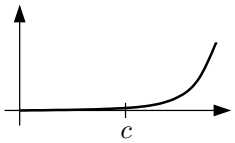


Abbildung 27: Skizze zu Satz 10.8

1. $u'(t) = \sqrt{u(t)},\ t \geq 0;\ u(0) = 0$ hat die Lösungen

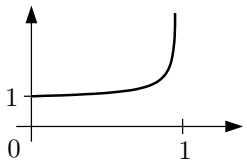
$$u_c(t) = \begin{cases} 0 & \text{für } 0 \leq t \leq c \\ \frac{1}{4}(t-c)^2 & \text{für } t > c \end{cases}$$



$\Rightarrow$ überabzählbar viele Lösungen!

2. $u'(t) = u^2(t),\ t \in [0,1]\,;\ u(0) = 1$ hat die Lösung

$$u(t) = \frac{1}{1-t}$$



$\Rightarrow$ Lösung geht gegen ∞ für $t \to 1$. Dieses Beispiel zeigt, dass Lösungen eventuell nur bis zu einer endlichen Zeit fortsetzbar sind.

Definition 10.10 (Lipschitzbedingung)**.** $f(t,x) : \mathbb{R} \times \mathbb{R}^m \to \mathbb{R}^m$ genügt einer Lipschitzbedingung, falls

$$\|f(t,x) - f(t,y)\| \leq L(t)\,\|x-y\| \qquad \forall\,(t,x)\,,(t,y) \in D$$

Hier ist $\|.\|$ die euklidsche Norm und $L := \sup_{t\in I} L(t)$. □

Satz 10.11 (Stabilitätssatz)**.** Mit zwei stetigen Funktionen f und g betrachten wir die beiden Anfangswertpobleme

$$\begin{aligned} u'(t) &= f\,(t,u(t))\,,\ t \in I; \qquad u\,(t_0) = u_0 \\ v'(t) &= g\,(t,v(t))\,,\ t \in I; \qquad v\,(t_0) = v_0 \end{aligned}$$

f genüge einer Lipschitzbedingung wie in Definition 10.10. Hier ist L die Lipschitzkonstante (unabhängig von t). Dann gilt für zwei beliebige Lösungen u und v:

$$\|u(t) - v(t)\| \leq e^{L(t-t_0)} \left\{ \|u_0 - v_0\| + \int_{t_0}^{t} \varepsilon(s)ds \right\};\quad t \in I$$

mit $\varepsilon(t) := \sup_{x\in\Omega} \|f(t,x) - g(t,x)\|$.

Beweis. Nach (Ran). Mit den integralen Lösungsdarstellungen gilt ($t \in I$):

$$\begin{aligned} u(t) - v(t) &= u(t_0) + \int_{t_0}^{t} f(s, u(s))\, ds - v(t_0) - \int_{t_0}^{t} g(s, v(s))\, ds \\ &= \int_{t_0}^{t} f(s, u(s)) - f(s, v(s))\, ds + \int_{t_0}^{t} f(s, v(s)) - g(s, v(s))\, ds + u_0 - v_0 \end{aligned}$$

Unter Beachtung von (dies ist trivial für skalares e, nicht jedoch für vektorwertiges e)

$$\begin{aligned} \left\| \int_{t_0}^{t} e(s) ds \right\| &= \lim_{N\to\infty} \left\| \sum_{i=1}^{N} e(t_i)(t_i - t_{i-1}) \right\| \\ &\leq \lim_{N\to\infty} \sum_{i=1}^{N} \|e(t_i)\| (t_i - t_{i-1}) \\ &= \int_{t_0}^{t} \|e(s)\|\, ds \end{aligned}$$

folgt also für $e(t) = u(t) - v(t)$:

$$\begin{aligned} \|e(t)\| &\leq \int_{t_0}^{t} \|f(s, u(s)) - f(s, v(s))\|\, ds + \int_{t_0}^{t} \|f(s, v(s)) - g(s, v(s))\|\, ds + \|u_0 - v_0\| \\ &\leq L \int_{t_0}^{t} \|e(s)\|\, ds + \int_{t_0}^{t} \varepsilon(s)\, ds + \|u_0 - v_0\| \end{aligned}$$

Die Behauptung ergibt sich von hier aus mit Lemma 10.12. □

Lemma 10.12 (Gronwall)**.** Es sei

- $w(t) \geq 0$ stückweise stetig
- $a(t) \geq 0$ integrabel
- $b(t) \geq 0$ nicht fallend

und

$$w(t) \leq \int_{t_0}^{t} a(s)w(s)ds + b(t); \qquad t \geq t_0.$$

Dann gilt:

$$w(t) \leq \exp\left(\int_{t_0}^{t} a(s)ds \right) b(t); \qquad t \geq t_0$$

Beweis. Nach (Ran). Setze

$$\begin{aligned} \varphi(t) &= \int_{t_0}^{t} a(s)w(s)ds \\ \psi(t) &= w(t) - \int_{t_0}^{t} a(s)w(s)ds \leq b(t) \end{aligned}$$

Für $\varphi(t)$ gilt offensichtlich

$$\varphi'(t) = a(t)w(t); \quad \varphi(t_0) = 0$$

also:

$$a(t)\psi(t) = a(t)w(t) - a(t)\int_{t_0}^{t} a(s)w(s)ds = \varphi'(t) - a(t)\varphi(t)$$

Damit ist $\varphi(t)$ Lösung des linearen Anfangswertproblems. Es ist also

$$\begin{aligned} \varphi'(t) &= a(t)\varphi(t) + a(t)\psi(t); \qquad t \geq t_0 \\ \varphi(t_0) &= 0 \end{aligned}$$

Dieses Anfangswertproblem hat die Lösung (nachrechnen!)

$$\varphi(t) = \exp\left(\int_{t_0}^{t} a(s)ds\right) \cdot \int_{t_0}^{t} \exp\left(-\int_{t_0}^{s} a(r)dr\right) a(s)\psi(s)ds.$$

Diese schätzen wir nun ab:

$$\begin{aligned} \varphi(t) &= \exp\left(\int_{t_0}^{t} a(s)ds\right) \cdot \int_{t_0}^{t} \exp\left(-\int_{t_0}^{s} a(r)dr\right) a(s) \underbrace{\psi(s)}_{\leq b(t) \text{ da } t \geq s,\ b \text{ nicht fallend}} ds \\ &\leq b(t) \exp\left(\int_{t_0}^{t} a(s)ds\right) \cdot \int_{t_0}^{t} \exp\left(-\int_{t_0}^{s} a(r)dr\right) a(s)ds \\ &= b(t) \exp\left(\int_{t_0}^{t} a(s)ds\right) \cdot \int_{t_0}^{t} \left(-\frac{\mathrm{d}}{\mathrm{d}s} \exp\left(-\int_{t_0}^{s} a(r)dr\right)\right) ds \\ &= b(t) \exp\left(\int_{t_0}^{t} a(s)ds\right) \cdot \left[-\exp\left(-\int_{t_0}^{s} a(r)dr\right)\right]_{t_0}^{t} \\ &= b(t) \exp\left(\int_{t_0}^{t} a(s)ds\right) \cdot \left(-\exp\left(-\int_{t_0}^{t} a(r)dr\right) + 1\right) \\ &= b(t) \exp\left(\int_{t_0}^{t} a(s)ds\right) - b(t) \end{aligned}$$

Damit gilt schließlich:

$$w(t) \leq \int_{t_0}^{t} a(s)w(s)ds + b(t) = \varphi(t) + b(t) = b(t) \exp\left(\int_{t_0}^{t} a(s)ds\right)$$

□

Anwendung im Beweis zu Satz 10.11:

$$w(t) = \|e(t)\|; \quad a(t) = L; \quad b(t) = \int_{t_0}^{t} \varepsilon(s)ds + \|u_0 - v_0\|$$

Folgerung 10.13 (Eindeutige Lösung)**.** Bei Lipschitz-stetigem f ist die Lösung aus Satz 10.7 bzw. Satz 10.8 eindeutig.

Beweis. Seien u und v zwei Lösungen:

$$\begin{aligned} u' &= f\left(t, u\left(t\right)\right); \qquad u\left(t_0\right) = u_0 \\ v' &= f\left(t, v\left(t\right)\right); \qquad v\left(t_0\right) = v_0 = u_0 \end{aligned}$$

Dann gilt nach Satz 10.11 wegen $\varepsilon(s) = 0$ und $u_0 - v_0 = 0$ auch $\|u(t) - v(t)\| = 0$. □

11 Einschrittverfahren

11.1 Eulersches Polygonzugverfahren (explizites Eulerverfahren)

Wir betrachten das Anfangswertproblem für $u : I \to \mathbb{R}^m$

$$\begin{aligned} u'(t) &= f(t, u(t)), \qquad t \in I = [t_0, t_0 + T] \\ u(t_0) &= u_0 \end{aligned} \tag{11.1}$$

$f : \mathbb{R} \times \mathbb{R}^m \to \mathbb{R}^m$ sei stetig und gehorche einer Lipschitzbedingung im zweiten Argument. Nach Folgerung 10.13 ist die Lösung also eindeutig.

Zur numerischen Lösung wählen wir eine Unterteilung

$$t_0 < t_1 < \ldots < t_n < \ldots < t_N = t_0 + T$$

und setzen

$$I_n := [t_{n-1}, t_n], \quad h_n := t_n - t_{n-1}, \quad h = \max_{1 \le n \le N} h_n.$$

Das explizite Eulerverfahren ist gegeben durch die Vorschrift

$$y_n^h = y_{n-1}^h + h_n f\left(t_{n-1}, y_{n-1}^h\right); \qquad n = 1, \ldots, N. \tag{11.2}$$

y_n^h approximiert $u(t_n)$ und $y^h = \left(y_0^h, y_1^h, \ldots, y_N^h\right)^T$ fasst alle Werte in einem Vektor (*Gitterfunktion*) zusammen. L_h ist ein *Differenzenoperator* gegeben durch

$$\left(L_h y^h\right)_n := h_n^{-1}\left(y_n^h - y_{n-1}^h\right) - f\left(t_{n-1}, y_{n-1}^h\right), \qquad n = 1, \ldots, N.$$

Damit ist (11.2) äquivalent zu $L_h y^h = 0$.
Anmerkung: L_h ist nicht unbedingt linear, da f nicht linear sein muss.

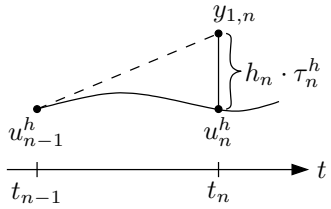
Nun wollen wir die Konvergenz des Verfahrens zeigen. Dazu definieren wir den *lokalen Abschneidefehler*:

$$\tau_n^h = \left(L_h u^h\right)_n = h_n^{-1}\left(u_n^h - u_{n-1}^h\right) - f\left(t_{n-1}, u_{n-1}^h\right) \tag{11.3}$$

Hierbei ist $u^h = \left(u(t_0), u(t_1), \ldots, u(t_N)\right)^T$ die exakte Lösung von (11.1) ausgewertet an den Gitterpunkten. Beachte, dass τ_n^h im Systemfall ein Vektor mit m Komponenten ist. Eine zweite Interpretation von τ_n^h ist die folgende:

$$\tau_n^h = h_n^{-1}\Bigg[u_n^h - \underbrace{\left(u_{n-1}^h + h_n f\left(t_{n-1}, u_{n-1}^h\right)\right)}_{=:\, y_{1,n}}\Bigg] \tag{11.4}$$

$y_{1,n}$ entsteht durch einen Schritt des Eulerverfahrens ausgehend von u_{n-1}^h.



Für den lokalen Abschneidefehler des expliziten Eulerverfahrens gilt ((Ran)):

$$\begin{aligned}
\tau_n^h &= h_n^{-1}\left(u_n^h - u_{n-1}^h\right) - f\left(t_{n-1}, u_{n-1}^h\right) \\
&= h_n^{-1}\int_{t_{n-1}}^{t_n} u'(t)dt - u'\left(t_{n-1}\right) \\
&= h_n^{-1}\left\{u'\left(t_n\right)t_n - u'\left(t_{n-1}\right)t_{n-1} - \int_{t_{n-1}}^{t_n} tu''(t)dt\right\} - u'\left(t_{n-1}\right) \\
&= h_n^{-1}\left\{u'\left(t_n\right)t_n - u'\left(t_{n-1}\right)t_{n-1} - \left(t_n - t_{n-1}\right)u'\left(t_{n-1}\right) - \int_{t_{n-1}}^{t_n} tu''(t)dt\right\} \\
&= h_n^{-1}\left\{t_n\left[u'\left(t_n\right) - u'\left(t_{n-1}\right)\right] - \int_{t_{n-1}}^{t_n} tu''(t)dt\right\} \\
&= h_n^{-1}\int_{t_{n-1}}^{t_n}\left(t_n - t\right)u''(t)dt
\end{aligned}$$

Damit erhält man als Abschätzung für die Norm:

$$\begin{aligned}
\left\|\tau_n^h\right\| &= \left\|h_n^{-1}\int_{t_{n-1}}^{t_n}\left(t_n - t\right)u''(t)dt\right\| \\
&\leq h_n^{-1}\int_{t_{n-1}}^{t_n}\left(t_n - t\right)\left\|u''(t)\right\|dt \\
&\leq h_n^{-1}\max_{t\in I_n}\left\|u''(t)\right\|\underbrace{\int_{t_{n-1}}^{t_n}\left(t_n - t\right)}_{=\frac{1}{2}h_n^2} \\
&= \frac{1}{2}h_n\max_{t\in I_n}\left\|u''(t)\right\|
\end{aligned}$$

Zur Abschätzung des globalen Fehlers $e_n^h = y_n^h - u_n^h$ leitet man eine Rekursionsgleichung

für den Fehler her.

$$\begin{aligned} e_n^h &= y_n^h - u_n^h \\ &= \underbrace{y_{n-1}^h + h_n f\left(t_{n-1}, y_{n-1}^h\right)}_{\text{Formel für } y_n^h} - \underbrace{\left(u_n^h - u_{n-1}^h - h_n f\left(t_{n-1}, u_{n-1}^h\right)\right)}_{h_n \tau_n^h} \\ &\quad \underbrace{-u_{n-1}^h - h_n f\left(t_{n-1}, u_{n-1}^h\right)}_{\text{Ergänzung abziehen}} \\ &= \underbrace{y_{n-1}^h - u_{n-1}^h}_{e_{n-1}^h} + h_n \left(f\left(t_{n-1}, y_{n-1}^h\right) - f\left(t_{n-1}, u_{n-1}^h\right)\right) - h_n \tau_n \end{aligned}$$

Mit Lipschitzbedingung und Dreiecksungleichung gilt:

$$\left\|e_n^h\right\| \leq \left\|e_{n-1}^h\right\| + h_n L \left\|e_{n-1}^h\right\| + h_n \left\|\tau_n\right\|$$

Abspulen der Rekursion:

$$\left\|e_n^h\right\| \leq \left\|e_0^h\right\| + L \sum_{\nu=0}^{n-1} h_{\nu+1} \left\|e_\nu^h\right\| + \sum_{\nu=1}^{n} h_\nu \left\|\tau_\nu\right\| \tag{11.5}$$

Dies ist vergleichbar mit der Stabilität im Kontinuierlichen und erfordert ein *diskretes Gronwall–Lemma*.

Lemma 11.1 (Diskreter Gronwall)**.** Seien $(w_n)_{n\geq 0}$, $(a_n)_{n\geq 0}$ und $(b_n)_{n\geq 0}$ Folgen nichtnegativer Zahlen, für die gilt:

$$\begin{aligned} w_0 &\leq b_0, \\ w_n &\leq \sum_{\nu=0}^{n-1} a_\nu w_\nu + b_\nu, \qquad n \geq 1. \end{aligned} \tag{11.6}$$

Ist $(b_n)_{n\geq 0}$ nicht fallend, so gilt die Abschätzung

$$w_n \leq \exp\left(\sum_{\nu=0}^{n-1} a_\nu\right) b_n, \qquad n \geq 1. \tag{11.7}$$

Beweis. Nach (Ran). Definiere zwei Folgen (S_n) und (d_n):

$$\begin{aligned} S_n &= \sum_{\nu=0}^{n-1} a_\nu w_\nu + b_n & S_0, &= b_0, \\ d_n &= S_n - w_n, & d_0 &= b_0 - w_0. \end{aligned}$$

Das heißt, S_n ist die rechte Seite von (11.6) und d_n ist die Differenz von rechter und linker Seite in (11.5). Dann gilt für $n \geq 1$:

$$S_n - S_{n-1} = \sum_{\nu=0}^{n-1} a_\nu w_\nu + b_n - \sum_{\nu=0}^{n-2} a_\nu w_\nu + b_{n-1} = a_{n-1} w_{n-1} + b_n - b_{n-1}.$$

Mittels Induktion über n zeigen wir

$$S_n \leq \exp\left(\sum_{\nu=0}^{n-1} a_\nu\right) b_n$$

1. $n = 0$: $S_0 = b_0 = e^0 b_0 \quad \surd$

2. $n - 1 \to n$:

$$\begin{aligned}
S_n &= S_{n-1} + a_{n-1} \underbrace{w_{n-1}}_{\leq S_{n-1}} + b_n - b_{n-1} \\
&\leq (1 + a_{n-1})\, S_{n-1} + b_n - b_{n-1} \\
&\leq \underbrace{(1 + a_{n-1}) \exp\left(\sum_{\nu=0}^{n-2} a_\nu\right)}_{\geq 1} b_{n-1} + \underbrace{b_n - b_{n-1}}_{\geq 0} \\
&\leq (1 + a_{n-1}) \exp\left(\sum_{\nu=0}^{n-2} a_\nu\right) (b_{n-1} + b_n - b_{n-1}) \\
&\leq \exp\left(a_{n-1}\right) \exp\left(\sum_{\nu=0}^{n-2} a_\nu\right) b_n \\
&= \exp\left(\sum_{\nu=0}^{n-1} a_\nu\right) b_n
\end{aligned}$$

Im vorletzten Schritt wurde die Reihenentwicklung von e^x eingesetzt:

$$e^x = \sum_{k=0}^{\infty} \frac{x^k}{k!} = 1 + x + \sum_{k=2}^{\infty} \frac{x^k}{k!} \geq 1 + x$$

□

Anwendung auf (11.5):

$$\underbrace{\left\|e_n^h\right\|}_{w_n} \leq \sum_{\nu=0}^{n-1} \underbrace{Lh_{\nu+1}}_{a_\nu} \underbrace{\left\|e_\nu^h\right\|}_{w_\nu} + \underbrace{\sum_{\nu=1}^{n} h_\nu \left\|\tau_\nu\right\| + \underbrace{\left\|e_0^h\right\|}_{b_0}}_{b_n}$$

$$\leq \exp\left(\underbrace{\sum_{\nu=0}^{n-1} Lh_{\nu+1}}_{L(t_n - t_0)}\right) \left\{ \left\|e_0^h\right\| + \sum_{\nu=1}^{n} h_\nu \left\|\tau_\nu\right\| \right\}$$

Also:

$$\max_{1\leq n\leq N} \left\|e_n^h\right\| \leq \left\|e_N^h\right\| = e^{LT} \left\{ \left\|e_0^h\right\| + \underbrace{\max_{1\leq \nu\leq N} \left\|\tau_\nu\right\|}_{\leq \frac{h}{2} \max_{t\in I} \|u''(t)\|} T \right\}$$

Damit gilt die a-priori-Fehlerabschätzung

$$\max_{1\leq n\leq N} \left\|e_n^h\right\| \leq e^{LT} \left\{ \left\|e_0^h\right\| + \frac{T}{2} h \max_{t\in I} \left\|u''(t)\right\| \right\} \quad = O(h) \tag{11.8}$$

Zusammenfassend gilt für das explizite Eulerverfahren:

- Die globale Konvergenzordnung ist gleich der lokalen Konvergenzordnung, nämlich $O(h)$.
- Das Resultat (11.8) erfordert höhere Differenzierbarkeit der Lösung u.
- Ansonsten ist nur die Lipschitz-Stetigkeit von f erforderlich.

11.2 Taylor- und Runge-Kutta-Verfahren

Ziel: Höhere Konvergenzordnung $O\left(h^m\right)$.
Idee: Taylorentwicklung der Lösung:

$$u(t) = \sum_{r=0}^{R} \frac{h^r}{r!} u^{(r)}(t-h) + \frac{h^{R+1}}{(R+1)!} u^{(R+1)}(\xi); \qquad \xi \in [t-h, t] \tag{11.9}$$

Wegen $u'(t) = f(t, u(t))$ gilt $u^{(r)}(t) = \frac{\partial^{r-1}}{\partial t^{r-1}} f(t, u(t)) =: f^{(r-1)}(t, u(t))$. Das R-stufige Taylorverfahren entsteht durch Weglassen des Restglieds:

$$y_n = y_{n-1} + h_n \underbrace{\sum_{r=1}^{R} \frac{h_n^{r-1}}{r!} f^{(r-1)}\left(t_{n-1}, y_{n-1}\right)}_{F(h_n; t_{n-1}, y_n, y_{n-1})}; \qquad n \geq 1 \tag{11.10}$$

$F(h_n; t_{n-1}, x, y)$ ist die allgemeine *Verfahrensfunktion*, die auch den impliziten Fall beinhaltet. Dies erfordert das Lösen eines nichtlinearen Gleichungssystems. Analog zu oben definiert man den lokalen Abschneidefehler:

$$\tau_n^h := \left(L_h u^h\right)_n = h_n^{-1}\left(u_n^h - u_{n-1}^h\right) - F\left(h_n; t_{n-1}, u_n^h, u_{n-1}^h\right) \tag{11.11}$$

Definition 11.2 (Konsistenz)**.** Das Einschrittverfahren (11.10) heißt *konsistent* mit dem Anfangswertproblem, falls

$$\max_{1\le n\le N_h} \left\|\tau_n^h\right\| \to 0 \quad \text{für} \quad h \to 0$$

bzw. *konsistent mit Konsistenzordnung* m, falls

$$\max_{1\le n\le N_h} \left\|\tau_n^h\right\| = O\left(h^m\right) \quad \text{für} \quad h \to 0.$$

□

Bemerkung 11.3. Durch Einsetzen von (11.9) in τ_n^h erhält man für Taylorverfahren:

$$\tau_n^h \overset{(11.9)}{=} h_n^{-1}\Bigg(\underbrace{u_{n-1}^h + h_n \sum_{r=1}^{R} \frac{h_n^{r-1}}{r!} f^{(r-1)}\left(t_{n-1}, u_{n-1}^h\right) + \frac{h_n^{R+1}}{(R+1)!} u^{(R+1)}(\xi)}_{= u_n^h \text{ wegen (11.9)}} - u_{n-1}^h\Bigg)$$
$$- \sum_{r=1}^{R} \frac{h_n^{r-1}}{r!} f^{(r-1)}\left(t_{n-1}, u_{n-1}^h\right) = O\left(h_n^R\right)$$

also $m = R$. □

Die praktische Realisierung des Verfahrens erfordert die Berechnung höherer Ableitungen $\frac{\partial^r}{\partial t^r} f(t, u(t))$. Dies ist für jedes Anfangswertproblem extra durchzuführen.

Skalarer Fall: $f(t,x) : \mathbb{R}\times\mathbb{R} \to \mathbb{R}$. Anwendung der Kettenregel:

$$\begin{aligned} f^{(1)}\big(t, u(t)\big) &= \frac{\mathrm{d}}{\mathrm{d}t} f\big(t, u(t)\big) \\ &= \frac{\partial f}{\partial t}\big(t, u(t)\big) + \frac{\partial f}{\partial u}\big(t, u(t)\big) \cdot \frac{\mathrm{d}}{\mathrm{d}t} u(t) \\ &= f_t\big(t, u(t)\big) + f_x\big(t, u(t)\big) f\big(t, u(t)\big). \end{aligned}$$

Es sind also keine Ableitungen der Lösung u erforderlich.

$$\begin{aligned} f^{(2)}\big(t, u(t)\big) &= \frac{\mathrm{d}}{\mathrm{d}t} f'\big(t, u(t)\big) \\ &= \Big(f_{tt} + f_{tx} f + \left(f_{xt} + f_{xx} f\right) f + f_x\left(f_t + f_x f\right)\Big)(t, u(t)) \\ &= \Big(f_{tt} + 2 f_{tx} f + f_{xx} f^2 + f_x f_t + \left(f_x\right)^2 f\Big)\,(t, u(t)). \end{aligned}$$

- Sehr aufwändig!
- Für Systeme ist f_x eine Jacobimatrix.

Alternativ verwendet man numerische Differentiation, zum Beispiel:

$$\begin{aligned} f^{(1)}\big(t,u(t)\big) &= h^{-1}\left\{f\big(t+h,u(t+h)\big) - f\big(t,u(t)\big)\right\} + O(h) \\ &= h^{-1}\left\{f\Big(t+h,u(t)+hf\big(t,u(t)\big)+O\left(h^2\right)\Big) - f\big(t,u(t)\big)\right\} + O(h) \\ &= h^{-1}\left\{f\Big(t+h,u(t)+hf\big(t,u(t)\big)\Big) - f\big(t,u(t)\big)\right\} + O(h) \end{aligned}$$

Dies legt die folgende allgemeine Form nahe:

Definition 11.4 (explizite Runge-Kutta-Verfahren)**.** Explizite Runge-Kutta-Verfahren haben die Form

$$y_n = y_{n-1} + h_n F(h_n; t_{n-1}, y_{n-1}), \qquad F\left(h;t,x\right) = \sum_{r=1}^{R} c_r k_r\left(h;t,x\right)$$

$$\text{mit} \quad k_1 = f(t,x) \quad \text{und} \quad k_r = f\left(t + ha_r, x + h\sum_{s=1}^{r-1} b_{rs}k_s\right); \quad r = 2,\ldots,R$$

mit geeignet gewählten Konstanten a_r, b_{rs} und c_r. Diese Konstanten bestimmt man so, dass die Konvergenzordnung möglichst groß wird. □

Beispiel 11.5 (Beispiele für unterschiedliche R)**.**

$R = 1$: Explizites Eulerverfahren. Für Konsistenz ist $c_r = 1$ notwendig. Das explizite Eulerverfahren ist also das einzige explizite Runge-Kutta-Verfahren der Stufe 1.

$R = 2$: Taylorentwicklung um den Punkt (t,x) ergibt:

$$\begin{aligned} F\left(h;t,x\right) &= c_1 f\left(t,x\right) + c_2 f\left(t + a_2 h, x + h b_{21} f\left(t,x\right)\right) \\ &= c_1 f\left(t,x\right) + c_2 f\left(t,x\right) + c_2 a_2 h f_t\left(t,x\right) + c_2 h b_{21} f\left(t,x\right) f_x\left(t,x\right) + O\left(h^2\right) \\ &\overset{!}{=} f + \frac{h}{2}\underbrace{\left(f_t + f_x f\right)}_{=\,f'} + O\left(h^2\right) \end{aligned}$$

Koeffizientenvergleich ergibt

$$c_1 + c_2 = 1, \quad c_2 a_2 = \frac{1}{2}, \quad c_2 b_{21} = \frac{1}{2}$$

also 3 Bedingungen für 4 Parameter ($\rightarrow$ nicht eindeutig). Eine Möglichkeit ist

$$c_1 = c_2 = \frac{1}{2}, \quad a_2 = b_{21} = 1 \qquad (\textit{Verfahren von Heun})$$

□

Für die maximal erreichbare Ordnung in Abhängigkeit der Stufenzahl kann man zeigen, siehe (Sim) :

Stufenzahl R	1	2	3	4	5	6	7	8
max. Ordnung	1	2	3	4	4	5	6	6

Da mit $R = 5$ auch nur Ordnung 4 erreichbar ist, benutzt eines der beliebtesten Runge-Kutta-Verfahren die Stufenzahl 4.

Für Systeme ist die Sache noch komplizierter. Es können die Konstanten aus dem skalaren Fall verwendet werden, aber ab $R \geq 5$ ist die Ordnung im Allgemeinen verringert (gegenüber dem skalaren Fall).

Beispiel 11.6 (implizite Verfahren)**.** Der Vollständigkeit wegen wollen wir auch einige implizite Einschrittverfahren benennen. Der nachfolgende Satz ist auch auf implizite Verfahren anwendbar.

implizites Eulerverfahren $y_n = y_{n-1} + h_n f(t_n, y_n)$

Trapezregel $y_n = y_{n-1} + \frac{h_n}{2}\left(f(t_n, y_n) + f(t_{n-1}, y_{n-1})\right)$

Mittelpunktregel $y_n = y_{n-1} + h_n f\left(t_n + \frac{1}{2}h_n, \frac{1}{2}(y_{n-1} + y_n)\right)$

□

11.3 Konvergenz von Einschrittverfahren

Wir zeigen hier die Konvergenz für konsistente explizite als auch implizite Einschrittverfahren.

Satz 11.7 (Konvergenz von Einschrittverfahren)**.** Die Funktion f sei Lipschitz-stetig im zweiten Argument mit Konstante L und das Verfahren sei konsistent mit Ordnung m. Dann gilt:

$$\max_{1\leq n\leq N_h} \left\| u_n - y_n^h \right\| \leq c\frac{e^{LT}-1}{L} h_{\max}^m; \qquad h_{\max} = \max_{1\leq n\leq N_h} h_n.$$

Beweis. Nach (Sim). Die Verfahrensfunktion deckt auch den impliziten Fall ab: $y_n = y_{n-1} + h_n F(h_n; t_{n-1}, y_n, y_{n-1})$.

Für den Parameter n, $0 \leq n \leq N$, definieren wir die Funktionen $u_n(t)$ für $t \geq t_n$ als Lösung des Anfangswertproblems

$$u_n'(t) = f(t, u_n(t)), \qquad u_n(t_n) = y_n,$$

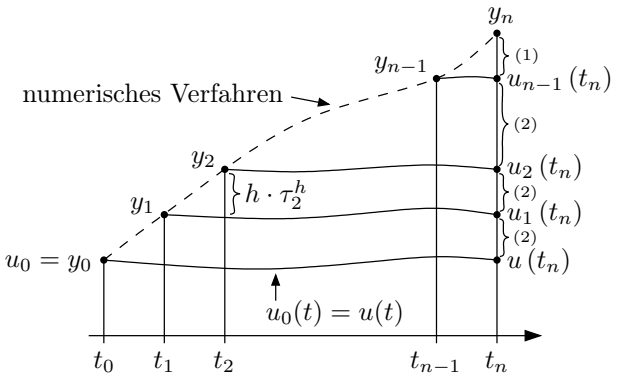
d. h. für u_n dient die numerische Lösung y_n zur Zeit t_n als Startwert.

Dann gilt für den Fehler e_n:

$$\begin{aligned} e_n &= u(t_n) - y_n \\ &= \underbrace{u_0(t_n)}_{=u} - u_1(t_n) + u_1(t_n) \ldots - u_{n-1}(t_n) + u_{n-1}(t_n) - y_n \\ &= \sum_{i=0}^{n-2} [u_i(t_n) - u_{i+1}(t_n)] + u_{n-1}(t_n) - y_n \end{aligned}$$

Normen bilden und Dreiecksungleichung anwenden liefert:

$$\|e_n\| \leq \Big\| \underbrace{u_{n-1}(t_n) - y_n}_{(1)} \Big\| + \sum_{i=0}^{n-2} \Big\| \underbrace{u_i(t_n) - u_{i+1}(t_n)}_{(2)} \Big\|$$

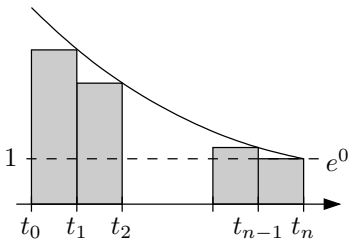


Mit Satz 10.11 gilt:

$$\begin{aligned}
\|e_n\| &\leq \|u_{n-1}(t_n) - y_n\| + \sum_{i=0}^{n-2} e^{L(t_n - t_{i+1})} \|u_i(t_{i+1}) - y_{i+1}\| \\
&= \sum_{i=0}^{n-1} e^{L(t_n - t_{i+1})} \|u_i(t_{i+1}) - y_{i+1}\| \\
&= \sum_{i=0}^{n-1} e^{L(t_n - t_{i+1})} \left\| h_{i+1} \tau_{i+1}^h \right\|
\end{aligned}$$

Mit Konsistenz: $\left\|\tau_n^h\right\| \leq c h_n^m \leq c h_{\max}^m$

$$\begin{aligned}
\|e_n\| &\leq c h_{\max}^m \sum_{i=0}^{n-1} h_{i+1} e^{L(t_n - t_{i+1})} \\
&\leq c h_{\max}^m \int_{t_0}^{t_n} e^{L(t_n - t)} dt \\
&= c h_{\max}^m \left[-L^{-1} e^{L(t_n - t)} \right]_{t_0}^{t_n} \\
&= c h_{\max}^m \left(-L^{-1} e^0 + L^{-1} e^{L(t_n - t_0)} \right) \\
&= c \frac{e^{L(t_n - t_0)} - 1}{L} h_{\max}^m.
\end{aligned}$$



Wegen $\|e_{n-1}\| \leq \|e_n\|$ gilt $\max_{1 \leq n \leq N} \|e_n\| \leq c \frac{e^{LT} - 1}{L} h_{\max}^m$. □

Bemerkung 11.8.

- Dies gilt für alle expliziten und impliziten Einschrittverfahren.

- Aus Konsistenz folgt globale Konvergenz mit der selben Ordnung.
- Die Konstante e^{LT} ist ziemlich pessimistisch. Es existieren andere Beweise mit optimistischeren Resultaten, aber weiteren Voraussetzungen.
- Konvergenz gilt für festes T und $h \to 0$.

□

12 Schrittweitensteuerung für Einschrittverfahren

Problemstellung: Wie wählt man die Schrittweite h_n, so dass der globale Fehler $e_n = y_n - u(t_n)$ kontrolliert wird? Das Ziel ist also

$$\max_{t_n \in I} \|e_n\| \leq \text{TOL},$$

wobei TOL eine vorgegebene Zahl ist.

Die Darstellung in diesem Abschnitt folgt (Ran).

12.1 Ein anderer Zugang zur Konvergenz

Definition 12.1 (L-Stetigkeit)**.** Ein Einschrittverfahren heißt *L-stetig*, falls für eine Verfahrensfunktion F gilt:

$$\|F(h;t,x,y) - F(h;t,\tilde{x},\tilde{y})\| \leq L\{\|x-\tilde{x}\| + \|y-\tilde{y}\|\}$$

für beliebige Punkte (t,x), $(t,\tilde{x})$, (t,y), $(t,\tilde{y}) \in D$.

x und $\tilde{x}$ entfallen bei expliziten Verfahren. Runge-Kutta-Verfahren sind immer L-stetig, wenn f Lipschitz-stetig ist.

Satz 12.2 (Konvergenzsatz)**.** Das Einschrittverfahren sei L-stetig und konsistent. Dann ist das Verfahren konvergent, falls $\|y_0 - u(t_0)\| \to 0$, und es gilt die a–priori–Fehlerabschätzung

$$\|y_n - u(t_n)\| \leq e^{L(t_n - t_0)} \left\{ \|y_0 - u_0\| + \sum_{\nu=1}^{n} h_\nu \|\tau_\nu\| \right\}, \quad 0 \leq n \leq N, \tag{12.1}$$

wobei im impliziten Fall $h \leq \frac{1}{2}L^{-1}$ erforderlich ist.

Beweis.

1. Einschrittverfahren:

$$y_n = y_{n-1} + h_n F(h_n; t_{n-1}, y_n, y_{n-1})$$

2. Definition des lokalen Abschneidefehlers:

$$\begin{aligned} \tau_n &= \left(L_h u^h\right) = h_n^{-1}(u_n - u_{n-1}) - F(h_n; t_{n-1}, u_n, u_{n-1}) \\ \Leftrightarrow u_n &= u_{n-1} + h_n F(h_n; t_{n-1}, u_n, u_{n-1}) + h_n \tau_n \end{aligned}$$

Differenz liefert

$$y_n - u_n = y_{n-1} - u_{n-1} + h_n\left(F(h_n; t_{n-1}, y_n, y_{n-1}) - F(h_n; t_{n-1}, u_n, u_{n-1})\right) - h_n\tau_n$$

Normen bilden, Dreiecksungleichung und L-Stetigkeit anwenden:

$$\begin{aligned} \|e_n\| &\leq \|e_{n-1}\| + h_n L \left(\|e_n\| + \|e_{n-1}\|\right) + h_n \tau_n \\ \Leftrightarrow \|e_n\| &\leq \frac{1+h_n L}{1-h_n L} \|e_{n-1}\| + \frac{1}{1-h_n L} h_n \|\tau_n\| \end{aligned}$$

Mit

$$\frac{1+h_n L}{1-h_n L} = \frac{1-h_n L}{1-h_n L} + \frac{2h_n L}{1-h_n L} \leq 1 + h_n L + \frac{(h_n L)^2}{2!} + \frac{(h_n L)^3}{3!} + \ldots = e^{h_n L}$$

wegen

$$\frac{2h_n L}{1-h_n L} \leq h_n L \;\Leftrightarrow\; h_n L \leq \frac{1}{2} \wedge 1 - h_n L > 0 \;\Leftrightarrow\; h_n \leq \frac{1}{2} L^{-1}$$

und

$$\frac{1}{1-h_n L} \leq \frac{1}{1-h_n L} + \frac{h_n L}{1-h_n L} = \frac{1+h_n L}{1-h_n L} \leq e^{h_n L}$$

erhält man:

$$\|e_n\| \leq e^{h_n L} \left(\|e_{n-1}\| + h_n \|\tau_n\|\right)$$

Rekursive Anwendung liefert

$$\begin{aligned} \|e_n\| &= \underbrace{\left(\prod_{k=1}^{n} e^{h_k L}\right)}_{= e^{L(t_n - t_0)}} \|e_0\| + \sum_{\nu=1}^{n} \underbrace{\left(\prod_{k=\nu}^{n} e^{h_k L}\right)}_{\leq e^{L(t_n - t_0)}} h_\nu \|\tau_\nu\| \\ &\leq e^{L(t_n - t_0)} \left(\|e_0\| + \sum_{\nu=1}^{n} h_\nu \|\tau_\nu\|\right) \end{aligned}$$

□

Auch hieraus folgt die globale Konvergenz. Allerdings ist die Bedingung $h_n \leq \frac{1}{2} L^{-1}$ unschön und nicht notwendig.

Im expliziten Fall gilt

$$\|e_n\| \leq (1 + h_n L) \|e_{n-1}\| + h_n \|\tau_n\| \text{ und } (1 + h_n L) \leq e^{h_n L}.$$

Aus Satz 12.2 folgt, dass der lokale Abschneidefehler τ_n den globalen Fehler kontrolliert. Also sind Abschätzungen von τ_n notwendig. Bei Taylorverfahren gilt auf Grund der Konstruktion (vergleiche Bemerkung 11.3):

$$\|\tau_n\| \leq \frac{1}{(m+1)!} h^m \max_{t_0 \leq t \leq t_n} \left\| u^{(m+1)}(t) \right\|$$

Selbst unter Ausnutzung von $u^{(m+1)}(t) = f^{(m)}(t, u(t))$ und Kenntnis von Schranken an u ist dies sehr schwierig auszuwerten und zudem ungenau. In Abschnitt 12.3 werden wir die *Methode der Schrittweitenhalbierung* verwenden, um a–posteriori, d.h. aus dem berechneten y_n, eine Näherung für τ_n zu bestimmen.

Wir gehen zunächst der Frage nach, wie bei Kenntnis von τ_n die Schrittweite h_n bestimmt werden kann.

12.2 Bestimmung der Schrittweite

In der Fehlerabschätzung (12.1) taucht die Größe $K(t_n) = e^{L(t_n - t_0)}$ auf. Dies ist oft sehr pessimistisch, etwa bei eine linearen gewöhnlichen Differentialgleichung deren Eigenwerte alle negativen Realteil haben. Im Folgenden nehmen wir an, dass K eine Konstante ist (im strengen Sinn ist also die Abschätzung nicht rigoros).

Betrachten wir den lokalen Abschneidefehler des Taylor-Verfahrens etwas genauer. Aus Bemerkung 11.3 folgt für ein m-stufiges Verfahren:

$$\begin{aligned}\tau_n^h &= \frac{h_n^m}{(m+1)!}u^{(m+1)}(t_{n-1}) + \frac{h_n^{m+1}}{(m+2)!}u^{(m+2)}(\xi') \\ &= \underbrace{\frac{u^{(m+1)}(t_n)}{(m+1)!}}_{\tau^m(t_n)} h_n^m + \underbrace{\frac{h_n^{m+1}}{(m+1)!}\left(\frac{1}{m+2}u^{(m+2)}(\xi') - u^{(m+2)}(\xi'')\right)}_{O(h_n^{m+1})}.\end{aligned}$$

Wir können den lokalen Abschneidefehler also schreiben als

$$\tau_n^h = \tau^m(t_n)\, h_n^m + O\left(h_n^{m+1}\right) \tag{12.2}$$

wobei die *Hauptabschneidefunktion* $\tau^m(t_n)$ unabhängig von h_n ist und *nur* von t_n abhängt. Für andere Einschrittverfahren, etwa Runge–Kutta–Verfahren, gelten auch solche Darstellungen.
Wir betrachten zwei Strategien zur Bestimmung der Schrittweite:

1. Verteile den Fehler gleichmäßig auf das Zeitintervall. Mit der Näherung $\tau_n^h \approx \tau^m(t_n)h_n^m$ und der Abkürzung $\tau_n^m = \tau^m(t_n)$ setzen wir

$$Kh_n^m \left\|\tau_n^m\right\| \approx \frac{\mathrm{TOL}}{T} \quad \Rightarrow \quad h_n \approx \sqrt[m]{\frac{\mathrm{TOL}}{KT\left\|\tau_n^m\right\|}}, \tag{12.3}$$

   denn dann gilt in der Fehlerabschätzung (12.1):

$$\max_{t_n \in I}\|e_n\| \approx K\sum_{t_n\in I} h_n \underbrace{\left(h_n^m\left\|\tau_n^m\right\|\right)}_{\approx \frac{\mathrm{TOL}}{KT}} \approx K\frac{\mathrm{TOL}}{T}\underbrace{\sum_{t_n\in I} h_n}_{=T} = \mathrm{TOL}$$

   Die Anzahl der Zeitschritte N berechnet sich dann zu

$$N = \sum_{t_n\in I} h_n h_n^{-1} \overset{(12.3)}{\approx} \sum_{t_n\in I} h_n \sqrt[m]{\frac{KT\left\|\tau_n^m\right\|}{\mathrm{TOL}}} = \sqrt[m]{\frac{KT}{\mathrm{TOL}}}\sum_{t_n\in I} h_n \sqrt[m]{\left\|\tau_n^m\right\|}$$

   Aus (12.2) sehen wir, dass für Taylorverfahren $\tau_n^m \doteq u^{(m+1)}(t_{n-1})$ ist, also

$$N \approx \sqrt[m]{\frac{KT}{\mathrm{TOL}}}\int_I \sqrt[m]{\left\|u^{(m+1)}\right\|}dt$$

   Für Runge–Kutta–Verfahren gilt dies näherungsweise auch.

2. Verteile den Fehler gleichmäßig auf alle Schritte. Setze

$$Kh_n^{m+1} \left\|\tau_n^m\right\| \approx \frac{\mathrm{TOL}}{N} \quad \Rightarrow \quad h_n \approx \sqrt[m+1]{\frac{\mathrm{TOL}}{KN \left\|\tau_n^m\right\|}},$$

mit der unbekannten Schrittanzahl N. Dann gilt:

$$\max_{t_n \in I} \|e_n\| \approx K \sum_{t_n \in I} \underbrace{h_n h_n^m}_{= h_n^{m+1}} \left\|\tau_n^m\right\| = \frac{\mathrm{TOL}}{N} \underbrace{\sum_{t_n \in I} 1}_{= N} = \mathrm{TOL}$$

Für die Anzahl der Zeitschritte erhält man

$$N = \sum_{t_n \in I} h_n h_n^{-1} \approx \sum_{t_n \in I} h_n \sqrt[m+1]{\frac{KN \left\|\tau_n^m\right\|}{\mathrm{TOL}}} = \sqrt[m+1]{\frac{KN}{\mathrm{TOL}}} \sum_{t_n \in I} h_n \sqrt[m+1]{\left\|\tau_n^m\right\|}$$

Mit $\tau_n^m \approx u^{(m+1)}(t_{n-1})$ liefert Auflösen nach N (Vorsicht: N ist a–priori unbekannt und muss iterativ angepasst werden!):

$$N \approx \sqrt[m]{\frac{K}{\mathrm{TOL}}} \left(\int_I \sqrt[m+1]{\left\|u^{(m+1)}\right\|} dt \right)^{\frac{m+1}{m}}$$

Die zweite Strategie ist besser, wenn $\sqrt[m+1]{\left\|u^{(m+1)}\right\|}$ noch integrierbar ist, $\left\|u^{(m+1)}\right\|$ aber nicht mehr.

Beispiel 12.3. $m = 1$ (Eulerverfahren), $u^{(2)} = t^{-2}$. t^{-2} ist nicht integrierbar, $\sqrt{t^{-2}} = t^{-1}$ aber schon.

12.3 Schätzung der τ_n^m

Um obige Strategien anwenden zu können benötigen wir gute Schätzungen für τ_n^m. Es seien $y_0, \ldots, y_n$ bereits berechnet und es sei nun h_{n+1} zu bestimmen. Wir wählen eine Schätzschrittweite H (z.B. $H = 2h_n$) und berechnen für den vorläufigen Zeitpunkt $t_{n+1} = t_n + H$

- y_{n+1}^H mit einem Schritt des Verfahrens und Schrittweite H sowie
- $y_{n+1}^{\frac{H}{2}}$ mit zwei Schritten des Verfahrens und Schrittweite $\frac{H}{2}$.

Wir beschränken uns auf den expliziten skalaren Fall. Für die Fehler gilt dann:

$$\begin{aligned} y_{n+1}^H - u(t_{n+1}) &= e_n + H\left(F(H; t_n, y_n) - F(H; t_n, u_n)\right) - H\tau_{n+1}^H \\ &= e_n + H\left(F(H; t_n, u_n) + e_n \frac{\partial F}{\partial x}(H; t_n, \xi) - F(H; t_n, u_n)\right) - H\tau_{n+1}^H \\ &= (1 + O(H))\, e_n - H^{m+1}\tau^m(t_{n+1}) + O\left(H^{m+2}\right) \end{aligned} \tag{12.4}$$

Entsprechend beträgt der Fehler nach einem Schritt mit $\frac{H}{2}$:

$$y_{n+\frac{1}{2}}^{\frac{H}{2}} - u\left(t_n + \frac{H}{2}\right) = (1 + O(H))\, e_n - \left(\frac{H}{2}\right)^{m+1} \tau^m(t_n + H/2) + O\left(H^{m+2}\right) \quad (12.5)$$

Und nach dem zweiten Schritt mit $\frac{H}{2}$:

$$\begin{aligned}
&y_{n+1}^{\frac{H}{2}} - u(t_{n+1}) \\
&= y_{n+\frac{1}{2}}^{\frac{H}{2}} - u\left(t_n + \frac{H}{2}\right) + \frac{H}{2}\left\{F\left(\frac{H}{2}; t_{n+\frac{1}{2}}, y_{n+\frac{1}{2}}^{\frac{H}{2}}\right) - F\left(\frac{H}{2}; t_{n+\frac{1}{2}}, u_{n+\frac{1}{2}}\right)\right\} - \frac{H}{2}\tau_{n+1}^{\frac{H}{2}} \\
&= (1 + O(H))\left\{y_{n+\frac{1}{2}}^{\frac{H}{2}} - u\left(t_n + \frac{H}{2}\right)\right\} - \left(\frac{H}{2}\right)^{m+1} \tau^m(t_{n+1}) + O\left(H^{m+2}\right) \\
&= (1 + O(H))\left\{(1 + O(H))\, e_n - \left(\frac{H}{2}\right)^{m+1} \underbrace{\tau^m(t_{n+1} - H/2)}_{\tau^m(t_{n+1}) + O(H)} + O\left(H^{m+2}\right)\right\} \\
&\quad - \left(\frac{H}{2}\right)^{m+1} \tau^m(t_{n+1}) + O\left(H^{m+2}\right) \\
&= (1 + O(H))\, e_n - 2\left(\frac{H}{2}\right)^{m+1} \tau^m(t_{n+1}) + O\left(H^{m+2}\right)
\end{aligned} \quad (12.6)$$

Differenz (12.6) − (12.4) liefert:

$$y_{n+1}^{\frac{H}{2}} - y_{n+1}^{H} = O(H)\, e_n - \tau^m(t_{n+1}) \underbrace{\left\{2\left(\frac{H}{2}\right)^{m+1} - H^{m+1}\right\}}_{H^{m+1}(2^{-m}-1)} + O\left(H^{m+2}\right)$$

Auflösen nach $\tau^m(t_{n+1})$:

$$\tau^m(t_{n+1}) = \frac{y_{n+1}^{\frac{H}{2}} - y_{n+1}^{H}}{H^{m+1}(1 - 2^{-m})} + O\left(H^{-m}\right) e_n + O(H) \quad (12.7)$$

Nun wird gefordert, dass:

- H klein genug ist, so dass $O(H)$ klein gegen den ersten Term ist.
- $e_n = 0$ ist, das heißt, man nimmt y_n als exakt an und berechnet $\tau^m(t_{n+1})$ damit an anderer Stelle.

Alternativ kann man $e_n = O\left(H^{m+1}\right)$ und damit $O\left(H^{-m}\right) e_n = O(H)$ annehmen. Dass dies gerechtfertigt sein kann, werden wir im Folgenden sehen.

12.4 Adaptiver Algorithmus

1. Setze $y_0 = u_0$, wähle Startschrittweite h_0 und setze $n = 0$.

2. Es sei y_n berechnet, mit der zugehörigen Schrittweite h_n. Wähle $H = 2h_n$ und setze vorläufig $t_{n+1} = t_n + H$.

3. Berechne y_{n+1}^H und $y_{n+1}^{\frac{H}{2}}$. Bestimme

$$\tilde{\tau}_{n+1}^m = \frac{y_{n+1}^{\frac{H}{2}} - y_{n+1}^H}{H^{m+1}\left(1 - 2^{-m}\right)}$$

   und daraus z.B. nach der ersten Strategie

$$h_{n+1} = \sqrt[m]{\frac{\mathrm{TOL}}{KT \left\| \tilde{\tau}_{n+1}^m \right\|}}$$

4. Prüfe, ob $h_{n+1} \ll H/2$, z.B. $h_{n+1} \leq H/4$.

   ja: Die Schrittweite H war zu groß. Setze $H = 2h_{n+1}$ und gehe zu Schritt 3, oder beende, falls $H < h_{\min}$.

   nein: Setze $h_{n+1} = H$, $t_{n+1} = t_n + H$ und akzeptiere $y_{n+1} = y_{n+1}^{\frac{H}{2}}$, d.h. die bessere der beiden Näherungen (hier kann auch $h_{n+1} \gg h_n$ sein). Gehe zu Schritt 2.

Bemerkung 12.4. Unter Annahme einer asymptotischen Entwicklung des *globalen* Fehlers (vergleiche auch mit den Extrapolationsverfahren bei der numerischen Integration)

$$y_{n+1}^H - u\left(t_{n+1}\right) = a^m\left(t_{n+1}\right) H^m + O\left(H^{m+1}\right)$$

kann aus den beiden vorhandenen Näherungen y_{n+1}^H und $y_{n+1}^{\frac{H}{2}}$ mittels

$$\tilde{y}_{n+1} := \frac{2^m y_{n+1}^{\frac{H}{2}} - y_{n+1}^H}{2^m - 1}$$

eine um eine Ordnung bessere Näherung berechnet werden:

$$\tilde{y}_{n+1} - u\left(t_{n+1}\right) = O\left(H^{m+1}\right).$$

Dies zeigt, dass $O\left(H^{-m}\right) e_n = O\left(H\right)$ sein kann. Die weitere Ausnutzung dieser Idee führt zu den *Extrapolationsverfahren.*

Bemerkung 12.5. Nachteil des Verfahrens: Der Zusatzaufwand durch die Fehlerabschätzung beträgt in der Standardvariante 50%. Besser sind *eingebettete Runge–Kutta–Verfahren* oder *Mehrschrittverfahren.*

13 Stabilität und steife Probleme

13.1 Lineare Stabilitätsanalyse

Wir betrachten das skalare, lineare Anfangswertproblem mit $\lambda \in \mathbb{C}$

$$\begin{aligned} u'(t) &= \lambda u(t), \\ u(0) &= u_0, \end{aligned} \tag{13.1}$$

mit der exakten Lösung $u(t) = u_0 e^{\lambda t}$. Das explizite Eulerverfahren angewandt auf (13.1) lautet

$$\begin{aligned} y_n &= y_{n-1} + h_n \lambda y_{n-1} \\ &= (1 + h_n \lambda) y_{n-1} \\ &= \prod_{i=1}^{n} (1 + h_i \lambda) u_0 \end{aligned}$$

bei $y_0 = u_0$. Bei konstanter Schrittweite gilt also

$$|y_n| = |1 + h\lambda|^n |y_0|.$$

Sei nun $\mathrm{Re}(\lambda) \leq 0$. Dann ist die Lösung wegen

$$|u(t)| = |u_0 e^{(\alpha + i\beta)t}| = e^{\alpha t} |e^{i\beta t}| |u_0| \leq |u_0|$$

beschränkt. Das numerische Verfahren liefert eine beschränkte Folge von Werten, falls

$$|1 + h\lambda|^n \leq 1 \quad \Leftrightarrow \quad |1 + h\lambda| \leq 1.$$

Für $\lambda \in \mathbb{R}$ und $\lambda < 0$ führt dies speziell auf

$$\begin{aligned} |1 + h\lambda| \leq 1 &\Leftrightarrow -1 \leq 1 + h\lambda \leq 1 \\ &\Leftrightarrow h \leq -\frac{2}{\lambda} \end{aligned}$$

Wir folgern: Obwohl $|u(t)| \leq |u_0|$ für $\mathrm{Re}(\lambda) \leq 0$ ist, gilt $|y_n| \leq |y_0|$ nur unter einer zusätzlichen Bedingung an die Schrittweite h.

Für das implizite Eulerverfahren erhält man

$$\begin{aligned} y_n &= y_{n-1} + h\lambda y_n \\ \Leftrightarrow (1 - h\lambda) y_n &= y_{n-1} \\ \Leftrightarrow y_n &= \frac{1}{1 - h\lambda} y_{n-1} \end{aligned}$$

und damit für $\lambda \in \mathbb{R}$, $\lambda < 0$

$$\left| \frac{1}{1 - h\lambda} \right| \leq 1 \quad \Leftrightarrow \quad |1 - h\lambda| \geq 1$$

für alle h wegen $h > 0$, $\lambda < 0$. Dies motiviert die folgende Definition.

Definition 13.1 (Absolute Stabilität)**.** Ein Einschrittverfahren heißt *absolut stabil* für ein $h\lambda \neq 0$, wenn es angewandt auf das skalare Testproblem (13.1) für $\mathrm{Re}\,(\lambda) \leq 0$ beschränkte Näherungen erzeugt, d.h.

$$\sup_{n\geq 0} |y_n| < \infty.$$

Für Einschrittverfahren gilt in der Regel

$$y_n = \omega(h\lambda) y_{n-1}$$

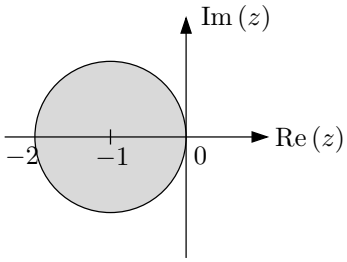
und $\omega(z)$ heißt *Stabilitätsfunktion* oder *Verstärkungsfaktor*. ω ist polynomial für explizite und rational für implizite Verfahren. Man bezeichnet

$$\mathrm{SG} = \left\{ z = h\lambda \in \mathbb{C} \middle| \, |\omega(z)| \leq 1 \right\}$$

als das Gebiet absoluter Stabilität oder *Stabilitätsgebiet* einer Einschrittformel. □

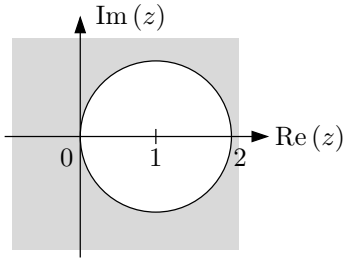
Beispiel 13.2 (Beispiele für Stabilitätsgebiete)**.**

1. Explizites Eulerverfahren: $\omega(z) = 1 + z$



Stabilitätsintervall: $[-2, 0]$
Der Rand des Kreises gehört zum Stabilitätsgebiet.

2. Implizites Eulerverfahren: $\omega(z) = \frac{1}{1-z}$



$$\frac{1}{|1-z|} \leq 1 \Leftrightarrow |1-z| \geq 1$$

□

Für die Taylormethode mit R Stufen gilt:

$$y_n = y_{n-1} + h \sum_{r=1}^{R} \frac{h^{r-1}}{r!} f^{(r-1)}\left(t_{n-1}, y_{n-1}\right)$$

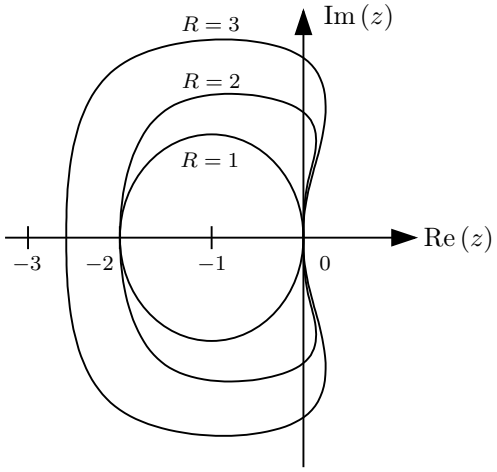
Mit

$$\begin{aligned} f^{(0)} &= f\left(t, u\left(t\right)\right) = \lambda u\left(t\right) \\ f^{(1)} &= \frac{\mathrm{d}}{\mathrm{d}t} f\left(t, u\left(t\right)\right) = \lambda u'\left(t\right) = \lambda^2 u\left(t\right) \\ f^{(2)} &= \frac{\mathrm{d}}{\mathrm{d}t} f^{(1)}\left(t, u\left(t\right)\right) = \lambda^2 u'\left(t\right) = \lambda^3 u\left(t\right) \\ &\vdots \end{aligned}$$

erhält man

$$\begin{aligned} y_n &= y_{n-1} + h \sum_{r=1}^{R} \frac{h^{r-1}}{r!} \lambda^r y_{n-1} \\ &= \left(1 + \sum_{r=1}^{R} \frac{(h\lambda)^r}{r!}\right) y_{n-1} \\ &= \underbrace{\left(\sum_{r=0}^{R} \frac{(h\lambda)^r}{r!}\right)}_{\omega(h\lambda)} y_{n-1} \end{aligned}$$

Wir sehen, dass die Funktion $\omega(z)$ eine Approximation der Exponentialfunktion ist. Die Stabilitätsgebiete haben folgende Form:



Bemerkung 13.3. Dies gilt analog für explizite Runge–Kutta–Verfahren mit $m \leq 4$, die ja Approximationen der Taylormethoden sind.

Für $R \leq 4$ reicht das Stabilitätsgebiet bis $\approx -2,78$ in die linke Halbebene.

Für $\mathrm{Re}\left(\lambda\right) \ll 1$ ist also eine entsprechend kleine Schrittweite erforderlich, damit $h\lambda \in$ SG ist. In gewisser Hinsicht optimal wären Methoden ohne eine solche Beschränkung.

Definition 13.4 (A-Stabilität)**.** Ein Einschrittverfahren heißt *A-stabil*, falls

$$\left\{z \in \mathbb{C} \middle| \operatorname{Re}(z) \leq 0\right\} \subseteq \mathrm{SG}$$

gilt. □

Man kann zeigen:

- Es gibt keine expliziten Einschrittverfahren, die A-stabil sind.
- Das implizite Eulerverfahren und die Trapezregel

$$y_n = y_{n-1} + \frac{h}{2}\left(f\left(t_{n-1}, y_{n-1}\right) + f\left(t_n, y_n\right)\right) \tag{13.2}$$

  sind A-stabil.

Bisher galt alles Gesagte nur für das skalare, lineare Modellproblem. Dies erweitert man wie folgt:

Hypothese 13.5. Ein Einschrittverfahren mit Stabilitätsgebiet SG $\subseteq \mathbb{C}$ ist numerisch stabil für ein allgemeines Anfangswertproblem, wenn die Schrittweiten h_n so gewählt werden, dass mit den Eigenwerten $\lambda(t)$ der Jacobimatrix $\frac{\partial f}{\partial x}\left(t, u\left(t\right)\right)$ unter der Annahme $\operatorname{Re}\left(\lambda(t)\right) \leq 0$ gilt:

$$h_n \lambda\left(t_n\right) \in \mathrm{SG}; \qquad n \geq 0.$$

Dies nennt man *lineare Stabilitätsanalyse.* □

Man kann zeigen, dass trotz $\operatorname{Re}\left(\lambda\left(t\right)\right) \leq 0$ die Lösung $\left\|u\left(t\right)\right\|$ exponentiell wachsen kann. Trotzdem wird diese Form der Stabilitätsanalyse häufig verwendet.

13.2 Steife Probleme

Betrachte das folgende lineare System (aus (SB05)):

$$\begin{pmatrix} u_1' \\ u_2' \end{pmatrix} = \underbrace{\begin{pmatrix} \frac{\lambda_1+\lambda_2}{2} & \frac{\lambda_1-\lambda_2}{2} \\ \frac{\lambda_1-\lambda_2}{2} & \frac{\lambda_1+\lambda_2}{2} \end{pmatrix}}_{A} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} \tag{13.3}$$

Die Eigenwerte der Matrix A sind, wie man nachrechnet, λ_1 und λ_2, mit den zugehörigen Eigenvektoren $e_1 = (1,1)^T$ und $e_2 = (1,-1)^T$. Die allgemeine Lösung ist also

$$u_1(t) = c_1 e^{\lambda_1 t} + c_2 e^{\lambda_2 t}$$
$$u_2(t) = c_1 e^{\lambda_1 t} - c_2 e^{\lambda_2 t}$$

Das explizite Eulerverfahren ist wie gehabt $y_n = y_{n-1} + hAy_{n-1}$. Mit der Darstellung $y_0 = \alpha_1 e_1 + \alpha_2 e_2$ für den Startwert y_0 gilt

$$y_n = \alpha_1 \left(1 + h\lambda_1\right)^n e_1 + \alpha_2 \left(1 + h\lambda_2\right)^n e_2.$$

Dies sieht man so (nur Induktionsschritt):

$$\begin{aligned}
y_n &= y_{n-1} + hAy_{n-1} \\
&= \alpha_1 (1 + h\lambda_1)^{n-1} e_1 + \alpha_2 (1 + h\lambda_2)^{n-1} e_2 \\
&\quad + hA \left(\alpha_1 (1 + h\lambda_1)^{n-1} e_1 + \alpha_2 (1 + h\lambda_2)^{n-1} e_2 \right) \\
&= \alpha_1 (1 + h\lambda_1)^{n-1} e_1 + \alpha_1 h (1 + h\lambda_1)^{n-1} \lambda_1 e_1 \\
&\quad + \alpha_2 (1 + h\lambda_2)^{n-1} e_2 + \alpha_2 h (1 + h\lambda_2)^{n-1} \lambda_2 e_2 \\
&= \alpha_1 (1 + h\lambda_1)^{n-1} e_1 (1 + h\lambda_1) + \alpha_2 (1 + h\lambda_2)^{n-1} e_2 (1 + h\lambda_2) \\
&= \alpha_1 (1 + h\lambda_1)^{n} e_1 + \alpha_2 (1 + h\lambda_2)^{n} e_2.
\end{aligned}$$

Damit bleibt $\|y_n\|$ für $n \to \infty$ nur dann beschränkt, falls

$$|1 + h\lambda_1| \leq 1 \quad \wedge \quad |1 + h\lambda_2| \leq 1,$$

also bei reellem λ_i, $\lambda_i < 0$:

$$h \leq -\frac{2}{\lambda_1} \quad \wedge \quad h \leq -\frac{2}{\lambda_2} \quad \Rightarrow \quad h \leq \min \left\{ -\frac{2}{\lambda_1}, -\frac{2}{\lambda_2} \right\}. \tag{13.4}$$

Angenommen, $\lambda_1 = -1$, $\lambda_2 = -1000$ und $c_1 = c_2 = 1$. Wegen $e^{-\frac{1000}{50}} \approx 2 \cdot 10^{-9}$ ist die Lösung für $t \geq \frac{1}{50}$ praktisch von e^{-t} dominiert und es sollte eine Schrittweite von $h \approx -\frac{2}{\lambda_1} = -2$ ausreichen. Die Stabilitätsbedingung (13.4) erfordert aber $h \leq -\frac{2}{1000}$. Beim impliziten Eulerverfahren ist dies nicht der Fall, man kann große Schrittweiten wählen. Allerdings erfordert eine gewünschte Genauigkeit natürlich auch eine entsprechend kleine Schrittweite. Zu Beginn bei $t = 0$ muss deshalb auch beim impliziten Eulerverfahren die Schrittweite $h \approx -\frac{2}{\lambda_2}$ gewählt werden.

Von einem *steifen System* spricht man,

- wenn die Eigenwerte stark unterschiedlich sind:

$$\frac{\max_j |\mathrm{Re}\,(\lambda_j)|}{\min_j |\mathrm{Re}\,(\lambda_j)|} \gg 1$$

- falls implizite Verfahren besser sind als explizite Verfahren.

- bei singulär gestörten Problemen:

$$\begin{aligned}
u_1'(t) &= f\,(t, u_1(t), u_2(t)) \\
\varepsilon u_2'(t) &= g\,(t, u_1(t), u_2(t))
\end{aligned}$$

  mit $\varepsilon \ll 1$. Für $\varepsilon = 0$ hätte man ein differentiell algebraisches System. DAEs sind deshalb grundsätzlich steif.

13.3 Verfahren zur Lösung steifer Probleme

Implizite Runge–Kutta–Verfahren

Implizite Runge–Kutta–Verfahren sind eine allgemeine Klasse von Einschrittverfahren zur Lösung steifer Probleme.

$$\begin{aligned} y_n &= y_{n-1} + h_n \sum_{r=1}^{R} c_r k_r \left(h_n; t_{n-1}, y_{n-1}\right) \\ \text{mit } k_r \left(h_n; t_{n-1}, y_{n-1}\right) &= f\left(t_{n-1} + a_r h_n, y_{n-1} + h_n \sum_{s=1}^{R} b_{rs} k_s \left(h_n; t_{n-1}, y_{n-1}\right)\right) \end{aligned} \tag{13.5}$$

- Die k_r sind durch ein voll gekoppeltes System der Größe $R \cdot m$ bestimmt $\Rightarrow$ hoher Aufwand.
- A-stabile Verfahren beliebig hoher Ordnung sind möglich.
- Die maximale Ordnung ist $2R$.
- Der Aufwand wird geringer, wenn man $b_{rs} = 0$ für $s > r$ fordert (*diagonal implizite Runge–Kutta–Verfahren*. Dann sind R Systeme der Größe m zu lösen.

Beispiel 13.6 (Gauß–Verfahren)**.** Das Gauß–Verfahren, A-stabil der Ordnung 4 mit nur 2 Stufen, lautet:

$$B = \begin{pmatrix} \frac{1}{4} & \frac{1}{4} - \frac{\sqrt{3}}{6} \\ \frac{1}{4} + \frac{\sqrt{3}}{6} & \frac{1}{4} \end{pmatrix}, \quad c = \begin{pmatrix} \frac{1}{2} \\ \frac{1}{2} \end{pmatrix}, \quad a = \begin{pmatrix} \frac{1}{2} - \frac{\sqrt{3}}{6} \\ \frac{1}{2} - \frac{\sqrt{3}}{6} \end{pmatrix}.$$

□

Lineare Mehrschrittverfahren

Lineare Mehrschrittverfahren haben die allgemeine Form

$$\sum_{r=0}^{R} \alpha_{R-r} y_{n-r} = h \sum_{r=0}^{R} \beta_{R-r} f\left(t_{n-r}, y_{n-r}\right) \tag{13.6}$$

Lineare Mehrschrittverfahren können hohe Ordnung erreichen und müssen dazu in jedem Schritt nur *ein* System der Größe m lösen. Sie sind also in der Regel effizienter als Runge-Kutta-Verfahren.

- Zur Berechnung von y_n werden die $y_{n-1}, \ldots, y_{n-R}$ benötigt.
- Für $\beta_R = 0$ ist das Verfahren explizit, sonst implizit.
- Das implizite Eulerverfahren und die Trapezregel lassen sich als Mehrschrittverfahren mit $R = 1$ interpretieren.

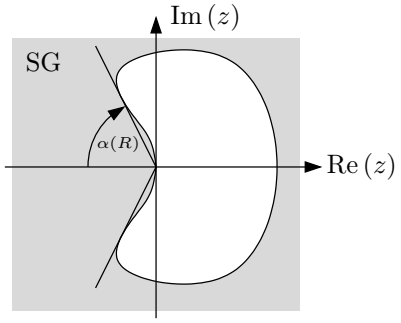
- Bezüglich der A-Stabilität gilt:
  - Kein explizites Mehrschrittverfahren ist A-stabil.
  - Kein implizites Mehrschrittverfahren mit Konvergenzordnung größer als 2 ist A-stabil.

- Das beliebteste Mehrschrittverfahren für steife Probleme ist BDF (*Backward Difference Formula*): Bei Parameter R setze ein Polynom p vom Grad $R+1$ an, so dass:

  1. $p\left(t_{n-r}\right) = y_{n-r}; \quad r = 0, \ldots, R$,
  2. $p'\left(t_n\right) = f\left(t_n, y_n\right)$.

  Dies genau $R+2$ Bedingungen für die $R+2$ unbekannten Koeffizienten. Die Auswertung des Polynoms an der Stützstelle t_n ergibt den gesuchten Wert der numerischen Lösung.

- BDF(R) ist A(α)-stabil.



- Lineare Mehrschrittverfahren erlauben eine effiziente Schrittweitenkontrolle ohne Zusatzaufwand (das gibt es allerdings auch für sog. eingebettete Runge-Kutta-Verfahren).

- BDF eignet sich sehr gut für DAEs vom Index 1.

Lösung der nichtlinearen Gleichungssysteme

Alle A-stabilen Verfahren sind implizit und erfordern die Lösung eines (eventuell nichtlinearen) Gleichungssystems. Betrachte das implizite Eulerverfahren:

$$y_n = y_{n-1} + hf\left(t_n, y_n\right) \tag{13.7}$$

Eine Möglichkeit ist die Fixpunktiteration, d.h. y_n ist offensichtlich Fixpunkt der Gleichung

$$y = g(y) \text{ mit } g(y) = y_{n-1} + hf\left(t_n, y\right).$$

Nach dem Banach'schen Fixpunktsatz konvergiert die Iteration

$$y^i = g\left(y^{i-1}\right),$$

wenn g eine Kontraktion ist:

$$\begin{aligned}\left\|g\left(y\right)-g\left(y'\right)\right\| &= \left\|y_{n-1}+hf\left(t_n,y\right)-\left(y_{n-1}+hf\left(t_n,y'\right)\right)\right\| \\ &= h\left\|f\left(t_n,y\right)-f\left(t_n,y'\right)\right\| \\ &\leq hL\left\|y-y'\right\| \quad \text{(Lipschitz-Stetigkeit von } f\text{)}.\end{aligned}$$

Also dann, wenn $hL < 1 \Leftrightarrow h < L^{-1}$. Dieser Schrittweitenbeschränkung wollten wir durch die Wahl impliziter Verfahren aber gerade entkommen! Daher ist dieses Vorgehen inpraktikabel. Stattdessen verwendet man die Newton–Iteration mit einer geeigneten Globalisierungsstrategie (gedämpftes Newton–Verfahren).

14 Modellierung mit partiellen Differentialgleichungen

Wir motivieren am Beispiel der Wärmeleitung, wie die kontinuumsmechanische Modellierung zu partiellen Differentialgleichungen führt. Dieses Vorgehen ist prototypisch für eine ganze Reihe von Anwendungen.

14.1 Begriffe aus der Vektoranalysis

Wir betrachten im Allgemeinen vektorwertige Funktionen $f : \mathbb{R}^n \to \mathbb{R}^m$

$$f(x) = (f_1(x), \ldots, f_m(x))^T .$$

$f : \mathbb{R}^n \to \mathbb{R}$ ist partiell differenzierbar nach x, falls

$$\frac{\partial f}{\partial x_i}(x) = \lim_{h \to 0} \frac{f(x_1, \ldots, x_i + h, \ldots, x_n) - f(x_1, \ldots, x_i, \ldots, x_n)}{h}$$

existiert. Man schreibt auch $\partial_i f(x)$.

Oft sind die Funktionen auf einer Teilmenge $\Omega \subset \mathbb{R}^n$, einem *Gebiet*, definiert. Ein Gebiet ist offen und zusammenhängend. $\partial\Omega$ bezeichnet den Rand des Gebietes und $\overline{\Omega}$ seinen Abschluss, d.h. $\overline{\Omega} = \Omega \cup \partial\Omega$.

Weiterhin bezeichnet

$$C^m(\Omega) = \left\{ f : \Omega \to \mathbb{R} \middle| \partial_{i_1}, \ldots, \partial_{i_m} f \text{ stetig}, i_k \in \{1, \ldots, n\} \right\}$$

die Menge der Funktionen, die auf $\overline{\Omega}$ m mal stetig differenzierbar sind. $C^0(\Omega)$ bezeichnet die auf Ω stetig differenzierbaren Funktionen.

Definition 14.1 (Gradient)**.** Für $f \in C^1\left(\overline{\Omega}\right)$, $\Omega \subset \mathbb{R}^n$ heißt

$$\nabla f(x) = (\partial_1 f(x), \ldots, \partial_n f(x))^T \tag{14.1}$$

Gradient von f. □

Definition 14.2 (Divergenz)**.** Für $f \in \left[C^1\left(\overline{\Omega}\right)\right]^n$, $\Omega \subset \mathbb{R}^n$ heißt

$$\nabla \cdot f(x) = \sum_{i=1}^{n} \partial_i f(x) \tag{14.2}$$

Divergenz von f. □

Partielle Integration: Für $v, w \in C^1\left(\overline{\Omega}\right)$ ist

$$\int_\Omega \left[\partial_i v(x)\right] w(x) dx = -\int_\Omega v(x) \partial_i w(x) dx + \int_{\partial\Omega} v(x) w(x) \left(\nu(x)\right)_i ds. \tag{14.3}$$

Dabei ist $\nu(x)$ die äußere Einheitsnormale im Punkt $x \in \partial\Omega$.

Folgerung 14.3. Aus der Formel für die partielle Integration ergibt sich für vektorwertige Funktionen

- Für $v \in \left[C^1\left(\overline{\Omega}\right)\right]^n$, $w \in C^1(\Omega)$ ist

$$\int_\Omega \left[\nabla \cdot v(x)\right] w(x) dx = -\int_\Omega v(x) \cdot \nabla w(x) dx + \int_{\partial\Omega} v(x) \cdot \nu(x) w(x) ds. \tag{14.4}$$

- *Gauß'scher Integralsatz*: Für $v \in \left[C^1\left(\overline{\Omega}\right)\right]^n$ ist

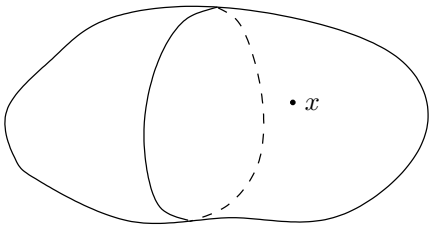
$$\int_\Omega \nabla \cdot v(x) dx = \int_{\partial\Omega} v(x) \cdot \nu(x) ds. \tag{14.5}$$

□

14.2 Modellierung des Wärmetransports

Als weiterführende Literatur verweisen wir auf (Fey63).

Gegeben sei ein Körper, der das Gebiet $\Omega \subset \mathbb{R}^3$ ausfüllt. Für das Zeitintervall $\Sigma = [a, b]$ soll der Temperaturverlauf berechnet werden, d.h. wir sind interessiert an $T(x, t)$ für $(x, t) \in \Omega \times \Sigma$.



Um diese Frage eindeutig beantworten zu können, sind *Anfangsbedingungen*

$$T(x, a) = T_0(x), \quad x \in \overline{\Omega}$$

und *Randbedingungen*, z.B.

$$T(x, t) = g(x, t), \quad (x, t) \in \partial\Omega \times \Sigma,$$

erforderlich. Weitere mögliche Randbedingungen werden später behandelt.

Grundlage dieses Vorgehens ist die Kontinuumshypothese, d.h. der Körper wird als Kontinuum betrachtet, bei dem jedem Punkt im mathematischen Sinne eine Eigenschaft (z.B. Temperatur) zugeordnet werden kann. Dies erfordert, dass die betrachtete Raum- und Zeitskala genügend groß ist.

Nun zur Physik.

Energie Betrachte ein beliebiges Teilgebiet $\omega \subseteq \Omega$. Zum Zeitpunkt $t \in \Sigma$ enthält das Volumen ω die Wärmeenergie $Q_\omega(t)$. Diese berechnet sich als

$$Q_\omega(t) = \int_\omega c(x)\rho(x)T(x,t)dx \tag{14.6}$$

$c(x)$:	spezifische Wärmekapazität	$\left[\frac{J}{kgK}\right]$
$\rho(x)$:	Massendichte	$\left[\frac{kg}{m^3}\right]$
$T(x,t)$:	Temperatur	$[K]$

Hierbei können neben Temperatur auch Wärmekapazität und Dichte in ω variieren.

Energieerhaltung Wir betrachten die zeitliche Änderung von Q_ω in einem beliebigen Zeitintervall $[t, t+\Delta t]$:

$$Q_\omega(t+\Delta t) - Q_\omega(t) = \left\{\begin{array}{c}\text{Energieeinspeisung/-verluste}\\ \text{im Gebiet } \omega\end{array}\right\} + \left\{\begin{array}{c}\text{Wärmefluss über}\\ \text{den Rand von } \omega\end{array}\right\}.$$

In Formeln:

$$\int_\omega c\rho T(x,t+\Delta t)\,dx - \int_\omega c\rho T(x,t)\,dx = \int_t^{t+\Delta t}\int_\omega f(x,t)\,dxdt - \int_t^{t+\Delta t}\int_{\partial\omega} q(x,t)\cdot\nu(x)\,dsdt.$$

$f(x,t)$:	Quellen-/Senkenterm	$\left[\frac{J}{sm^3}\right]$
$q(x,t)$:	Flussvektor	$\left[\frac{J}{sm^2}\right]$
$\nu(x)$:	äußere Einheitsnormale	

Das Vorzeichen des zweiten Terms rührt daher, dass $q\cdot\nu > 0$ eine Abnahme der Energie bedeutet.

Aus $\int_t^{t\Delta t} r(t)dt = \Delta t r(t) + O\left(\Delta t^2\right)$ folgt

$$\int_\omega \frac{c\rho T(x,t+\Delta t) - c\rho T(x,t)}{\Delta t}dx = \int_\omega f(x,t)\,dx - \int_{\partial\omega} q(x,t)\,\nu(x)\,ds + O(\Delta t).$$

Anwendung des Gauß'schen Integralsatzes und $\Delta t \to 0$ liefert:

$$\int_\omega \frac{\partial(c\rho T)}{\partial t}(x,t) + \nabla\cdot q(x,t) - f(x,t)\,dx = 0.$$

Da $\omega \subseteq \Omega$ beliebig gewählt war, folgert man, dass der Integrand punktweise verschwinden muss:

$$\frac{\partial(c\rho T)}{\partial t}(x,t) + \nabla\cdot q(x,t) = f(x,t) \qquad \forall\,(x,t)\in\Omega\times\Sigma. \tag{14.7}$$

Dies ist die mathematische Formulierung der Energieerhaltung im Rahmen der Kontinuumsannahme.

Wärmefluss: Nun benötigt man einen Ausdruck für den Fluss $q(x,t)$. Dieser setzt sich aus zwei Anteilen zusammen:

Konduktion: Auf molekularer Ebene entspricht die Wärmeenergie der Bewegungsenergie der Moleküle. Konduktion ist die Übertragung von Bewegungsenergie durch Stöße. Eine Modellannahme auf makroskopischer Ebene ist, dass Wärmeenergie in Richtung des größten Temperaturunterschiedes fließt. Wegen des zweiten Hauptsatzes der Thermodynamik fließt sie in Richtung der kleineren Werte.

$$\text{Fourier'sches Gesetz (1822):} \qquad q_c(x,t) = -\lambda \nabla T(x,t). \tag{14.8}$$

$$\lambda\text{: Wärmeleitfähigkeit} \quad \left[\tfrac{J}{smK}\right] = \left[\tfrac{W}{mK}\right]$$

λ ist im Allgemeinen ein ortsabhängiger Tensor $\lambda(x) = R^T(x) D(x) R(x)$ $(d_{ii} > 0)$ und symmetrisch positiv definit.

Konvektion: In Fluiden (dies umfasst auch Gase) wird Wärme auch durch Stoffbewegung transportiert.

$$q_t(x,t) = c(x,t)\,\rho(x,t)\,T(x,t)\,u(x,t) \tag{14.9}$$

$c(x,t)$:	spezifische Wärmekapazität	$\left[\tfrac{J}{kgK}\right]$
$\rho(x,t)$:	Massendichte des bewegten Stoffes	$\left[\tfrac{kg}{m^3}\right]$
$T(x)$:	Temperatur	$[K]$
$u(x,t)$:	Geschwindigkeit am Punkt (x,t)	$\left[\tfrac{m}{s}\right]$

Der Gesamtfluss ergibt sich dann zu

$$q(x,t) = q_c(x,t) + q_t(x,t). \tag{14.10}$$

Wärmeleitungsgleichung: Das vollständige mathematische Modell erhält man durch Einsetzen von (14.10) in (14.7):

$$\frac{\partial(c\rho T)}{\partial t}(x,t) + \nabla \cdot \{c(x,t)\,\rho(x,t)\,u(x,t)\,T(x,t) - \lambda(x)\,\nabla T(x,t)\} = f(x,t)$$
$$\forall (x,t) \in \Omega \times \Sigma, \tag{14.11}$$

mit der Anfangsbedingung

$$T(x,a) = T_0(x), \quad x \in \overline{\Omega},$$

und den Randbedingungen

$$\begin{aligned} T(x,t) &= g(x,t), && (x,t) \in \Gamma_D(t) \times \Sigma, && \Gamma_D(t) \subseteq \partial\Omega, \\ \{c\rho u T - \lambda \nabla T\} \cdot \nu &= Q(x,t), && (x,t) \in \Gamma_N(t) \times \Sigma, && \Gamma_N(t) = \partial\Omega \setminus \Gamma_D(t). \end{aligned}$$

Bemerkung 14.4. Während (14.9) und (14.7) im Rahmen der Kontinuumsmechanik exakt sind, ist (14.8) ein Modell, welches die Wirklichkeit mehr oder weniger stark approximiert ($\rightarrow$ Skalenübergang).

14.3 Weitere Anwendungen

Ladungsträgerfluss in einem Leiter

$$\begin{aligned} \frac{\partial (cu(x,t))}{\partial t} + \nabla \cdot i(x,t) &= f(x,t), \\ i(x,t) &= \frac{1}{r(x,t)} \nabla u(x,t). \end{aligned} \tag{14.12}$$

Anwendung: z.B. Signalausbreitung auf einem Neuron.

Stationäre Wärmeleitung

Durch die Annahmen

- $T = T(x)$, d.h. T ist zeitunabhängig
- $u = 0$, also kein Stoffffluss (Festkörper)

reduziert sich (14.11) auf die stationäre Diffusionsgleichung

$$\begin{aligned} -\nabla \cdot \{\lambda \nabla T(x)\} &= f(x), \quad && x \in \Omega, \\ T(x) &= g(x), && x \in \Gamma_D \subseteq \partial\Omega, \\ -\lambda \nabla T(x)\, \nu &= Q(x), && x \in \Gamma_N = \partial\Omega \setminus \Gamma_D. \end{aligned} \tag{14.13}$$

Mit $\lambda = I$ ergibt sich die *Poissongleichung*:

$$\begin{aligned} -\nabla \cdot \nabla T = -\Delta T &= f \quad \text{in } \Omega, \\ T &= g \quad \text{auf } \partial\Omega, \end{aligned} \tag{14.14}$$

mit

$$\Delta T = \sum_{i=1}^{n} \frac{\partial^2 T}{\partial x_i^2} \tag{14.15}$$

dem *Laplaceoperator*. Mit $f \equiv 0$ ergibt sich schließlich die *Laplacegleichung*:

$$\begin{aligned} \Delta T &= 0 \quad \text{in } \Omega, \\ T &= g \quad \text{auf } \partial\Omega. \end{aligned} \tag{14.16}$$

Grundwasserströmung

(Strömung von Wasser in einem porösen Medium)

Das Volumen ω besteht aus Sandkörnern (Volumenanteil $1-\Phi$) und Hohlraum (Volumenanteil Φ) mit der Porosität $\Phi(x) : \Omega \to (0,1)$. Der Hohlraum sei vollständig mit Wasser gefüllt. $\rho(x,t)$ entspricht der Massendichte $\left[\frac{kg}{m^3}\right]$ und (14.7) der Massenerhaltung.

$$\frac{\partial (\Phi \rho(x,t))}{\partial t} + \nabla \cdot \{\rho(x,t)\, u(x,t)\} = f(x,t). \tag{14.17}$$

Für die Geschwindigkeit fand Darcy 1856 den empirischen Zusammenhang

$$u(x,t) = -\frac{K}{\mu}(\nabla p - \rho g).$$

K:	Permeabilität	$\left[m^2\right]$
μ:	dynamische Viskosität	$[Pa\,s]$
p:	Druck	$[Pa] = \left[\frac{N}{m^2}\right]$
g:	Gravitationsvektor $(0,0,-9.81)^T$	$\left[\frac{m}{s^2}\right]$

Im inkompressiblen Fall ($\rho = const$) ergibt sich die stationäre Grundwassergleichung

$$-\nabla \cdot \left\{ \rho \frac{K}{\mu}(\nabla p - \rho g) \right\} = f \quad \text{in } \Omega. \tag{14.18}$$

Mögliche Randbedingungen wären

$p = g$	auf $\Gamma_D \subseteq \partial\Omega$	(Druckvorgabe),
$u(x,t)\,\nu = J$	auf $\Gamma_N = \partial\Omega \setminus \Gamma_D$	(Flussvorgabe).

15 Typeinteilung partieller Differentialgleichungen

15.1 Allgemeine Definition

Die allgemeine implizite partielle Differentialgleichung m-ter Ordnung ($m \geq 1$) für eine gesuchte Funktion $u : \mathbb{R}^n \to \mathbb{R}$ lautet:

$$F\left(\frac{\partial^m u}{\partial x_1^m}(x), \ldots, \frac{\partial^m u}{\partial x_n^m}(x), \frac{\partial^{m-1} u}{\partial x_1^{m-1}}(x), \ldots, u(x)\right) = 0. \tag{15.1}$$

u heißt Lösung von (15.1), wenn

1. alle partiellen Ableitungen von u bis zur Ordnung m existieren und
2. (15.1) für jeden Punkt $x \in \Omega$ erfüllt ist.

Das Definitionsgebiet Ω ist offen und zusammenhängend. Oft werden weitere Bedingungen an den Rand $\partial\Omega$ gestellt (Lipschitzstetigkeit, Kegelbedingung, Differenzierbarkeit, ...).

Um die Eindeutigkeit der Lösung zu garantieren, sind zusätzliche Bedingungen, die sogenannten *Rand-* bzw. *Anfangsbedingungen* erforderlich.

Partielle Differentialgleichungen erlauben keine einheitliche Lösungstheorie wie die gewöhnlichen Differentialgleichungen. Selbst die Frage, welche Rand- / Anfangsbedingungen zu einer eindeutigen und stabilen Lösung führen, muss von Fall zu Fall beantwortet werden. Etwas Licht ins Dunkel bringt die Typeinteilung. Zuvor betrachten wir aber einige Beispiele.

15.2 Elementare partielle Differentialgleichungen

Potential–Gleichung oder Laplace–Gleichung

Die Gleichung

$$\Delta u(x) = \sum_{i=1}^{n} \frac{\partial^2 u}{\partial x_i^2}(x) = 0 \tag{15.2}$$

heißt *Laplace–Gleichung*. (15.2) entsteht aus der Wärmeleitungsgleichung unter der Annahme der Stationarität, $\lambda = I$ und $u = 0$. Die Laplace–Gleichung ist mit einer Randwertvorgabe zu versehen. Diese kann verschiedene Formen annehmen:

Dirichlet–Randbedingung: $$u(x) = \varphi(x) \quad x \in \Gamma_D \subseteq \Omega, \tag{15.3a}$$

Neumann–Randbedingung: $$\frac{\partial u}{\partial n}(x) = \psi(x) \quad x \in \Gamma_N, \tag{15.3b}$$

Robin–Randbedingung: $$\alpha(x)\frac{\partial u}{\partial n} + \beta(x)\, u(x) = \eta(x) \quad x \in \Gamma_R. \tag{15.3c}$$

- Hat man nur Dirichlet–Randbedingungen, so spricht man vom *Dirichlet–Problem*.

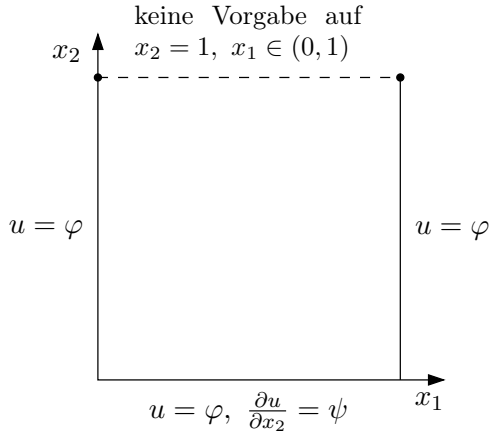
- Hat man nur Neumann–Randbedingungen, so ist die Lösung nur bis auf eine additive Konstante bestimmt.
- An jedem Punkt des Randes muss *eine* der Randbedingungen (15.3a), (15.3b) oder (15.3c) vorgegeben werden. Deshalb spricht man auch von einem *Randwertproblem.*

Wellengleichung

Die Gleichung

$$\frac{\partial^2 u}{\partial x_n^2} = \underbrace{\sum_{i=1}^{n-1} \frac{\partial^2 u}{\partial x_i^2}}_{=:\ \Delta_{n-1} u} \quad \text{in } \Omega \tag{15.4}$$

heißt *Wellengleichung.* Für den Fall $\Omega = (0,1)^2$ sind etwa folgende Randwertvorgaben zulässig:



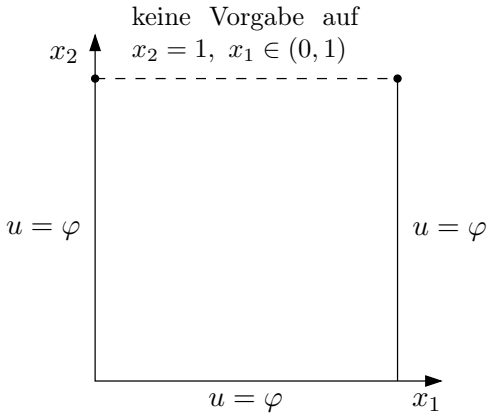
- Hierbei ist eine der beiden Richtungen ausgezeichnet (hier x_2, in der Anwendung entspricht x_2 der Zeit).
- Die Vorgaben auf $x_2 = 0$ heißen *Anfangswerte.*
- Auf $x_2 = 1$ sind keine Vorgaben zulässig.
- Die Wellengleichung erhält man nicht aus der allgemeinen Wärmeleitungsgleichung.

Wärmeleitungsgleichung

Die kanonische Form der Wärmeleitungsgleichung lautet

$$\frac{\partial u}{\partial x_n} = \sum_{i=1}^{n-1} \frac{\partial^2 u}{\partial x_i^2} \quad \text{in } \Omega. \tag{15.5}$$

Man erhält sie für $x_n = t$, $\lambda = I$ und $c = \rho = 1$ aus der Herleitung in Abschnitt 14.2. Für den Fall $\Omega = (0,1)^2$ sind etwa die folgenden Randvorgaben zulässig:



- Wie bei der Wellengleichung ist hier die x_2-Richtung ausgezeichnet.
- Die Vorgabe auf $x_2 = 0$ nennt man wieder Anfangswertvorgabe.
- Wegen der ersten Ableitung in x_2 (x_n) ist nur eine Vorgabe möglich.

Bisher waren alle Gleichungen 2. Ordnung.

Transportgleichung

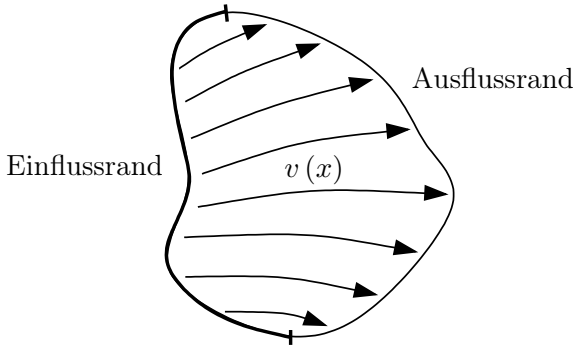
Sei $v : \mathbb{R}^n \to \mathbb{R}^n$ ein gegebenes Vektorfeld. Dann heißt die Gleichung 1. Ordnung

$$\nabla \cdot \{v(x)\, u(x)\} = 0 \quad \text{in } \Omega \tag{15.6}$$

Transportgleichung. Man erhält (15.6) aus der allgemeinen Wärmetransportgleichung für $\lambda = 0$ (nur Konvektion). Zu beachten ist, dass der stationäre und der instationäre Fall sich im Prinzip nicht unterscheiden.

Sei $\nu(x)$ die äußere Einheitsnormale an $\partial\Omega$. Dann ist folgende Randwertvorgabe zulässig:

$$u(x) = \varphi(x) \quad \text{auf } \left\{x \in \partial\Omega \middle| v(x)\,\nu(x) < 0\right\}$$



15.3 Sachgemäß gestellte Probleme

In welchem Sinne sind die oben gestellten Probleme zulässig?

Definition 15.1 (Sachgemäß gestellt)**.** Eine partielle Differentialgleichung heißt *sachgemäß gestellt*, falls

1. eine eindeutige Lösung existiert und
2. diese stetig von den Daten (Randbedingung, Anfangsbedingung, rechte Seite) abhängt. □

Sei unser Problem

$$A(u) = f \quad \text{in } \Omega$$

mit dem Differentialoperator $A : X \to Y$ (z.B. $X = C^2\left(\overline{\Omega}\right),\ Y = C^0\left(\overline{\Omega}\right)$). Dann lauten die Bedingungen aus Definition 15.1

1. Zu jedem f existiert genau ein u, so dass $A(u) = f$.
2. Es gibt eine Konstante C, die nur von A abhängt, so dass $\|u\| \leq C\|f\|$. Hierbei ist eine geeignete Norm zu wählen.

Die oben genannten Probleme sind sachgemäß gestellt. Dabei können kleine Änderungen große Wirkungen haben. Zum Beispiel ist die "umgedrehte" Wärmeleitung

$$\frac{\partial u}{\partial x_n} = -\sum_{i=1}^{n-1} \frac{\partial^2 u}{\partial x_i^2} \quad \text{in } \Omega \tag{15.7}$$

mit den Vorgaben aus Abschnitt 14.2 *nicht* sachgemäß gestellt.

15.4 Typeinteilung

Der folgende Satz zeigt, dass die oben genannten elementaren Gleichungen mehr als bloße Beispiele sind.

Satz 15.2. Im Fall $n = 2$, $m = 2$ (2. Ordnung, 2 Raumdimensionen) ist die allgemeine lineare partielle Differentialgleichung

$$\underbrace{\sum_{i,j=1}^{2} a_{ij} \frac{\partial^2 u}{\partial x_i \partial x_j}(x)}_{\text{Hauptteil}} + \sum_{i=1}^{2} a_i \frac{\partial u}{\partial x_i}(x) + a_0(x) = 0 \quad \text{in } \Omega \tag{15.8}$$

in jedem Punkt $x \in \Omega$ bis auf einen Term niederer Ordnung auf einen der drei Typen Potentialgleichung, Wellengleichung oder Wärmeleitungsgleichung transformierbar.
Beweis: Wir führen die Transformation $\psi : \Omega \to \Omega'$ mit $\xi = \psi(x)$ ein. Dabei sei $\psi(x)$ in jedem Punkt invertierbar und

$$(S(x))_{\alpha\beta} = \frac{\partial \psi_\alpha}{\partial x_\beta}$$

sei die Jacobimatrix der Transformation. Wir setzen $u(x) = \tilde{u}(\psi(x))$ und wollen nun eine partielle Differentialgleichung für die unbekannte Funktion $\tilde{u}(\xi)$ aufstellen. Dabei wollen wir die Randbedingungen außer Acht lassen. Es gilt:

$$\partial_i u(x) = \frac{\partial \tilde{u}}{\partial x_i}(\psi(x)) = \sum_{k=1}^{n} \frac{\partial \tilde{u}}{\partial \xi_k}(\psi(x)) \frac{\partial \psi_k}{\partial x_i}(x),$$

$$\frac{\partial^2 u}{\partial x_j \partial x_i}(x) = \sum_{k=1}^{n} \left\{ \sum_{l=1}^{n} \frac{\partial^2 \tilde{u}}{\partial \xi_l \partial \xi_k}(\psi(x)) \frac{\partial \psi_l}{\partial x_j}(x) \frac{\partial \psi_k}{\partial x_i}(x) + \frac{\partial \tilde{u}}{\partial \xi_k}(\psi(x)) \frac{\partial^2 \psi_k}{\partial x_j \partial x_i}(x). \right\}$$

Einsetzen in (15.8) liefert:

$$\sum_{i=1}^{n}\sum_{j=1}^{n} \left\{ a_{ij}(x) \left[\sum_{k=1}^{n}\sum_{l=1}^{n} \frac{\partial^2 \tilde{u}}{\partial \xi_l \partial \xi_k}(\psi(x)) \frac{\partial \psi_l}{\partial x_j}(x) \frac{\partial \psi_k}{\partial x_i}(x) + \sum_{k=1}^{n} \frac{\partial \tilde{u}}{\partial \xi_k}(\psi(x)) \frac{\partial^2 \psi_k}{\partial x_j \partial x_i}(x) \right] \right\}$$
$$+ \sum_{i=1}^{n} a_i(x) \left[\sum_{k=1}^{n} \frac{\partial}{\partial \xi_k} \tilde{u}(\psi(x)) \frac{\partial}{\partial x_i} \psi_k(x) \right] + a_0(x) = 0.$$

Vertauschen der Summen liefert schließlich:

$$\sum_{k=1}^{n}\sum_{l=1}^{n} \frac{\partial^2 \tilde{u}}{\partial \xi_k \partial \xi_l}(\psi(x)) \underbrace{\left\{ \sum_{i=1}^{n} \underbrace{\frac{\partial \psi_k}{\partial x_i}(x)}_{S_{ki}} \left[\sum_{j=1}^{n} a_{ij}(x) \underbrace{\frac{\partial \psi_l}{\partial x_j}(x)}_{S_{lj}} \right] \right\}}_{\tilde{a}_{kl}(x) = (SAS^T)_{k,l}(x)}$$
$$+ \sum_{k=1}^{n} \frac{\partial \tilde{u}}{\partial \xi_k}(\psi(x)) \underbrace{\left\{ \sum_{i=1}^{n} a_i(x) \frac{\partial \psi_k}{\partial x_i}(x) \right\}}_{\tilde{a}_k(x)} + a_0(x) = 0.$$

Nach Einsetzen von $x = \psi^{-1}(\xi)$ haben wir (15.8) auf ein äquivalentes System

$$\sum_{k=1}^{n}\sum_{l=1}^{n} \tilde{a}_{kl}(\xi) \frac{\partial^2 \tilde{u}}{\partial \xi_k \partial \xi_l} + \sum_{k=1}^{n} \tilde{a}_k(\xi) \frac{\partial \tilde{u}}{\partial \xi_k}(\xi) + \tilde{a}_0(\xi) = 0$$

mit $\tilde{A} = SAS^T$ transformiert. Wegen $\frac{\partial^2 u}{\partial x_i \partial x_j} = \frac{\partial^2 u}{\partial x_j \partial x_i}$ kann man A als symmetrisch annehmen. A besitzt einen vollständigen Satz von Eigenvektoren und ist reell diagonalisierbar. Darüberhinaus existiert eine unitäre Matrix Q (d.h. $Q^T Q = I$), so dass $A = Q^T D Q$. Die Diagonalmatrix D enthält gerade die Eigenwerte $\lambda \in \mathbb{R}$.
Mittels der Diagonalmatrix $\tilde{D}$

$$\tilde{d}_{ii} = \begin{cases} \frac{1}{\sqrt{|d_{ii}|}} & d_{ii} \neq 0 \\ 1 & \text{sonst} \end{cases}$$

wählen wir die Transformation ψ an jeder Stelle x so, dass $S = \tilde{D}Q$. Damit gilt dann

$$\tilde{A} = SAS^T = \left(\tilde{D}Q\right) Q^T DQ \left(Q^T \tilde{D}\right) = \tilde{D}D\tilde{D},$$

also

$$\tilde{a}_{ii} = \begin{cases} 1 & d_{ii} > 0 \\ -1 & d_{ii} < 0 \\ 0 & d_{ii} = 0 \end{cases},$$

Man kann auch zeigen, dass für jede zulässige Transformation ψ die Vorzeichen der Eigenwerte von $\tilde{A}$ mit denen von A übereinstimmen (Trägheitssatz von Sylvester). □

Definition 15.3 (Typeinteilung)**.** (15.8) heißt

1. *elliptisch in* x, falls alle Eigenwerte von $A(x)$ gleiches Vorzeichen besitzen und kein Eigenwert 0 ist.
2. *hyperbolisch in* x, falls kein Eigenwert von $A(x)$ 0 ist, $n-1$ Eigenwerte gleiches Vorzeichen besitzen und ein Eigenwert das entgegengesetzte Vorzeichen hat.
3. *parabolisch in* x, falls genau ein Eigenwert 0 ist, die übrigen Eigenwerte gleiches Vorzeichen besitzen und $\text{Rang}\left(A(x), a(x)\right) = n$ ist mit $a(x) = \left(a_1(x), \ldots, a_n(x)\right)^T$.
4. *elliptisch / hyperbolisch / parabolisch in* Ω, falls sin in jedem Punkt $x \in \Omega$ elliptisch / hyperbolisch / parabolisch ist. □

Folgerung 15.4 (Zweidimensionaler Fall)**.** In zwei Raumdimensionen ist die Einteilung eindeutig. Es gibt zwei Eigenwerte λ_1, λ_2.

1. $\lambda_1\lambda_2 > 0$: elliptisch
2. $\lambda_1\lambda_2 < 0$: hyperbolisch
3. $\lambda_1\lambda_2 = 0$: parabolisch, wobei man hier fordert, dass

$$\text{Rang}\begin{pmatrix} a_{11} & a_{12} & a_1 \\ a_{21} & a_{22} & a_2 \end{pmatrix} = 2$$

ist. Dies schließt Fälle wie $\frac{\partial u}{\partial x_1} + \frac{\partial u}{\partial x_2} = 0$ oder $\frac{\partial^2 u}{\partial x_1 \partial x_2} + \frac{\partial u}{\partial x_1} = 0$ aus. □

Bemerkung 15.5. Der Typ hängt nur vom Hauptteil der Gleichung, also den Koeffizienten vor den höchsten Ableitungen, ab. □

16 Finite Differenzen für die Poissongleichung

Das Finite–Differenzen–Verfahren ist das einfachste Verfahren zur numerischen Lösung partieller Differentialgleichungen.

16.1 Differenzenformeln

Mittels Taylorreihenentwicklung erhält man

$$\begin{aligned} u(x+h) &= u(x) + hu'(x) + \frac{h^2}{2}u''\left(x+\vartheta^+ h\right) \\ \Leftrightarrow u'(x) &= \frac{u(x+h)-u(x)}{h} - \frac{h}{2}u''\left(x+\vartheta^+ h\right), \quad \vartheta^+ \in [0,1]. \end{aligned} \tag{16.1}$$

Ebenso erhält man bei Entwicklung von $u(x-h)$:

$$\begin{aligned} u(x-h) &= u(x) - hu'(x) + \frac{h^2}{2}u''\left(x-\vartheta^- h\right) \\ \Leftrightarrow u'(x) &= \frac{u(x)-u(x-h)}{h} + \frac{h}{2}u''\left(x-\vartheta^- h\right), \quad \vartheta^- \in [0,1]. \end{aligned} \tag{16.2}$$

Bis auf einen Fehler $O(h)$ kann man die Ableitung also durch Funktionswerte ausdrücken. (16.1) heißt *Vorwärtsdifferenz* und (16.2) *Rückwärtsdifferenz*. Durch Kombination verschiedener Entwicklungen kann man den Fehler weiter verkleinern:

$$u(x+h) = u(x) + hu'(x) + \frac{h^2}{2}u''(x) + \frac{h^3}{6}u'''\left(x+\vartheta^+ h\right), \tag{16.3}$$

$$u(x-h) = u(x) - hu'(x) + \frac{h^2}{2}u''(x) - \frac{h^3}{6}u'''\left(x+\vartheta^- h\right). \tag{16.4}$$

(16.3) - (16.4) liefert die *zentrale Differenz* für die erste Ableitung:

$$u'(x) = \frac{u(x+h)-u(x-h)}{2h} - \frac{h^2}{12}\left(u'''\left(x+\vartheta^+ h\right) + u'''\left(x+\vartheta^- h\right)\right) \tag{16.5}$$

Schließlich kann man durch Linearkombination verschiedener Entwicklungen auch Näherungen für Ableitungen höherer Ordnung erzeugen. So liefert (16.3) + (16.4) bei Entwicklung bis zur 4. Ordnung

$$u(x+h) + u(x-h) = 2u(x) + h^2u''(x) + \frac{h^4}{24}\left(u^{(4)}\left(x+\vartheta^+ h\right) + u^{(4)}\left(x+\vartheta^- h\right)\right)$$

und damit die zentrale Differenz für die zweite Ableitung:

$$u''(x) = \frac{u(x+h)-2u(x)+u(x-h)}{h^2} - \frac{h^2}{24}\left(u^{(4)}\left(x+\vartheta^+ h\right) + u^{(4)}\left(x+\vartheta^- h\right)\right). \tag{16.6}$$

Lemma 16.1. Es sei $u \in C^k(C)\left(\overline{\Omega}\right)$ mit $k = 2, 3, 4$. Dann gilt

$$\begin{aligned}
\frac{1}{h}\left(u(x+h) - u(x)\right) &= u'(x) + hR \qquad \text{mit } |R| \le \frac{1}{2}\|u\|_{C^2(\overline{\Omega})} \\
\frac{1}{h}\left(u(x) - u(x-h)\right) &= u'(x) + hR \qquad \text{mit } |R| \le \frac{1}{2}\|u\|_{C^2(\overline{\Omega})} \\
\frac{1}{2h}\left(u(x+h) - u(x-h)\right) &= u'(x) + h^2 R \qquad \text{mit } |R| \le \frac{1}{6}\|u\|_{C^3(\overline{\Omega})} \\
\frac{1}{h^2}\left(u(x-h) - 2u(x) + u(x+h)\right) &= u''(x) + h^2 R \qquad \text{mit } |R| \le \frac{1}{12}\|u\|_{C^4(\overline{\Omega})}
\end{aligned}$$

□

Bemerkung 16.2. h^2–Konvergenz setzt gleiche Auswerteabstände voraus! □

16.2 Finite Differenzen in einer Raumdimension

Wir beschränken uns zunächst auf das eindimensionale Randwertproblem

$$\begin{aligned}
-u''(x) &= f(x), \qquad x \in \Omega = (0,1), \\
u(0) &= \varphi_0, \\
u(1) &= \varphi_1.
\end{aligned} \tag{16.7}$$

Das Gebiet Ω wird nun in N gleichlange Intervalle zerteilt:

$$x_i = ih, \qquad h = \frac{1}{N}.$$

Durch Ansetzen der Differenzenformel in den Gitterpunkten erhält man

$$\begin{aligned}
-\frac{1}{h^2}\left[u(x_{i-1}) - 2u(x_i) + u(x_{i+1})\right] + O\left(h^2\right) &= f(x_i), \quad i = 1, \ldots, N-1, \\
u(x_0) &= \varphi_0, \\
u(x_N) &= \varphi_1.
\end{aligned} \tag{16.8}$$

Durch Streichen des Fehlerterms erhält man ein lineares Gleichungssystem für die Näherungswerte $u_h(x_i)$ in den Gitterpunkten:

$$\begin{aligned}
\frac{1}{h^2}\left[-u_h(x_{i-1}) + 2u_h(x_i) - u_h(x_{i+1})\right] &= f(x_i), \quad i = 1, \ldots, N-1, \\
u_h(x_0) &= \varphi_0, \\
u_h(x_N) &= \varphi_1.
\end{aligned} \tag{16.9}$$

Mittels $\overline{\Omega}_h = \left\{ih \in \overline{\Omega} \middle| i \in \mathbb{Z}\right\}$ kann man sich $u_h : \overline{\Omega}_h \to \mathbb{R}$ als eine *Gitterfunktion* vorstellen. Wahlweise kann u_h auch als Vektor interpretiert werden. Eliminiert man die beiden

Randwerte, so ergibt sich das lineare Gleichungssystem

$$\underbrace{\frac{1}{h^2}\begin{pmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{pmatrix}}_{=:L_h} \underbrace{\begin{pmatrix} u_h(x_1) \\ u_h(x_2) \\ \vdots \\ u_h(x_{N-2}) \\ u_h(x_{N-1}) \end{pmatrix}}_{=:u_h} = \underbrace{\begin{pmatrix} f(x_1) + \frac{\varphi_0}{h^2} \\ f(x_2) \\ \vdots \\ f(x_{N-2}) \\ f(x_{N-1}) + \frac{\varphi_1}{h^2} \end{pmatrix}}_{=:q_h}. \tag{16.10}$$

L_h ist eine symmetrisch positiv definite Tridiagonalmatrix.

16.3 Der n-dimensionale Fall

Die Differenzenformeln kann man auch für partielle Ableitungen heranziehen:

$$\frac{\partial^2 u}{\partial x_i^2}(x) = \frac{1}{h}\left[u(x + he_i) - 2u(x) + u(x - he_i)\right] + O\left(h^2\right) \tag{16.11}$$

mit e_i dem Einheitsvektor in Koordinatenrichtung i. Sei nun die Poissongleichung

$$\begin{aligned} -\Delta u(x) &= f(x), \quad x \in \Omega = (0,1)^n, \\ u(x) &= \varphi(x), \quad x \in \partial\Omega, \end{aligned} \tag{16.12}$$

im n-dimensionalen Einheitswürfel zu lösen. Das Gitter $\overline{\Omega}_h$ besteht nun aus den Punkten

$$\overline{\Omega}_h = \left\{(i_1 h, \ldots, i_n h) \in \overline{\Omega} \middle| i_k \in \mathbb{Z}, 1 \le k \le n\right\}.$$

Weiter sei

$$\Omega_h = \left\{(i_1 h, \ldots, i_n h) \in \Omega \middle| i_k \in \mathbb{Z}, 1 \le k \le n\right\} \subset \overline{\Omega}_h$$

und

$$\partial\Omega_h = \overline{\Omega}_h \setminus \Omega_h.$$

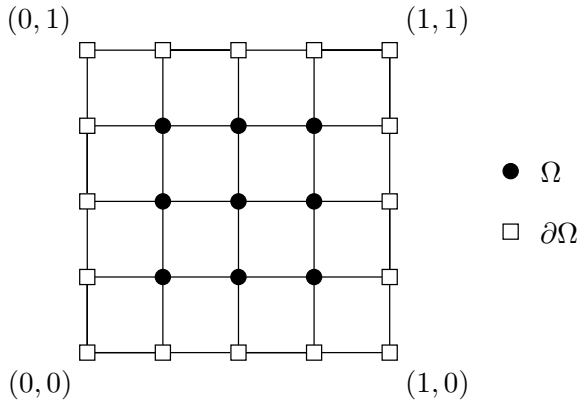
Schließlich hat man wieder die Identität

$$\begin{aligned} \frac{1}{h^2}\left\{2n\,u(x) - \sum_{i=1}^{n}\left[u(x + he_i) + u(x - he_i)\right]\right\} + O\left(h^2\right) &= f(x), \quad x \in \Omega_h, \\ u(x) &= \varphi(x), \quad x \in \partial\Omega_h. \end{aligned} \tag{16.13}$$

Bezeichnet man $u_h : \overline{\Omega}_h \to \mathbb{R}$ wieder als Gitterfunktion der Näherungen, so setzt man

$$\begin{aligned} \frac{1}{h^2}\left\{2d u_h(x) - \sum_{i=1}^{d}\left[u_h(x + he_i) + u_h(x - he_i)\right]\right\} &= f(x), \quad x \in \Omega_h, \\ u_h(x) &= \varphi(x), \quad x \in \partial\Omega_h. \end{aligned} \tag{16.14}$$

Das lineare Gleichungssystem hat nach Elimination der Randwerte $(N-1)^n$ Unbekannte. Wir illustrieren den Fall $n=2,\ N=4$ genauer:



Das lineare Gleichungssystem hat nach Elimination der Randwerte folgende Gestalt:

$$\underbrace{\frac{1}{h^2}\left(\begin{array}{ccc|ccc|ccc}
4 & -1 & & -1 & & & & & \\
-1 & 4 & -1 & & -1 & & & & \\
 & -1 & 4 & & & -1 & & & \\
\hline
-1 & & & 4 & -1 & & -1 & & \\
 & -1 & & -1 & 4 & -1 & & -1 & \\
 & & -1 & & -1 & 4 & & & -1 \\
\hline
 & & & -1 & & & 4 & -1 & \\
 & & & & -1 & & -1 & 4 & -1 \\
 & & & & & -1 & & -1 & 4
\end{array}\right)}_{L_h}
\underbrace{\left(\begin{array}{c}
u_h(h,h) \\ u_h(2h,h) \\ u_h(3h,h) \\ u_h(h,2h) \\ u_h(2h,2h) \\ u_h(3h,2h) \\ u_h(h,3h) \\ u_h(2h,3h) \\ u_h(3h,3h)
\end{array}\right)}_{u_h}$$

$$= \underbrace{\left(\begin{array}{c}
f(h,h) + \frac{\varphi(h,0)}{h^2} + \frac{\varphi(0,h)}{h^2} \\
f(2h,h) + \frac{\varphi(2h,0)}{h^2} \\
f(3h,h) + \frac{\varphi(3h,0)}{h^2} + \frac{\varphi(1,h)}{h^2} \\
f(h,2h) + \frac{\varphi(0,2h)}{h^2} \\
f(2h,2h) \\
f(3h,2h) + \frac{\varphi(1,2h)}{h^2} \\
f(h,3h) + \frac{\varphi(h,1)}{h^2} + \frac{\varphi(0,3h)}{h^2} \\
f(2h,3h) + \frac{\varphi(2h,1)}{h^2} \\
f(3h,3h) + \frac{\varphi(3h,1)}{h^2} + \frac{\varphi(1,3h)}{h^2}
\end{array}\right)}_{q_h} \tag{16.15}$$

- Wegen der lexikographischen Anordnung der Gitterpunkte ist L_h eine 5–Band–Matrix.

- L_h ist dünnbesetzt: Maximal $2n + 1$ Nichtnullelemente pro Zeile.
- L_h ist symmetrisch und positiv definit (was nicht unmittelbar einsichtig ist).

Für eine genauere Behandlung der Methode der Finiten Differenzen verweisen wir auf das Buch (Hac86).

Literatur

[Bal96] Ball, Peter: *Introduction to Discrete Event Simulation*. http://www.dmem.strath.ac.uk/~pball/simulation/simulate.html, 1996.

[Bas08] Bastian, Peter: *Numerik partieller Differentialgleichungen*. Vorlesungsskript, Universität Stuttgart, 2008.

[BC84] Banks, J. und J.S. Carson: *Discrete-Event System Simulation*. Prentice-Hall, 1984.

[BCP96] Brenan, K. E., S. L. Campbell und L. R. Petzold: *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations*. SIAM, 1996.

[Bur] Buroojy, Nicholas: *Cellular Automata*. http://falconnet.peddie.org/students/2007/nburoojy/projects/cellular/.

[Fey63] Feynman, Richard: *The Feynman Lectures on Physics: Volume 2*, Band 2 der Reihe *The Feynman Lectures on Phsyics*. Addison-Wesley, 1963.

[FHP86] Frisch, U., B. Hasslacher und Y. Pomeau: *Lattice-gas automata for the Navier-Stokes equation*. Physical Review Letters, 56:1505–1508, apr 1986.

[Hac86] Hackbusch, W.: *Theorie und Numerik elliptischer Differentialgleichungen*. Teubner, 1986. http://www.mis.mpg.de/scicomp/articleshackbusch_d.html.

[Kra97] Krabs, Werner: *Mathematische Modellierung*. Teubner, 1997.

[NS92] Nagel, K. und M. Schreckenberg: *A cellular automaton model for freeway traffic*. Journal de Physique I France 2, Seiten 2221–2229, 1992.

[Ran] Rannacher, R.: *Numerische Mathematik 1 (Numerik gewöhnlicher Differentialgleichungen)*. http://numerik.iwr.uni-heidelberg.de/~lehre/notes.

[Ran06] Rannacher, R.: *Einführung in die Numerische Mathematik (Numerik 0)*. http://numerik.iwr.uni-heidelberg.de/~lehre/notes, 2006.

[SB05] Stoer, J. und R. Bulirsch: *Numerische Mathematik II*. Springer, 5. Auflage, 2005.

[Sim] Simeon, B.: *Numerik gewöhnlicher Differentialgleichungen*. http://www-m2.ma.tum.de/~simeon/numerik4/skript.html.

[Unb81] Unbehauen, R.: *Elektrische Netzwerke*. Springer, 1981.

[Yan81] Yannakakis, Mihalis: *Computing the Minimum Fill-In is NP-Complete*. SIAM Journal on Algebraic and Discrete Methods, 2(1):77–79, 1981.

Index

A-stabil, 118
Abschneidefehler, 99, 104
Anfangsbedingung, 124
Anfangswertproblem, 93

CG-Verfahren, 83
    Algorithmus, 86
    Konvergenz, 87
Charakterisierungssatz, 79
Cholesky-Zerlegung, 73

DAE, 92
Defekt, 81
deterministisches Modell, 14
Dirichlet, 129
diskrete Systeme, 15
diskretes Modell, 14
Divergenz, 123
dynamisches Modell, 14

Einschrittverfahren
    Konvergenz, 106
elektrische Bauelemente, 35
elektrische Feldstärke, 33
Energieerhaltung, 125
ereignisgesteuerte Systeme, 15
Eulerverfahren
    explizit, 99
    implizit, 106
explizite Form, 92

Fill-In, 71
Fortsetzungssatz, 93
Fourier'sches Gesetz, 126

Galerkin-Gleichungen, 83
Game of Life, 25
Gauß'scher Integralsatz, 124
Gauß-Seidel-Verfahren, 81
Gebiet, 123
Gitterfunktion, 136
Gradient, 123
Gradientenverfahren, 82
Gronwall, Lemma von, 95
    diskret, 101

Hauptabschneidefunktion, 111
Heun, 105

Jacobimatrix, 118

Kirchhoff, 37
Knotenpotentialverfahren, 43
Kondition, 63
Konduktion, 126
Konsistenz, 104
kontinuierliches System, 15
Kontinuumshypothese, 124
Konvektion, 126
Konvergenzsatz, 109
Krylovraum, 84

L-Stetigkeit, 109
Ladungsdichte, 35
Laplace, 127
Laplace–Gleichung, 129
Laplacegleichung, 127
Line Search, 80
lineare Stabilitätsanalyse, 118
Lipschitzbedingung, 94

mathematisches Modell, 11
Mehrfrontenmethode, 76
Mehrschrittverfahren
    linear, 120
Mehrskaligkeit, 14
Mittelpunktregel, 106
Modell, 11

Nagel-Schreckenberg, 27
Nested Dissection, 73
Neumann, 129

Parallelschwingkreis, 52
partielle Integration, 123

Peano, Existenzsatz, 93
Poissongleichung, 127
Potential, 34
Potential–Gleichung, 129

Rückwärtsdifferenz, 135
Randbedingung, 124, 129
Randwertproblem, 91, 130
RC-Glied, 41
Reihenschwingkreis, 42
Reverse Cuthill-McKee, 72
Robin, 129
Runge-Kutta-Verfahren, 105
    implizit, 120

sachgemäß gestellt, 132
semi-explizite Form, 92
Spannung, 33
Spektralradius, 79
Stabilitätsfunktion, 116
Stabilitätsgebiet, 116
Stabilitätssatz, 94
steifes System, 119
Strömungssimulation, 28
Strom, 34
Stromdichte, 35
System, 11
Systemklassifikation, 15

Transportgleichung, 131
Trapezregel, 106
Typeinteilung, 134

Verfahrensfunktion, 104
Verkehrssimulation, 27
Verstärkungsfaktor, 116
Vorwärtsdifferenz, 135

Wärmeenergie, 125
Wärmeleitungsgleichung, 126, 130
Wellengleichung, 130
Widerstandsnetzwerk, 51

zellulärer Automat, 23
zentrale Differenz, 135